CASSYS v. 1.5.3 Reference Guide

Physical Models and C# Engine Guide

Abhijeet Pai & Didier Thevenard,
Canadian Solar O&M

TABLE OF CONTENTS

1 Introduction

The following document describes the physical models and used to develop the C#-source code available at CASSYS.

Please Note: This document is currently a work in progress, and is updated as new methods are added to the program.

2 Grid Connected Mode - General Program Flow

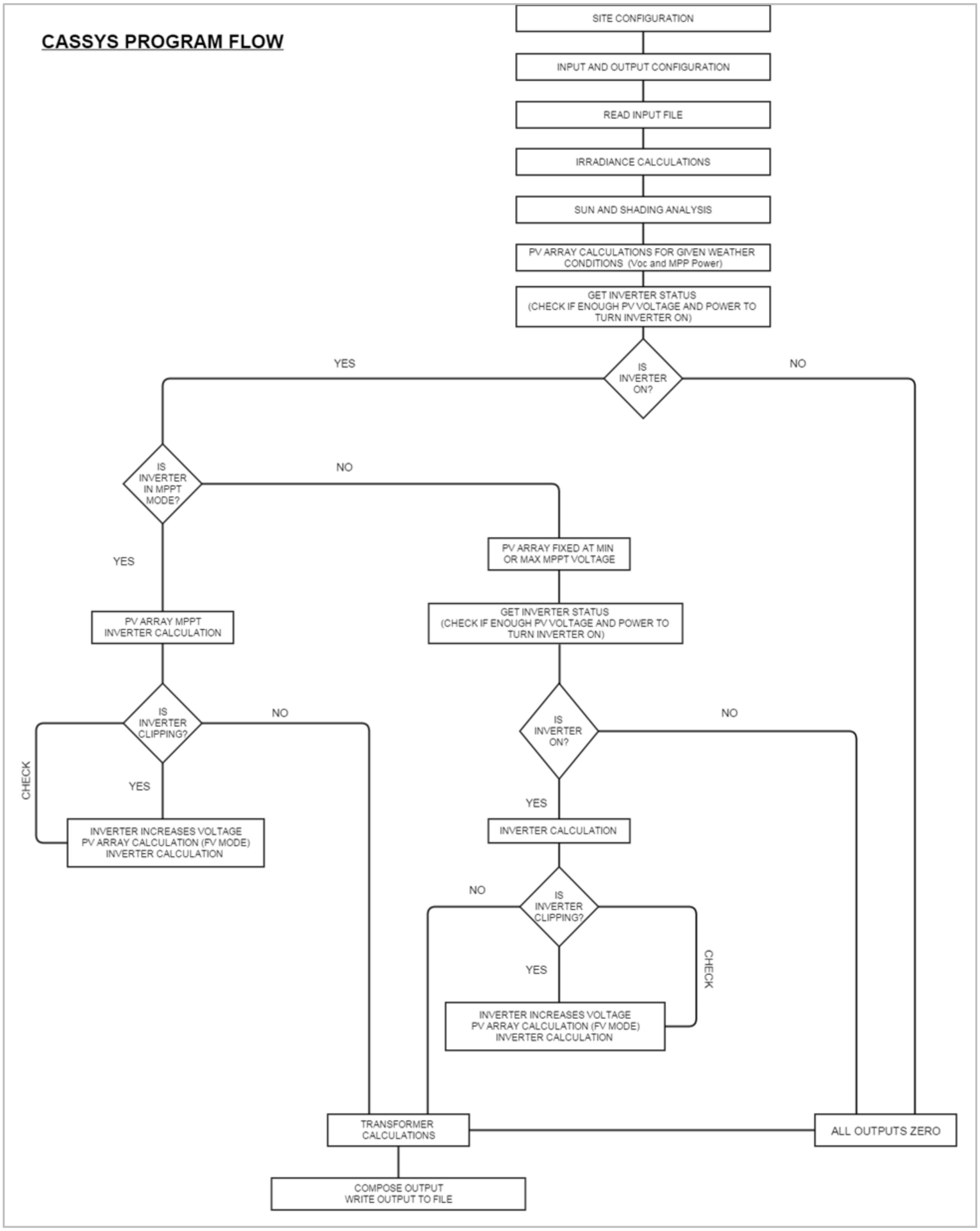


Figure 1: Grid Connected Mode - CASSYS v.1.1.0 Program Flow

3 List of Classes

A typical simulation flow is shown in Figure 1. Version 1.5.0 of CASSYS uses the following classes to complete a simulation:

Classes that simulate physical objects or devices:

1) ACWiring.cs
2) BackTilter.cs
3) GridConnectedSystem.cs
4) GroundShading.cs
5) HorizonShading.cs
6) Inverter.cs
7) LossDiagram.cs
8) PVArray.cs
9) RadiationProc.cs
10) Shading.cs
11) Simulation.cs
12) SpectralEffects.cs
13) Splitter.cs
14) Sun.cs
15) Tilter.cs
16) Tracker.cs
17) Transformer.cs

Classes that contain mathematical constructs:

18) ASTM E2848: ASTME2848.cs
19) Daily Astronomical Calculations: Astro.cs
20) Linear and Quadratic Interpolation: Interpolate.cs
21) Solar Radiation on a Tilted Surface: Tilt.cs

Classes that support program function, configuration and error checking:

22) Input File Reading: MetReader.cs
23) Site and Simulation Settings: ReadFarmSettings.cs
24) Error Logging: ErrorLogger.cs
25) Batch or Interface Mode initialization: CASSYS.cs
26) Utilities and Unit Conversions: Utilities.cs

4 Classes and Physical Model Guide

Parameters for each class are obtained from the .CSYX file which is defined by the user through the CASSYS MS Excel-based interface. Parameters remain unchanged during the entirety of the simulation, and provide their relevant classes (i.e. their devices or models) with key defining characteristics. The inputs are then processed and reported to an output file selected by the user. The physical models, the parameters used, and the treatment of the input values to obtain relevant outputs are discussed in this section.

4.1 AC Wiring *ACWiring.cs*

AC Wiring represents the wires in the Photovoltaic system from the inverters to the transformer. This class is used to calculate the ac wiring loss caused due to wire resistance. Each instance of AC wiring is affiliated with a respective inverter within the system.

4.1.1 Inputs

ArrayNum	*Number of subarrays connected to respective inverter*
PNomArrayDC	*Nominal power produced by subarray*
SimInverter	*Instance of Inverter*

4.1.2 Outputs

ACWiringLoss	*Losses due to wiring resistance between Inverter and Transformer [W]*

4.1.3 Parameters

itsACWiringLossPC	*AC wiring loss specified as a percentage [%]*
itsACWiringRes	*AC wiring loss as a resistance [ohms]*

These parameters are used to model and simulate the effects of AC wiring and the losses due to resistance and calculate the related outputs.

4.1.4 Modelling AC Wiring Losses

The AC wiring losses are the losses between the inverter and the input of the grid connected transformer. There are two ways to specify AC wiring losses in the interface: as a percent of the STC rating of the inverter (which is the DC array size attached to the inverter, multiplied by the maximum input power efficiency), or as a percent of the inverter nominal power. The second approach is the preferred one; it corresponds to the way electrical engineers would actually size the cables. The first approach is provided only for compatibility with the way AC wiring losses are treated in PVsyst.

First method (percentage of STC rating of inverter)

The AC wiring loss of the inverter is calculated by determining an effective resistance in the wiring following the inverter based on the STC rating.

$$R_{AC} = \frac{AC\ Loss_{\%}\ V_{out}}{P_{AC,STC}/(V_{out} * \sqrt{Ph})} \quad \textbf{(1)}$$

Where,
$AC\ Loss_{\%}$: STC loss percentage specified by the user [%]
V_{out}: Output voltage of the inverter [V]
$P_{AC,STC}$: $\eta * P_{DC,STC}$ – Product of the maximum efficiency of the inverter (medium voltage level, if three curves), and the STC size of the array
Ph: Number of output phases of the inverter

The AC wiring losses are then given by:

$$AC_{Losses} = \sqrt{Ph} * ({I_{out}}^2) * R_{AC} \quad \textbf{(2)}$$

Where,
I_{out}: Output current during operation

Second method (percentage of nominal rating of inverter)

The first equation above is replaced with:

$$R_{AC} = \frac{AC\ Loss_{\%}\ V_{out}}{P_{AC,nom}/(V_{out} * \sqrt{Ph})} \quad \textbf{(3)}$$

Where,
$AC\ Loss_{\%}$: loss as a percentage of nominal inverter power specified by the user [%]
V_{out}: Output voltage of the inverter [V]
$P_{AC,nom}$: Nominal AC power of the inverter
Ph: Number of output phases of the inverter

The second equation is unchanged.

4.2 Back Tilter *BackTilter.cs*

The BackTilter class determines the irradiance that reaches the back side of the array. It calculates and combines diffuse irradiance, irradiance reflected from the front of the array behind, irradiance reflected from the ground, and any direct irradiance the back may receive. When determining shading and front-panel-reflection, CASSYS is capable of distinguishing between single row and multi-row systems. The model used has been adapted from the bifacial irradiance model proposed by NREL [1].

4.2.1 Inputs

HDif	*Horizontal diffuse irradiance (W/m^2)*
TDifRef	*Diffuse irradiance reflected from front of array (W/m^2)*
HGlo	*Horizontal global irradiance (W/m^2)*
FrontGroundGHI	*Global horiz. irradiance profile of patch of ground before row*
RearGroundGHI	*Global horiz. irradiance profile of patch of ground behind row*
AveGroundGHI	*Avg. global irradiance on patch of ground behind row (W/m^2)*
Month	*Current month*
BackSH	*Fraction of back side of array that is unshaded*

TDir	*Plane-of-array beam irradiance on back of array (W/m²)*

For Tracking Configurations:

Panel Tilt	*Tilt of the modules relative to the ground (rad)*
Clearance	*Distance from lowest point of array to ground (m)*

4.2.2 Outputs

IAMDir	*Incidence angle modifier (IAM) for beam irradiance (unitless)*
IAMDif	*IAM for diffuse irradiance (unitless)*
IAMRef	*IAM for front- and ground-reflected irradiances (unitless)*
IrrInhomogeneity	*Inhomogeneity of global irradiance across back of array (%)*
Albedo	*Albedo for the current month*
BGlo	*Effective global irradiance on back of array (W/m²)*
BackGlo[]	*Effective global irr. for each cell row on back of array (W/ m²)*
BDir	*Effective beam irradiance on back of array (W/m²)*
BDif	*Effective diffuse irradiance on back of array (W/m²)*
BFroRef	*Effective front-reflected irradiance on back of array (W/m²)*
BGroRef	*Effective ground-reflected irradiance on back of array (W/m²)*

4.2.3 Parameters

Panel Tilt	*Tilt of the modules relative to the ground (rad)*
Clearance	*Distance from lowest point of array to ground (m)*
Array BW	*Bandwidth of the array (m)*
Pitch	*Distance between consecutive rows of modules (m)*
NumCellRows	*Number of cells across the width of the array*
NumRows	*Number of PV rows in the system*

4.2.4 Equations and Model Descriptions

To account for the non-uniformity of irradiance across the back of the array, the model calculates the irradiance at the location of each row of cells. A view factor model is used where the 180° field-of-view of each cell is divided into 1° segments [2]. The angles β_{sky} and β_{ground} determine broader segmentation corresponding to different components of irradiance, shown by Figure 2. These segments differ depending on which cell row i is computed.

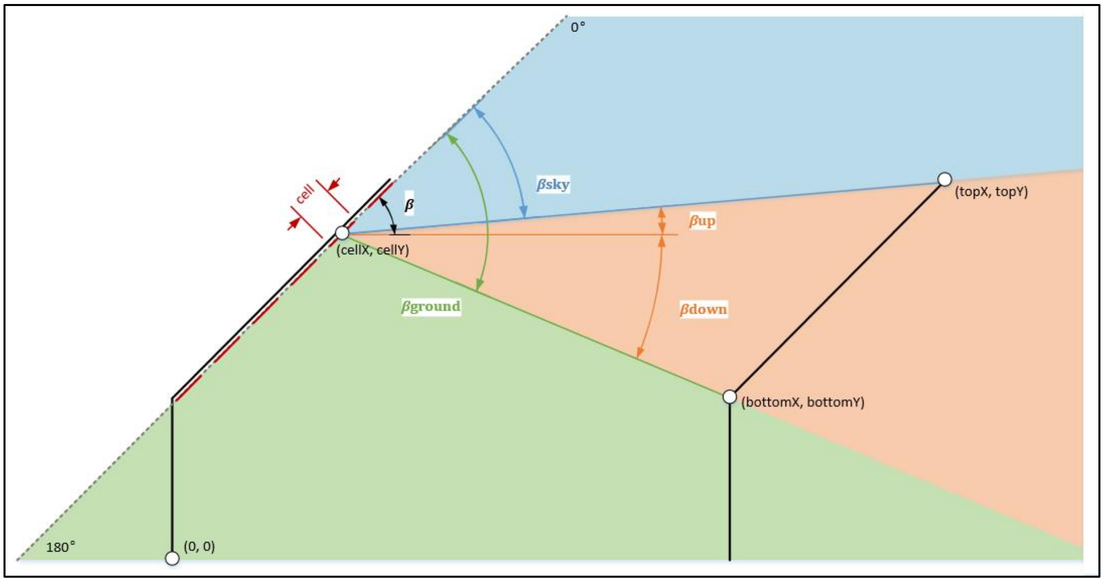


Figure 2: Back Side Irradiance Segmentation

A view factor, which is the fraction of the irradiance leaving one surface that is incident on another, is the basis of the model. For calculating the view factor between two angles θ_1 and θ_2, where $\theta_2 > \theta_1$, formula (4) from the *RadiationProc* class is used throughout:

$$VF(\theta_1, \theta_2) = \frac{1}{2} \cdot [\cos(\theta_1) - \cos(\theta_2)] \quad \textbf{(4)}$$

4.2.4.1 Model of the Diffuse Component

The diffuse component for cell i ($B_{i,dif}$) is straightforward to calculate, as the product of the view factor and the horizontal diffuse irradiance (H_{dif}):

$$B_{i,dif} = VF(0, \beta_{i,sky}) \cdot H_{dif} \quad \textbf{(5)}$$

4.2.4.2 Model of the Front-Reflected Component

For the front-reflected component, the source of irradiance is the effective diffuse component reflected off the front of the array to the rear ($T_{dif_{ref}}$). In accordance with [2], CASSYS assumes that the beam reflected from the front of the panel is specular and does not reach the back of the row ahead. The reflection factor (R) is obtained from equation (5.1.3) of [3], assuming that the surface of reflectance is glass and the angle of incidence is 60°. Equation (6) calculates this front-reflected component, where T_{dif} is the tilted diffuse irradiance on the front after shading, IAM_{dif} is the incidence angle modifier for diffuse irradiance, and L_{soil} is the soiling loss.

$$T_{dif_{ref}} = T_{dif} \cdot (1 - IAM_{dif} \cdot (1 - R)) \cdot (1 - L_{soil}) \quad \textbf{(6)}$$

Thus the front-reflected component for cell i ($B_{i,front}$) is calculated in equation (7).

$$B_{i,front} = VF(\beta_{i,sky}, \beta_{i,ground}) \cdot T_{dif_{ref}} \quad \textbf{(7)}$$

4.2.4.3 Model of the Ground-Reflected Component

Whereas the other components all involve a homogeneous source of irradiance, the ground-reflected component varies significantly based on the portion of ground viewed, due to shading. Therefore a more precise method, illustrated in Figure 3, is employed.

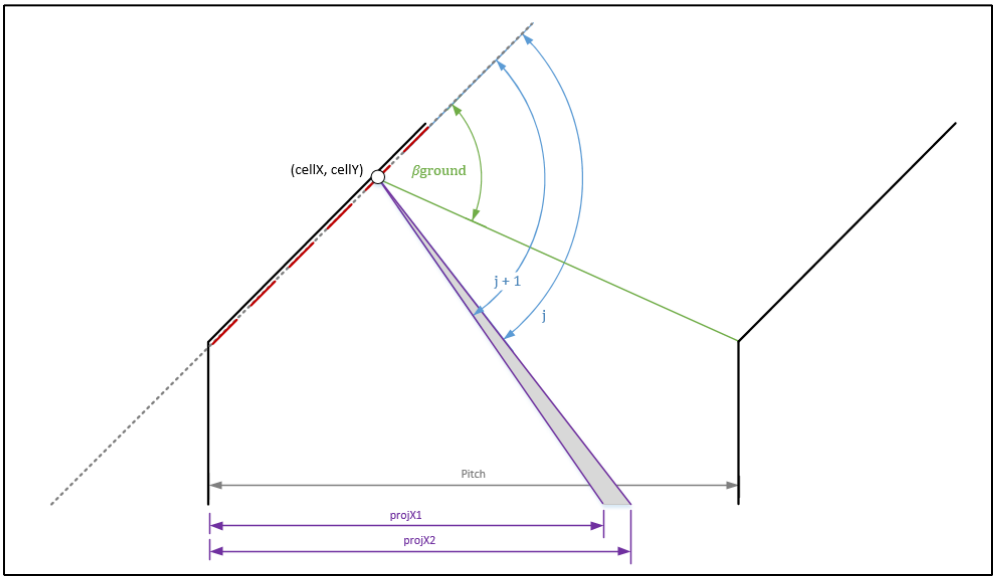


Figure 3: Back Side Ground Reflected View Factor

Each 1° segment (j, $j+1$) of the field-of-view from β_{ground} to 180° is projected onto the ground, and the view factor computed for that individual segment. The ground irradiance for the (j, $j+1$) segment ($groundGHI_j$) is found by indexing the ground irradiance profile given as input. When the projection is the length of the pitch or greater, an average value is used.

Whether the projection falls inside or outside of the pitch, CASSYS assumes that all ground patches throughout the array rows will have the same irradiance profile. Therefore, with transposed indices, $groundGHI_j$ may always be found. Hence the ground-reflected component for cell i ($B_{i,ground}$) is calculated in equation (8), where ρ is the albedo and $proj_{X2} - proj_{X1}$ is the width of the projection.

$$B_{i,ground} = \sum_{j=\beta_{ground}}^{179} VF(j, j+1) \cdot \frac{groundGHI_j}{proj_{X2} - proj_{X1}} \cdot \rho \qquad \textbf{(8)}$$

4.2.4.4 Model of the Beam Component

In the rare case that direct sunlight reaches the back of the array, the beam irradiance (T_{dir}) is obtained through the same transposition model as in equation (106) of the *Tilter* class. The beam component for cell i ($B_{i,dir}$) is then calculated in equation (9), including the back shading fraction ($SH_{i,back}$) as a loss:

$$B_{i,dir} = T_{dir} \cdot (1 - SH_{i,back}) \qquad \textbf{(9)}$$

4.2.4.5 *Effective Global Irradiance for One Cell Row*

Before the irradiance components are summed, they are corrected by their incidence angle modifiers and for other losses. The final value $B_{i,glo}$ represents the effective global irradiance for the cell row i, where L_{struct} is the loss from back structural blocking.

$$B_{i,glo} = \left[B_{i,dif} \cdot IAM_{dif} + \left(B_{i,front} + B_{i,ground}\right) \cdot IAM_{ref} + B_{i,dir} \cdot IAM_{dir}\right] \cdot \left(1 - L_{struct}\right) \qquad \textbf{(10)}$$

For the diffuse and reflected components, CASSYS uses the same angle of 60° to compute IAM_{dif} and IAM_{ref} in accordance with [3]. For the beam component, IAM_{dir} is obtained using the angle of incidence. For more on the incidence angle modifier, see Section 4.8.7.

4.2.4.6 *Effective Global Irradiance for Back of Array*

At this point, the only calculation left is to average the irradiance values across the cells on the back of the array, given by equation (11):

$$B_{glo} = \sum_{i=1}^{numCellRows} \frac{B_{i,glo}}{numCellRows} \qquad \textbf{(11)}$$

4.3 Grid Connected System *GridConnectedSystem.cs*

Grid Connected System is responsible for handling all calculations relating to a grid connected solar site.

4.3.1 Inputs

RadProc	*Instance of RadiationProc*
SimMet	*Climate information from input file*

4.3.2 Outputs

Global_POA_Irradiance_Corrected_for_Shading	*[Wm^{-2}]*
Near_Shading_Loss_for_Global	*[Wm^{-2}]*
Near_Shading_Loss_for_Beam	*[Wm^{-2}]*
Near_Shading_Loss_for_Diffuse	*[Wm^{-2}]*
Near_Shading_Loss_for_Ground_Reflected	*[Wm^{-2}]*
Global_POA_Irradiance_Corrected_for_Incidence	*[Wm^{-2}]*
Radiation_Soiling_Loss	*[Wm^{-2}]*
Radiation_Spectral_Loss	*[Wm^{-2}]*
Incidence_Loss_for_Global	*[Wm^{-2}]*
Incidence_Loss_for_Beam	*[Wm^{-2}]*
Incidence_Loss_for_Diffuse	*[Wm^{-2}]*
Incidence_Loss_for_Ground_Reflected	*[Wm^{-2}]*
Profile_Angle	*[degrees]*
Near_Shading_Factor_on_Global	*[unitless]*
Near_Shading_Factor_on_Beam	*[unitless]*
Near_Shading_Factor_on__Diffuse	*[unitless]*
Near_Shading_Factor_on_Ground_Reflected	*[unitless]*
IAM_Factor_on_Global	*[unitless]*
IAM_Factor_on_Beam	*[unitless]*
IAM_Factor_on__Diffuse	*[unitless]*

IAM_Factor_on_Ground_Reflected	*[unitless]*
Effective_Irradiance_in_POA	*[Wm^{-2}]*
Interrow_Albedo	*[unitless]*
Average_Ground_GHI	*[Wm^{-2}]*
IAM_Factor_on_Beam_Back	*[unitless]*
IAM_Factor_on_Beam_Diffuse_Back	*[unitless]*
Effective_Back_Diffuse_Irradiance	*[Wm^{-2}]*
Effective_Back_Front_Reflected_Irradiance	*[Wm^{-2}]*
Effective_Back_Ground_Reflected_Irradiance	*[Wm^{-2}]*
Effective_Back_Beam_Irradiance	*[Wm^{-2}]*
Effective_Back_Global_Irradiance	*[Wm^{-2}]*
ShowBackIrradianceProfile	*[Wm^{-2}]*
Back_Global_Irradiance_Inhomogeneity	*[unitless]*
Bifacial_Gain	*[Wm^{-2}]*
Array_Nominal_Power	*[kW]*
Array_Soiling_Loss	*[kW]*
Modules_Array_Mismatch_Loss	*[kW]*
Ohmic_Wiring_Loss	*[kW]*
Module_Quality_Loss	*[kW]*
Effective_Energy_at_the_Output_of_the_Array	*[kW]*
Calculated_Module_Temperature_deg_C	*[degrees]*
Difference_between_Module_and_Ambient_Temp._deg_C	*[degrees]*
PV_Array_Current	*[A]*
PV_Array_Voltage	*[V]*
Available_Energy_at_Inverter_Output	*[kW]*
AC_Ohmic_Loss	*[kW]*
Inverter_Efficiency	*[%]*
Inverter_Loss_Due_to_Low_Voltage_Threshold	*[kW]*
Inverter_Loss_Due_to_Low_Power_Threshold	*[kW]*
Inverter_Loss_Due_to_High_Power_Threshold	*[kW]*
Inverter_Loss_Due_to_High_Voltage_Threshold	*[kW]*
External_transformer_loss	*[kW]*
Power_Injected_into_Grid	*[kW]*
Energy_Injected_into_Grid	*[kWh)*
PV_Array_Efficiency	*[%]*
AC_side_Efficiency	*[%]*
Overall_System_Efficiency	*[%]*
Normalized_System_Production	*[unitless]*
Array_losses_ratio	*[unitless]*
Inverter_losses_ratio	*[unitless]*
AC_losses_ratio	*[unitless]*
Performance_Ratio	*[unitless]*
System_Loss_Incident_Energy_Ratio	*[unitless]*
Sub_Array_Performance	*[kW,V,A]*

4.3.3 Parameters

SimPVA	*Array of PVArray instances*
SimInv	*Array of Inverter instances*

SimACWiring	*Array of ACWiring instances*
SimTransformer	*Instance of Transformer class*
SimBackTilter	*Instance of BackTilter class*
SimSpectral	*Instance of SpectralEffects class*
SimGround	*Instance of GroundShading class*
SimHorizon	*Instance of HorizonShading class*
SimShading	*Instance of Shading class*

These parameters are used to model and simulate a grid-connected solar facility and calculate the related outputs. The list of outputs available is limited by the CASSYS Interface.

4.4 Ground Shading *GroundShading.cs*

This class calculates the shading effects on the beam and diffuse components of ground irradiance based on the sun position throughout the day. When determining shading, CASSYS is able to differentiate between single row and multi-row systems. The model used has been adapted from the bifacial irradiance model proposed by NREL [1].

4.4.1 Inputs

Sun Zenith	*Zenith angle of the sun (rad)*
Sun Azimuth	*Azimuth angle of the sun (rad)*
HDir	*Beam component of irradiance ($W \cdot m^{-2}$)*
HDif	*Diffuse component of irradiance ($W \cdot m^{-2}$)*

Tracking Configurations:

Clearance	*Distance from lowest point of array to ground (m)*
Panel Tilt	*Tilt of the modules installed relative to the ground (rad)*
Panel Azimuth	*Azimuth angle of the modules relative to True South (rad)*

4.4.2 Outputs

FrontGroundGHI	*Global horiz. irr. profile of patch of ground before row*
RearGroundGHI	*Global horiz. irr. profile of patch of ground behind row*
AveGroundGHI	*Avg. global irr. on patch of ground behind row (W/m^2)*

4.4.3 Parameters

Array BW	*Bandwidth of the array (m)*
Pitch	*Distance between consecutive rows of modules (m)*
Clearance	*Distance from lowest point of array to ground (m)*
Panel Tilt	*Tilt of the modules installed relative to the ground (rad)*
Panel Azimuth	*Azimuth angle of the modules relative to True South (rad)*
TransFactor	*Fraction of light that is transmitted through array*
NumGroundSegs	*Number of segments into which to divide patches of ground*
NumRows	*Number of PV rows in the system*

4.4.4 Equations and Model Descriptions

The ground shading model is supported for the unlimited row configuration as well as single axis tracking systems. The unlimited row configuration is an array configuration in which panels are placed on racking with rows of such racks placed behind one another. CASSYS considers the length of such rows to be large enough that any edge effects can be neglected in its shading model.

For both the beam and diffuse component, the amount of irradiance received by each segment of the ground is dependent on the panel tilt, pitch, and ground clearance.

4.4.4.1 Shading Model of the Beam Component

Calculating the effects of shading on the beam component involves the parameters, inputs, and measurements illustrated in Figure 4.

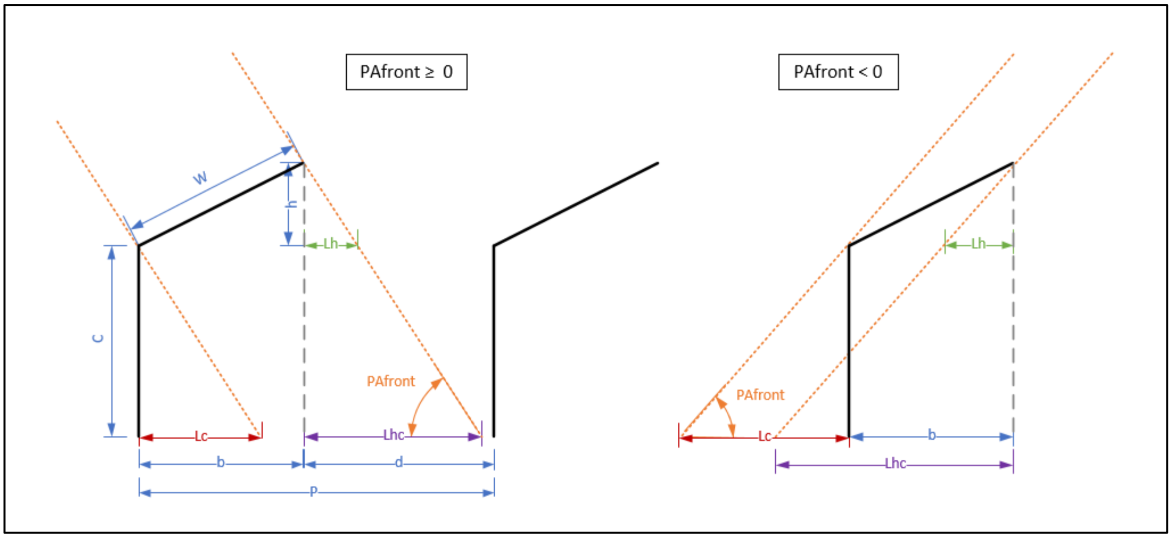


Figure 4: Ground Shadow Geometry

Using the known collector width (W) and row tilt (β), the panel height (h) and base (b) are found:

$$h = W \cdot \sin(\beta) \tag{12}$$

$$b = W \cdot \cos(\beta) \tag{13}$$

The three shadow lengths, Lh, Lc, and Lhc are calculated in equations (14)-(16) based on the current front profile angle of the sun from equation (139). When PA_{front} is positive, it indicates that the sun is in front of the panels, while negative PA_{front} indicates the sun is behind the panels. The lengths Lh, Lc, and Lhc will be positive or negative, accordingly.

$$Lh = \frac{h}{\tan(PA_{front})} \tag{14}$$

$$Lc = \frac{C}{\tan(PA_{front})} \tag{15}$$

$$Lhc = \frac{(h + C)}{\tan(PA_{front})} \tag{16}$$

These correspond to the projection of different heights of the beam irradiance onto the ground, and the values determine the shadowed area within the space from row to row. In particular, Lh provides the necessary checks for the shading limit angle of the panels. The pitch distance is divided into n ground segments and each is evaluated to be within a shadow segment or not. Equation (17) gives the formulation for ground shading (GS) values:

$$GS_i = \begin{cases} 1 & if\ segment\ i\ is\ shadowed \\ 0 & if\ segment\ i\ is\ not\ shadowed \end{cases} \quad for\ 1 \leq i \leq n \tag{17}$$

4.4.4.2 Shading Model of the Diffuse Component

Calculating the effect of shading on the diffuse component involves computing the sky view factors for any openings between panels that are seen by each segment of ground. The calculation for a single opening in the sky, for a single ground segment, is based on various cases of geometry as illustrated by Figure 5.

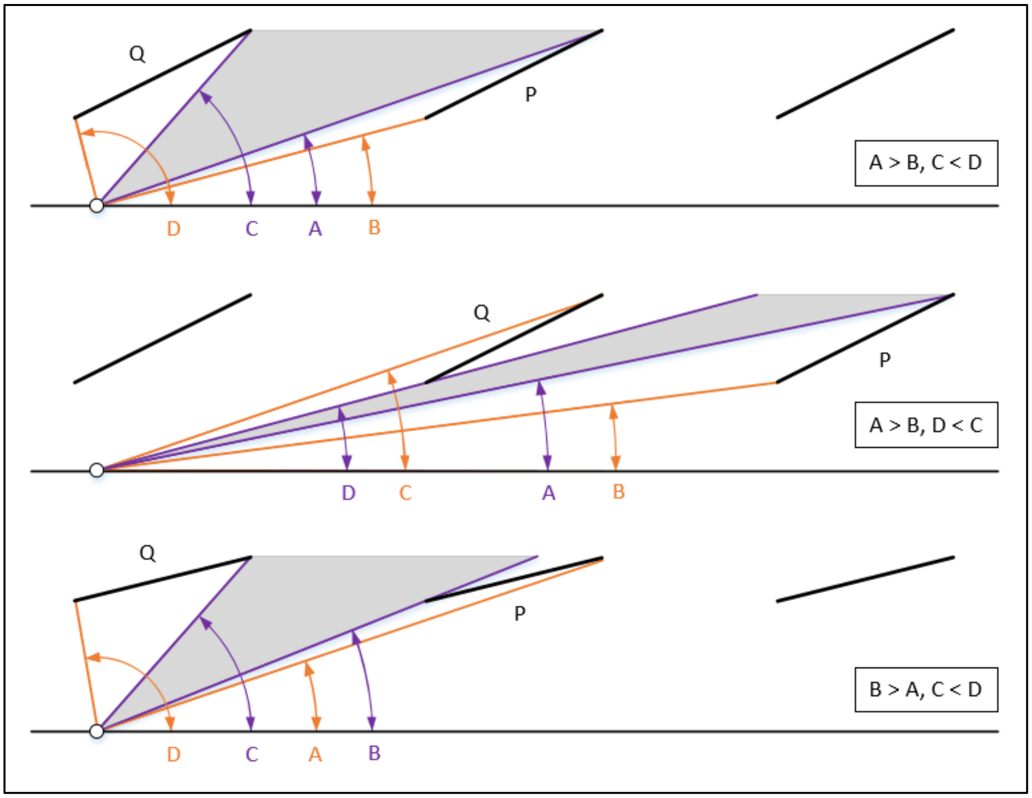


Figure 5: Sky View Factor Geometry

The view factor (VF_{ij}) is calculated for the j^{th} opening of sky that can be viewed by the i^{th} ground segment as per equations (18)-(20).

$$\beta_1 = \max(A, B) \tag{18}$$

$$\beta_2 = \min(\mathrm{C}, \mathrm{D}) \tag{19}$$

$$VF_{ij} = \frac{1}{2} \cdot (\cos\beta_1 - \cos\beta_2) \tag{20}$$

The model accounts for openings of sky ahead and behind each ground segment. More precisely, any sky opening where $\beta_2 > \beta_1$ and the contribution is greater than 1% of the total diffuse irradiance for that ground segment is counted in the sum. This yields the view factor sum for the i^{th} ground segment (VF_i) in equation (21). For more information regarding sky view factors, please see [1].

$$VF_i = \sum_{j=1}^{m} VF_{ij} \tag{21}$$

4.4.4.3 Shading Model of the Global Ground Irradiance

The ground shading and sky view factor values combined with the horizontal beam (H_B) and horizontal diffuse (H_D) irradiance components give an irradiance value for each segment of ground (G_i) in equation (22). The transmission factor (τ) accounts for a small amount of beam irradiance received by the ground, though shaded, through the panel. If the panels are completely opaque, $\tau = 0$.

$$G_i = \begin{cases} (H_D \cdot VF_i) + H_B & if\ GS_i = 0 \\ (H_D \cdot VF_i) + (H_B \cdot \tau) & if\ GS_i = 1 \end{cases} \tag{22}$$

4.5 Horizon Shading — *HorizonShading.cs*

This class calculates the effects of the horizon or far away objects on the beam, diffuse, and ground reflected components. The user can decide whether or not to define a horizon, and if a horizon is defined, up to 360 points can be used to define it. If the horizon is defined, a factor is applied to the beam, diffuse, and ground reflected irradiance values respectively.

4.5.1 *Inputs*

Sun Zenith	*Zenith Angle of the Sun (rad)*
Sun Azimuth	*Azimuth Angle of the Sun (rad)*
POA Beam Irradiance	*Beam component of POA irradiance ($W \cdot m^{-2}$)*
POA Diffuse Irradiance	*Diffuse component of POA irradiance ($W \cdot m^{-2}$)*
POA Ground-Reflected Irr.	*Ground reflected component of POA irradiance ($W \cdot m^{-2}$)*
Plane of Array Tilt	*The tilt of the modules (rad)*
Plane of Array Azimuth	*The azimuthal orientation of the modules (rad)*
Horizon Azimuth Definition	*The array of azimuth points of the horizon definition (rad)*
Horizon Elevation Definition	*The array of elevation points of the horizon definition (rad)*

4.5.2 *Outputs*

Horizon Shaded Beam POA Irradiance	*The POA beam irradiance with horizon factor applied ($W \cdot m^{-2}$)*
Horizon Shaded Diffuse POA Irradiance	*The POA diffuse irradiance with horizon factor applied ($W \cdot m^{-2}$)*
Horizon Shaded Ground Reflected POA Irradiance	*The POA ground reflected irradiance with horizon factor applied ($W \cdot m^{-2}$)*

4.5.3 Equations and Model Descriptions

4.5.3.1 Interpolation of Horizon

The horizon is defined by a series of azimuthal angles and their corresponding elevation angles. The interpolation between azimuthal angles is done using the *Linear(…)* method from the *Interpolate* class (section 5.3).

4.5.3.2 Calculate Beam Factor

The horizon shading factor on the beam portion of the irradiance is either a 1 or a 0 based on whether the sun is above or below the horizon. There is a tolerance of 0.25° as the angular diameter of the solar disk in the sky is 0.5°. If the sun's zenith position is more than that tolerance below the horizon elevation value, the beam irradiance is 0.

4.5.3.3 Calculation of Diffuse Factor

The diffuse factor can be calculated as the ratio of the fraction of the sky that is visible to a tilted solar panel with a far shading horizon to the sky visible to a tilted panel with no far shading horizon.

$$Diffuse\ Factor = \frac{ViewFactor_{tilted,horizon}}{ViewFactor_{tilted,no\ horizon}} \tag{23}$$

Since the horizon is normally relatively low it is more convenient to calculate $ViewFactor_{tilted,horizon}$ as the difference between the view factor in the absence of horizon and the view factor of the sky that is below the horizon:

$$Diffuse\ Factor = \frac{ViewFactor_{tilted,no\ horizon} - ViewFactor_{tilted,below\ horizon}}{ViewFactor_{tilted,no\ horizon}} \tag{24}$$

The derivation of the view factor of the visible sky on an oriented surface requires a spherical coordinate system to simplify the calculations. The shading horizon lies at such a distance that the physical size of the solar array is negligible, further simplifying the calculations. The View Factor of the sky with no shading onto a tilted panel has a known value (see [3], p. 95), shown below:

$$ViewFactor_{tilted,no\ horizon} = \frac{1 + \cos(\beta)}{2} \tag{25}$$

where β is the slope of the panel. The View Factor, from the solar surface, of the sky hidden by the horizon is given by the equation below (see [4], ch. 13]:

$$ViewFactor = \frac{1}{\pi} \iint \cos(\alpha) \sin(\theta)\, d\theta d\phi \tag{26}$$

where the integration is made over all directions of the solar dome hidden from the surface by the horizon, represented by their azimuth angle ϕ and their zenith angle θ. Angle α is the incidence angle of the solar rays coming from a particular direction onto the surface. α is a trigonometric combination of the angular coordinates of the system and the orientation of the surface ([3], p. 15):

$$\cos(\alpha) = \cos(\theta)\cos(\beta) + \sin(\theta)\sin(\beta)\cos(\phi - \gamma) \quad \textbf{(27)}$$

where γ is the azimuth of the panel. Using the above expression for α, the integral can be expressed in terms of the integrating variables and the orientation constants of the surface. Replacing the $\cos(\alpha)$ in Eq. with the expanded expression in Eq. , the integral expression for the view factor can be rewritten and rearranged:

$$ViewFactor = \frac{1}{\pi}\left[\cos(\beta)\iint \cos(\theta)\sin(\theta)\,d\theta d\phi + \sin(\beta)\iint \sin^2(\theta)\cos(\phi-\gamma)\,d\theta d\phi\right] \quad \textbf{(28)}$$

The amount of sky "hidden" from the panel is different when one is integrating the region "in front" of the panel surface compared to when integrating the region "behind" the panel surface. The reason for this is that the panel itself acts as a "horizon" that limits the sky that can be viewed from its surface. For any azimuth angle that hits the plane of the panel (as opposed to azimuth angles that sweep the space in front of it) the portion of the sky that is hidden from the panel is restricted to zenith angles between the horizon and a limiting angle θ_{limit} calculated from the following expression:

$$\cot(\theta_{limit}) = \tan(\beta)\cos(\gamma - \phi) \quad \textbf{(29)}$$

Because θ_{limit} is dependent on the azimuthal coordinate, ϕ, a numerical computation of the integrals will be necessary. The remaining "shading" View Factor is calculated from the previously shown integral between the following bounds:

$$\phi = [\gamma - \pi, \gamma + \pi]$$

$$\theta = \left[\frac{\pi}{2} - \beta_{horizon}, \theta_{limit}\right] when \left(\frac{\pi}{2} - \beta_{horizon}\right) < \theta_{limit}$$

where $\beta_{horizon}$ is the elevation angle of the horizon. Defining $\theta_{horizon} = \frac{\pi}{2} - \beta_{horizon}$ and substituting these bounds into the View Factor integral yields:

$$ViewFactor = \frac{1}{\pi}\left[\cos(\beta)\left\{\int_{\gamma-\pi}^{\gamma+\pi}\int_{\theta_{horizon}}^{\theta_{limit}} \cos(\theta)\sin(\theta)\,d\theta d\phi\right\} + \sin(\beta)\left\{\int_{\gamma-\pi}^{\gamma+\pi}\int_{\theta_{horizon}}^{\theta_{limit}} \sin^2(\theta)\cos(\phi-\gamma)\,d\theta d\phi\right\}\right] \quad \textbf{(30)}$$

From the expanded integral form, the two components that need to be solved are the $\cos(\theta)\sin(\theta)$ section and the $\sin^2(\theta)\cos(\phi - \gamma)$ section. The integration of these types of integrals is shown below:

$$\int_{\gamma-\pi}^{\gamma+\pi} \int_{\theta_{horizon}}^{\theta_{limit}} \cos(\theta)\sin(\theta)\,d\theta d\phi$$

Using a substitution for the Double Sine identity, $\sin(2\theta) = 2\sin(\theta)\cos(\theta)$, the first integral can be rewritten:

$$\int_{\gamma-\pi}^{\gamma+\pi} \int_{\theta_{horizon}}^{\theta_{limit}} \cos(\theta)\sin(\theta)\,d\theta d\phi$$
$$= \int_{\gamma-\pi}^{\gamma+\pi} \int_{\theta_{horizon}}^{\theta_{limit}} \frac{1}{2}\sin(2\theta) d\theta d\phi$$
$$= \int_{\gamma-\pi}^{\gamma+\pi} \left[-\frac{1}{4}\cos(2\theta)\right]_{\theta_{horizon}}^{\theta_{limit}} d\phi$$
$$= \frac{1}{4}\int_{\gamma-\pi}^{\gamma+\pi} \cos(2\theta_{horizon}) - \cos(2\theta_{limit})\,d\phi$$

Moving on to the $\sin^2(\theta)\cos(\phi-\gamma)$ integral, the reduction formula for power of sines is used to simplify the solution.

$$\int \sin^2(\theta) d\theta = \frac{\theta - \cos(\theta)\sin(\theta)}{2} = \frac{\theta}{2} - \frac{\sin(2\theta)}{4}$$

This expression is substituted into the integral:

$$\int_{\gamma-\pi}^{\gamma+\pi} \int_{\theta_{horizon}}^{\theta_{limit}} \sin^2(\theta)\cos(\phi-\gamma)\,d\theta d\phi$$
$$= \int_{\gamma-\pi}^{\gamma+\pi} \cos(\phi-\gamma)\,d\phi \left[\frac{\theta}{2} - \frac{\sin(2\theta)}{4}\right]_{\theta_{horizon}}^{\theta_{limit}}$$
$$= \frac{1}{4}\int_{\gamma-\pi}^{\gamma+\pi} \cos(\phi-\gamma)\,[2(\theta_{limit}-\theta_{horizon}) + \sin(2\theta_{horizon}) - \sin(2\theta_{limit})] d\phi$$

Finally the complete expression for the view factor is:

$$ViewFactor = \frac{1}{\pi}\left[\cos(\beta)\left\{\frac{1}{4}\int_{\gamma-\pi}^{\gamma+\pi}\cos(2\theta_{horizon}) - \cos(2\theta_{limit})\,d\phi\right\}\right.$$
$$\left. + \sin(\beta)\left\{\frac{1}{4}\int_{\gamma-\pi}^{\gamma+\pi}\cos(\phi-\gamma)\,[2(\theta_{limit}-\theta_{horizon}) + \sin(2\theta_{horizon}) - \sin(2\theta_{limit})]d\phi\right\}\right] \tag{31}$$

This integral is then evaluated numerically, summing the results for all values of ϕ where the "horizon" created by the tilt of the panel is greater than $\theta_{horizon}$. This then forms the value of $ViewFactor_{tilted,below\ horizon}$ and is subbed in to eq. , giving the final diffuse factor.

4.5.3.4 *Calculate Ground Reflected Factor*

The ground reflected factor is calculated as a combination of the diffuse and beam irradiance factors. As will be seen later in equation (116) the ground reflected irradiance is generally assumed to be proportional to the global horizontal irradiance. To calculate the ground reflected factor, the global horizontal irradiance is split into its beam and diffuse components, then the beam factor and diffuse factor calculations are calculated with a tilt of 0, as they are with reference to the ground. The factors are then applied to the beam and diffuse horizontal irradiance values. The ground reflected factor is then the fraction of the sum of the beam and diffuse horizontal irradiance values multiplied by the respective horizon factors over the sum of the unaffected beam and diffuse horizontal values.

$$Ground\ Reflected\ Factor = \frac{(H_D * DiffFactor + H_B * BeamFactor)}{(H_D + H_B)} \tag{32}$$

4.6 Inverter — *Inverter.cs*

An Inverter converts DC power from a Photovoltaic (PV) Array to AC power, which is then transferred to the grid.

4.6.1 *Inputs*

$DCPwr_{In}$	*Input power from the photovoltaic array (kW)*
V_{in}	*Voltage dictated by the Inverter (V)*

4.6.2 *Outputs*

$ACPwr_{OUT}$	*Output power from the photovoltaic array (kW)*
V_{InDC}	*Voltage dictated by the Inverter (V)*
Efficiency	*Efficiency of the Inverter*
Losses	*Losses from the Inverter (Efficiency related)*
I_{Out}	*Current produced (A, AC Single Phase)*
ON	*Boolean - if the inverter is ON or OFF*
Bipolar	*Boolean - if the inverter uses bipolar inputs or not*
Clipping	*Boolean - if the inverter is currently demonstrating power limitation behaviour*

4.6.3 Parameters

PNom AC	*Nominal AC Power delivered by the Inverter (can be changed to reflect de-rating)*
MPPT Tracking	*Boolean - specifies If the inverter performs max. power point tracking, or if the inverter is operating in fixed-voltage mode*
MPPT Window Min. Voltage	*Minimum value for MPPT (V) Window*
MPPT Window Max. Voltage	*Maximum value for MPPT (V) Window*
Num. Inverters	*Number of Inverters in the Sub-Array*
Threshold Power	*Power required to turn the Inverter ON*
Min. Voltage	*Min. DC-side voltage required to turn the Inverter ON (user defined)*
Max. Voltage	*Max. DC-voltage specification of the Inverter*
Output Voltage	*Output Voltage of the Inverter (dictated by the Grid or step-up transformer)*
OutputPhases	*Number of phases (AC) at inverter output*
Three Efficiency Curves	*Boolean - denotes either Single or Three Efficiency Curves*
Low Voltage	*Voltage threshold for low voltage efficiency curve [V]*
Med Voltage	*Voltage threshold for medium voltage efficiency curve [V]*
High Voltage	*Voltage threshold for high voltage efficiency curve [V]*
LowEff	*Jagged Array[P_{in}][P_{AC}], Low Voltage Efficiency Curve*
MedEff	*Jagged Array [P_{in}][P_{AC}], Medium Voltage Efficiency Curve*
HighEff	*Jagged Array [P_{in}][P_{AC}], High Voltage Efficiency Curve*
Single Efficiency Curve	*Jagged Array [P_{in}][P_{AC}], used if Three Efficiency Curve is false*
Nom. Output Power	*Nominal Output Power (W AC)*
Wiring Resistance	*AC wiring resistance(Ω) translated from STC % Loss*
Threshold Power	*Minimum power required to turn the Inverter ON given sufficient Voltage (Min. Voltage)*

These parameters are used to simulate the behaviour and output of the inverter. This information can be exported from an .OND file to the CASSYS database, or by using the manufacturer's datasheet in conjunction with the "Add an Inverter" button in the database (see CASSYS Interface User Manual).

4.6.4 Equations and Model Description

Inverters are responsible for the conversion of DC power into AC power that is transferred to the grid. To achieve this in a safe and controlled manner, the inverter controls the voltage of the PV array based on its operation mode. The inverter must first determine if it has sufficient voltage and power to turn on. Hence the ON or OFF state of the inverter is determined first:

4.6.4.1 Determining ON or OFF State of the Inverter

The following flow chart (Figure 6) summarizes the process to determine if the Inverter has sufficient voltage and Power to turn ON.

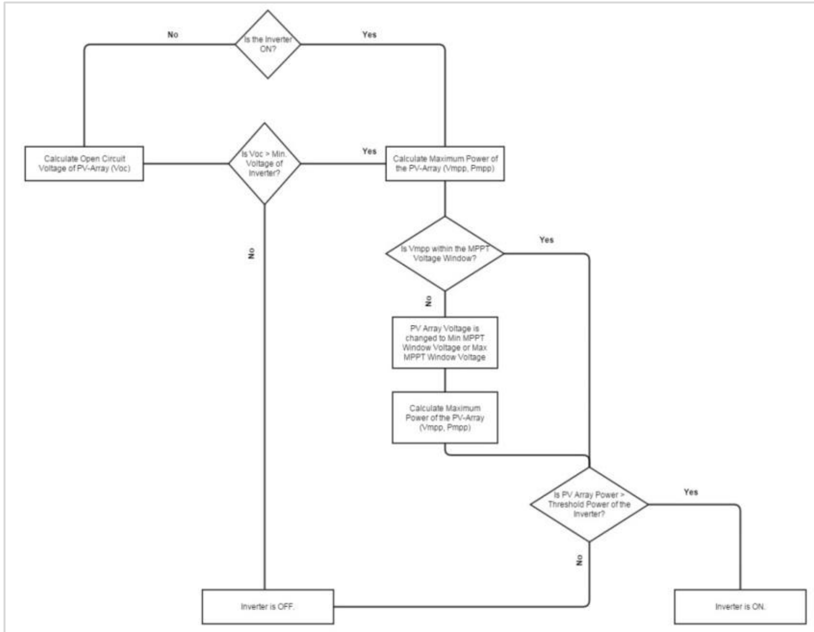


Figure 6: Determining the ON or OFF State of the Inverter

4.6.4.2 DC-Side Voltage Control

The DC-side voltage is controlled by the Inverter based on the operation mode of the Inverter. The impact on DC-side voltage for each operation mode is explained below. For all of these cases, the voltage from the array is divided by two if the inverter uses bipolar inputs.

If the Inverter is OFF

Based on Figure 6, if the Inverter is OFF resulting from insufficient voltage from the PV Array (i.e. V_{OC} < Min. Voltage of Inverter), the Inverter will report the V_{OC} as the DC-side voltage. If the Inverter has sufficient voltage to turn ON, a maximum power point tracking (MPPT) voltage determination is made. This is explained in the next section.

Maximum Power Point Tracking Voltage

If the Inverter has sufficient voltage to turn on, the maximum voltage (V_{MPP}) for given meteorological conditions is calculated. If the V_{MPP} is within the MPPT Voltage Window (V_{MPPmin}, V_{MPPmax}) the DC-side voltage remains at V_{MPP}. If $V_{MPP} < V_{MPPmin}$ or $V_{MPP} > V_{MPPmax}$, the voltage of the DC-array is shifted to V_{MPPmin} or V_{MPPmax} respectively; this value is then used to calculate the power produced by the array.

If $V_{MPP} < V_{MPPmin}$ and the inverter turns off or remains off, the produced power by the array is then counted as a loss and is equal to the array Pmpp. On the other hand if $V_{MPP} > V_{MPPmax}$ the adjusted voltage causes a loss in power equal to the difference between the array Pmpp and the actual array power.

Power Limiting (or Clipping) Mode

If the calculated output power exceeds the PNom AC of the Inverter for a given input power, the limitation behaviour of the Inverter is activated. The Inverter decreases the power of the PV array to achieve an output power that does not exceed PNom AC; it increases the voltage of the PV Array and the corresponding efficiency is then re-calculated (as long as it does not exceed the maximum voltage of the inverter). The final adjusted PV Array voltage is the voltage at which the Inverter limits power output to its nominal value. The voltage for which the input power is sufficiently reduced to no longer activate the power limitation mode is found using the bisection method along the PV array's Power-Voltage curve.

In decreasing the power of the array there is loss in possible power attained from the array which is then accounted for in loss due to high power threshold as the difference between the potential power if no clipping and the adjusted power. In the opposing case, if the array is producing power, but it is an insufficient amount to turn on the inverter, then the power produced is not converted to usable power. This is accounted for in the loss due to minimum power threshold and is equal to the calculated Pmpp.

4.6.4.3 Modelling Efficiency

Inverter efficiency is the ratio of the AC output power (P_{OUT}) and the DC input power (P_{IN}). An inverter's efficiency varies based on input power, the DC voltage, and the inverter temperature (heat related de-rating of Inverters is commonly observed).

The efficiency curve below is provided for some inverters based on the California Energy Commission (CEC) which was adopted from the Sandia and BEW (now DNV-KEMA) test protocol. For more information regarding the test protocol and a list of test results for various inverters, please see [5].

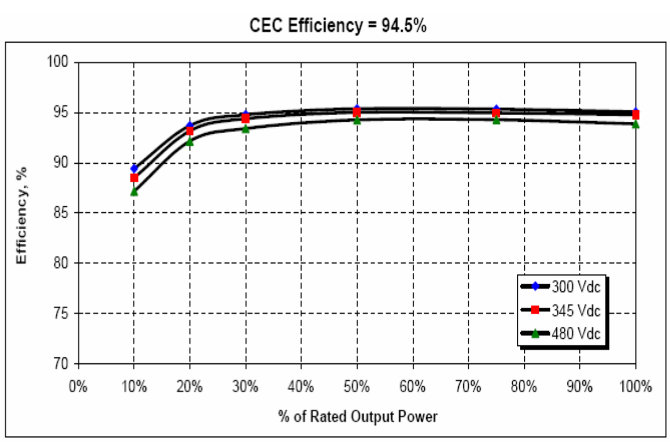


Figure 7: Voltage dependent Efficiency Curve for a sample Inverter [5]

The inverter efficiency curves used in CASSYS are defined by the efficiency measured at 8 different input power points. The inverter output is determined by using a single or a voltage-dependent

(8-points each for three voltage levels: Low, Medium and High) efficiency curve based on the information available in the database.

For Single 8-point Efficiency Curve Inverters:

The curve is first translated to a $P_{OUT} = f(P_{IN})$ array using the efficiency values given. The efficiency value for a given input power is then calculated using linear interpolation (see 5.3.1.1).

For Voltage-Dependent Efficiency Curve Inverters:

For a given P_{IN}, the resultant efficiency value from the efficiency curve of each voltage level is calculated using linear interpolation similar to the previous case. This information is then used to create a voltage level vs efficiency curve (at given P_{IN}). The operating voltage of the PV array is then used as the interpolant to determine the efficiency level from the new curve using quadratic interpolation (see 5.3.1.2).

4.7 Loss Diagram — ***LossDiagram.cs***

The Loss Diagram class is responsible for the summation of all instantaneous losses caused by a grid connected system as well as significant intermediate values. It is also responsible for outputting these values to the interface, where they are turned into a percentage and are graphed to give a visual representation of the system losses.

4.7.1 Inputs

At the irradiance level	
Horizontal_Global_Irradiance	*Instantaneous horiz. global radiation ($W{\cdot}m^{-2}$)*
Global_Irradiance_in_Array_Plane	*Instantaneous global radiation in POA ($W{\cdot}m^{-2}$)*
FarShading_Global_Loss	*Instantaneous horizon shading losses ($W{\cdot}m^{-2}$)*
Near_Shading_Loss_for_Global	*Instantaneous near shading losses ($W{\cdot}m^{-2}$)*
Radiation_Soiling_Loss	*Instantaneous soiling losses ($W{\cdot}m^{-2}$)*
Incidence_Loss_for_Global	*Instantaneous incidence angle loss ($W{\cdot}m^{-2}$)*
Bifacial_Gain	*Instantaneous bifacial gain ($W{\cdot}m^{-2}$)*
Radiation_Spectral_Loss	*Instantaneous spectral correction gain($W{\cdot}m^{-2}$)*
Effective_Irradiance	*Instantaneous effective radiation ($W{\cdot}m^{-2}$)*
PV array level	
Array_Nominal_Power	*Instantaneous nominal array power (kWh)*
Power_Loss_Due_to_Temperature	*Instantaneous loss due to temperature (kWh)*
Energy_Loss_Due_to_Irradiance	*Instantaneous loss due to radiation (kWh)*
Module_Quality_Loss	*Instantaneous module quality loss (kWh)*
Module_LID_Loss	*Instantaneous loss due to LID (kWh)*
Module_Ageing_Loss	*Instantaneous loss to module ageing (kWh)*
Modules_Array_Mismatch_Loss	*Instantaneous mismatch loss (kWh)*
Ohmic_Wiring_Loss	*Instantaneous DC wiring loss (kWh)*
Inverter level	
Virtual_Inverter_Input_Energy	*Instant. virtual inverter input power (kWh)*

Inverter_Loss_Due_to_Low_Power_Threshold	*Instant. loss due to low power limit (kWh)*
Inverter_Loss_Due_to_High_Power_Threshold	*Instant. loss due to high power limit (kWh)*
Inverter_Loss_Due_to_Low_Voltage_Threshold	*Instant. loss due to low voltage limit (kWh)*
Inverter_Loss_Due_to_High_Voltage_Threshold	*Instant. loss due to high voltage limit (kWh)*
Effective_Energy_at_the_Output_of_the_Array	*Instant. actual energy at inverter input (kWh)*
DCAC_Conversion_Losses	*Instant. DC to AC conversion losses (kWh)*
Available_Energy_at_Inverter_Output	*Inst. power available at inverter output (kWh)*
Transformer level	
AC_Ohmic_Loss	*Instant. AC wiring loss (kWh)*
NightTime_Energizing_Loss	*Instant. night time transformer loss (kWh)*
External_transformer_loss	*Instant. day time transformer loss (kWh)*
Power_Injected_into_Grid	*Instant. power to the grid (kWh)*

4.7.2 Outputs

At the irradiance level	
Horizontal_Global_Radiation	*Total horizontal global radiation ($W \cdot m^{-2}$)*
Global_Radiation_in_POA	*Total global radiation in POA ($W \cdot m^{-2}$)*
Horizon_Shading_Losses	*Total horizon shading losses ($W \cdot m^{-2}$)*
Near_Shading_Losses	*Total near shading losses ($W \cdot m^{-2}$)*
Soiling_Losses	*Total soiling losses ($W \cdot m^{-2}$)*
Incidence_Angle_Losses	*Total incidence angle loss ($W \cdot m^{-2}$)*
Bifacial_Gain	*Total effective bifacial gain ($W \cdot m^{-2}$)*
Spectral_Losses	*Total spectral effects losses ($W \cdot m^{-2}$)*
Effective_Radiation	*Total effective radiation ($W \cdot m^{-2}$)*
PV array level	
PV_Array_Nominal_Energy	*Total nominal array power (kWh)*
Energy_Loss_Due_to_Temperature	*Total loss due to temperature (kWh)*
Energy_Loss_Due_to_Irradiance_Level	*Total loss due to radiation (kWh)*
Module_Quality_Losses	*Total module quality loss (kWh)*
Module_LID_Losses	*Total loss due to LID (kWh)*
Module_Ageing_Losses	*Total loss due to ageing (kWh)*
Mismatch_Losses	*Total mismatch loss (kWh)*
DC_Wiring_Losses	*Total DC wiring loss (kWh)*
Inverter level	
Virtual_Inverter_Input	*Total virtual inverter input power(kWh)*
Energy_Lost_to_Input_Power_too_Low	*Total loss due to low power limit(kWh)*
Energy_Lost_to_Input_Power_too_High	*Total loss due to high power limit (kWh)*
Energy_Lost_to_Input_Voltage_too_Low	*Total loss due to low voltage limit (kWh)*
Energy_Lost_to_Input_Voltage_too_High	*Total loss due to high voltage limit (kWh)*
Actual_Inverter_Input_Energy	*Total actual energy at inverter input (kWh)*
DCAC_Conversion_Losses	*Total DC to AC conversion losses (kWh)*
Inverter_Output	*Total power available at inverter output (kWh)*
Transformer level	
AC_Wiring_Losses	*Total AC wiring loss (kWh)*

Night-Time_Energization_Losses	*Total night time transformer loss (kWh)*
External_Transformer_Losses	*Total day time transformer loss (kWh)*
Power_Injected_into_Grid	*Total power to the grid (kWh)*

4.7.3 Parameters

The parameters used in this class are equivalent to those used in the Grid Connected Systems class as all of the above inputs are first calculated in the Grid Connected System class.

4.7.4 Equations and Modelling Descriptions

4.7.4.1 Soiling Loss

Soiling loss is the loss of irradiance on the panels due to dirty or snow-covered panels (this loss is counted as an irradiance loss, rather than a panel loss, because it happens before the irradiance reaches the front surface of the panels). The soiling calculation is based on a user input and can be seen modelled in the following equation:

$$L_{Soiling} = H_{IAM} \times L_{Soil\,,\%} \tag{33}$$

H_{IAM}: IAM global tilted irradiance
$L_{Soil\,,\%}$: Soiling loss in percent (user input)

4.7.4.2 Loss Due to Temperature

Loss due to temperature is the loss of potential power output created by the module due to cell efficiencies at varying temperatures.

$$L_{Temp} = -\left(\frac{\alpha}{100}\right)(T_{mod} - 25)\left(P_{DC,nom} \times \frac{H_{T,eff}}{1000}\right) \tag{34}$$

α: mPmpp, i.e. maximum power temperature coefficient, defined for the module (%/K)
T_{mod}: Temperature of the module either based on an input measurement or in reference to the cell temperature shown calculated in equation (43).
$P_{DC,nom}$: Nominal DC power produced by the PV array
$H_{T,eff}$: Effective irradiance available for power conversion in POA, this specific calculation can be seen in equation (41).

4.7.4.3 Loss Due to Radiation

Loss due to radiation is the loss of potential output power due to varying cell efficiencies at varying levels of irradiance.

$$L_{Rad} = \left(P_{DC,nom} \times \frac{H_{T,eff}}{1000}\right) - L_{Temp} - P_{MPP} \tag{35}$$

$P_{DC,nom}$: Nominal DC power produced by the PV array
$H_{T,eff}$: Effective irradiance available for power conversion in POA, this specific calculation can be seen in equation (41).

L_{Temp}: Loss due to temperature shown in the equation (34).
P_{mpp}: Maximum power the module can produce [W]

4.7.4.4 Virtual Inverter Energy

Virtual energy at the inverter input is essentially the available power output at the array after losses. The system models this value using the following equation:

$$E_{Virt} = P_{DC,nom} - L_{Rad} - L_{Soiling} - L_{Temp} - L_{modquality} - L_{LID} - L_{Ageing} - L_{mismatch} - L_{ohmic} \quad (36)$$

OR

$$E_{Virt} = P_{DC,eff} + L_{P,High} + L_{P,Low} + L_{V,High} + L_{V,Low} \quad (37)$$

$P_{DC,nom}$: Nominal DC power produced by the PV array
$L_{Soiling}$: Radiation loss due to soiling
L_{Temp}: Radiation loss due to module temperature
$L_{modquality}$: Module quality loss (53)
L_{LID}: Module light-induced degradation (LID) loss (54)
L_{Ageing}: Module ageing loss (55)
$L_{mismatch}$: Losses as a result of mismatch (57)
L_{ohmic}: DC wiring losses (59)

$P_{DC,eff}$: Effective power at the output of the array:

$$P_{DC,eff} = P_{DC,Out} - L_{modquality} - L_{LID} - L_{Ageing} - L_{mismatch} - L_{ohmic} \quad (38)$$

$P_{DC,Out}$: Output power of array based on radiation hitting the solar cells
$L_{P,High}$: Loss due to clipping
$L_{P,Low}$: Loss due to power below minimum inverter power threshold
$L_{V,High}$: Loss due to array voltage too high
$L_{V,Low}$: Loss due to array voltage below minimum inverter voltage threshold

Threshold losses are explained briefly above in *Inverter* section.

4.8 Photovoltaic Array ***PVArray.cs***

The PV Array Class evaluates the performance of a solar module using the "standard" or one-diode model as described in Equations and Model Description 4.8.4. The STC condition parameters for the module are obtained from module data-sheets or a .PAN file. Module behaviour is calculated for a number of non-STC operating conditions such as open circuit, fixed voltage, and maximum point tracking. Values are then converted from Module to Array level and losses are applied in accordance with user input values.

4.8.1 Inputs

Shaded POA Beam Component	*Beam component of plane of array irradiance ($W \cdot m^{-2}$)*
Shaded POA Diffuse Component	*Diffuse component of plane of array irradiance ($W \cdot m^{-2}$)*
Shaded POA Ground-Reflected Component	*Ground reflected component of plane of array irradiance ($W \cdot m^{-2}$)*
Incidence Angle	*Angle of Incidence of the Beam Component (Radians)*
Wind Speed	*Wind Speed ($m \cdot s^{-1}$) if provided by user*
Measured Module Temperature	*(°C) if provided by user*
Month of the Year	*For Soiling Percentage to be applied on irradiance*

4.8.2 Outputs

Effective Global and Components of POA Irradiance	*Global, Beam, Diffuse and Ground reflected irradiance in POA after gains and losses have been applied ($W \cdot m^{-2}$)*
Bifacial Gain	*Effective irradiance gained due to backside collection ($W \cdot m^{-2}$)*
RadEff	*Total irradiance available to be converted into power ($W \cdot m^{-2}$)*
TDifRef	*Diffuse irradiance that is reflected from front of panel ($W \cdot m^{-2}$)*
Vout	*Voltage produced by (MPP mode) or dictated by the Inverter (Fixed-Voltage mode) of the array (V)*
Iout	*Current produced by the array under MPP or Fixed-Voltage Mode (A)*
Pout	*Power produced by the DC Array (W)*
TModule	*Temperature of the array (Calculated/Measured based, °C)*
Soiling Loss	*Power lost due to soiling of the array (W)*
Mismatch Loss	*Power lost to mismatch of module I-V curves in the array (W)*
Module Quality Loss	*Power lost to difference between actual and nameplate rating of modules (W) [can be positive or negative]*
Module LID Loss	*Power lost to light induced degradation [LID] (W)*
Module Ageing Loss	*Power lost to ageing of modules (W)*
Ohmic Losses	*Power lost to resistance in the wires from array to inverter (W)*
Nominal DC Array Power	*Power of the array under STC conditions*

4.8.3 Parameters

Due to the contextual and extensive nature of the parameters required for the modelling of the PV Array, the description of each variable is presented alongside the appropriate equation.

4.8.4 Equations and Model Description

A photovoltaic module is a non-linear DC electrical device which converts the energy of light into electricity by the photovoltaic effect. An equivalent circuit model, the single diode model (SDM), is extensively used to determine the performance of a non-ideal solar cell under various illumination and temperature conditions. Figure 8 shows the equivalent circuit for the model.

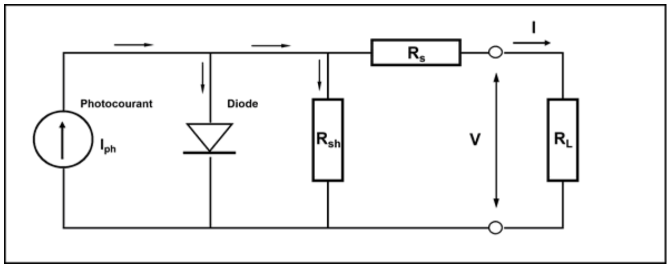


Figure 8: Equivalent circuit for the Single Diode Model (SDM) of a Solar Cell

I-V characteristics of the solar module can be completely described by equation (39).

$$I = I_{\Phi} - I_0 \left[e^{q\frac{(V+IR_s)}{N_{cells}\gamma k T_c}} - 1 \right] - \frac{V + IR_s}{R_{sh}} \quad \textbf{(39)}$$

Where,

I_{Φ}: Photocurrent of the module [A]

I_0: Reverse saturation current of the module [A]

R_s: Series Resistance of the module [Ω]

R_{sh}: Shunt Resistance of the module [Ω]

γ: The diode ideality factor

T_c: Temperature of the cell [K]

N_{cells}: Number of cells in the module

V: Operating voltage of the module [V]

I : Current produced by the module [A]

q: Elementary Charge [C], Constant: $1.602 \times 10^{-19} C$

Parameters of the module are obtained from the module database provided as part of the user interface. For details on adding modules that are not in the database, please see the CASSYS User Manual. The configuration of the panels occurs in the *void Config(..)* method, which then determines the missing parameters required to simulate array performance using the SDM equivalent circuit (see 4.8.5).

4.8.5 SDM Parameter Determination *void CalcGammaIPhiIrsRef(..)*

The first five parameters listed above depend upon the type of module selected for simulation. The only available information to determine these parameters is the I-V curve obtained under standard test conditions (STC). Since the modules do not always operate under STC conditions, the unknown parameters are solved for operation at reference conditions (lending a *Ref* subscript to each parameter).

The values for $R_{series,Ref}$ and $R_{sh,Ref}$ are obtained directly from the database included in CASSYS and can also be specified from other data sources (e.g. .PAN files for modules).

The remaining three variables $(I_{0,Ref}, I_{\Phi,Ref}, \gamma_{Ref})$ are determined by solving the SDM equation for the three known operating points of the module at STC, i.e. at open-circuit (V_{OC}, 0), at short-circuit (0, I_{SC}), and the maximum power point (V_{MPP},I_{MPP}). These yield three equations and three unknowns which are easily reduced to a function of γ_{Ref} through substitution and elimination. Once γ_{Ref} is obtained using the Newton-Raphson root-finding algorithm, it is substituted back into the previous equations and a solution for each parameter $(I_{0,Ref}, I_{\Phi,Ref}, \gamma_{Ref})$ is obtained. For more details see 4.8.11.

The dependence on temperature and irradiance of each of these factors is discussed in a later section. First, the method used to determine the effective irradiance reaching a panel and the cell temperature model are discussed.

4.8.6 Effective Irradiance on Panel Surface *void CalcEffectiveIrradiance(..)*

The effective irradiance available for electricity conversion is the plane-of-array irradiance after incidence angle effects and soiling have been taken into account, bifacial gain (if any) has been added, and spectral losses have taken effect. It is therefore:

$$G_{bi} = B_{glo} \cdot biFactor \tag{40}$$

$$H_{T,eff} = \left[\left(IAM_{dir} \cdot H_{T,dir} + IAM_{dif} \cdot H_{T,dif} + IAM_{ref} \cdot H_{T,ref}\right) \cdot (1 - L_{soil}) + G_{bi}\right] \cdot \left(1 - L_{spec}\right) \tag{41}$$

Where,

B_{glo} : global irradiance on back side of array [W/m^2] (see 4.2.4.6),

$biFactor$: module bifaciality factor, expressed as a percentage,

G_{bi} : bifacial gain [W/m^2],

IAM_x : Incidence Angle Modifier [unitless], with the subscript *x* indicating the direct, diffuse and reflected components of irradiance (see section 4.8.7),

$H_{T,x}$: plane of array irradiance [W/m^2], with the subscript *x* indicating the direct, diffuse and reflected components, after shading has been taken into account,

L_{soil} : soiling loss for the current month, expressed as a fraction (see 4.8.8),

L_{spec} : spectral effects loss (see 4.12.4.1)

4.8.7 Incidence Angle Modifier (IAM)

4.8.7.1 ASHRAE Model:

The incidence angle of the beam component of the irradiance affects the amount of light reaching the cell due to reflective losses. ASHRAE has proposed a convenient one parameter method of describing the efficiency of transmission for any incidence angle of incoming light [6]. This parameter (b_0) is specified by the user (based on module-specific measurements) and the Incidence Angle Modifier (IAM) is calculated using the relationship shown in (42):

$$IAM = 1 - b_0\left(\frac{1}{\cos(IA)} - 1\right) \tag{42}$$

Where,

IAM : Incidence Angle Modifier [unitless]

b_0 : Single defining parameter for the IAM curve (ASHRAE)

IA : Incidence Angle of the Plane of Array Incidence Component (radians)

The Incidence Angle for the Plane of Array Beam Component is calculated based on the sun's position for a given time step. To see how the Incidence Angle for the Plane of Array Beam Component is calculated in detail, please see 5.3.1.3. Due to the scattered nature of the diffuse and ground-reflected components of the plane of array irradiance, an "averaged" incidence angle is used to determine the incidence angle modifier for these components. CASSYS uses the same angle of 60° to calculate IAM for the diffuse and ground-reflected components in accordance with [3].

4.8.7.2 User-defined IAM Profile:

For some panels, the ASHRAE IAM profile is inadequate. It is then possible for users to define their own IAM profile, by defining the IAM vs. incidence angle curve through a series of points. Equation (42) is then replaced with a Bezier interpolation between the user-defined points.

4.8.8 Soiling Losses

Soiling losses account for the accumulation of dirt, snow, or sand on the panels. Users can specify values on a yearly or monthly basis. The corresponding IAM is applied to every component of plane of array irradiance and after which the user specified soiling loss percentage is applied to the resultant irradiance. This is the irradiance used in further calculations.

4.8.9 Effective Cell Temperature *void CalcTemperature(..)*

Cell temperature is modelled using the equation provided below. The equation finds its basis in the Faiman Temperature Model for module temperature [7] but is modified with module absorption and efficiency to determine the cell temperature. The equation requires an assumption regarding the absorption of the modules which is assigned to 0.9 in the program.

$$T_c = T_a + \frac{\alpha H_{POA}(1 - \eta_{module})}{U_0 + U_1 * WS} \tag{43}$$

Where,

T_c : Cell temperature [°C]

T_a : Ambient air temperature [°C]

α : Absorption coefficient of the module [default value of 0.9 is used]

H_{POA} : Incident irradiance after soiling and IAM losses are applied [W m^{-2}]

η_{module} : Efficiency of the module in the sub-array (see (44))

U_0 : Constant Convective heat transfer coefficients [W m^{-2} K^{-1}]

U_1 : Wind Dependent Convective heat transfer coefficient [W m^{-3} s^{-1} K^{-1}]

WS : Wind speed [m/s]

The efficiency of the module is calculated at reference conditions using the following equation:

$$\eta_{module} = \frac{P_{ref}}{H_{ref} * A_{module}} \tag{44}$$

Where,

P_{ref}: Rated power of the module at reference conditions (W)

H_{ref}: Irradiance at reference conditions (W·m^2)

A_{module}: Individual module area (m^2)

4.8.10 I-V Curve Parameters at Effective Conditions ***void CalcIVCurveParameters(…)***

4.8.10.1 Diode Ideality Factor ***void CalcGammaCoeff()***

The diode ideality factor varies with temperature and is determined in order to best match the temperature correction factor of the maximum power point defined the module database (or PAN file). The determination is done using the Power at reference conditions and by assuming a temperature of 25°C higher than reference. This evaluation of the gamma coefficient assumes a linear dependence of the ideality factor on temperature. The following equation is used to adjust the ideality factor at reference conditions to a calculated cell temperature, T_C:

$$\gamma = \gamma_{ref} - \gamma_{coeff}(T_c - T_{ref}) \tag{45}$$

4.8.10.2 Reverse Saturation Current

The reverse saturation current of the module is dependent upon cell temperature T_C and the relationship is defined using physical principles as below [3]:

$$I_0 = I_{0,ref}\left(\frac{T_c}{T_{c,ref}}\right)^3 \exp\left[\frac{qE_g}{\gamma k}\left(\frac{1}{T_{c,ref}} - \frac{1}{T_c}\right)\right] \tag{46}$$

Where,

I_0 : Reverse Saturation Current

$I_{0,ref}$: Reverse Saturation Current at reference conditions

T_c : Calculated cell temperature [K] (see 4.8.9)

$T_{c,ref}$: Cell temperature at reference conditions (or at T_{ref}) [K]

q: Elementary Charge [C], Constant: $1.602 \times 10^{-19} C$

γ: Temperature adjusted diode ideality factor [1/K]

E_g: Band-gap of the cell material (CASSYS only simulation Si based solar modules, 1.12 eV)

k: Boltzmann Constant, 1.381 x 10^{-23} m^2 kg s^{-2} K^{-1}

4.8.10.3 Shunt Resistance

CASSYS uses the PVsyst irradiance-dependent shunt resistance model (47) which requires the definition of an exponential decay constant and a shunt resistance value at 0 irradiance, $R_{sh}(0)$. Review of prior work [8] shows that the best exponential parameter for any technology is 5.5 whereas, the $R_{sh}(0)$ is typically *4 x* $R_{sh,ref}$. CASSYS uses the values in the database for both of these parameters and uses the $R_{sh}(0) = 4 * R_{sh,ref}$ rule as a default if the value is not defined for a selected module.

$$R_{sh} = R_{sh,base} + \left[R_{sh}(0) - R_{sh,base}\right]\exp\left[-R_{sh,exp}\frac{H}{H_{ref}}\right] \quad \textbf{(47)}$$

Where,

R_{sh} : Irradiance adjusted shunt resistance of the module [Ω]

$R_{sh,base}$: Fitting parameter [Ω] defined as,

$$R_{sh,base} = R_{sh,ref} - R_{sh}(0) * \exp(-R_{sh,exp})/(1 - \exp[-R_{sh,exp}]) \quad \textbf{(48)}$$

$R_{sh,exp}$: Exponential decay rate fixed to 5.5 in the program

H: Effective Plane of Array Irradiance [Wm^{-2}]

H_{ref}: Irradiance used at Reference Conditions [Wm^{-2}]

4.8.10.4 Photo-generated Current

Photo-generated current is affected by temperature which is calculated using the current temperature coefficient ($I_{sc,T-coeff}$) listed in the database for the module selected) and the irradiance through a proportionality relationship. The temperature coefficient is typically specified by module manufacturers in module datasheets. The equation to calculate effective photo-generated current at given temperature and irradiance takes the following form:

$$I_{\Phi} = \frac{H}{H_{ref}}\left[I_{\Phi,\mathrm{ref}} + I_{sc,T-coeff}(T_c - T_{ref})\right] \quad \textbf{(49)}$$

Where,

I_{Φ}: Temperature and irradiance adjusted photo-current [A]

$I_{\Phi,\mathrm{ref}}$: Photo-current determined at reference conditions [see 4.8.54.8.5]

$I_{T-coeff}$: Current temperature coefficient [A/°C]

T_c: Calculated cell temperature [K] (see 4.8.9)

T_{ref}: Cell temperature at reference conditions (or at T_{ref}) [K]

4.8.10.5 Open-Circuit Voltage

Open-circuit voltage is affected by temperature and the relationship is best summarized by the voltage temperature coefficient ($V_{oc,T-coeff}$). To calculate the effective open-circuit voltage at given cell temperature one can use the following:

$$V_{oc} = V_{oc,ref} + V_{oc,T-coeff} * (T_c - T_{ref}) \quad \textbf{(50)}$$

Where,

V_{oc}: Temperature adjusted open-circuit voltage of the module [V]

$V_{oc,ref}$: Open-circuit voltage at reference conditions [V]

$V_{T-coeff}$: Open-circuit voltage temperature coefficient [V/°C]

T_c: Calculated cell temperature [K] (see 4.8.9)

T_{ref}: Cell temperature at reference conditions (or at T_{ref}) [K]

H: Effective Plane of Array Irradiance [Wm^{-2}]
H_{ref}: Irradiance used at Reference Conditions [Wm^{-2}]

4.8.11 Array to Inverter Wiring Resistance

The wiring resistance is specified by the user for the entire sub-array in terms of a percentage loss at maximum production by the modules in the sub-array ($R_{\%,STC}$). This loss percentage is translated to an equivalent wiring resistance R_w 'seen' by each module, using the following equation:

$$R_w = \frac{N_s}{N_p} R_{\%,STC} \left(\frac{P_{mpp}}{{I_{mpp}}^2} \right) \quad (51)$$

Where,
N_s: Number of modules in series in each string of the sub-array
N_p: Number of modules in parallel in the sub-array
P_{mpp}: Maximum power the module can produce [W]
I_{mpp}: Current of the module at the maximum power point [A]

This resistance is treated as an additional resistance in series with the series resistance. The effective equation for the SDM is then changed to:

$$I = I_\Phi - I_0 \left[e^{q\frac{[V+I(R_s+R_w)]}{N_{cells}\gamma k T_c}} - 1 \right] - \frac{V + I(R_s+R_w)}{R_{sh}} \quad (52)$$

4.8.12 Calculating Module Performance *void CalcAtMaximumPowerPoint()*

After the effective I-V curve parameters and the revised SDM equations are established, the module performance can be calculated based on the operating mode of the array (i.e. if the array is allowed to operate at its maximum power point (MPP) or at a fixed-voltage).
Due to the implicit nature of the revised SDM equation, the Newton-Raphson root-finding algorithm is used to determine the current produced by the module at any voltage V. For more information on how this algorithm is implemented, please see [9] and [10].
The maximum power of the module is then determined by the golden-ratio search algorithm [10].

4.8.13 Calculating Array Performance *void CalcModuleToArray(…)*

The calculations in 4.8.12 are done at the module level and must be translated to the array using the configuration of the modules. The current produced by the number of modules in parallel will add, whereas voltage for number of modules in series will add. The effective current and voltage are determined after applying the following losses to the array performance. The losses currently accepted by CASSYS (v 0.9) are discussed briefly in the next few sub-sections.

4.8.13.1 Module Quality Loss, LID Loss and Ageing Loss

The user provides a percentage loss ($L_{\%,modquality}$) at standard conditions reflecting losses in module quality due to differences between actual and nameplate rating of the modules (and possibly other causes). Losses due to LID and module aging are treated separately (see below). The loss is calculated using the following:

$$L_{modquality} = I_{array} \cdot L_{\%,modquality} \cdot V_{array} \quad (53)$$

The LID loss is entered by the user as a percentage ($L_{\%,LID}$) of module power that the module typically degrades by in the first few weeks of exposure to sunlight. The loss due to LID is calculated using the following:

$$L_{LID} = I_{array} \cdot L_{\%,LID} \cdot V_{array} \quad (54)$$

Finally the ageing loss is entered by the user as a percentage ($L_{\%,Ageing}$) of module power that the module typically degrades by with each passing year. The loss due to LID is calculated using the following:

$$L_{Ageing} = I_{array} \cdot L_{\%,Ageing} \cdot V_{array} \cdot (y + 0.5) \quad (55)$$

where y is the number of years since the beginning of the simulation: $y = 0$ in the first year, $y = 1$ in the second year, etc. Please note that y is an integral number, that is, ageing is considered to be constant during each individual year of the simulation (another way of calculating ageing would have been to calculate it on a daily basis, starting from 0% at day 1 and reaching $L_{\%,LID}$ after 365 days; however this method has the disadvantage of being sensitive to the start date of the simulation, that is, a year-long simulation starting in July would not give the same results as a year-long simulation starting in January; for that reason CASSYS assumes that ageing can be represented by a single degradation number that is constant throughout the year). Please note also the addition of 0.5 to y. In essence this signifies that in the first year, modules are assumed to have degraded by half of the yearly ageing.

These percentage losses due to quality, LID and ageing are then applied to the current produced by the array and a new current value is determined:

$$I_{array,1} = I_{array} \left(1 - L_{\%,modquality} - L_{\%,LID} - L_{\%,Ageing} \cdot (y + 0.5)\right) \quad (56)$$

4.8.13.2 Mismatch Loss

The user provides a percentage loss occurring due to the mismatch of module current-voltage characteristics results. The losses resulting from this mismatch would be different if the array is operating at its maximum power point (MPP) or if it forced by the inverter to operate at a fixed voltage (FV). Hence, the user can provide a percentage loss at each condition. Typically the fixed voltage operating mode incurs a higher performance penalty. The percentage loss is applied to the current produced by the array based on the operating mode.

$$L_{mismatch} = I_{array} \cdot L_{\%\, mismatch,FV/MPP} \cdot V_{array} \tag{57}$$

This percentage is then applied to the current produced by the array and a new current value is determined:

$$I_{array,2} = I_{array,1} \left(1 - L_{\%\, mismatch,FV/MPP}\right) \tag{58}$$

4.8.13.3 Ohmic Losses

Given the effective wiring resistance (R_w) for the entire sub-array (see 4.8.11), the ohmic loss is calculated using ohms law:

$$L_{ohmic} = I^2_{array,2} \cdot R_w \tag{59}$$

4.8.13.4 Soiling Losses

As discussed earlier (4.8.13.4), the soiling loss percentage specified by the user is directly applied to the incident irradiance. To estimate the power lost to soiling, the program calculates the losses using the following:

$$L_{soiling} = I_{array} \cdot \frac{L_{\%,soiling}}{(1 - L_{\%,soiling})} \cdot V_{array} \tag{60}$$

4.8.13.5 Power Produced by the Array

Since all other losses are accounted for by reducing the effective current produced by the array, the ohmic loss is applied in the end. The total power produced by the array after all is then given by:

$$P_{array} = V_{array} \cdot I_{array} - L_{ohmic} \tag{61}$$

4.9 Radiation Processing *RadiationProc.cs*

Radiation processing is responsible for handling all radiation related calculations within the solar site.

4.9.1 Inputs

SimMet	*Climate information from input file*

4.9.2 Outputs

SimSun	*Instance of Sun class*
SimTracker	*Instance of Tracker class*
SimHorizonShading	*Instance of HorizontalShading class*
SimTilter	*Instance of Tilter class*
SimTilterOpposite	*Instance of Tilter class*

Timestamp_Used_for_Simulation	*[yyyy-mm-dd hh:mm:ss]*
Sun_Zenith_Angle	*[degrees]*
ET_Irrad	*Extraterrestial Irradiance [Wm^{-2}]*
Albedo	*[-]*
Normal_beam_irradiance	*[Wm^{-2}]*
Horizontal_global_irradiance	*[Wm^{-2}]*
Horizontal_diffuse_irradiance	*[Wm^{-2}]*
Horizontal_beam_irradiance	*[Wm^{-2}]*
Global_Irradiance_in_Array_Plane	*[Wm^{-2}]*
Beam_Irradiance_in_Array_Plane	*[Wm^{-2}]*
Beam_Irradiance_in_Array_Back	*[Wm^{-2}]*
Diffuse_Irradiance_in_Array_Plane	*[Wm^{-2}]*
Ground_Reflected_Irradiance_in_Array_Plane	*[Wm^{-2}]*
Tracker_Slope	*[degrees]*
Tracker_Azimuth	*[degrees]*
Tracker_Rotation_Angle	*[degrees]*
Collector_Surface_Slope	*[degrees]*
Collector_Surface_Azimuth	*[degrees]*
Incidence_Angle	*[degrees]*

4.9.3 Parameters

SimSplitter	*Instance of Splitter class*
pyranoTilter	*Instance of Tilter class*
negativeIrradFlag	*Boolean used to track the presence of negative Irradiance*

These parameters are used in combination with output variables to model irradiance and calculate radiation related outputs.

4.10 Shading *Shading.cs*

This class calculates the shading factors on the beam, diffuse and ground-reflected components of incident irradiance based on the sun position throughout the day resulting from a near shading model. Shading models are available for panels arranged in an unlimited rows or a fixed tilt configuration. If the unlimited row model is to be used, the model can be further customized to use a linear shading model or a cell based (step-wise) shading model. For bifacial modelling, the shading class also calculates the shading factor on the beam component in the back of the panel.

4.10.1 Inputs

Sun Zenith	*Zenith angle of the sun (rad)*
Sun Azimuth	*Azimuth angle of the sun (rad)*
POA Beam Irradiance	*Beam component of POA irradiance (W·m^{-2})*
POA Diffuse Irradiance	*Diffuse component of POA irradiance (W·m^{-2})*
POA Ground-Reflected Irradiance	*Ground reflected component of POA irradiance (W·m^{-2})*

4.10.2 Outputs

Beam Shading Factor (SF)	*Shading factor applied to POA beam irradiance*

Back Beam SF	*Shading factor applied to back side beam irradiance*
Diffuse SF	*Shading factor applied to POA diffuse irradiance*
Ground-Reflected SF	*Shading factor applied to POA ground-reflected irradiance*
Front Profile Ang	*Profile angle of the sun from front of panel [see (64)]*
Back Profile Ang	*Profile angle of the sun from back of panel [see (65)]*
Shaded Global POA Irradiance	*Sum of all components after respective shading factors are applied ($W{\cdot}m^{-2}$)*
Shaded Beam POA Irradiance	*POA beam irradiance with SF applied ($W{\cdot}m^{-2}$)*
Shaded Diffuse POA Irradiance	*POA diffuse irradiance with SF applied ($W{\cdot}m^{-2}$)*
Shaded Ground-Reflected POA Irradiance	*POA ground-reflected irradiance with SF applied ($W{\cdot}m^{-2}$)*

4.10.3 Parameters

Fixed-Tilt Configuration:

Plane of Array Tilt	*Tilt of the modules installed relative to the ground (rad)*
Plane of Array Azimuth	*Azimuth angle of the modules relative to True South (rad)*

For the Unlimited Rows Configuration, the following are required in addition to the above (see Figure 9):

Front Shading Limit Angle	*Shading limit angle for front of panel*
Back Shading Limit Angle	*Shading limit angle for back of panel*
Width of Active Area	*Width of the collector (m)*
Pitch	*Distance between consecutive rows of modules (m)*
Number of Rows	*Number of rows in the farm, Unlimited Rows Configuration*

The above parameters allow for the calculation of a linear shading model (with respect to Profile Angle), for the non-linear shading model, the following additional parameters are required (see Figure 9):

Number of Modules	*Number of modules in width of active area*
Cell Size	*Size of the cell in the transverse direction (cm)*

4.10.4 Shading Model for Unlimited Rows Configuration

The unlimited row configuration is an array configuration in which panels are placed on racking with a fixed tilt with rows of such racks placed behind one another. CASSYS considers the length of such rows to be large enough that any edge effects can be neglected in its shading model. This assumption reduces the calculation of the shading factor at different times of the day to a simple geometrical construct. The user can specify the following parameters regarding the row orientation summarized in Figure 9.

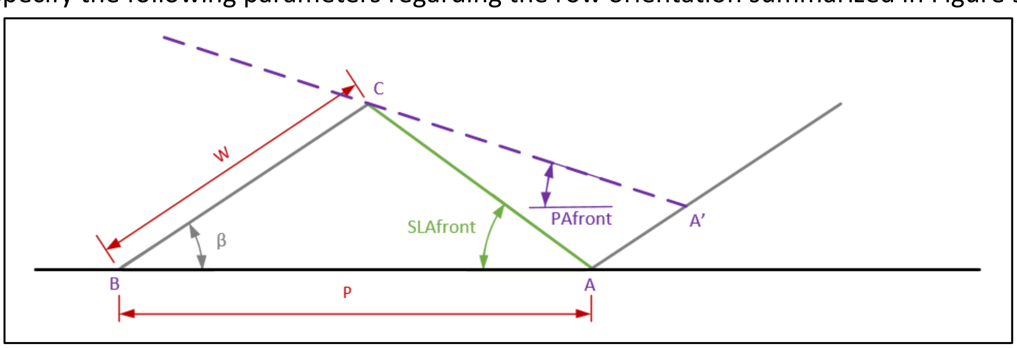


Figure 9: Unlimited Rows Orientation and Shading Analysis: Module Front Side

The front shading limit angle (SLA_{front}) is the angle formed by the top of the collector to the bottom of the collector in the following row. It is determined using the known values of collector width (W), row tilt (β), and pitch (P). This calculation is done in the interface and provided to the user.

$$SLA_{front} = \arctan\left[\frac{W \cdot \sin(\beta)}{P - W \cdot \cos(\beta)}\right] \tag{62}$$

Similarly, the back shading limit angle (SLA_{back}) is the angle formed by the top of the collector to the bottom of the collector in the preceding row. This is relevant in modelling back side irradiance for bifacial modules.

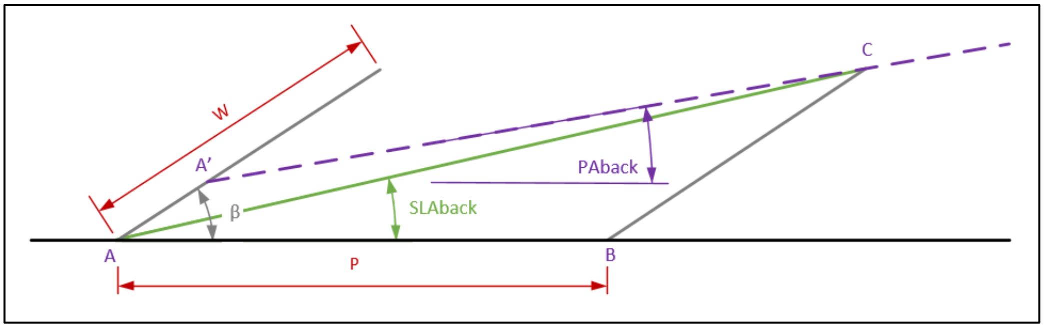


Figure 10: Unlimited Rows Orientation and Shading Analysis: Module Back Side

$$SLA_{back} = \arctan\left[\frac{W \cdot \sin(\beta)}{P + W \cdot \cos(\beta)}\right] \tag{63}$$

Another significant angle is the profile angle of the sun at a given time. The profile angle (PA) is defined as the projection of the solar altitude angle (complement of zenith angle) on a vertical plane perpendicular to the array [3]. For a given sun zenith (θ_Z) and azimuth (γ_S), and for a collector azimuth (γ), the front profile angle (PA_{front}) is calculated with equation (64).

$$PA_{front} = \arctan\left[\frac{\tan\left(\frac{\pi}{2} - \theta_Z\right)}{\cos(\gamma_S - \gamma)}\right] \tag{64}$$

Since the collector azimuth is defined by the azimuth of the front side of the module, the back profile angle (PA_{back}) is found by adding π to the collector azimuth. As shown by equation (65), PA_{back} is found to be the negation of PA_{front}.

$$PA_{back} = \arctan\left[\frac{\tan\left(\frac{\pi}{2} - \theta_Z\right)}{\cos(\gamma_S - (\gamma + \pi))}\right] = \arctan\left[\frac{\tan\left(\frac{\pi}{2} - \theta_Z\right)}{-\cos(\gamma_S - \gamma)}\right] = -PA_{front} \tag{65}$$

4.10.4.1 Linear Shading Factor on the POA Beam Component

For shading to occur, the profile angle of the sun must be lower than the shading limit angle. If this is true, the length of AA’ is the portion of a row shaded by the preceding row. To determine the length of the shaded portion on the front of the module, the following equations (66)-(69) are used:

$$AC = \frac{\mathrm{W} \cdot \sin(\beta)}{\sin\left(SLA_{front}\right)} \tag{66}$$

$$\angle CAA' = \pi - \beta - SLA_{front} \tag{67}$$

$$\angle CA'A = \pi - CAA' - (SLA_{front} - PA_{front}) \tag{68}$$

$$AA' = \frac{AC \cdot \sin(SLA_{front} - PA_{front})}{\sin(CA'A)} \tag{69}$$

The shaded fraction for the beam component is given by the ratio of the collector width and AA' calculated from equation (69). The calculations for the back shaded fractions in equations (70)-(73) are analogous:

$$AC = \frac{\mathrm{W} \cdot \sin(\beta)}{\sin(SLA_{back})} \tag{70}$$

$$\angle CAA' = \pi - \beta - SLA_{back} \tag{71}$$

$$\angle CA'A = \pi - CAA' - (SLA_{back} - PA_{back}) \tag{72}$$

$$AA' = \frac{AC \cdot \sin(SLA_{back} - PA_{back})}{\sin(CA'A)} \tag{73}$$

The shaded fraction for the beam component is, again, given by the ratio of the collector width and AA' calculated from equation (73).

4.10.4.2 Cell-based Shading Model on the POA Beam Component

In the cell-based shading model, the entire section (modules placed alongside in the length of the row) is considered inactive as soon as the first cell of that module is completely shaded. Once the length of the shaded section is calculated from either (69) or (73), the model determines the number of cells under this section (74).

$$N_{cells\ shaded} = \frac{AA'}{Cell\ Size} \tag{74}$$

When the last cell of section *S* is shaded by the preceding row, the shading fraction is the ratio of the section number and the total number of sections (or modules) in the active area width.

$$SF_S = \frac{S}{N_{modules,Active\ Area}} \tag{75}$$

The shading fraction maximizes for a given section at SF_s once a module is completely shaded, and follows a linear pattern until the first cell in each module section is completely shaded.

4.10.4.3 Shading factor on the POA Diffuse Component

The shaded fraction for the diffuse component is calculated using the view factor of the sky for a tilted surface obstructed by the row in front of it. For more information regarding view factor integrals, please see [4].

$$Diffuse\ Shading\ Fraction = \frac{1 + \cos(SLA_{front})}{2} \quad \textbf{(76)}$$

4.10.4.4 Row Block Factor Shading factor on the POA Ground-Reflected Component

Due to the arrangement of the rows, only the rows other than the first row will experience shading from preceding rows. CASSYS adjusts for this by scaling the shading factors based on a row block factor. The row block factor is given by (77).

$$Row\ Block\ Factor = \frac{Number\ of\ Rows - 1}{Number\ of\ Rows} \quad \textbf{(77)}$$

This is applied to the diffuse and beam shading factors as defined above. That is, the first row is receiving the full irradiance (beam or diffuse) whereas all other rows receive the irradiance multiplied by the shading factor. The program uses a weighted average of the two. After some rearranging, the plane of array beam irradiance after shading is given by:

$$(1 - RBF \cdot SF) \cdot H_{POA,beam} \quad \textbf{(78)}$$

where *RBF* is the row block factor and *SF* is the beam shading factor calculated in section 4.10.4.1, and $H_{POA,beam}$ is the plane of array beam irradiance. For diffuse irradiance, the expression is:

$$\left(1 - RBF \cdot \frac{1 - \cos(SLA_{front})}{2}\right) \cdot H_{POA,dif} \quad \textbf{(79)}$$

where $H_{POA,dif}$ is the plane of array diffuse irradiance. The minus sign in front of the cosine is not an error and results from rearranging eqn. (76); if the rows are far apart and SLA_{front} is essentially zero, then eqn. (79) reduces to $H_{POA,dif}$ which is what is expected. Similarly if there is an infinite number of rows then *RBF* becomes 1 and eqn. (79) reduces to eqn. (76) times $H_{POA,dif}$.
Finally the shading factor for the ground-reflected component is given by (80).

$$Ground\ Reflected\ Shading\ Factor = 1 - Row\ Block\ Factor \quad \textbf{(80)}$$

That is, only the first rows sees ground-reflected irradiance, subsequent rows basically see nothing. In systems with a large number of rows, the row block factor is close to 1 and in first approximation there is no reflected irradiance reaching the front of the panels.
For non-tracking systems, from the definition of the diffuse and ground-reflected shading factor, it is clear that these remain constant over the year, and hence are calculated only once by the program.

4.11 Simulation — ***Simulation.cs***

This class is responsible for determining the system mode and managing the various interactions between the other classes. Simulation calls specific methods to initialize, configure, and calculate different aspects of a site based on the system mode.

4.12 Spectral Effects *SpectralEffects.cs*

This class is responsible for the simulation of "spectral" effects, using a relatively simple model. Given a "spectral" effect curve based on clearness index (k_t), the clearness correction value is determined.

4.12.1 Inputs

HGlo	*Horizontal global irradiance, measured/calculated (W/m^2)*
NExtra	*Extraterrestrial normal irradiance, measured (W/m^2)*
Sun Zenith	*Zenith angle of the sun (rad)*

4.12.2 Outputs

Clearness Correction	*Clearness correction value*

4.12.3 Parameters

SpectralClearnessIndexStr	*The array of clearness index points defined by the curve*
SpectralClearnessCorrectionStr	*The array of clearness correction values defined by the curve*

4.12.4 Equations and Model Descriptions

4.12.4.1 Interpolation and Application of Clearness Correction

The clearness correction curve is defined by a series of clearness indices and their corresponding correction values. The clearness index is first calculated by the *GetClearnessIndex(…)* method from the *Sun* class, and then the interpolation between clearness indices is done using the *Linear(…)* method from the *Interpolate* class to obtain the clearness correction. Within the *CalcEffectiveIrradiance(…)* method of the *PVArray* class, the amount of irradiance that reaches the PV cell is modified by a spectral loss factor as in equation (41). The sign of the clearness correction value is switched so that a positive value acts as a gain and a negative value acts as a loss.

4.13 Splitter *Splitter.cs*

This class calculates beam and diffuse irradiance, given global horizontal irradiance (or its components).

4.13.1 Inputs

Sun Zenith	*Zenith angle of the sun [radians]*
Global Horizontal Irradiance	*Measured/Calculated (W·m^{-2})*
Diffuse Horizontal Irradiance	*Measured (W·m^{-2})*
Direct Normal Irradiance	*Measured (W·m^{-2})*
Direct Horizontal Irradiance	*Measured (W·m^{-2})*
Normal Extraterrestrial Irradiance	*Measured (W·m^{-2})*

4.13.2 Outputs

Global Horizontal Irradiance	*(W·m^{-2})*
Diffuse Horizontal Irradiance	*(W·m^{-2})*
Direct Normal Irradiance	*(W·m^{-2})*
Direct Horizontal Irradiance	*(W·m^{-2})*

4.13.3 Equations and Model Description

The algorithm used by the Splitter class depends on the solar radiation components (e.g. global, diffuse or direct) provided as input.

4.13.3.1 When only global irradiance is provided

When only global irradiance is provided, the first step of the algorithm is to calculate the beam and diffuse components of irradiance. The *GetClearnessIndex(…)* and *GetDiffuseFraction(…)* methods from the *Sun* class are called to obtain the diffuse fraction k_d as in equation (88). With the provided global irradiance on the horizontal (H_g), the diffuse and beam irradiance on the horizontal, H_d and H_b, are calculated:

$$H_d = H_g \cdot k_d \tag{81}$$

$$H_b = H_g - H_d \tag{82}$$

Finally, normal beam irradiance $H_{b,n}$ is calculated according to horizontal beam irradiance and z the zenith angle of the sun:

$$H_{b,n} = H_b / \cos(z) \tag{83}$$

In early morning or late afternoon, the formulae above may lead to very large values of beam irradiance. To circumvent that issue, the program sets H_b to zero and $H_d = H_g$ whenever the zenith angle exceeds 87.5°.

Finally, to prevent possible issues, particularly when the program is used with radiation data measured in the field, beam normal irradiance is limited its clear sky value $H_{b,cs}$. This latter value is calculated using the ASHRAE model ([11], ch. 14) as

$$H_{b,cs} = H_{0,n} \cdot \exp\left(-\tau_b \cdot m^{ab}\right) \tag{84}$$

where m is the air mass and $H_{0,n}$ is the normal extraterrestrial irradiance measured perpendicularly to the rays of the sun. The beam pseudo optical depth τ_b and the airmass exponent ab receive the values 0.245 and 0.668. These values were derived for Flagstaff, AZ, for the month of June, and correspond to one of the highest beam/extraterrestrial ratios that can be reasonably expected worldwide.

4.13.3.2 When both global and diffuse irradiances are provided

This is the easiest case. Beam irradiance is simply calculated through eqs. (82) and (83).

4.13.3.3 When both global and beam horizontal irradiances are provided

This is again an easy case. Diffuse irradiance is calculated as:

$$H_d = H_g - H_b \tag{85}$$

Other formulae such as (83) are unchanged.

4.13.3.4 When both global and beam normal irradiances are provided

This case is similar to the previous one, except that beam horizontal irradiance is first calculated as:

$$H_b = H_{b,n} \cdot \cos(z) \tag{86}$$

4.13.3.5 When both global and beam horizontal irradiances are provided

The same equations are used, the first step being now the calculation of global irradiance:

$$H_g = H_b + H_d \tag{87}$$

4.13.3.1 When both global and beam normal irradiances are provided

This is the same case as the previous one, with the addition of equ. (86).

4.14 Sun Sun.cs

The Sun class is an object used to compute solar zenith and azimuth angles, and other quantities related directly to earth-sun geometry. For the conventions used throughout CASSYS for solar angles, please refer to Appendix: Angle Conventions.

4.14.1 Inputs

Day of the year	*1 to 365 (Leap Year: 29 Feb and March 1 is treated as the same day)*
Hour of the Day	*24-Hour Decimal Format (11.75 is 11:45)*

4.14.2 Outputs

Zenith	*Zenith angle of sun [rad]*
Azimuth	*Azimuth of sun [rad, >0 facing E]*
AirMass	*Air mass [unitless]*
NExtra	*Extraterrestrial normal Irradiance [W/m2]*
AppSunsetHour	*Apparent sunset hour (given array tilt)*
AppSunriseHour	*Apparent sunrise hour (given array tilt)*
TrueSunsetHour	*True sunset hour*
TrueSunriseHour	*True sunrise hour*

4.14.3 Parameters and Cached Variables

Cached variables are used to speed up some calculations.

Site Lat	*Latitude of the site (°N is positive, °S is negative)*
Site Long	*Longitude of the site (°E is positive, °W is negative)*
Site Meridian Longitude	*Reference Meridian for Local Standard Time at Site (°E is Positive), required when doing calculations is in local time*
Slope of Collector	*Slope of the plane of array (radian)*
Current Day	*Cached, Current day of year (to speed up daily calculations)*
Current Declination	*Cached, Declination for Current Day*
Current extraterrestrial irradiance	*Cached, Extra-terrestrial radiation for current day*

4.14.4 Equations and Model Descriptions

This class is mostly a wrapper for methods found in the Astro.cs class (for calculation of declination, sun position, etc.) and in the Tilt.cs class (for calculation of apparent sunrise and sunset). Please refer to sections 5.2 and 5.3.1.3 for a description of the mathematical models used.

Note that internally all the angular parameters and variables are stored in radians. This makes it straightforward to make use of trigonometric equations without having to convert angles between degrees and radians.

4.14.4.1 Calculation of Clearness Index and Diffuse Fraction

This is done through the use of Orgill and Hollands' formula (see [3], p. 81). The diffuse fraction k_d is defined as the ratio of horizontal diffuse irradiance to horizontal global irradiance:

$$\begin{aligned} k_d &= 1.0 - 0.249 \cdot k_t && \text{if } k_t < 0.35 \\ k_d &= 1.557 - 1.84 \cdot k_t && \text{if } 0.35 < k_t < 0.75 \\ k_d &= 0.177 && \text{if } k_t > 0.75 \end{aligned} \tag{88}$$

where k_t is the clearness index defined as:

$$k_t = \frac{H_g}{H_{0,n} \cdot \cos(z)} \tag{89}$$

with H_g the global irradiance on the horizontal, $H_{0,n}$ the normal extraterrestrial irradiance measured perpendicularly to the rays of the sun, and z the zenith angle.

4.15 Trackers *Tracker.cs*

This class is responsible for determining the array orientation when trackers are used in solar plants. The tracking mode depends on the inputs provided by the user. The effect of backtracking strategies is also calculated in this class.

4.15.1 Inputs

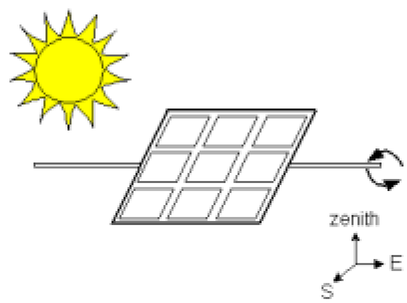
Tracker Axis Tilt	*Angle between tracking axis and horizontal (°)*
Tracker Axis Azimuth	*Angle of proj. horizontal tracker axis rel. to true South (°)*
Minimum Tilt	*Minimum slope of tracker surface wrt. horizontal (°)*
Maximum Tilt	*Maximum slope of tracker surface wrt. horizontal (°)*
Minimum Rotation Angle	*Minimum angle of rotation of surface about tracker axis (°)*
Maximum Rotation Angle	*Maximum angle of rotation of surface about tracker axis (°)*
Minimum Azimuth	*Minimum angle of horizontal proj. of normal to module surface and true South (°)*
Maximum Azimuth	*Maximum angle of horizontal proj. of normal to module surface and true South (°)*
Sun Azimuth	*Angle between horizontal proj. of a ray from sun and true South (°)*
Sun Zenith	*Angle between elevation of sun and vertical (°)*
Tracker Pitch	*Distance between two rows of trackers (m)*
Tracker Bandwidth	*Width of single tracker array (m)*
Tracker Clearance	*Ground clearance of tracker (m)*

4.15.2 Outputs

Tracker Azimuth	*Angle of proj. horizontal axis relative to true South (°)*
Tracker Slope	*Angle between tracking axis and horizontal (°)*
Surface Azimuth	*Angle between horizontal proj. of normal to module surface and true South (°)*
Surface Slope	*Angle of tracker surface wrt. horizontal (°)*
Surface Clearance	*Distance from lowest point of tracker surface to ground (m)*
Rotation Angle	*Angle of rotation of surface about tracker axis (°)*

4.15.3 Equations and Model Descriptions

4.15.3.1 E-W Elevation Tracker



E-W elevation trackers allow the panel to track the sun's movement vertically throughout the day. The tracker's tilt angle β is given by the following equation, where θ_z is sun zenith, γ_s is sun azimuth, and γ_a is axis azimuth.

$$\beta = \arctan(\tan(\theta_z) * \cos(\gamma_s - \gamma_a)) \tag{90}$$

If the calculated slope is negative, an angle of 180 degrees is added.

If the sun azimuth is greater than the axis azimuth, then the surface azimuth γ is given by

$$\gamma = \gamma_a + \frac{\pi}{2} \tag{91}$$

If the sun azimuth is less that the axis azimuth, then the surface azimuth γ is given by

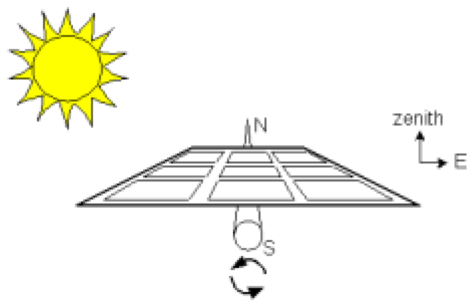
$$\gamma = \gamma_a - \frac{\pi}{2} \tag{92}$$

When backtracking is selected, the outputted surface slope is adjusted by the backtracking correction angle, ω_c , which is calculated from the tracker pitch, TP, and the tracker bandwidth, TW. The equation for finding the correction angle is found below

$$\omega_c = \arccos\left(TP * \frac{cos(\beta)}{TW}\right) \tag{93}$$

The outputted surface angle β is then adjusted by the correction angle.

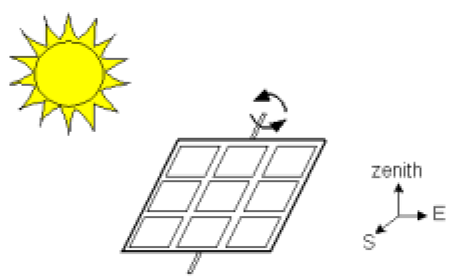
4.15.3.2 N-S Axis Tracker



N-S axis trackers allow the panel to track the sun's movement from east to west, rotating on an axis from north to south (axis azimuth of 0^o). The model used to calculate the tilt angle is the same as with the E-W elevation tracker, except that the axis azimuth value is 0.

With backtracking enabled, the same correction angle from the E-W tracker is calculated and applied.

4.15.3.3 Tilt and Roll Tracker



Tilt and Roll trackers allow the panel to follow the sun by rotating on a tilted axis with a user defined azimuth, γ_a, and tilt angle, β_a. The rotation angle, R, of the tracker is given by the equation

$$R = \arctan\left[\frac{\sin(\theta_z) * \sin(\gamma_s - \gamma_a)}{\cos(\theta_z) * \cos(\beta_a) + \sin(\theta_z) * \sin(\beta_a) * \cos(\gamma_s - \gamma_a)}\right] \tag{94}$$

The slope of the module, β, is calculated with the equation

$$\beta = \arccos(\cos(R) * \cos(\beta_a)) \tag{95}$$

The surface azimuth, γ, is calculated using one of the three equations below depending on the rotation angle. If the rotation angle is between -180 degrees and -90 degrees then the azimuth is calculated with

$$\gamma = \gamma_a - \arcsin\left(\frac{\sin(R)}{\sin(\beta)}\right) - \pi \tag{96}$$

If the rotation angle is between 180 degrees and 90 degrees then the azimuth is calculated with

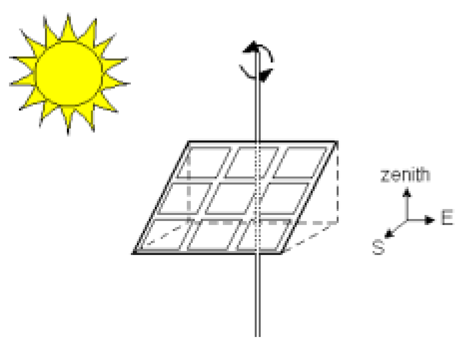
$$\gamma = \gamma_a - \arcsin\left(\frac{\sin(R)}{\sin(\beta)}\right) + \pi \tag{97}$$

If the rotation angle is outside either of those ranges then the azimuth is calculated with

$$\gamma = \gamma_a + \arcsin\left(\frac{\sin(R)}{\sin(\beta)}\right) \tag{98}$$

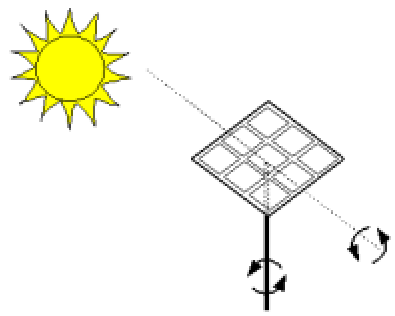
In the reference papers [12] for these equations, an azimuthal angle was defined between 0 and 360 degrees clockwise from true north, whereas CASSYS uses an azimuthal angle of -180 through 180 degrees, with $\pm$ 180 degrees being true north. Due to this difference of angle convention, 360 degrees will be added or subtracted to the CASSYS angle in order to rectify this difference.

4.15.3.4 Azimuth Tracking



Azimuth trackers follow the sun's azimuth exactly. With this type of tracker, the slope of the panel is fixed but the surface azimuth is equal to that of the sun within the limits on the surface azimuth defined by the user.

4.15.3.5 Two Axis Tracking



Two axis trackers follow both the sun's zenith and azimuth. The tilt of the surface is equal to the zenith of the sun and the surface azimuth is equal to the azimuth of the sun, which keeps the incidence angle of the beam at 0 degrees to the normal within the azimuthal and tilt limits defined by the user.

4.16 Transformer **Transformer.cs**

CASSYS models two types of losses for the grid-connected transformer, the core or iron losses and the ohmic losses.

4.16.1 Inputs

Input Power	*Power going from Inverters to the low-side of the transformer (W)*
Global Loss	*Total loss of the transformer at Rated Transformer Power (W), provided by the user*
Core or Iron-Loss	*Loss under no-load conditions (W), provided by the user*
Nightly Disconnected	*Boolean, if the transformer is designed to disconnect at night*

4.16.2 Outputs

Power Produced	*Power delivered to the grid by the transformer (W)*
Ohmic Losses	*Losses due to internal resistance of the windings (see* (101)*)*
Total Losses	*Losses due to the core and ohmic losses*

4.16.3 Parameters

4.16.3.1 Core Losses

Some of the input power on the primary side of the transformer is used to compensate for the core losses (composed of hysteresis and eddy currents) occurring within the transformer. The user can specify the power lost to core losses for a grid-connected transformer (or a combination of inverter transformers and grid-connected transformers) directly. Since this loss occurs under no-load conditions, this is modelled as a constant loss by CASSYS as long as the transformer is not disconnected.

4.16.3.2 Ohmic Losses

Input power is also lost to the internal resistance of the primary and secondary windings. This loss is also called the copper loss of the transformer. This is typically dissipated as heat and can be modelled with an effective resistance. The ohmic losses are zero when no power is flowing through the transformer, and is maximal when the transformer is working at its full nominal power. In CASSYS the user specifies the full load loss, which is the sum of core loss and ohmic loss at full load.

4.16.3.3 Nightly Disconnect

Core losses of the transformer could add up to a substantial amount over a year (~1% of annual production). Hence, some sites are designed to disconnect the transformer at night. CASSYS models this by disconnecting the transformer when there is no power output available from the site.

4.16.4 Modelling Transformer Loss

Transformer losses are the sum of the iron and resistive/inductive losses. The core losses are constant and the ohmic losses are quadratic. The following definition of the global loss of the transformer at its nominal rated power can be used:

$$L_{Global,Nom} = L_{IronLoss,Nom} + L_{OhmicLoss,Nom} \tag{99}$$

This provides the global losses at nominal power of the transformer as:

$$L_{OhmicLoss,Nom} = L_{Global,Nom} - L_{IronLoss,Nom} \tag{100}$$

Both parameters in the right hand side are provided by the user. This relationship is then used to determine the power loss from an input power, P_{Input} using

$$L_{OhmicLoss,P_{Input}} = \left(L_{Global,Nom} - L_{IronLoss,Nom}\right) \cdot \left(\frac{P_{Input}}{P_{Nom,T}}\right)^2 \tag{101}$$

where $P_{Nom,T}$ is the nominal power of the transformer. The loss at any input power is then calculated using a relationship similar to as the iron losses remain constant:

$$L_{Global,P_{Input}} = L_{IronLoss,Nom} + L_{OhmicLoss,P_{Input}} \tag{102}$$

<u>*Note: PVsyst transformer model*</u>

The interface (not the engine) also offers an option to switch back and forth between the transformer model used by CASSYS, based on the transformer's electrical characteristics (no load loss and full load loss) and PVsyst's model, where losses are expressed as a percentage of the hypothetical AC capacity at STC.. For reference, the equations used are:

$$P_{AC,STC} = \eta * P_{DC,STC} \tag{103}$$

$$L_{IronLoss,Nom} = f_{IronLoss,STC} \cdot P_{AC,STC} \tag{104}$$

$$L_{OhmicLoss,Nom} = f_{OhmicLoss,STC} \cdot P_{AC,STC} \cdot \left(\frac{P_{Nom,T}}{P_{AC,STC}}\right)^2 \quad \textbf{(105)}$$

where $P_{DC,STC}$ is the total DC power of the plant, η is the inverter efficiency (extrapolated to STC power divided by the total number of inverters, based on the efficiency curve of the first inverter only), $P_{AC,STC}$ is the hypothetical total AC power of the plant if the array was producing its nominal STC power and the inverters were not clipping, $f_{IronLoss,STC}$ is the iron loss expressed as a percentage of hypothetical STC power, and $f_{OhmicLoss,STC}$ is the ohmic loss expressed as a percentage of hypothetical STC power.

4.17 Transposition models — Tilter.cs

The Tilter class is used to calculate plane of array irradiance knowing horizontal irradiance (i.e. solar radiation *transposition*). It includes two widely used models, the Hay model and the Perez model.

4.17.1 Inputs

Normal Direct Irradiance	*Beam normal irradiance incident upon the array (W/m²)*
Horizontal Diffuse Irradiance	*Horizontal diffuse irradiance (W/m²)*
Normal Extraterrestrial Irr.	*Extraterrestrial irradiance measured perpendicularly to the rays of the sun (W/m²)*
Sun Zenith	*Zenith angle of the sun (radians)*
Sun Azimuth	*Azimuth angle of the sun (radians)*
Air Mass	*Air mass (-)*
Month number	*Month (1-12)*

4.17.2 Outputs

Global Irradiance in Tilted Plane	*Plane-of-array global irradiance (W/m²)*
Beam Irradiance in Tilted Plane	*Plane-of-array beam irradiance (W/m²)*
Diffuse Irradiance in Tilted Plane	*Plane-of-array diffuse irradiance (W/m²)*
Reflected Irradiance in Tilted Plane	*Plane-of-array ground-reflected irradiance (W/m²)*
Angle of Incidence	*Angle of incidence of beam irradiance on the array (radians)*

4.17.3 Parameters and Cached Variables

Tilt Algorithm	HAY or PEREZ
Surface Slope	Slope of the array (radians)
Surface Azimuth	Azimuth of the array (radians, 0 = S, W > 0)
Monthly Albedo	Average albedo value for each month [0-1]

4.17.4 Equations and Model Descriptions

Both models calculate plane-of-array radiation as the sum of three components: beam, diffuse and ground-reflected. The models differ only in the way they calculate plane-of-array diffuse irradiance. The calculation of beam and ground-reflected irradiances is identical in both models.

4.17.4.1 Beam and ground-reflected irradiances in the plane of array

Beam irradiance in the plane of the array $H_{b,t}$ is calculated from beam normal irradiance $H_{b,n}$ through a simple geometrical relationship:

$$H_{b,t} = H_{b,n} \cdot \cos\theta \tag{106}$$

where θ is the angle of incidence of beam irradiance on the array.

4.17.4.2 Diffuse irradiance in the plane of array – Hay model

The Hay model [3] calculates diffuse irradiance in the plane of array as:

$$H_{d,t} = H_d \cdot \left[(1 - AI) \cdot \frac{1 + \cos\beta}{2} + AI \cdot \frac{\cos\theta}{\cos z}\right] \tag{107}$$

where H_d is the diffuse irradiance on the horizontal, β is the slope of the array, θ is the angle of incidence of beam irradiance on the array, and z is the zenith angle. AI is the anisotropy index defined as:

$$AI = \frac{H_{b,n}}{H_{0,n}} \tag{108}$$

where $H_{0,n}$ is the extraterrestrial normal irradiance.

4.17.4.1 Diffuse irradiance in the plane of array – Perez model

The Perez model [13] is slightly more complicated than the Hay model. It calculates diffuse irradiance in the plane of array as:

$$H_{d,t} = H_d \cdot \left[(1 - F_1) \cdot \frac{1 + \cos\beta}{2} + F_1 \cdot \frac{a}{b} + F_2 \cdot \sin\beta\right] \tag{109}$$

a and b are two empirical coefficients defined as

$$a = \max(0, \cos\theta) \tag{110}$$

$$b = \max(\cos 85°, \cos z) \tag{111}$$

where θ is the incidence angle and z is the zenith angle.

F_1 and F_2 are also empirical coefficients. Their definition relies on the concepts of clearness ε and brightness Δ, defined as:

$$\epsilon = 1 + \frac{H_{b,n}/H_d}{1 + \kappa \cdot z^3} \tag{112}$$

$$\Delta = m\frac{H_d}{H_{0,n}} \tag{113}$$

where $H_{b,n}$ is the beam normal irradiance, H_d is the diffuse horizontal irradiance, κ is a constant equal to 5.535 10^{-6}, z is the zenith angle, m is the air mass, and $H_{0,n}$ is the extraterrestrial normal irradiance. F_1 and F_2 are calculated as:

$$F_1 = \max[0, f_{11} + f_{12} \cdot \Delta + f_{13} \cdot z\,] \quad \textbf{(114)}$$

$$F_2 = f_{21} + f_{22} \cdot \Delta + f_{23} \cdot z \quad \textbf{(115)}$$

where coefficients f_{11} to f_{23} are defined through a look-up table based on the value of ϵ:

Range of ϵ	f_{11}	f_{12}	f_{13}	f_{21}	f_{22}	f_{23}
0.000 to 1.065	-0.008	0.588	-0.062	-0.060	0.072	-0.022
1.065 to 1.230	0.130	0.683	-0.151	-0.019	0.066	-0.029
1.230 to 1.500	0.330	0.487	-0.221	0.055	-0.064	-0.026
1.500 to 1.950	0.568	0.187	-0.295	0.109	-0.152	-0.014
1.950 to 2.800	0.873	-0.392	-0.362	0.226	-0.462	0.001
2.800 to 4.500	1.132	-1.237	-0.412	0.288	-0.823	0.056
4.500 to 6.200	1.060	-1.600	-0.359	0.264	-1.127	0.131
above 6.200	0.678	-0.327	-0.250	0.156	-1.377	0.251

4.17.4.1 Ground-reflected irradiance in the plane of array

Ground-reflected irradiance in the plane of the array $H_{r,t}$, it is approximated by

$$H_{r,t} = H_g \cdot \rho \cdot \frac{1 - \cos\beta}{2} \quad \textbf{(116)}$$

where H_g is the global irradiance on the horizontal, β is the slope of the array, and ρ is the albedo.

4.17.4.2 Global irradiance in the plane of array

Once beam, diffuse and reflected components of irradiance $H_{b,t}$, $H_{d,t}$ and $H_{r,t}$, are calculated through equations (106), (107) or (109) (depending whether the Hay or Perez model is used), and (116), plane-of-array global irradiance H_t is calculated by summing all three components:

$$H_t = H_{b,t} + H_{d,t} + H_{r,t} \quad \textbf{(117)}$$

5 Classes with Mathematical Constructs

The following classes are static classes and contain mathematical constructs that enable the calculation of several different values that are used in the main classes above.

5.1 ASTM E2848 ***ASTME2848.cs***

ASTM E2848 is a standard which uses multiple regression to approximate the AC Power output from a solar farm.

5.1.1 Inputs

SimMet	*Climate information from input file (must at least contain plane of array irradiance in W/m² and ambient temperature in °C; optionally, wind speed in m/s)*

5.1.2 Outputs

ACPower	*Power output produced by solar farm (kW)*
Energy_Injected_Into_Grid	*Energy injected into grid (kWh)*

5.1.3 Parameters

itsPmax	*Maximum power the site can produce (kW)*
itsA1	*Regression factor a1 (1000 m²)*
itsA2	*Regression factor a2 (1000 m⁴/W)*
itsA3	*Regression factor a3 (1000 m²/°C)*
itsA4	*Regression factor a4 (1000 ms)*
itsEAF	*Array of length 12 holding monthly Empirical Adjustment Factors (unitless)*

These parameters are used to simulate the power output of the site in accordance with the ASTM E2848 standard [14].

5.1.4 Equations and Model Description

ASTM E2848 class is responsible for calculating the power and energy injected into the grid for a solar system as per the ASTM E2848 standard. The *EAF* (Empirical Adjustment Factor) is a monthly value that is used to increase accuracy of the power calculation and account for external factors that can hamper or enhance the site's production (for example snow or seasonal factors).

$$ACPower = max\left[min\left[E \times (a_1 + a_2 E + a_3 T_a + a_4 v) \times EAF_{monthly}, P_{max}\right], 0\right] \quad \textbf{(118)}$$

$ACPower$: Power generated by the site [kW]
E: Plane-of-array Irradiance [W/m²]
a_1: Regression factor a1
a_2: Regression factor a2
a_3: Regression factor a3
a_4: Regression factor a4
$EAF_{monthly}$: Empirical Adjustment factor for the current month [unitless; 1 = no adjustment]
T_a: Ambient temperature [°C]
v: wind speed [m/s]
P_{max}: Maximum power the site can produce [kW]

The output is limited to the maximum power of the site (P_{max}), as is the case for sites with large inverter loading ratios (that is, when the inverters are 'clipping'). The maximum power should also account for transformer losses, and should be determined using the power available at the high-side of the main transformer. Output is also limited to positive values, as the model can potentially lead to negative values when certain values of parameters a_1 to a_4 are used.

5.2 Daily Astronomical Calculations *Astro.cs*

The Astro class contains a set of methods to calculate sun position, air mass, day length, etc.
This class is a static class and does not require any parameters from the user.

5.2.1 *List of Methods*

private static double _GetDayAngle(…)
public static double GetDeclination(…)
public static double GetHourAngle (…)
public static void CalcSunPosition (…)
public static void CalcSunPositionHourAngle (…)
public static double GetEccentricityCorrFactor (…)
public static double GetSunEarthDistance (…)
public static double GetATmsST (…)
public static double GetLATime (…)
public static double GetNormExtra (…)
public static double GetAirMass (…)
public static double GetSunsetHourAngle (…)
public static double GetDayLength (…)

5.2.2 *Physical Model and Methods Description*

5.2.2.1 *GetDayAngle: calculation of the day angle*

The day angle is an auxiliary quantity used in the calculation of declination. It is defined as:

$$DA = \frac{2 \cdot \pi \cdot (n-1)}{365} \tag{119}$$

where n is the day of year (January 1 = 1, February 1 = 32, etc.).

5.2.2.1 *GetDeclination: calculation of declination*

Declination is the angular position of the sun at solar noon with respect to the plane of the equator ([3], p. 13). North is positive. It is given as [15]:

$$\begin{aligned}\delta = 0.006918 \; - \; 0.399912 \cdot \cos(DA) + \; 0.070257 \cdot \sin(DA) \\ - 0.006758 \cdot \cos(2 \cdot DA) + \; 0.000907 \cdot \sin(2 \cdot DA) \\ - 0.002697 \cdot \cos(3 \cdot DA) \; + \; 0.00148 \cdot \sin(3 \cdot DA)\end{aligned} \tag{120}$$

where DA is the day angle.

5.2.2.1 *GetHourAngle: calculation of hour angle*

The hour angle HA is the angular displacement of the sun east or west of the local meridian due to the rotation of the earth, and has a value of 15° per hour. It is expressed in degrees as:

$$HA = 15 \cdot (t - 12) \tag{121}$$

where t is the solar time. The hour angle is negative in the morning and positive in the afternoon.

The Astro.cs class also includes the reciprocal function, *GetTimeHA*, which calculates the time given the hour angle.

5.2.2.1 CalcSunPosition: calculation of the sun's position (zenith and azimuth angle)

CalcSunPosition is a wrapper function. It calculates the sun declination and hour angle and calls *CalcSunPositionHourAngle*.

5.2.2.1 CalcSunPositionHourAngle: calculation of the sun's position (zenith and azimuth angle)

The function calculates the sun zenith and azimuth given declination and hour angle. The zenith angle z is the angle between the vertical and a line to the sun. Its cosine is given by [3]:

$$\cos(z) = \cos(\phi) \cdot \cos(\delta) \cdot \cos(HA) - \sin(\phi) \cdot \sin(\delta) \tag{122}$$

where ϕ is the latitude, δ is the declination, and HA is the hour angle.

The azimuth angle γ is the angular displacement from south of the projection, on the horizontal plane, of the earth/sun line. It is positive for afternoon hours and negative for morning hours. It is defined by its sine and cosine [16] [17]:

$$\sin(\gamma) = \sin(HA) \cdot \cos(\delta) \,/\, \sin(z) \tag{123}$$

$$\cos(\gamma) = (\cos(HA) \cdot \cos(\delta) \cdot \sin(\phi) - \sin(\delta) \cdot \cos(\phi)) / \sin(z) \tag{124}$$

where z is the zenith angle.

5.2.2.1 GetEccentricityCorrFactor: calculation of the sun's eccentricity factor

The eccentricity correction factor describes the eccentricity of the earth's orbit around the sun. It is approximated by [15]:

$$\begin{aligned}\varepsilon = 1.000110 \ + \ 0.034221 \cdot \cos(DA) + \ 0.00128 \cdot \sin(DA) \\ + 0.000719 \cdot \cos(2 \cdot DA) + \ 0.000077 \cdot \sin(2 \cdot DA)\end{aligned} \tag{125}$$

where DA is the day angle.

5.2.2.1 GetSunEarthDistance: calculation of the distance between the sun and the earth

The earth-sun distance d is simply the average sun-earth distance (also known as the Astronomical Unit or AU) divided by the eccentricity correction factor:

$$d = AU / \varepsilon \tag{126}$$

5.2.2.1 GetATmsST: Calculation of the solar to standard time difference

GetATmsST is a function that calculates the apparent time minus standard time, that is, the difference between the solar time (the time based on the apparent motion of the sun across the sky, with noon corresponding to the time when the sun reaches its zenith) and the clock time (exclusive of daylight savings time). It is given in minutes by the following equation:

$$Solar\ time - Standard\ time = 4 \cdot (\lambda_{st} - \lambda_{loc}) + E \quad (127)$$

where λ_{st} is the longitude of the standard meridian (corresponding to the standard time zone), expressed in degrees, λ_{loc} is the local longitude, also expressed in degrees, and E is the *equation of time:*

$$E = 229.18 * (0.000075 + 0.001868 \cdot \cos(DA) - 0.032077 \cdot \sin(DA) - 0.014615 \cdot \cos(2 \cdot DA) - 0.04089 \cdot \sin(2 \cdot DA)) \quad (128)$$

with DA the day angle.

5.2.2.1 GetLATime and GetLSTime: conversions between standard and solar time

These functions simply calculate the Local Apparent Time (a.k.a. solar time) or the Local Standard Time, given the other quantity and the difference between solar and standard time.

5.2.2.2 GetNormExtra: calculation of normal extraterrestrial irradiance

Normal extraterrestrial irradiance $H_{0,n}$ is the irradiance at the top of the atmosphere, measured perpendicularly to the rays of the sun. It is given by

$$H_{0,n} = H_x \cdot \varepsilon \quad (129)$$

where H_x is the solar constant (1367 W/m^2) and ε is the eccentricity correction factor, given by equation (125).

5.2.2.3 GetAirMass: calculation of the air mass

Air mass is the ratio of the length of the optical path through the atmosphere to the path length vertically upwards. It is given by Kasten's formula [18]:

$$AM = \frac{1}{\cos(z) + 0.15 \cdot (93.885 - z)^{-1.253}} \quad (130)$$

where z is the zenith angle [expressed in degrees].

5.2.2.4 GetSunsetHourAngle: calculation of the sunset or sunrise hour angle

This function calculates the hour angle at sunset (or sunrise, since the day is symmetrical with respect to solar noon). The sunset hour angle ω_s is simply given by its cosine [3]:

$$\cos(\omega_s) = -\tan(\phi) \cdot \tan(\delta) \quad (131)$$

where ϕ is the latitude and δ is the declination

5.2.2.5 GetDayLength: calculation of day length

The length of the day can be directly calculated from the sunset hour angle through:

$$L = 2 \cdot \omega_s / 15 \quad (132)$$

where L is the day length in hours and ω_s is the sunset hour angle expressed in degrees.

5.2.2.6 GetDistance: calculation of the distance between two points on the surface of the earth

This function first calculates the angular distance ξ between the two points according to [19]:

$$\cos(\xi) = \sin(\phi_1) \cdot \sin(\phi_2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\lambda_1 - \lambda_2) \quad \textbf{(133)}$$

Then the distance d between the two points is calculated as:

$$d = R \cdot \xi \quad \textbf{(134)}$$

where R is the mean earth radius (6,371 km).

5.3 Interpolate *Interpolate.cs*

This class contains 2 methods that enable Linear and Quadratic Interpolation for arrays of varied lengths. This class is used exclusively to model Inverter Efficiency (see: 4.6.4).

5.3.1 Equations and Methods Description

5.3.1.1 Linear Interpolation *public static double Linear (…)*

For a given interpolant x_v, CASSYS uses linear interpolation using two arrays x and y each with n entries to determine y_v. First, the program determines the interval number k, such that $x_k < x_v < x_{k+1}$. The following formulae can then be used.

$$y_v = y_k \frac{(x_{k+1} - x_v)}{(x_{k+1} - x_k)} + y_{k+1} \frac{(x_v - x_k)}{(x_{k+1} - x_0)} \quad \textbf{(135)}$$

If x_v lies outside of these bounds, an extrapolation formula is used, ensuring the continuity and smoothness of the overall curve. The following formula is used with *k = 0* when extrapolating to the left and *k = n-1* when extrapolating to the right:

$$y_v = y_i + (x_v - x_{k+1}) \cdot \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)} \quad \textbf{(136)}$$

5.3.1.2 **Quadratic Interpolation** *public static double Quadratic (…)*

27) CASSYS only allows the use of quadratic interpolation [10] for three input values (see 4.6). This allows for the use of the Lagrange interpolating polynomial of degree 2 to obtain $P(x)$ as below:

$$P(x) = \frac{(x - x_2)(x - x_2)}{(x_1 - x_2)(x_1 - x_3)} y_0 + \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} y_1 + \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} y_2 \quad \textbf{(137)}$$

5.3.1.3 Bezier Interpolation *public static double Bezier (…)*

CASSYS allows users to specify a variable number of incidence angles and incidence angle modifiers for a given PV system. The interpolation between the points provided is done using the Bezier interpolation method (see PVArray.cs). For succinctness, the reader is advised to follow the description of this method provided in [10] which explains the process of obtaining the curves in good detail.

5.4 Tilt *Tilt.cs*

The Tilt class contains a set of methods to calculate solar radiation on tilted surfaces, as well as the object to encapsulate them. For the conventions used throughout CASSYS for solar angles, please refer to Appendix: Angle Conventions.

5.4.1 List of Methods

public static double GetCosIncidenceAngle (…)
public static double GetIncidenceAngle (…)
public static double GetProfileAngle (…)
public static double GetApparentSunsetHourAngleEquator (…)
public static void CalcApparentSunsetHourAngle (…)

5.4.2 Physical Models and Descriptions

5.4.2.1 GetCosIncidenceAngle: calculation of the cosine of the incidence angle

The incidence angle θ is the angle between the beam radiation on the array and the normal to the array. Its cosine is calculated as [3]:

$$\cos(\theta) = \cos(z) \cdot \cos(\beta) + \sin(z) \cdot \sin(\beta) \cdot \cos(\gamma - \gamma_s) \tag{138}$$

where z is the solar zenith angle, β is the slope of the array, γ is the solar azimuth angle and γ_s is the azimuth of the surface.

5.4.2.2 GetIncidenceAngle: calculation of the incidence angle

This function simply calls *GetCosIncidenceAngle* and takes the arc cosine of the result.

5.4.2.3 GetProfileAngle: calculation of the profile angle

The profile angle $\alpha_p$is the projection of the solar altitude angle (i.e. the complement of the zenith angle) on a vertical plane perpendicular to the array. Its tangent is calculated as [3]:

$$\tan(\alpha_p) = \frac{\tan(\pi/2 - z)}{\cos(\gamma - \gamma_s)} \tag{139}$$

Where z is the zenith angle, γ is the solar azimuth angle and γ_s is the azimuth of the surface

5.4.2.1 GetApparentSunsetHourAngleEquator: calculation of the apparent sunset or sunrise hour angle for a tilted surface facing the equator

The apparent sunset hour angle ω_{ss} is the hour angle at which direct rays from the sun strike the front of the array. It is calculated as ([3], p. 109):

$$\omega_{ss} = \min \begin{cases} \operatorname{acos}(-\tan\phi \cdot \tan\delta) \\ \operatorname{acos}(-\tan(\phi - s \cdot \beta) \cdot \tan\delta) \end{cases} \tag{140}$$

Where ϕ is the latitude, δ the declination, β the slope of the surface, and s is equal to +1 in the Northern hemisphere and -1 in the Southern hemisphere.

5.4.2.2 CalcApparentSunsetHourAngle: calculation of the apparent sunset and sunrise hour angles for a tilted surface of any orientation

This function computes the apparent sunrise and sunset hour angles on a tilted surface of any orientation, i.e. the hour angle at which direct rays from the sun first strike the surface during the morning and last strike it in the evening. This is a generalization of the formula above, although it is much more complicated. The algorithm is as follows ([3], Eq. 2.20.5e to 2.20.5i). First, calculate quantities A, B and C defined as:

$$\mathrm{A} = \cos\beta + \tan\phi \cdot \cos\gamma \cdot \sin\beta \tag{141}$$

$$\mathrm{B} = \cos\omega_s \cdot \cos\beta + \tan\delta \cdot \cos\gamma \cdot \sin\beta \tag{142}$$

$$\mathrm{C} = \frac{\sin\beta \cdot \sin\gamma}{\cos\phi} \tag{143}$$

where β is the slope of the array, ϕ is the latitude, γ is the solar azimuth angle, δ is the declination, and ω_s is the true sunset angle (see Eq. (132)). Then sunrise and sunset hour angles ω_{sr} and ω_{ss} are calculated as:

$$\omega_{sr} = s \cdot min\left[\omega_s, \operatorname{acos}\frac{AB + C\sqrt{A^2 - B^2 + C^2}}{A^2 + C^2}\right] \tag{144}$$

with

$$\begin{gathered} s = -1 \text{ if } (A > 0 \; and \; B > 0) \; or \; (A \geq B) \\ s = +1 \text{ otherwise} \end{gathered} \tag{145}$$

and

$$\omega_{ss} = s \cdot min\left[\omega_s, \operatorname{acos}\frac{AB - C\sqrt{A^2 - B^2 + C^2}}{A^2 + C^2}\right] \tag{146}$$

with

$$\begin{gathered} s = +1 \text{ if } (A > 0 \; and \; B > 0) \; or \; (A \geq B) \\ s = -1 \text{ otherwise} \end{gathered} \tag{147}$$

These expressions can only be calculated when the discriminant appearing in the square root is positive. If this is not the case, then it's either a case where the surface is always or never illuminated during the day. The calculation of the angle of incidence at noon enables to distinguish between these two possibilities: if the incidence angle is greater than $\pi/2$ then the surface is never illuminated, otherwise it is always illuminated from true sunrise to true sunset.

*****

6 Bibliography

[1] B. Marion, S. MacAlpine, C. Deline, A. Asgharzadeh, F. Toor, D. Riley, J. Stein and C. Hansen, "A Practical Irradiance Model for Bifacial PV Modules," National Renewable Energy Laboratory, 2017.

[2] B. Marion, S. Ayala and C. Deline, "Bifacial PV View Factor model," National Renewable Energy Laboratory, [Online]. Available: https://github.com/NREL/bifacialvf.

[3] J. A. Duffie and W. A. Beckman, Solar Engineering of Thermal Processes 2 Ed., New York: Wiley-Interscience Publication, 1991.

[4] F. Incropera and D. De Witt, Fundamentals of Heat and Mass Transfer, Toronto: John Wiley & Sons., 1990.

[5] Sandia Corporation, "pvpmc.sandia.gov," PVPerformance Modelling Collaborative, 2014. [Online]. Available: https://pvpmc.sandia.gov/modeling-steps/dc-to-ac-conversion/cec-inverter-test-protocol/. [Accessed 23 March 2015].

[6] Y. Goswami, F. Kreith and J. Kreider, Principles of Solar Engineering, Philadelphia: Taylor & Francis, 2000.

[7] D. Faiman, "Assessing the outdoor operating temperature of photovoltaic modules," *Progress in Photovoltaics,* vol. 16, no. 4, pp. 307-315, 2008.

[8] A. Mermoud and T. Lejeune, "Performance assessment of a simulation model for PV modules of any available technology," in *25th European Photovoltaic Solar Energy Conference*, Valencia, Spain, 2010.

[9] V. Quaschning and R. Hanitsh, "Numerical Simulation of Current-Voltage Characteristics of Photovoltatic Systems with Shaded Solar Cells," *Solar Energy,* vol. 56, no. 6, pp. 513 - 520, 1996.

[10] C. Gerald and P. Wheatley, Applied Numerical Analysis, New Delhi: Pearson, 2003.

[11] ASHRAE, ASHRAE Handbook - Fundamentals, ASHRAE, Atlanta, GA, 2013.

[12] W. F. Marion and A. P. Dobos, "Rotation Angle for the Optimum Tracking of One-Axis Trackers," National Renewable Energy Laboratory, 2013.

[13] R. Perez, P. Ineichen, R. Seals, J. Michalsky and R. Stewart, "Modeling daylight availability and irradiance components from direct and global irradiance," *Solar Energy,* vol. 44, no. 5, pp. 271-289, 1990.

[14] ASTM, "Standard Test Method for Reporting Photovoltaic Non-Concentrator System Performance," *ASTM E2848,* 2011.

[15] M. Iqbal, An introduction to solar radiaition, Academic Press Canada, 1993.

[16] J. Braun and J. Mitchell, "Solar Geometry for fixed and tracking surfaces," *Solar Energy,* vol. 31, no. 5, pp. 439-444, 1983.

[17] J. D. McQuiston and F. C. Parker, Heating, Ventilating, and Air Conditioning, John Wiley & Sons, 1982.

[18] F. Kasten, "A New Table and Approximation Formula for the Relative Optical Air Mass.," *Archiv. Fuer Meteorol., Geophys. und Biokl., B,* vol. 14, pp. 206-223, 1965.

[19] J. Meeus, Astronomical Algorithms, Willmann-Bell, 1991.

7 Appendix: Angle Conventions

This appendix summarizes angle conventions used throughout the CASSYS program. All angles (including latitudes and longitudes) are expressed in radians:

7.1 General angles

- Latitude: angular location north or south of the equator, north positive; [-π/2, π /2].
- Longitude: angular location east of Greenwich meridian; [-π, π].
- Declination: angular position of the sun at solar noon, with respect to the plane of the equator, north positive; [-0.4093, 0.4093] (i.e. [-23.45, 23.45] in degrees).

7.2 Solar angles

- Hour Angle: the angular displacement of the sun east or west of the local meridian (15 degrees per hour); morning negative, afternoon positive; [-π, π].
- Zenith Angle: angle between the vertical and the line to the sun, i.e. the angle of incidence of beam radiation on a horizontal surface; [0, π]. Values greater than π /2 indicate that the sun is below the horizon.
- Solar Azimuth Angle: the angular displacement from south of the projection of beam radiation on the horizontal plane; displacements east of south are negative and west of south are positive; [-π, π].

7.3 Orientation of a receiving surface

- Slope: angle between the normal to the surface and the zenith; [0, [(slope > π /2 means surface is facing the ground).
- Collector Azimuth Angle: the deviation of the projection on a horizontal plane of the normal to the surface from the local meridian, with zero due south, east negative, and west positive; [-π, π].
- Angle Of Incidence: the angle between the beam radiation on a surface and the normal to that surface; [0, π] (greater than π/2 means the rays of the sun strike the back of the surface).

7.4 Using CASSYS as a Dynamically Linked Library (DLL)

7.4.1 Python

Python supports the use of C# dll files. The following steps are required to use the CASSYS DLL in python:

1. Install “Python for .NET”

2. Ensure the classes and functions you wish to access from CASSYS are public
3. Make the following changes within the solution settings:
    - Application >> output type = Class library
    - Build >> Register for COM interop = unchecked
4. Compile the CASSYS C# file
5. Move CASSYS.dll (found in bin\debug) to the C:\Python27 directory of your local machine
6. Create python file. A sample python file can be found below.
7. Run python file through command prompt. This can be done as follows:
    - Open command prompt
    - Navigate to folder with python file
    - Run the following command: C:\python27\python SampleFile.py

```
import clr
clr.AddReference('C:/python27/CASSYS.dll')
from CASSYS import CASSYSClass

args = ["Sample Site - Toronto.csyx", "Toronto_ClimateFile_CWEC.csv", "outputFile.csv"]

my_instance = CASSYSClass()

my_instance.Main(args)
```

Figure 11 Accessing CASSYSClass and executing the main method through Python using CASSYS.dll

7.4.2 C#

A DLL can also be used within C#. To access the classes within the CASSYS.dll, the DLL solution must be added the project as a reference. This is done as follows:

1. Within the solution explorer right click on the references tab and select “add reference”
2. Click browse and select the CASSYS.dll from wherever you chose to save it

All public classes and their respective public parameters/methods can now be accessed within the C# project. Sample code can be found below:

```
using System;
using CASSYS;

namespace TrialDLL
{
    class Program
    {
        static void Main(string[] args)
        {
            String[] arguments = new String[1];
            arguments[0] = "fullFilePath/Sample Site - Toronto.csyx";

            CASSYSClass.Main(arguments);
        }
    }
}
```

Figure 12 Accessing CASSYSClass and executing the main method through C# using CASSYS.dll

7.4.3 Excel VBA

A DLL can be used within VBA. The following steps are required to use the CASSYS DLL in python:

1. Make the following changes within the CASSYS C# solution settings:
    - Application >> output type = Class library
    - Build >> Register for COM interop = checked
2. Compile the CASSYS C# file
3. Create a .tlb file. This can be done as follows:
    - Run command prompt as administrator
    - Navigate to the directory with the CASSYS.dll file
    - run the following lines of code:
    - C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe CASSYS.dll
    - C:\Windows\Microsoft.NET\Framework\v4.0.30319\RegAsm.exe -tlb -codebase CASSYS.dll
4. Add CASSYS.dll as a reference in your VBA workbook. This is done as follows:
    - Open the workbook and enter the development environment
    - In the ribbon select Tools >> References
    - Click browse and select the CASSYS.tlb file you created
    - Find "CASSYS" in the list of available References and ensure it has a checkmark in front of it

All public classes and their respective public parameters/methods can now be accessed within the C# project. Sample code can be found below:

```
Sub foo()
    Dim myInstance As CASSYS.CASSYSClass
    Set myInstance = CreateObject("CASSYS.CASSYSClass")

    Dim args As Variant
    args = Array("siteFile.csyx", "climateFilePath.csv", "outputFilePath.csv")

    myInstance.Main (args)
End Sub
```

Figure 13 Accessing CASSYSClass and executing the main method through VBA using CASSYS.dll and CASSYS.tlb