



## Bachelorarbeit

# Modellierung eines Wärmeschichtspeichers mit Solareinbindung in Excel/VBA

Teil: Solare Anbindung

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Bachelor of Science

eingereicht am  
Institut für Energietechnik und Thermodynamik  
an der  
Technischen Universität Wien

von Autor

Blabensteiner Rainer  
Matrikelnummer: 0625030

unter Anleitung von

Projektass. Dipl.-Ing. Dietrich Wertz

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Werner Andreas

Wien, am 21. November 2011

---

(Unterschrift Verfasser/in)

---

(Unterschrift Betreuer/in)



Ich versichere, dass ich diese Bachelorarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Wien , den 21. November 2011

Blabensteiner Rainer

## Zusammenfassung

Die Modellierung eines Thermischen Wärmeschichtspeicher ist ein komplexes und umfangreiches Unterfangen mit vielen Randbedingungen. Um diesem System Herr zu werden, soll diese Arbeit in die theoretischen Hintergründe der Modellierungen Einblick gewähren. Die Auswertung zwischen Modell und Simulation wurde mit Excel/VBA umgesetzt, wobei mit dem in dieser Arbeit entstandenen Simulationstool *SolarSimulation* eine einfache Auslegung eines Schichtspeichers mit Solareinbindung möglich gemacht werden soll. Dazu ist in einer ausführlichen Bedienungsanleitung erklärt, wie mit den einzugebenden Randbedingungen des Systems umzugehen ist.

## abstract

The modeling of a thermal stratified heat storage seems to be really complex, not at least because of its amount of boundary conditions. For better understanding, this work is made to provide the theoretical background of the modeling. It has to be mentioned, that the *SolarSimulation* i.e. the simulation tool is a simplification of the correlation between the model and simulation, all implemented in Excel/VBA. The following detailed manual explains the usage of the whole programm, as described above.

# Inhaltsverzeichnis

<b>Titelseite</b>	i
<b>Erklärung zur Bachelorarbeit</b>	iii
<b>Zusammenfassung</b>	iv
<b>Inhaltsverzeichnis</b>	vi
<b>Outline der Bachelorarbeit</b>	vii
<b>1. Einleitung</b>	1
1.1. Motivation . . . . .	1
1.2. Zielsetzung . . . . .	1
1.3. Überblick über das Simulationstool „Solarsimulation“ . . . . .	1
1.4. Ausblick . . . . .	2
<b>2. Modellbildung</b>	3
2.1. Modellierung des Gesamtsystems . . . . .	3
2.2. Modell: Sonnenstand . . . . .	4
2.2.1. Sonnenazimut . . . . .	4
2.2.2. Sonnenhöhe . . . . .	4
2.3. Modell: Einstrahlung auf Kollektor . . . . .	5
2.3.1. Strahlungsaufteilung und Globalstrahlung . . . . .	5
2.3.2. Globalstrahlung . . . . .	5
2.3.3. Klimadatenkatalog . . . . .	6
2.3.4. Winkelabhängigkeit der direkten Strahlung auf die geneigte Fläche	7
2.4. Modell: Energieabgabe des Kollektors . . . . .	7
<b>3. Mathematische Modellbildung der Solareinbindung</b>	9
3.1. Berechnungsmodelle für Sonnenstand . . . . .	9
3.1.1. Berechnungsmodell 1 . . . . .	9
3.1.2. Berechnungsmodell 2 und 3 . . . . .	10
3.1.3. Vergleich der Sonnenstandsberechnungsmodelle . . . . .	11
3.2. Berechnungsmodell der Kollektoreinstrahlung . . . . .	11
3.2.1. Einstrahlungsmodell auf die horizontale Fläche . . . . .	11
3.2.2. Einstrahlungsmodell auf die geneigte Fläche . . . . .	13
3.2.3. Strahlungdatensatz KS und ER . . . . .	17
3.3. Berechnungsmodelle für den Winkelfaktor . . . . .	17
3.4. Berechnungsmodelle für die Kollektorwirkungsgradkurve . . . . .	18
<b>4. VBA - Programm</b>	20
4.1. Allgemeines zur Programmiersprache . . . . .	20

## Inhaltsverzeichnis

4.2.	Handhabung und Anwendung des Programms . . . . .	20
4.2.1.	Werkzeugkasten „neues Projekt“ . . . . .	21
4.2.2.	Werkzeugkasten „Berechnung“ . . . . .	23
4.2.3.	Werkzeugkasten „Visualisierung“ . . . . .	23
4.2.4.	Werkzeugkasten „Daten Ausgabe“ . . . . .	24
4.2.5.	Projektdaten Eingabe . . . . .	24
4.3.	Aufbau und Unterteilung . . . . .	25
4.3.1.	Module und Klassen im Programmcode . . . . .	26
4.3.2.	Die wichtigsten Subs und Funktionen . . . . .	27
4.4.	<i>MS Excel Makro</i> tauglich machen . . . . .	30
4.5.	Weitere Ausblicke zur Programmiersprache VBA . . . . .	30
<b>A.</b>	<b>Struktogramme</b>	<b>31</b>
A.1.	Struktogramme DIN5034 . . . . .	31
A.2.	Struktogramme Sunae . . . . .	32
<b>B.</b>	<b>Tabellen</b>	<b>34</b>
B.1.	Albedo . . . . .	34
B.2.	Parametertabellen für Perez Algorithmus . . . . .	34
<b>C.</b>	<b>Diagramme von SolarSimulation</b>	<b>35</b>
<b>D.</b>	<b>Bilder</b>	<b>40</b>
D.1.	Allgemeine Abbildungen . . . . .	40
<b>E.</b>	<b>Code des VBA Programmes SolarSimulation</b>	<b>42</b>
E.1.	1. Kapitel Allgemeine Konstanten und 2. Allg. Deklarationen . . . . .	42
E.2.	3. Kapitel Allgemeine Funktionen und Subs . . . . .	45
E.3.	4. Kapitel Initialisierung des Programmes . . . . .	51
E.4.	5. Kapitel Hauptprogrammsequenzen . . . . .	53
E.5.	6. Kapitel Visualisierende Programmsequenzen . . . . .	71
E.6.	7. Kapitel Daten Ein- Ausgabe Erzeugung . . . . .	97
	<b>Abkürzungsverzeichnis</b>	<b>112</b>
	<b>Abbildungsverzeichnis</b>	<b>113</b>
	<b>Literaturverzeichnis</b>	<b>115</b>

# Outline der Bachelorarbeit

## **Kapitel 1: Einleitung**

Einleitende Worte, Zielsetzung sowie Motivation und Überblick sollen den Rahmen der Arbeit beschreiben.

## **Kapitel 2: Modellbildung**

Das Physikalische Modell des gesamten Systems wird in einzelne Modelle unterteilt und dargestellt.

## **Kapitel 3: Mathematisches Modellbildung der Solareinbindung**

Mathematische Formulierung der Modelle und genauere Beschreibung der einzelnen Terme.

## **Kapitel 4: VBA - Programm**

Allgemeine Worte sowie eine kleine Anleitung sollen die Bedienung erläutern, weiters wird ein Überblick des VBA-Codes dargestellt um den Umgang mit dem Programm zu erleichtern.

## **Anhang A: Struktogramme**

Veranschaulichung des Prozedurverlaufs DIN5034 und Sunae Algorithmus.

## **Anhang B: Tabellen**

Tabellarische Auflistung von Albedo und Perez Parametern.

## **Anhang C: Diagramme von SolarSimulation**

Erzeugte Diagramme des VBA-Programmes vom Teil „Solare Anbindung“.

## **Anhang D: Bilder**

Allgemeine Bilder

## **Anhang E: Code des VBA Programmes SolarSimulation**

VBA-Code des Modules Kollektor.

# Kapitel 1.

## Einleitung

Im Zuge dieser Arbeit soll die Modellierung eines Wärmeschichtspeichers mit Solaranlagenbindung entstehen. Diese Modellbildung wird in einem *VBA* Programm simuliert und berechnet. Um den Rahmen einer Bachelorarbeit nicht zu sprengen, wurde die Arbeit in zwei Teilsysteme aufgeteilt. Der Teil thermischer Schichtspeicher *TESS* wird in der Bachelorarbeit von Herrn Martin Ortner [7] behandelt und somit in dieser Arbeit nicht im Detail beschrieben. Die Modellbildung sowie die mathematische Beschreibung und Erklärungen zum VBA - Programm für den Teil der solaren Anbindung an den *TESS*, sind in dieser Arbeit zu finden. Um einen besseren Überblick beider Arbeiten zu gewähren, wird diese Niederschrift an die inhaltliche Struktur von Herrn Martin Ortner angelehnt.

### 1.1. Motivation

Mit steigender Nachfrage nach regenerativen Energiequellen wird die Technik ständig gefordert, effizientere Methoden zu entwickeln. Ein Weg, den diese Arbeit verfolgt ist, durch thermische Schichtspeicherung von solarer Wärme und der gezielten, an die Nachfrage der Verbraucher angepassten Abgabe, eine Effizienzsteigerung zu erzielen und somit ein größeres Energieeinsparungspotential zu erlangen. Ein Weg um projektierte Anlagen möglichst effizient zu gestalten, ist die vorherige Einschätzung des Energieertrages. Dazu bieten sich Simulationsprogramme an, mit denen erste Prognosen über einen längeren Zeitraum, ohne größere finanzielle und technische Aufwände, erstellt werden können.

### 1.2. Zielsetzung

Ziel der beiden Bachelorarbeiten ist es, ein Simulationstool<sup>1</sup> eines Wärmeschichtspeichers mit Solareinbindung in EXCEL/VBA zu modellieren. Dieses Simulationstool<sup>2</sup> sollte einen benutzerfreundlichen Aufbau besitzen und mit den klassischen Excel-Befehlen<sup>3</sup> weiter bearbeitbar sein. Die Ergebnisse sollen anhand von Diagrammen und deren Datensätzen anschaulich gemacht werden. Insbesondere ein Diagramm, welches die Schichtung des *TESS* visuell darstellt und vom Teil *TESS* erstellt wird, soll dazu dienen.

### 1.3. Überblick über das Simulationstool „Solarsimulation“

Um eine Optimierung der zu projektierenden Anlagen zuzulassen, wird hierbei auf viele Eckdaten Rücksicht genommen.

---

<sup>1</sup>mit dem Namen *Solarsimulation*

<sup>2</sup>Basierend auf einem Makro, das im Hintergrund einer Excel-Anwendung arbeitet.

<sup>3</sup>ohne VBA Programmierkenntnisse

## **Überblick:**

- Stündliche Auswertung sämtlicher berechneter Daten
- Diagramm zur Verbesserung der Kollektorausrichtung
- Sonnenstandsdiagramm zur Festlegung der Sonnenscheindauer
- Projektspezifische Standorte, Auswahl der Anlage mit Rücksicht auf die Klimadaten<sup>4</sup>
- unterschiedliche Solar-Kollektortypen können berechnet werden
- variable Kollektorfläche
- Benötigte Heizenergie wird durch Einlesen eines Datensatzes individuell an die Projekte angepasst
- Geometrie des Speichers
- Variable Kesselleistung der Zusatzheizung
- Visualisierung des geschichteten Speichers
- spezifische Regelparameter des *TESS*

## **1.4. Ausblick**

Weitere Ausbaumöglichkeiten des Simulationstools sind beispielsweise die Abschattungen von anderen Gebäuden bzw. Bäumen auf den Kollektor oder die Aufwärmung von verschiedenen ausgerichteten Gebäudewänden durch Sonneneinstrahlung<sup>5</sup>.

Für diese Weiterentwicklung ist aber zu überlegen welche Programmiersprache verwendet wird, da Excel/VBA im Hinblick auf numerische Modelle und große Datenmengen Grenzen gesetzt sind. Ebenfalls ist zu berücksichtigen, dass von Microsoft ein inoffizielles Support-Ende für *VBA* angekündigt wurde<sup>6</sup>.

---

<sup>4</sup>insgesamt 12 Standorte, davon ein variabler und 11 fixe Standorte

<sup>5</sup>z.B. zur Berechnung der Kühllast oder anderer Energiebedarfsberechnungen.

<sup>6</sup>genaueres siehe 4.5

# Kapitel 2.

## Modellbildung

Um die Simulation in VBA möglich zu machen, erfolgt zuerst eine Modellierung des Gesamtsystems der Solaranlage. Diese besteht aus zwei Teilen, welche wie oben erwähnt in zwei Arbeiten aufgeteilt wurden.<sup>1</sup>

### 2.1. Modellierung des Gesamtsystems

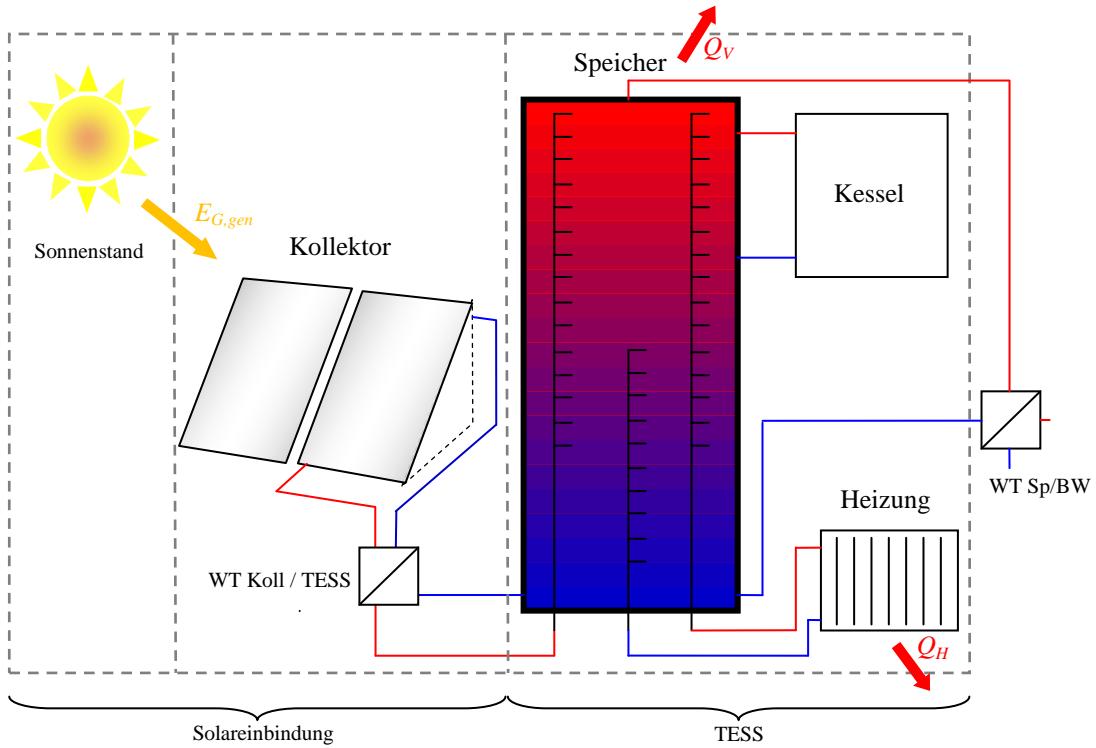


Abbildung 2.1.: Schematische Einteilung der Solaranlage in die Teilsysteme

In Abbildung 2.1 werden die modellierten Komponenten in das Teilsystem **TESS** und Solareinbindung unterteilt. Dies ist nötig um den großen programmiertechnischen Aufwand umzusetzen. Um die Schichtung des **TESS**, die als Ziel der Simulation dient, darstellen zu können, benötigt man Randwerte wie den Sonnenstand, die Einstrahlung auf den Kollektor und die damit verbundene stündliche Leistung, mit welcher der **TESS** über das Jahr beschickt wird. Diese Randwerte liefert das Teilsystem Solareinbindung, welches in die Modelle Sonnenstand, Einstrahlung auf Kollektor und Kollektor Einspeisung unter-

<sup>1</sup>Die Beschreibung des Teilsystems **TESS** erfolgt in der Arbeit von Herrn Martin Ortbauer siehe [7]

teilt wird. Eine nähere Betrachtung der Modelle soll im Folgenden anschaulich gemacht werden.

## 2.2. Modell: Sonnenstand

Die Berechnung des Sonnenstandes ist eine Grundlage für die weiteren Simulationen des Programms, nur mit Hilfe des Sonnenstandes lassen sich Aussagen über die Einstrahlung am Kollektor treffen. Unter dem Begriff „Sonnenstand“ ist der Punkt auf der Sonnenbahn definiert, an dem die Sonne beobachtet werden kann. Dieser Punkt wird durch zwei Winkel, den Sonnenazimut und die Sonnenhöhe beschrieben.

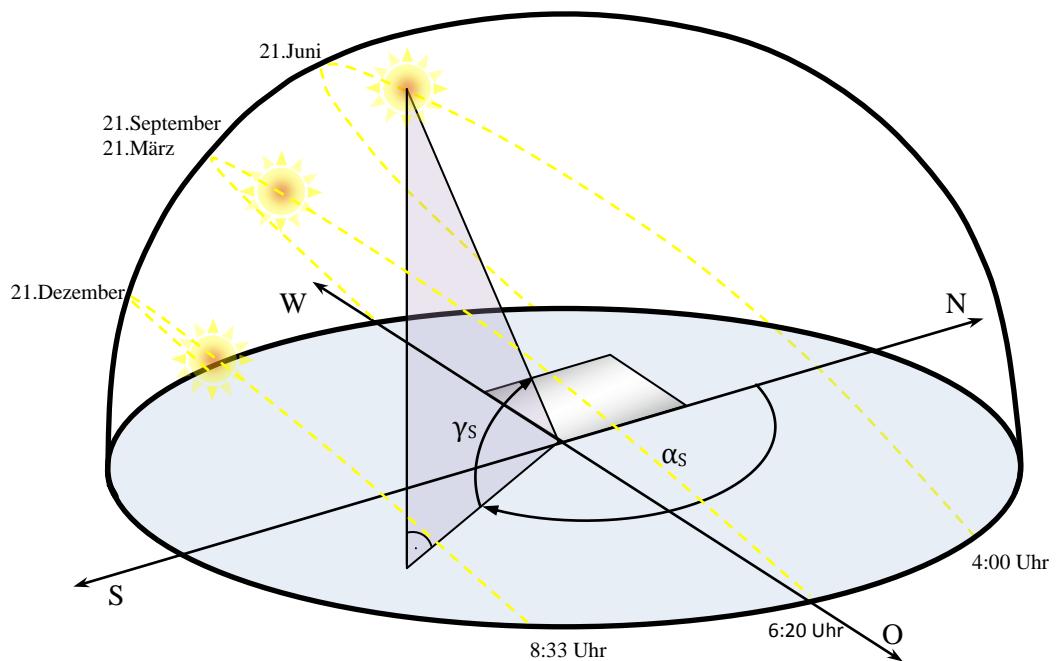


Abbildung 2.2.: Skizze des Verlaufs der Sonnenbahn

### 2.2.1. Sonnenazimut

Der Sonnenazimut  $\alpha_S$  oder auch Azimut (im Englischen Azimuth) siehe Abbildung 2.3, wird nach DIN 5034 Teil 2 [8] als Winkel zwischen der geographischen Nordrichtung und dem Vertikalkreis durch den Sonnenmittelpunkt ( $0^\circ$  bis  $360^\circ$ ), abhängig von Tageszeit, Jahreszeit und geographischer Breite des betreffenden Ortes verstanden. ( $0^\circ = \text{N}$ ,  $90^\circ = \text{O}$ ,  $180^\circ = \text{S}$ ,  $270^\circ = \text{W}$ )

### 2.2.2. Sonnenhöhe

Die Sonnenhöhe  $\gamma_S$  oder nur Höhe (im Englischen Elevation) in Abbildung 2.4, ist nach DIN 5034 Teil 2 [8] der Winkel zwischen dem Sonnenmittelpunkt und der Horizontalebene, vom Beobachter aus betrachtet, abhängig von Tageszeit, Jahreszeit und geographischer Breite des betreffenden Ortes.

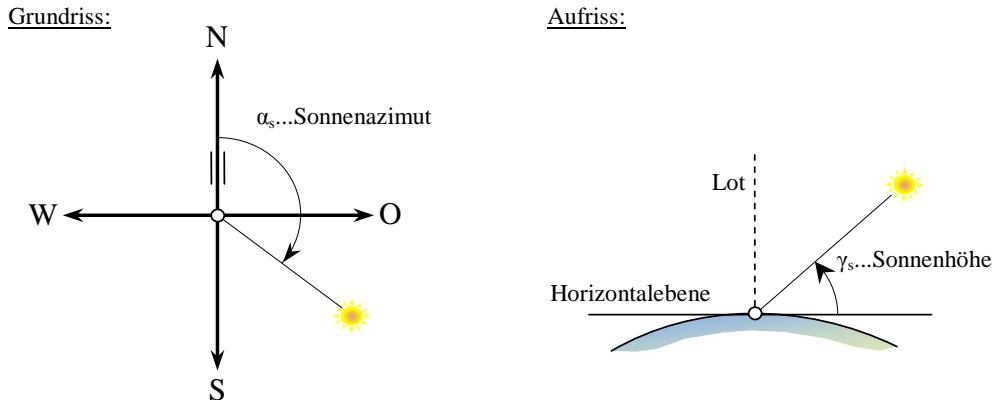


Abbildung 2.3.: Azimut

Abbildung 2.4.: Sonnenhöhe

## 2.3. Modell: Einstrahlung auf Kollektor

Die Gestaltung von solarthermischen Systemen hängt von der sorgfältiger Bewertung der Solarstrahlung an bestimmten Standorten ab. Dieses Kapitel enthält anschauliche Erklärungen der Einstrahlung auf die geneigte sowie die horizontale Fläche.

### 2.3.1. Strahlungsaufteilung und Globalstrahlung

Die Leistung der Sonne, die durch Fusion von Wasserstoffkernen zu Helium frei wird, beträgt ca.  $3,8 \cdot 10^{23}$  kW. Diese Energie wird durch Wärmeleitung und Konvektion vom Kern der Sonne zu den äußeren Schichten transportiert und dort von der Oberfläche emittiert. Die effektive Oberflächentemperatur der Sonne beträgt ca. 5778 K. Die emittierte Energie der Sonne wird als Spektrum elektromagnetischer Wellen freigesetzt, beginnend bei langwelligen Radiowellen über sichtbares Licht bis hin zur Röntgenstrahlung. Die Strahlung, die in die Atmosphäre eintritt, kann wie in Abb. 2.5 schematisch dargestellt werden. Was hier aber nicht weiter ausgeführt wird. Für genauere Erläuterungen verweist Quaschning[16] auf Kleemann und Meliß[6]. Es resultiert eine direkte Strahlung, eine Strahlung durch Brechung in Wolken und eine von der Erdoberfläche reflektierte Strahlung (Albedo), die zusammen als Gesamteinstrahlung auf den geneigten Kollektor definiert werden.

### 2.3.2. Globalstrahlung

Als Globalstrahlung wird die Summe der eintreffenden Solarstrahlung auf die horizontale Fläche definiert. Die Summe setzt sich wie in 2.6 dargestellt aus folgenden Teilen zusammen:

- über direkten Weg eintreffende Solarstrahlung (Direktstrahlung), und
- Strahlung, die durch Dispersion an Wolken, Wasser- und Staubteilchen entsteht und als Diffusstrahlung bezeichnet wird.

Diese Größen werden auch von den meisten Wetterstationen mit Messeinrichtungen aufgenommen und können, wie in diesem Programm, zur Simulation verwendet werden. Ein Charakteristikum für die Einstrahlung der Sonne auf die Erdoberfläche ist die

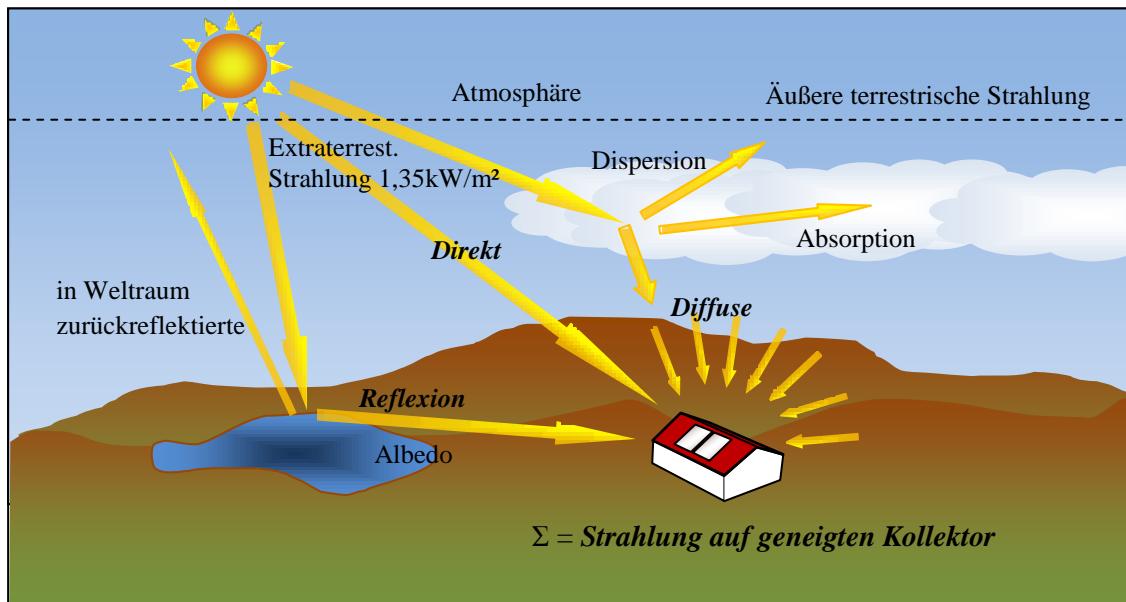


Abbildung 2.5.: Schematische Darstellung der Sonneneinstrahlung auf einen Kollektor

Globalstrahlung. Diese Größe ist wichtig, um aussagekräftige regionale und lokale Einstrahlung der Sonne als numerischen Wert festzuhalten. Die Daten werden in der Regel von Wetterstationen aufgezeichnet oder können mit Phyrhelimeter und Pyranometer mit zusätzlichen Datenloggern zu Klimadatensätzen zusammengefasst und protokolliert werden.

### 2.3.3. Klimadatenkatalog

Zu Simulationszwecken oder realistischen Prognosen eignen sich überwiegend Klimadatensätze, die über mehrere Jahre aufgezeichnet, analysiert und dann zu Durchschnittswerten gebildet werden. Die im Programm verwendeten Klimadatensätze wurden in Perioden von 1991 bis 2005 ausgewertet. Der Klimadaten-Katalog im Tabellenblatt **Projektname\_Klimadaten** nach [1], beinhalten folgende Standorte mit Angabe der Seehöhe:

1. Wien, Hohe Warte SH198m
2. Wien, Innere Stadt SH171m
3. Klagenfurt SH450m
4. Bregenz SH424m
5. Eisenstadt SH184m
6. Lienz SH659m
7. Linz SH263m
8. St.Pölten SH270m
9. Innsbruck, Uni SH578m
10. Großengersdorf SH153m

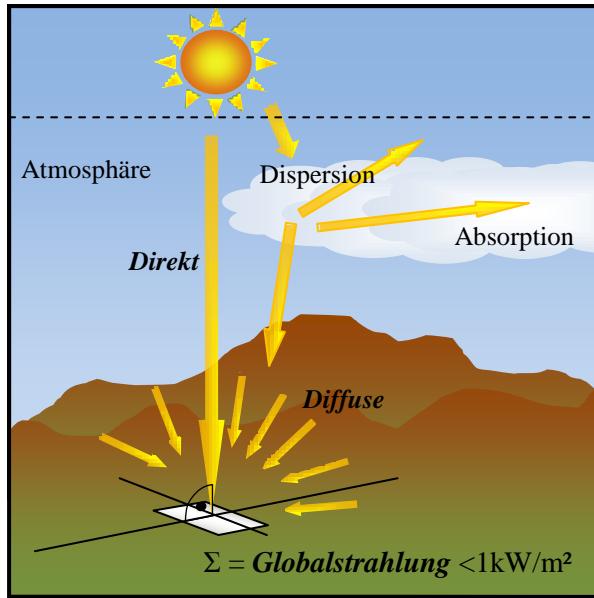


Abbildung 2.6.: Schematische Darstellung der Globalstrahlung

11. Graz, Uni SH366m

12. Vergleichsstandort

Die angegebenen Standorte sind im Programm genau in dieser Reihenfolge in der Ribbonleiste SolarSimulation unter „Standorte der Anlage“ im Dropdown-Menü auszuwählen. Näheres siehe Kapitel 4.2.1

#### 2.3.4. Winkelabhängigkeit der direkten Strahlung auf die geneigte Fläche

Die aus direkter Strahlung auf den geneigten Kollektor eintreffende Strahlung wird nur dann maximal vom Kollektor absorbiert, wenn die Direktstrahlung der Sonne normal auf die geneigte Fläche steht.<sup>2</sup> Dies trifft auf feststehende Kollektoren im Allgemeinen nicht zu, da durch die Winkelabweichung der Direktstrahlung auf die Normalen des Kollektors, definiert als Einfallswinkel, eine Reflexion an der transparenten Abdeckung<sup>3</sup> zur Folge hat. Die damit verbundene nicht vollständige Absorbtion der Direktstrahlung, wird mittels dem Winkelfaktor, der im mathematischen Modell siehe 3.3 beschrieben wird, berücksichtigt.

### 2.4. Modell: Energieabgabe des Kollektors

Das letzte wichtige Modell zur Energieübertragung an die Systemgrenze des *TESS* kann wie folgt beschrieben werden. Die Aufgabe der Strahlungsumwandlung übernimmt der im Kollektor integrierte Absorber, der eine wärmeleitende, gut absorbierende Fläche besitzt. In der Regel besteht der Absorber aus aneinander gereihten Rohren in welchen ein Wärmeträgermedium<sup>4</sup> fließt. Der Energiefloss durch einen Kollektor kann prinzipiell in

<sup>2</sup>Man spricht von optimaler Kollektorausrichtung bzw. Anlagen, die sich nach dem Sonnenstand ausrichten.

<sup>3</sup>In der Regel ein Solar-Sicherheitsglas.

<sup>4</sup>in unserem Fall Wasser mit oder ohne Frostschutz, wird vom Programm nicht berücksichtigt

zwei Teilschritte getrennt werden. In Abbildung 2.7 sind diese Energieflüsse schematisch dargestellt.

Der erste Schritt wird durch die optischen Eigenschaften des Kollektors bestimmt. Dieser beschreibt den Weg der solaren Strahlung durch die transparente Abdeckung und über Reflexion bis zu dem Moment, wo die Strahlung in Wärme umgewandelt und vom Absorber aufgenommen wird.

Der zweite Schritt berücksichtigt das thermische Verhalten des Kollektors. In diesem werden die thermischen Verluste ab der Wandlung der Strahlung am Absorber bis zu dem Punkt, an dem das Fluid den Kollektor verlässt, betrachtet.

Da dieser Energiefluss stark von der Bauweise des Kollektortyps abhängt, ist es sinnvoll, einen Wirkungsgrad zu verwenden. Der nach technischem Letztstand bekannte Wirkungsgrad wurde aus EN 12975-2 [9] entnommen und wird im mathematischen Modell siehe 3.4 formuliert. Durch ihn soll eine Annäherung an das reale Betriebsverhalten eines Solarkollektors modelliert werden.

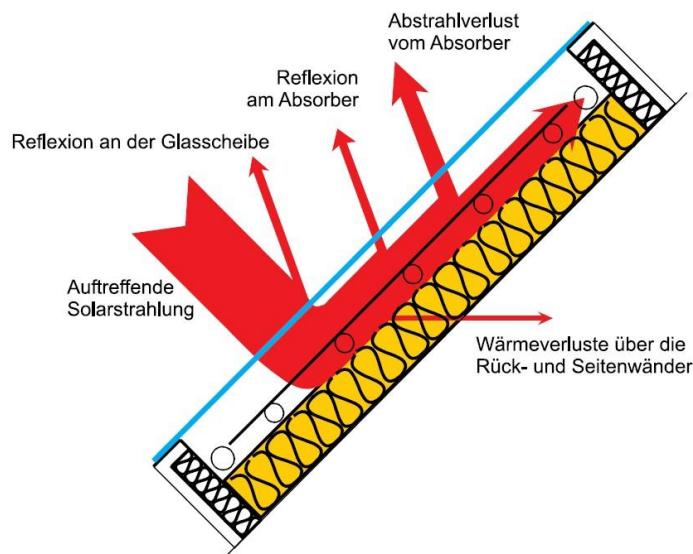


Abbildung 2.7.: Energiefluss durch den Kollektor mit optischen und thermischen Verlusten[2]

# Kapitel 3.

## Mathematische Modellbildung der Solareinbindung

### 3.1. Berechnungsmodelle für Sonnenstand

#### 3.1.1. Berechnungsmodell 1

Das erste Berechnungsmodell stützt sich auf den Formalismus vom Klimadatenkatalog des Bundesministeriums für Bauten und Technik [18]. Mit diesem Modell kann eine genaue Berechnung von Azimut und Sonnenhöhe für Standorte nach beliebiger geographischer Lage und zu beliebigen Zeitpunkten innerhalb der Zeitzone Mitteleuropäische Zeit (MEZ), und damit für österreichisches Bundesgebiet errechnet werden.

$$\alpha_S = 180^\circ + \arctan \left( \frac{-\cos(\delta) \cdot \sin(\sigma)}{-\cos(\delta) \cdot \sin(\varphi) \cdot \cos(\sigma) - \sin(\delta) \cdot \cos(\varphi)} \right) \quad (3.1)$$

$$\gamma_S = \arcsin (\sin(\delta) \cdot \sin(\varphi) - \cos(\delta) \cdot \cos(\varphi) \cdot \cos(\sigma)) \quad (3.2)$$

In Formel 3.1 ist darauf zu achten, dass der *arctan* nur eine Winkelabweichung zwischen  $-90^\circ$  und  $+90^\circ$  liefert. Daher wird für Sonnenstände im Sommerhalbjahr vom 21.03 - 23.09 mit einer Winkelabweichung von über  $90^\circ$  aus Südrichtung eine Winkelkorrektur benötigt, die im Programmcode implementiert ist. Zur Berechnung von Sonnenazimut, Gleichung 3.1 und Sonnenhöhe 3.2, werden die Beziehungen aus folgenden Gleichungen 3.3 bis 3.8 benötigt. Sie ergeben fortlaufend angewendet die fehlenden Variablen der obigen Gleichungen 3.1 und 3.2.

$$\tau = \frac{360^\circ}{365 \cdot [d]} \cdot n \quad (3.3)$$

$\tau$  ... Hilfswinkel in  $[\circ]$

$n$  ... Tagesnummer eines Jahres (z.B.: 1.1 n=1; 32.12 n=365) in [d]

Mit der Zeitgleichung 3.4 lassen sich anschaulich die Differenzen zwischen Sonnenuhr und Mittlerer Ortszeit (MOZ) darstellen. Ein Beispiel ist in den Ergebnissen in Abbildung C.8 ersichtlich.

$$\begin{aligned} z = & 0,008 \cdot \cos(\tau) - 0,122 \cdot \sin(\tau) - \\ & - 0,052 \cdot \cos(2 \cdot \tau) - 0,157 \cdot \sin(2 \cdot \tau) - \\ & - 0,001 \cdot \cos(3 \cdot \tau) - 0,005 \cdot \sin(3 \cdot \tau) \end{aligned} \quad (3.4)$$

$z$  ... Zeitgleichung in [h]

Für die wahre Ortszeit gilt die Beziehung:

$$WOZ = MEZ + z + \frac{1 \cdot [h]}{15^\circ} \cdot \lambda - 1 \cdot [h] \quad (3.5)$$

$\tau$	...	Hilfswinkel in $[\circ]$
$WOZ$	...	wahre Ortszeit in [h]
$MEZ$	...	Mitteleuropäische Zeit (Normalzeit am Standort) in [h]
$\lambda$	...	Geo. Länge des Standortes in $[\circ]$
$\varphi$	...	Geo. Breite des Standortes in $[\circ]$

$$\sigma = \frac{15}{1 \cdot [h]} \cdot WOZ \quad (3.6)$$

$\sigma$  ... Stundenwinkel in  $[\circ]$

Als Winkel zwischen den Verbindungslinien Sonne-Erde und der Verbindungslinie Sonne-Frühlingspunkt (Stand der Erde 21. März) wird die ekliptikale Länge in Gleichung 3.7 nach[18] definiert.

$$\omega = \frac{0,98630}{1 \cdot [d]} \cdot (n - 2,8749 \cdot [d]) + 1,9137 \cdot \sin \left( \frac{0,98630}{1 \cdot [d]} \cdot (n - 2,8749 \cdot [d]) \right) + 102,06 \quad (3.7)$$

$\omega$  ... Ekliptikale Länge in  $[\circ]$

Der Winkel zwischen Äquatorebene und der Verbindungslinie Sonne-Erde in Gleichung 3.8 wird Sonnen-Deklination genannt. Eine vom *VBA* Programm *SolarSimulation*<sup>1</sup> erzeugtes Diagramm ist in Abbildung C.7 zu sehen. Die Sonnendeklination  $\delta$ , bewegt sich im Laufe eines Jahres zwischen  $+23^\circ 26,5'$  im Sommer und  $-23^\circ 26,5'$  im Winter.

$$\delta = \arcsin(-0,3979 \cdot \sin(\omega)) \quad (3.8)$$

$\delta$  ... Sonnen-Deklination in  $[\circ]$

### 3.1.2. Berechnungsmodell 2 und 3

Als Vergleich und zu Kontrollzwecken wurde das Berechnungsmodell 2 nach DIN5034 [8] und Berechnungsmodell 3, (Sunae - Algorithmus nach [12]) in das Programm implementiert. Für beide Berechnungsmodelle steht ein Struktogramm von Quaschning [16] im Anhang. Der Programmcode für diese beiden Modelle wurde aus einem Excel - *VBA* - File [15] in das Programm als Modul *Sonnenposition\_Quaschning* implementiert. Dieses ursprüngliche Modul konnte jedoch nur stündliche Werte berechnen, was für die Simulation nicht ausreicht. Damit es für die Simulation und zum Vergleich einsetzbar ist, wurde im Modul Kollektor, das Sub *AzimutSonnenhoehe\_DIN\_SUNAE* erstellt, welches eine Simulation über ein volles Jahr ermöglicht. Mit beiden Modellen lassen sich für beliebige geographische Standorte, sowie mit zusätzlicher Angabe der Zeitzonen, Standorte außerhalb Österreichs berechnen. Die Eingabe der Zeitzone muss im Tabellenblatt (**Projektname\_Eingabe**) getätigt werden. Ebenso muss hier die Wahl des Berechnungsmodells getroffen werden.

<sup>1</sup>Kann im Programm mit der Visualisierungstaste Sonnenstand erzeugt werden

### 3.1.3. Vergleich der Sonnenstandsberechnungsmodelle

Ein Vergleich dieser drei Modelle zeigt keinen wesentlichen Unterschied, es geht sogar daraus hervor, dass Modell 1 nach *BMT* und Modell 2 nach DIN ident sind. In Abbildung C.9 ist dies zu sehen, da die blaue Linie von der roten überdeckt wird.

## 3.2. Berechnungsmodell der Kollektoreinstrahlung

Das Berechnungsmodell der Kollektoreinstrahlung stützt sich auf folgende Berechnungsmodelle.

### 3.2.1. Einstrahlungsmodell auf die horizontale Fläche

Wie in 4.3.2 erläutert, lässt sich die Globalstrahlung auf die horizontale Fläche mit Gleichung 3.9 errechnen.

$$E_{G,hor} = E_{diff,hor} + E_{dir,hor} \quad (3.9)$$

$E_{G,hor}$	...	Globalstrahlung auf die horizontale Fläche in $[W/m^2]$
$E_{diff,hor}$	...	Diffusstrahlung auf die horizontale Fläche in $[W/m^2]$
$E_{dir,hor}$	...	Direktstrahlung auf die horizontale Fläche in $[W/m^2]$

Je nachdem, welche Strahlungsanteile unter Tabellenblatt **Klimadaten** => Vergleichsstandort (siehe Tabelle 4.1), angegeben sind, wird im Modul Kollektor der fehlende Term mit obiger Gleichung umgeformt und errechnet. Für den Fall, dass nur Globalstrahlung gegeben ist, kann mit den folgenden Gleichungen 3.9 bis 3.13 nach Quaschning[16], aus der Solarkonstanten  $E_{0,hor}$ , der Globalstrahlung  $E_{G,hor}$  auf die horizontale Fläche und der Sonnenhöhe  $\gamma_S$  eine Differenzierung zwischen diffuser Strahlung  $E_{diff,hor}$  und direkter Strahlung  $E_{dir,hor}$  errechnet werden.

$$k_T = \frac{E_{G,hor}}{E_{0,hor}} \quad (3.10)$$

$k_T$	...	Klarkeindex (clearness index) [-]
$E_{0,hor}$	...	Extraterrestrische Bestrahlungsstärke auf die horizontale Fläche $[W/m^2]$

Die auf die horizontale Fläche bezogene extraterrestrische Bestrahlungsstärke  $E_{0,hor}$ , kann aus der Sonnenhöhe  $\gamma_S$  und der Solarkonstanten  $E_0$  berechnet werden.

$$E_{0,hor} = E_0 \cdot \sin(\gamma_S) \quad (3.11)$$

$$E_0 \quad \dots \quad \text{Solarkonstante } [W/m^2]$$

Die Solarkonstante  $E_0$  ist die über viele Jahre gemittelte extraterrestrische Sonnenbestrahlungsstärke, die 1982 von der Weltorganisation für Meteorologie in Genf auf:

$$E_0 = 1367 \pm 2 W/m^2 \quad (3.12)$$

festgelegt wurde. Definiert wird sie als die, bei mittlerem Abstand Erde - Sonne, ohne den Einfluss der Atmosphäre, senkrecht auf die Erde auftreffende Strahlung. Durch Bahnenzentritizitäten schwankt der Abstand zur Erde, diese und mittelfristige Störungen der

Erbahn welche Einfluss auf die Bestrahlungsstärke haben, werden durch die Milanković-Zyklen [19] beschrieben. Diese Abweichungen von  $\pm 2W/m^2$  werden allerdings nicht vom Programm berücksichtigt und daher nicht näher behandelt.

Die Differenzierung der horizontalen Diffusstrahlung wird nach dem Klarheitsindex  $k_T$  evaluiert, Korrelation nach Reindl-Duffie- & Beckman[11].

$$\begin{aligned} E_{diff,hor} &= E_{G,hor} \cdot (1,020 - 0,254 \cdot k_T + 0,0123 \cdot \sin(\gamma_S)) && \text{für } k_T \leq 0,3 \\ E_{diff,hor} &= E_{G,hor} \cdot (1,400 - 1,749 \cdot k_T + 0,177 \cdot \sin(\gamma_S)) && \text{für } 0,3 < k_T < 0,78 \\ E_{diff,hor} &= E_{G,hor} \cdot (0,486 - 0,182 \cdot \sin(\gamma_S)) && \text{für } k_T \geq 0,78 \end{aligned} \quad (3.13)$$

$k_T$  ... Klarheitsindex (clearness index) [-]  
 $E_{0,hor}$  ... Solarkonstante [ $W/m^2$ ]

Um diese Aufteilung anschaulicher zu machen, kann nach Quaschning[17] ein Zwischenfaktor  $D_{Faktor}$  eingeführt werden. Man teile die obige Gl. 3.13 auf in Gl. 3.14 und das Produkt in Gl. 3.15. Die Abb. 3.1 zeigt dann die prozentuelle diffuse Strahlung in Abhängigkeit von Sonnenhöhe  $\gamma_s$  und  $k_T$  in einem Diagramm.

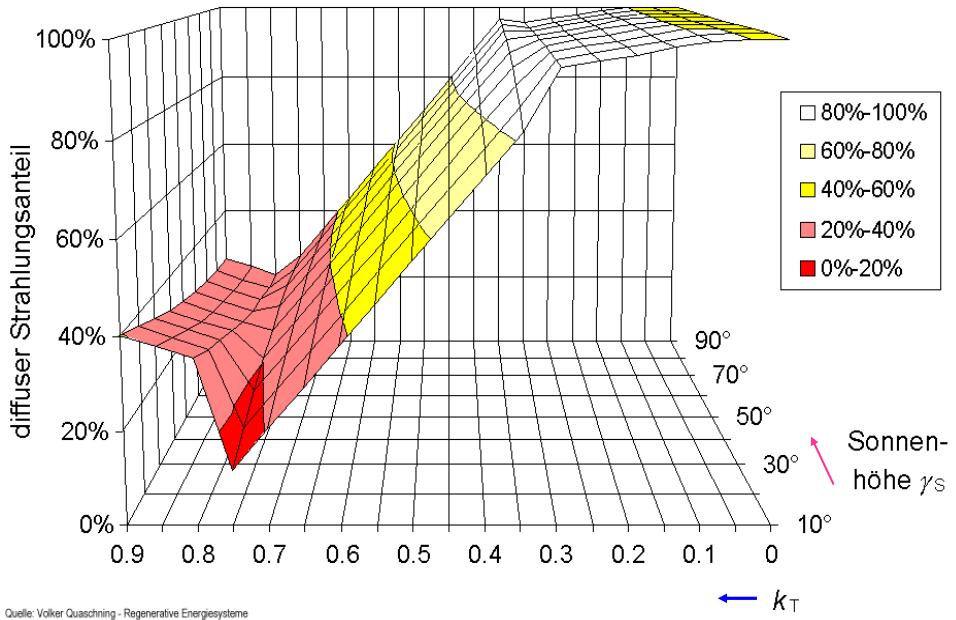


Abbildung 3.1.: Prozentuelle Aufteilung diffuser Strahlung

$$\begin{aligned} D_{Faktor} &= 1,020 - 0,254 \cdot k_T + 0,0123 \cdot \sin(\gamma_S) && \text{für } k_T \leq 0,3 \\ D_{Faktor} &= 1,400 - 1,749 \cdot k_T + 0,177 \cdot \sin(\gamma_S) && \text{für } 0,3 < k_T < 0,78 \\ D_{Faktor} &= 0,486 - 0,182 \cdot \sin(\gamma_S) && \text{für } k_T \geq 0,78 \end{aligned} \quad (3.14)$$

$$E_{diff,hor} = E_{G,hor} \cdot D_{Faktor} \quad (3.15)$$

Weitere Verfahren stellen Thoma [5] und Lanini [4] vor, die aus ähnlichen Korrelationsbedingungen ausgewertet wurden und ein Verfahren, das zur Bestimmung des diffusen Anteils auf verschiedene Schichten der Atmosphäre und des Erdbodens Beziehung nimmt.

### 3.2.2. Einstrahlungsmodell auf die geneigte Fläche

Aus den nun bekannten Horizontalstrahlungsdaten der Berechnung oder dem Klimadatenkatalog, kann die Einstrahlung auf die geneigte Fläche berechnet werden.

#### Einfallswinkel und Moduleinfallswinkel

Dazu werden zunächst die geometrischen Winkelbedingungen des geneigten Kollektors benötigt. Der Einfallswinkel  $\theta_{hor}$  und der Moduleinfallswinkel  $\theta_{gen}$  sind in Abb. 3.2 bildlich dargestellt, sie werden später zur Berechnung der Strahlungsaufteilung von Bedeutung sein.

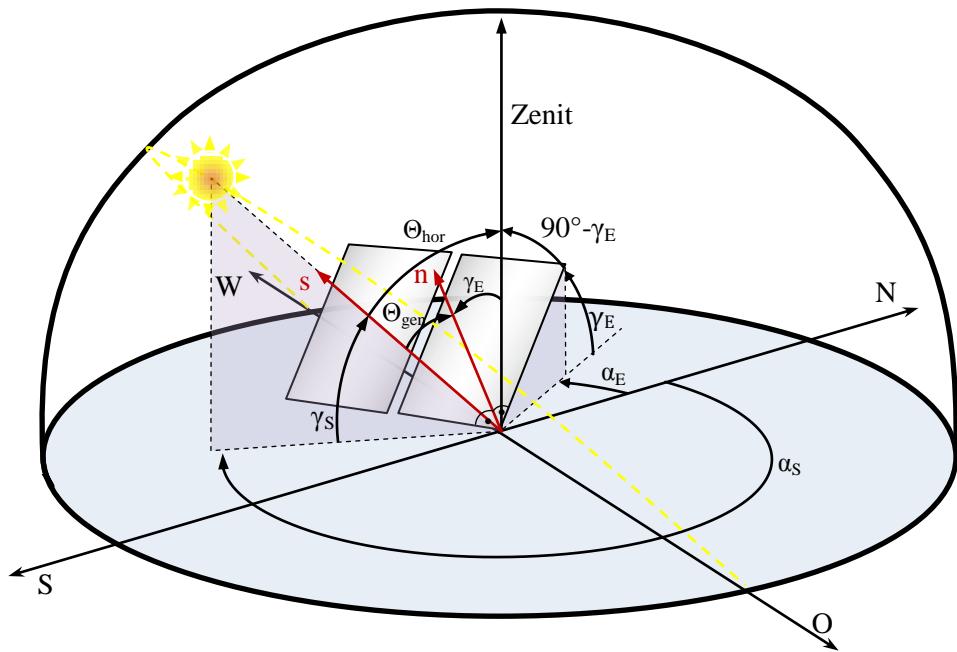


Abbildung 3.2.: Darstellung des Einfall- und Moduleinfallwinkel

$$\Theta_{hor} = \Theta_z = 90^\circ - \gamma_s \quad (3.16)$$

$$\begin{aligned}\Theta_{hor} &\dots \text{Einfallwinkel } [^\circ] \\ \Theta_z &\dots \text{Zenitwinkel } [^\circ]\end{aligned}$$

Schwieriger erweist sich die Transformation von Kugelkoordinaten in kartesische Koordinaten, Gl. 3.17 und Gl. 3.18, die für die geometrische Abhängigkeit des Moduleinfallwinkel benötigt wird. Der Moduleinfallwinkel ist der Sonneneinfallswinkel auf die geneigte Fläche, welcher sich zwischen dem Vektor  $\vec{s}$  in Sonnenrichtung und dem Normalvektor  $\vec{n}$  aufspannt.

$$\vec{s} = \begin{pmatrix} \cos(-\alpha_s) \cdot \cos(\gamma_s) \\ -\sin(-\alpha_s) \cdot \cos(\gamma_s) \\ \sin(\gamma_s) \end{pmatrix} = \begin{pmatrix} \cos(\alpha_s) \cdot \cos(\alpha_s) \\ \sin(\alpha_s) \cdot \cos(\gamma_s) \\ \sin(\gamma_s) \end{pmatrix} \quad (3.17)$$

$$\vec{n} = \begin{pmatrix} \cos(\pi + \alpha_E) \cdot \cos(\frac{\pi}{2} - \gamma_E) \\ \sin(\pi + \alpha_E) \cdot \cos(\frac{\pi}{2} - \gamma_E) \\ \sin(\frac{\pi}{2} - \gamma_E) \end{pmatrix} = \begin{pmatrix} -\cos(\alpha_E) \cdot \sin(\gamma_E) \\ -\sin(\alpha_E) \cdot \sin(\gamma_E) \\ \cos(\gamma_E) \end{pmatrix} \quad (3.18)$$

$\vec{s}$  ... Vektor in Sonnenrichtung [ $^{\circ}$ ]  
 $\vec{n}$  ... Normalvektor auf die geneigte Fläche [ $^{\circ}$ ]

Da beide Vektoren normiert sind, errechnet sich der Einfallswinkel  $\theta_{gen}$  der Sonnenstrahlung auf die geneigte Fläche mit dem Skalarprodukt beider Vektoren.

$$\begin{aligned} \theta_{gen} &= \arccos(\vec{s} \cdot \vec{n}) = \\ &= \arccos(-\cos(\gamma_s) \cdot \sin(\gamma_E) \cdot \cos(\alpha_s - \alpha_E) + \sin(\gamma_s) \cdot \cos(\gamma_E)) \end{aligned} \quad (3.19)$$

### Allgemeines zu isotropen und anisotropen Ansätzen

Bei den weiteren Berechnungsmodellen muss zunächst eine Unterteilung in isotrope und anisotrope Modelle erfolgen. Es wird beim isotropen Ansatz davon ausgegangen, dass im dreidimensionalen Raum, sprich in allen drei Himmelsrichtungen, immer der gleiche Anteil an Himmelsstrahlung vorherrscht. Diese Modelle sind nach Thoma[5] zwar einfach aufgebaut und liefern auch zuverlässige Ergebnisse, allerdings ausschließlich für grobe Abschätzungen oder bei bedecktem Himmel.

Um die Genauigkeit zu erhöhen, werden anisotrope Ansätze gewählt, da die Strahl-dichte vor allem bei klarem Himmel je nach Himmelsrichtung stark variieren kann. Diese Variationen wirken sich in der Helligkeitszunahme am Horizont und in Sonnenhöhe aus. Um dies zu berücksichtigen, werden die meisten anisotropen Ansätze in drei Bestandteile unterteilt. Als erster Teil wird der isotrope Anteil an Strahlung, welcher aus allen Himmelsrichtungen auf die Fläche einfällt, aufgezählt. Als zweites die Zunahme der Helligkeit in Sonnennähe und als drittes die Bodenreflexion und Helligkeitserhöhung in Horizontnähe, welche verstärkt an klaren Tagen auftritt. Als anisotrope Modelle zur Berechnung der Diffusstrahlung werden speziell die Modell von Klucher(1979) und Perez (1987) verwendet. Wobei nach Quaschning[16] das Diffusstrahlungsmodell nach Perez die exaktere Abschätzung liefern soll.

In diesem *VBA*-Programm werden für die diffuse Strahlung auf der geneigten Fläche drei Modelle errechnet, dazu zählen:

- isotropes Modell,
- Modell von Klucher
- Modell von Perez

### Strahlung auf der geneigten Fläche

Die Strahlung auf der geneigten Fläche lässt sich wie in Abb. 2.6 schematisch dargestellt, aus der Summe von diffus-, direkt- und reflektierter Strahlung in Gl. 3.20 zusammenfassen.

$$E_{G,gen} = E_{diff,gen} + E_{dir,gen} + E_{refl,gen} \quad (3.20)$$

$E_{G,gen}$	... Gesamtstrahlung auf der geneigten Fläche in $[W/m^2]$
$E_{diff,gen}$	... Diffusstrahlung auf der geneigten Fläche in $[W/m^2]$
$E_{dir,gen}$	... Direktstrahlung auf der geneigten Fläche in $[W/m^2]$
$E_{refl,gen}$	... Reflektierte Strahlung auf der geneigten Fläche in $[W/m^2]$

Eine Beziehung für die Direkte Strahlung auf der geneigten Fläche  $E_{dir,gen}$  leitet Quaschning[16] nach folgender Gl. 3.21 her.

$$E_{dir,gen} = E_{dir,hor} \cdot \left\{ \max \left( 0, \frac{\cos(\Theta_{gen})}{\sin(\gamma_s)} \right) \right\} \quad (3.21)$$

$\Theta_{gen}$  ... Einfallwinkel auf der geneigten Fläche  $[^\circ]$

### Reflektierte Strahlung auf der geneigten Fläche

Für die Einstrahlung von Bodenreflexion auf der geneigten Fläche  $E_{refl,gen}$  wird ein isotroper Ansatz nach Quaschning[16] verwendet.

$$E_{refl,gen} = E_{G,hor} \cdot A \cdot \frac{(1 - \cos(\gamma_E))}{2} \quad (3.22)$$

$A$  ... Albedo [-]

Albedo ist laut EN 9488[3] das Verhältnis der von einer Fläche reflektierten Strahlung zur einfallenden Strahlung. Der Albedo ist ein Messwert, falls keine Messungen vorliegen, können angenäherte Werte entsprechend der Umgebung aus der Tabelle in Anhang B.1 herangezogen werden. Falls die Umgebung unbekannt ist, soll ein Albedo  $A$  von 0,2 angewendet werden.

### Diffuse Strahlung auf der geneigten Fläche

Diffuse Strahlung, oft auch Himmelsstrahlung genannt, wird wie vorher erwähnt auf mehreren Arten errechnet. Der erste Ansatz, der im VBA-Programm errechnet wird, ist ein isotroper Ansatz nach Gleichung 3.23.

$$E_{diff,gen,iso} = E_{diff,hor} \cdot \frac{(1 + \cos(\gamma_E))}{2} \quad (3.23)$$

Als nächstes wird ein anisotroper Ansatz nach Klucher errechnet in Gleichung 3.24.

$$E_{diff,gen,aniso} = E_{diff,hor} \cdot \frac{(1 + \cos(\gamma_E))}{2} \cdot \left( 1 + F \cdot \sin^3\left(\frac{\gamma_E}{2}\right) \right) \cdot \left( 1 + F \cdot \cos^2(\theta_{gen}) \cdot \cos^3(\gamma_s) \right) \quad (3.24)$$

mit

$$F = 1 - \frac{E_{diff,hor}}{E_{G,hor}}^2 \quad (3.25)$$

$F$  ... zwischen Faktor [-]

Da wie erwähnt der Ansatz nach Perez et al. am genauesten sein soll, wird er mit folgender Gl. 3.26 in das VBA-Programm implementiert und dient im Anschluss auch für weitere Berechnungsschritte und Auswertungen, wohingegen das isotrope und anisotrope Modell nur als Vergleichsmodelle dienen.

$$E_{diff,gen} = E_{diff,hor} \cdot \left[ \frac{(1 + \cos(\gamma_E))}{2} \cdot (1 - F_1) + \frac{a}{b} \cdot F_1 + F_2 \cdot \sin(\gamma_E) \right] \quad (3.26)$$

- $a \dots \max(0; \cos(\theta_{gen}))$
- $b \dots \max(0, 087; \sin(\gamma_s))$
- $F_1 \dots \text{Horizontalhelligkeitsindex } [-]$
- $F_2 \dots \text{Sonnenumgebungshelligkeitsindex } [-]$

Der Horizontalhelligkeitsindex  $F_1$  und der Sonnenumgebungshelligkeitsindex  $F_2$  berücksichtigen vier grundlegende Komponenten, welche die Einstrahlungsbedingungen von „heiter“ bis „klar“ parametrieren. Diese vier Komponenten lauten:

1. Zenitwinkel bzw.  $\theta_{hor}$
2. Klarheitsindex  $\epsilon$
3. Helligkeitsindex  $\delta$
4. atmospheric precipitable water content , der in den Parametern der verschiedenen Himmelsklarheiten  $F_{11}$  bis  $F_{23}$  steckt

$$\epsilon = \frac{\frac{E_{diff,hor} + E_{dir,hor} \cdot \sin^{-1}(\gamma_s)}{E_{diff,hor}} + \kappa \cdot \theta_{hor}^3}{1 + \kappa \cdot \theta_{hor}^3} \quad (3.27)$$

$$\Delta = AM \cdot \frac{E_{diff,hor}}{E_0} \quad (3.28)$$

$$AM = \frac{1}{\sin(\gamma_s)} \quad (3.29)$$

- $\kappa \dots \text{Konstante } \kappa = 1,041$
- $AM \dots \text{Air Mass , relative Luftmasse}$

$$F_1 = F_{11}(\epsilon) + F_{12}(\epsilon) \cdot \Delta + F_{13}(\epsilon) \theta_{hor} \quad (3.30)$$

$$F_2 = F_{21}(\epsilon) + F_{22}(\epsilon) \cdot \Delta + F_{23}(\epsilon) \theta_{hor} \quad (3.31)$$

- $F_{11}, F_{12}, F_{13}$
- $F_{21}, F_{22}, F_{23} \dots \text{Parameter der verschiedenen Himmelsklarheiten}$

Die Parameter  $F_{11}$  bis  $F_{23}$  werden im VBA-Code automatisch aus der integrierten Tabelle B.2 in Abhängigkeit von  $\epsilon$  durch eine eigene Funktion namens „Kollektor.FParameterPerez<sup>24</sup>“ erzeugt. Sie wurden laut Quaschning, [16] von Perez et.al. aus fünf nordostamerikanischen Standorten gemessen und sollten noch auf den Standort Österreich angepasst werden.

---

<sup>24</sup>Dies ist die typische Schreibweise im **VBA** Code, um ein Sub oder eine Funktion zu nennen, vor dem Punkt wird das Modul genannt und nach dem Punkt der Name des Sub's.

### 3.2.3. Strahlungsdatensatz KS und ER

Aus den eben beschriebenen Einstrahlungsmodellen für die horizontale und geneigte Fläche lassen sich nun folgende zwei wichtige Gliederungen der Datensätze ableiten.

1. *KS*: Aus den bekannten Klimadaten auf die horizontale Fläche, die später im VBA-Programm mit *KS*(für Klimadatensatz) gekennzeichnet werden, können mit dem *EGF* die Strahlungsdaten auf die geneigte Fläche berechnet werden. Diesen Variablen wird auch ein *KS* angehängt.
2. *ER*: Es wird ausschließlich mit  $E_{G,hor}$  aus dem Klimadatensatz zuerst die Aufteilung in Diffus- und Direktstrahlung, mit dem Einstrahlungsmodell auf die horizontale Fläche berechnet. Diese erhalten im VBA-Programm die Abkürzung *ER*(für Errechnet). Aus diesem Datensatz wird ebenfalls mit dem *EGF* die Strahlungsdaten auf die geneigte Fläche berechnet und wieder mit *ER* gekennzeichnet.

Diese Gliederung wird benötigt um dem VBA-Programm sagen zu können, wann es welche Datensätze zu verwenden hat, was speziell im **Eingabefall 2** von Bedeutung ist, da in diesem nur die Horizontalstrahlung des **Vergleichsstandortes** ohne Aufteilung in Diffus- und Direktstrahlung gegeben ist. Näheres siehe [4.2.1](#).

## 3.3. Berechnungsmodelle für den Winkelfaktor

Der Winkelfaktor(„Incident Angle Modifier“ kurz IAM) spiegelt das produktspezifische Verhältnis des optischen Wirkungsgrades, bei aktuellem Einfallswinkel, zu demjenigen, bei senkrechtem Einfall der Direktstrahlung, wieder. Er kann nach [2] mit Gleichung 3.32 in Abhängigkeit der linearen und quadratischen Kollektorkennwerte bzw. des Einfallswinkel  $\theta_{gen}$  berechnet werden und ist in Abbildung 3.3 anschaulich dargestellt. Die meisten Prüfprotokolle beziehen sich auf einen Winkelfaktor von  $50^\circ$  Einfallswinkel<sup>3</sup>, dadurch kann der quadratische Term vernachlässigt ( $b_1=0$ ) und  $b_0$  anhand Gleichung 3.33 berechnet werden.

$$K_{\theta_{gen}} = 1 - b_0 \cdot \left( \frac{1}{\cos(\theta_{gen})} - 1 \right) - b_1 \cdot \left( \frac{1}{\cos(\theta_{gen})} - 1 \right)^2 \quad (3.32)$$

$$b_0 = \frac{1 - K_1(\theta_{Prf})}{\frac{1}{\cos(\theta_{Prf})} - 1} \quad (3.33)$$

$K_{\theta_{gen}}$	...	Winkelfaktor [-]
$b_0$	...	Linearer Kollektorkennwert [ $^\circ$ ]
$b_1$	...	Quadratischer Kollektorkennwert [ $^\circ$ ]
$K_1$	...	Winkelfaktor nach Prüfprotokoll [K]
$\theta_{Prf}$	...	Einfallswinkel des Prüfprotokolles [ $^\circ$ ]

Es gilt als Faustformel: je größer der Einfallswinkel<sup>4</sup>, desto kleiner der Energieertrag.

---

<sup>3</sup>Dieser Winkel kann im VBA-Programm verändert werden aber dazu später in [4.2.5](#)

<sup>4</sup>sprich Reflexion am Solar-Glas

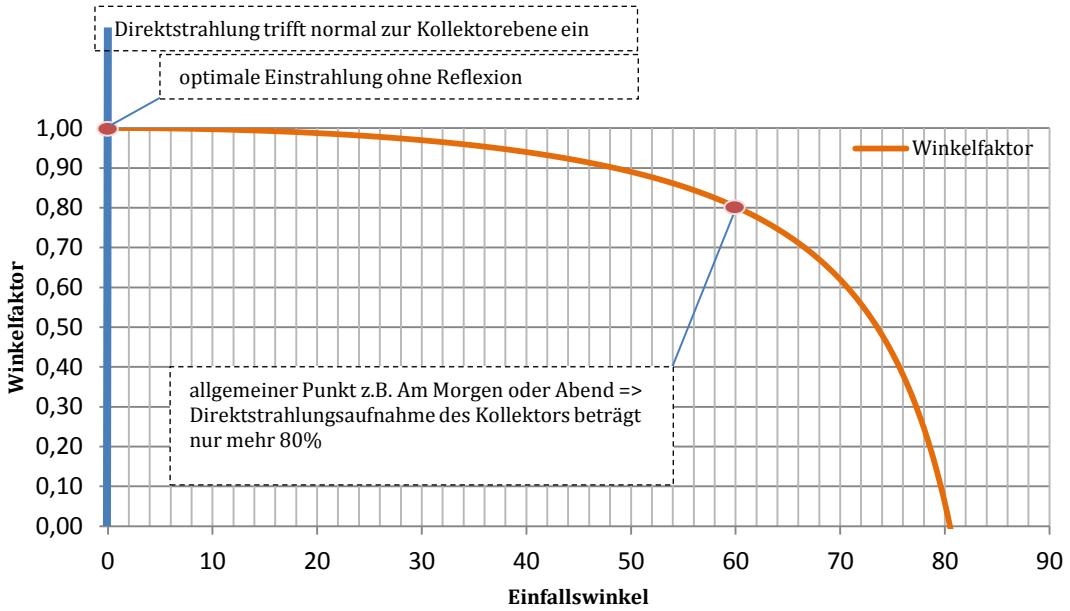


Abbildung 3.3.: Winkelfaktor  $K_{\theta_{gen}}$  eines Solarkollektors in Abhängigkeit vom Einfallsinkel

### 3.4. Berechnungsmodelle für die Kollektorwirkungsgradkurve

Die Gl. 3.34 der Kollektorwirkungsgradkurve ist das Resultat der Leistungsmessung an Kollektoren nach EN 12975-2[9]. Nach [14] beschreibt die darin beschriebene Messmethode die gemessene Nutzleistung eines Kollektors bei konstanter Einstrahlung und stationärem Volumenstrom. Durch Veränderung der Eintrittstemperatur werden fortlaufend mehrere Betriebspunkte vorgegeben und der Wirkungsgrad ermittelt. Danach werden die ermittelten Wertepaare in ein Diagramm eingetragen und ein Polynom zweiter Ordnung gefüttet. Die identifizierten Polynomkoeffizienten sind ident mit den Kennlinienparametern  $C_0$ ,  $C_1$  und  $C_2$  der Wirkungsgradkurve.

$$\eta_{Kollektor} = C_0 - C_1 \cdot \frac{(T_F - T_U)}{E_{G,gen}} - C_2 \cdot \frac{(T_F - T_U)^2}{E_{G,gen}} \quad (3.34)$$

$\eta_{Kollektor}$	...	Kollektorwirkungsgrad [-]
$C_0$	...	Konversionsfaktor des Kollektors [-]
$C_1$	...	Linearer Wärmeverlustkoeffizient [ $W/m^2K$ ]
$C_2$	...	Quadratischer Wärmeverlustkoeffizient [ $W/m^2K^2$ ]
$T_F$	...	mittlere Fluidtemperatur des Absorbers [K]
$T_U$	...	Umgebungstemperatur [K]

Die Umgebungstemperatur  $T_F$  wird vom Klimadatenkatalog je nach Standort in das Programm eingelesen. Die mittlere Fluidtemperatur des Absorbers wird anhand der Vor- bzw. Rücklauftemperaturen vom Modul Speicher bereitgestellt. Die Parameter  $C_0$ ,  $C_1$  und  $C_2$  sind auch auf den Prüfprotokollen der Solarkollektor-Hersteller zu finden und können dadurch für Vergleiche herangezogen werden. Einen Auszug dieser Werte und der verschiedenen Kollektortypen ist dem Tabellenblatt **allg. Informationen** zu entnehmen.

men. In Abbildung 3.4 sind solche Wirkungsgradkurven unterschiedlicher Einstrahlung dargestellt.

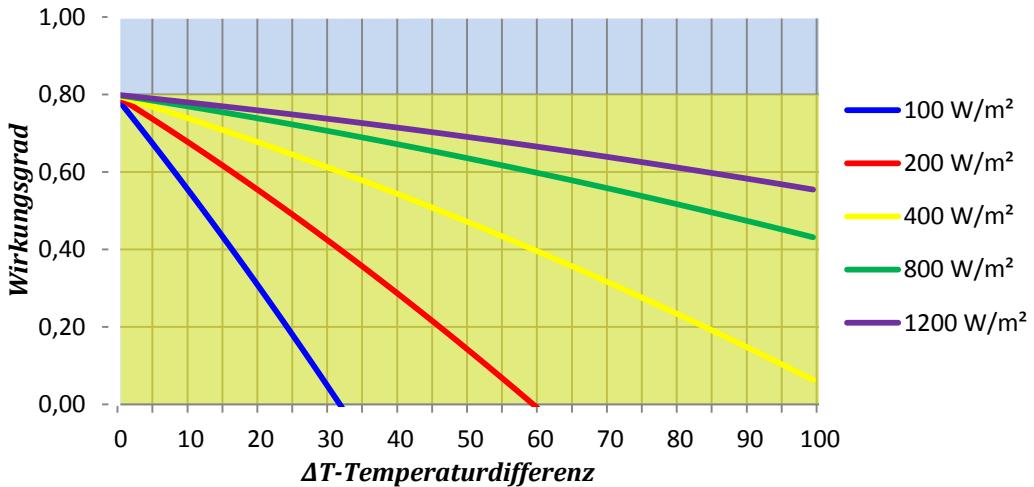


Abbildung 3.4.: Wirkungsgradkurven  $\eta_{Kollektor}$  eines Solarkollektors in Abhängigkeit vom  $\Delta T$  und Einstrahlung auf den Kollektor

Der Konversionsfaktor des Kollektors verschiebt im Wesentlichen den im Diagramm 3.5 blau dargestellten Bereich horizontal und wird in der Literatur meistens auch als der optische Verlust betrachtet. Der gelbe Bereich soll weiters den thermischen Verlust darstellen und der grüne Bereich unter der Kurve zeigt die nutzbare thermische Energie des Kollektors in Abhängigkeit des linearen und quadratischen Verlaufs der Koeffizienten  $C_1$  und  $C_2$ . Die Höhe der Einstrahlung spielt natürlich eine wesentliche Rolle und dadurch kommt es dazu, dass in den Morgen- und Abendstunden schlechtere Wirkungsgrade erreicht werden als zu Mittag.

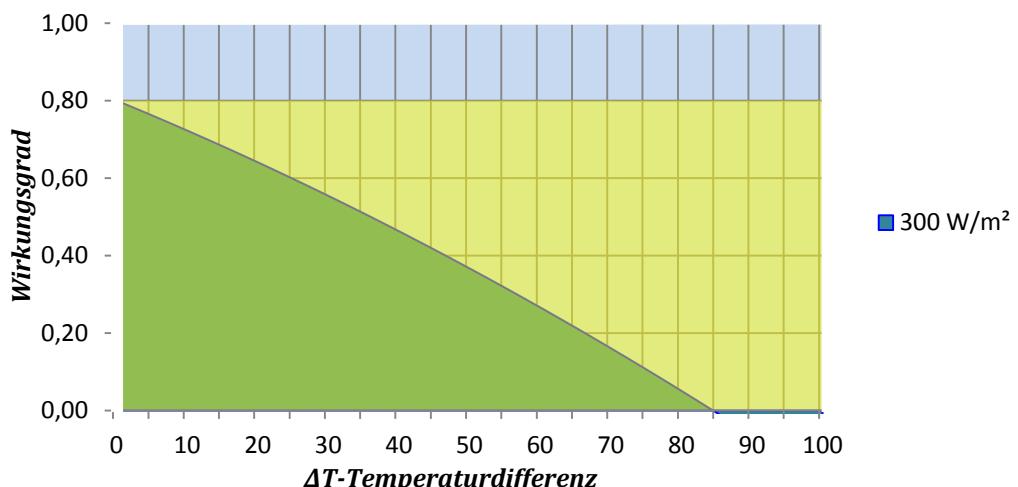


Abbildung 3.5.: Wirkungsgradverlauf  $\eta_{Kollektor}$  eines Solarkollektors mit Bereicheseinteilung

# Kapitel 4.

## VBA - Programm

### 4.1. Allgemeines zur Programmiersprache

Visual Basic for Applications (VBA) ist eine Skriptsprache, die zu den Microsoft-Office-Programmen gehört. Sie wurde aus dem von *MS* entwickelten BASIC-Dialekt Visual Basic (VB) abgeleitet und wurde zur Steuerung von Abläufen in den *MS*-Office-Anwendungen entwickelt. Die Programmiersprache wird nicht nur in den wichtigsten *MS*-Office Anwendungen wie Excel, Word und Access verwendet, sondern auch in der ganzen *MS*-Office Produktfamilie, sowie in anderen Anwendungsprogrammen der Hersteller Autodesk, Corel, Catia, Solid Works und vielen mehr. In unserer Arbeit wird ausschließlich *VBA* in *MS* Excel verwendet.

#### Zur Wahl der Programmiersprache

Es wurde vorgegeben, die Arbeit in *MS* Excel zu erstellen. Was den wesentlichen Vorteil birgt, die ausgegebenen Daten und Diagramme ohne Programmierkenntnisse in *MS* Excel weiter bearbeiten zu können. Diese Vorgabe ist aber mit Einschränkungen der Modellbildung des *TESS* verbunden<sup>1</sup>. Im Bereich Solareinbindung führt die Wahl lediglich zu Geschwindigkeitseinbußen, wenn man mehrere Projekte zugleich betrachtet oder vergleicht.

Weiters ist zu sagen, dass diese Programmiersprache relativ leicht zu erlernen ist. Ich selbst habe die Programmiersprache im Zuge dieser Arbeit erlernt, ohne Vorkenntnisse in VB oder dergleichen gehabt zu haben. Dabei waren mir Online-tutorials, Forumseinträge und die Suchengine Google eine große Hilfe.

### 4.2. Handhabung und Anwendung des Programms

Das Programm wird in der Datei Solarsimulation.xlsx mit Aufrufen unter *MS* Excel gestartet<sup>2</sup>. Dabei wird automatisch eine Registerkarte im Menüband mit den Namen Solarsimulation geladen, über welche die grundlegende Bedienung auf einer Menüleiste vorstatten geht. Eine Abbildung der gesamten Bedienungsoberfläche ist in [D.1](#) zu sehen. Weiters sollten beim Start des Originalfiles mindestens folgende Tabellenblätter vorhanden sein:

- Klimadaten
- Verbrauch
- allg. Informationen

---

<sup>1</sup>näheres siehe[7]

<sup>2</sup>MS Excel Makrotauglich machen siehe [4.4](#)

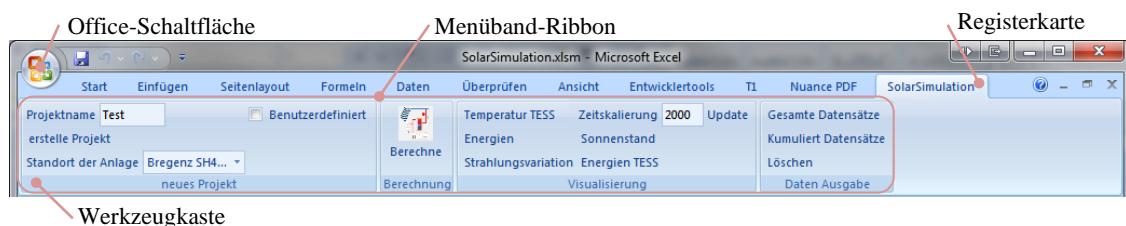


Abbildung 4.1.: Benutzeroberfläche im MS Excel Solarimulation

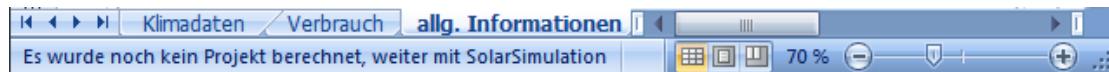


Abbildung 4.2.: Tabellenblätter nach Programmstart

Die Statusleiste unterhalb der Tabellenblätter zeigt die gegenwärtige Lage, Berechnungsdauer sowie zusätzliche Informationen und fehlerhafte Eingaben an. Sie sollte zur reibungslosen Anwendung des Programmes kontinuierlich beachtet werden.

#### 4.2.1. Werkzeugkasten „neues Projekt“

In Abbildung 4.1 befinden sich in dem Menüband der Werkzeugkasten „neues Projekt“, sowie die anderen drei Werkzeugkästen, die in 4.2.2 bis 4.2.4 beschrieben werden. Hier befinden sich alle Tools zur Erstellung eines neuen Projektes. Ein Projekt besteht aus vier Tabellenblättern, wobei jedes Tabellenblatt mit der Vorsilbe des Projektnamens erstellt wird.

- „Projektname“\_Eingabe
- „Projektname“\_Diagramme
- „Projektname“\_Ausgabe
- „Projektname“\_AusgabeKum

Unter **Projektname** wird der Name des Projektes eingetragen und mit dem darunter liegenden Button „**erstelle Projekt**“, wird dieses erzeugt. Ein Wechseln des Projekts erfolgt ganz einfach durch Auswahl eines Tabellenblattes mit anderem Projektnamen. Die weiteren Tools beziehen sich also ausschließlich auf den eingetragenen Namen des Projektes und werden auch in diesem durchgeführt. Um die fälschliche Bearbeitung eines anderen Projektes auszuschließen, muss immer darauf geachtet werden, welcher Name eingetragen ist.

Das Dropdown-Menü **Standort der Anlage** darunter, stellt die in Abschnitt 2.3.3 beschriebenen Standorte zur Auswahl. Mit der getroffenen Auswahl werden auch gleich die geographischen Daten, sprich Längengrad und Breitengrad, vom ausgewähltem Standort automatisch in das Tabellenblatt (Projektname)\_Eingabe eingetragen. Diese geographischen Daten können überschrieben werden, indem man das Kästchen „**Benutzerdefiniert**“ ankreuzt. Auch die Option, „Hinzufügen eines neuen Klimadatensatzes“, ist gegeben. Dazu muss im Dropdown-Menü des Anlagenstandortes der Ort **Vergleichsstandort** ausgewählt werden.

## Vergleichsstandort

Der Vergleichsstandort, der im Tabellenblatt Klimadaten (ganz rechts) zu finden ist, siehe auch Abbildung 4.3, dient einerseits zum Hinzufügen eigener Klimadaten und andererseits zu Vergleichszwecken.

Mit den Button **Select** wird die darunter liegende Spalte bis zum letzten Datensatz markiert. Mit der Tastenkombination „STRG + ALT + C“ kann der markierte Teil kopiert werden und mit „STRG + ALT + V“ in eine andere Spalte eingefügt werden. Der Button „**Bestimmte Spalte markieren**“ lässt die Wahl der zu markierenden Spalte offen und ist mit den Buchstaben der Spalte zu bestätigen. Falls eine Blattschutz-Aktivierung von Excel vergeben ist, kann diese mit dem Passwort „Klimadaten“ aufgehoben werden.

Da von vielen Wetterstationen oft nur die Datensätze für die Globalstrahlung bekannt sind und keine Aufteilung in diffuse und direkte Strahlung gemacht wird, erkennt das Programm die Eingabe und errechnet die fehlende Aufteilung je nach Eingabefall. Dabei gilt **Fall 1** als bekannter Standort, in dem alle Strahlungsarten bekannt sind und die Fälle 2 bis 9 nach Auflistung siehe Tabelle 4.1.

Vergleichsstandort		4651,304232								Bestimmte Spalte markieren				
12		132,4												
Vergleichsstandort														
geogr.L.	11,77000	geogr.B.	47,88	Für die Berechnung des Vergleichsstandort muss der Gelbe Bereich Ausgefüllt sein		Eghor	Ediffhor							
Monat	Tag	Stunde	TE	abDf_Aussen		GS	HS	DS	Nur für copy / paste zwecke					
		MEZ		°C		w/m²	w/m²	w/m²	Select	Select	Select	Select	Select	Select
			Select		Select	Select	Select	Select	Select	Select	Select	Select	Select	Select
1	1	1	-8,6		0,00	0,00	0,00							
1	1	2	-9,2		0,00	0,00	0,00							
1	1	3	-8,8		0,00	0,00	0,00							
1	1	4	-9,1		0,00	0,00	0,00							
1	1	5	-7,7		0,00	0,00	0,00							
1	1	6	-8,3		0,00	0,00	0,00							
1	1	7	-9,5		0,00	0,00	0,00							
1	1	8	-11,6		0,00	0,00	0,00							
1	1	9	-9,0		24,50	3,20	21,30							
1	1	10	-7,5		153,00	102,80	50,20							
1	1	11	-5,2		257,00	187,70	69,30							
1	1	12	-4,1		319,00	232,10	86,90							
1	1	13	-2,5		202,00	70,00	132,00							
1	1	14	-0,7		216,00	82,00	134,00							
1	1	15	-0,6		150,00	45,00	105,00							
1	1	16	-1,0		122,00	69,70	52,30							
1	1	17	-2,2		6,88	0,00	6,88							
1	1	18	-3,0		0,00	0,00	0,00							
1	1	19	-2,7		0,00	0,00	0,00							
1	1	20	-2,8		0,00	0,00	0,00							
1	1	21	-2,3		0,00	0,00	0,00							
1	1	22	-2,5		0,00	0,00	0,00							
1	1	23	-5,4		0,00	0,00	0,00							
1	1	24	-6,1		0,00	0,00	0,00							
1	2	1	-8,0		0,00	0,00	0,00							
1	2	2	-7,0		0,00	0,00	0,00						0,00	0,00
			-7,4		0,00	0,00	0,00							

Abbildung 4.3.: Tabellenblatt Klimadaten - Vergleichsstandort

## Einheiten

Die Eingabe der Einheiten von Globalstrahlung wird vom Programm überwacht. Es wird vorausgesetzt, dass Diffusstrahlung und Direktstrahlung die gleiche Einheit wie die Globalstrahlung besitzen (wird jedoch nicht überwacht). Es werden nur folgende Eingaben zugelassen:  $J/cm^2$  und  $W/m^2$ , auf Groß- und Kleinschreibung wird dabei nicht geachtet und zur Vereinfachung von  $^2$  kann einfach 2 geschrieben werden. Je nach Eingabe von  $J/cm^2$  oder  $W/m^2$  erfolgt eine automatische Umrechnung.

GS	HS	DS	Eingabefall	Arbeitsschritte des Programmes
leer	leer	leer	Fall 7	Programm Abbruch
leer	voll	leer	Fall 9	Programm Abbruch
leer	leer	voll	Fall 8	Programm Abbruch
leer	voll	voll	Fall 6	Globalstrahlung wird ergänzt
voll	voll	voll	Fall 5	Daten werden übernommen
voll	leer	voll	Fall 3	Diffusstrahlung wird ergänzt
voll	voll	leer	Fall 4	Direktstrahlung wird ergänzt
voll	leer	leer	Fall 2	HS und DS werden nach 3.2.1 berechnet

Tabelle 4.1.: Fallunterscheidung je nach Eingabe der Strahlungsaufteilung, des Vergleichsstandortes

$$E[W/m^2] = \frac{E[J/cm^2]}{(3600 \cdot 10^{-4})} \quad (4.1)$$

### Legende der Klimadaten

Unter dem in Excel verwendeten Tabellenblatt, Klimadaten sind die Abkürzungen wie folgt zu verstehen.

TE	... Außentemperatur [°C]
RF	... Relative Feucht [%]
GS	... Globalstrahlung auf die horizontale Fläche [J/cm²]
HS	... Himmels- oder diffuse Strahlung auf die horiz. Flächen [J/cm²]
DS	... Direktstrahlung auf die horizontale Flächen [J/cm²]
SD	... Zeitliche Auflösung der Messung [ $\frac{1}{10} \cdot h$ ]
WG	... Windgeschwindigkeit [m/s]
WR	... Windrichtung [360-teilig]
NS	... Niederschlag [mm = l/m²]
LD	... Luftdruck [hPa]

### 4.2.2. Werkzeugkasten „Berechnung“

Wie bereits erwähnt ist es wichtig darauf zu achten, in welchem Projekt man arbeiten möchte und welcher Projektname im Textfenster des Werkzeugkasten „neues Projekt“ eingetragen ist. Von der Statusleiste kann die Berechnungsdauer und der Projektname noch einmal kontrolliert werden.

### 4.2.3. Werkzeugkasten „Visualisierung“

In diesem Werkzeugkasten kann das simulierte Modell der Solaranlage mit dem TESS visualisiert werden. Jede Schaltfläche dieses Werkzeugkastens fügt ein Diagramm in das Tabellenblatt „Projektname\_Diagramme hinzu. Bei einigen wird zusätzlich ein Scrollbalken erzeugt, mit dem man das Zeitfenster verschieben kann und somit einen Tagesüberblick hat<sup>3</sup>. Die Schaltfläche **Zeitskalierung** wird in [7] näher beschrieben. Mit der

<sup>3</sup>Diagramme die über Scrollbalken gesteuert werden: 1) Temperaturverteilung im TESS, 2) Sonnenstand, 3) Azimut Vergleich und Sonnenhöhe Vergleich,

Taste **Update** werden alle Diagramme auf Aktualität überprüft und bei Änderungen neu initialisiert. Um ein Diagramm zu drucken, wird lediglich das gewünschte Diagramm markiert und mit einem Klick auf die Office-Schaltfläche => Drucken, wird ausschließlich dieses gedruckt.

#### 4.2.4. Werkzeugkasten „Daten Ausgabe“

Mit diesen Schaltflächen werden die Datensätze der zuvor berechneten Simulation ausgegeben. Die Schaltfläche **Gesamte Datensätze** erzeugt die Ausgabe in dem Tabellenblatt **Projektname\_Ausgabe**. Ident arbeitet die Schaltfläche **Kumulierte Datensätze**. Die Ausgabe erfolgt im Tabellenblatt **Projektname\_AusgabeKum**. Diese Datensätze sind die kumulierten Monatswerte. Mit der Schaltfläche **Löschen** kann der Zelleninhalt der Tabellenblätter gelöscht werden<sup>4</sup>. Die Zuweisung, welches Tabellenblatt gelöscht werden soll, wird mit Auswahl des Blattes getätigten. Um Fehlbedienung zu verhindern ist eine Sicherheitsabfrage eingebaut, in der die Löschung des Blattes explizit bestätigt werden muss.

#### 4.2.5. Projektdaten Eingabe

Die spezifischen Projektdaten werden bis auf wenige erwähnte Ausnahmen im erzeugten Tabellenblatt **Projektname\_Eingabe** eingetragen. Die spaltenförmige Einteilung spiegelt die Teilsysteme wieder, so sind im orangen Bereich die Daten des Kollektors einzugeben und im gelben Bereich die Eingabedaten des TESS.

Mit Auswahl des Standortes in dem Menüband, wird dieser direkt in das Tabellenblatt geschrieben, zusätzlich werden auch die geographischen Daten des Standortes eingetragen<sup>5</sup>.

Für den **Albedo** stehen, je nach Oberfläche der Umgebung, allgemeine Werte im Tabellenblatt **allg. Informationen** oder im Anhang siehe [B.1](#) bereit. Falls keine Information über die Umgebung bekannt ist, soll der Wert 0,2 verwendet werden. Unter **Azimut des Kollektors** und **Höhenwinkel des Kollektors** kann die geometrische Ausrichtung des Kollektors zur Nordachse, siehe Abbildung [4.4](#), beschrieben werden.

Mit den Eingabeflächen **C0 optischer Wirkungsgrad**, **C1 linearer Wärmeverlustkoeffizient**, und **C2 quadratischer Wärmeverlustkoeffizient** können die Prüfkenndaten der verschiedenen Kollektortypen eingebracht werden, ebenfalls der **Winkel faktor bei 50° Einfallwinkel** aus den Prüfdaten. Der **Einfallwinkel** sollte je nach Prüfbericht auf 50° eingestellt sein.

Die Taste darunter bietet, wie in [3.1](#) beschrieben, das Sonnenstandberechnungsmodell mit 1, 2 oder 3 zur Auswahl. Mit dem Eingabeparameter ja oder nein<sup>6</sup> der **Winkel faktor Berücksichtigung**, kann der Einfluss der Reflexion durch das Solarglas wie in [3.3](#), eingeschaltet werden.

Die letzte Eingabefläche **Diagrammausgabe Strahlungsvariation (KS oder ER)** berücksichtigt, mit welchem Datensatz der Globalstrahlung das Diagramm **Strahlungs variation** berechnet wird.

Im Eingabebereich **Sonnenstand Berechnungsmodell** wird die Wahl der Berechnung des Sonnenstandes, siehe [3.1](#), eingegeben. Die Eingabe darf nur mit 1, 2 oder 3

<sup>4</sup>Es werden keine Diagramme oder Scrollbalken entfernt, diese müssen einzeln gelöscht oder Ausgeschnitten werden

<sup>5</sup>wie in [4.2.1](#) erklärt, kann mit dem Kästchen **Benutzerdefiniert** die Eingabe der geographischen Daten überschrieben werden.

<sup>6</sup>es ist ausschließlich „ja“ oder „nein“ kleingeschrieben zulässig, sonst erfolgt ein Programmabbruch mit dem internen Fehler(4)

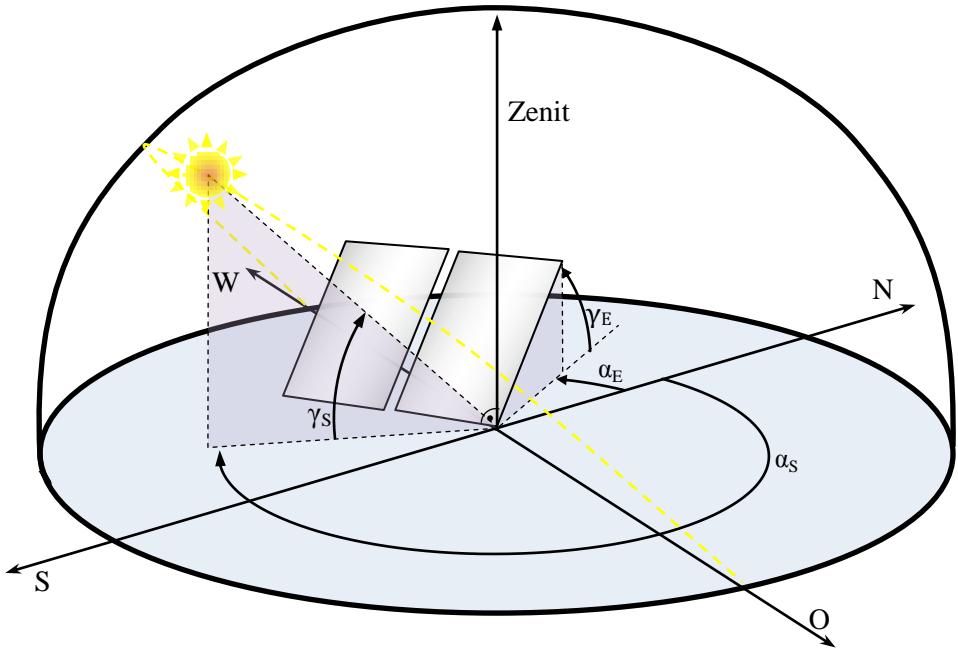


Abbildung 4.4.:  $\alpha_E$ ...Azimut des Kollektors,  $\gamma_E$ ...Höhe des Kollektors,  $\alpha_S$ ...Azimut der Sonne,  $\gamma_S$ ...Sonnenhöhe

erfolgen, da sonst ein unerwarteter Fehler in der Statusleiste zu erwarten ist und das Projekt nicht berechnet wird.

- Berechnungsmodell 1: Sonnenstand nach [BMT](#)
- Berechnungsmodell 2: Sonnenstand nach DIN5034 [A.1](#)
- Berechnungsmodell 3: Sonnenstand nach SUNAE-Algorithmus [A.2](#)

Näheres zu den Eingabedaten vom TESS ist in [7] zu finden.

### 4.3. Aufbau und Unterteilung

Ein [MS](#) Excel File mit Standard Dateiendung \*.xls beinhaltet keine Makros, nur [MS](#) Excel Dateien mit der Endung \*.xlsm können Makros speichern. In Abbildung 4.5 ist ein schematischer Aufbau des [MS](#) Excel/VBA Programmes Solarsimulation.xlsm zu sehen. Es soll verdeutlichen, welche Module, Klassen und \*.xml Dateien im Hintergrund implementiert sind und untereinander agieren. Zusätzlich ist auch die Aufteilung der Teilsysteme TESS und Solareinbindung zu sehen. Hier ist zu bemerken, dass bei den allgemeinen Algorithmen diese Aufteilung nicht zutrifft, da diese in Zusammenarbeit mit Herrn Ortbauer Martin entstanden sind und beide Module der Teilsysteme darauf zugreifen.

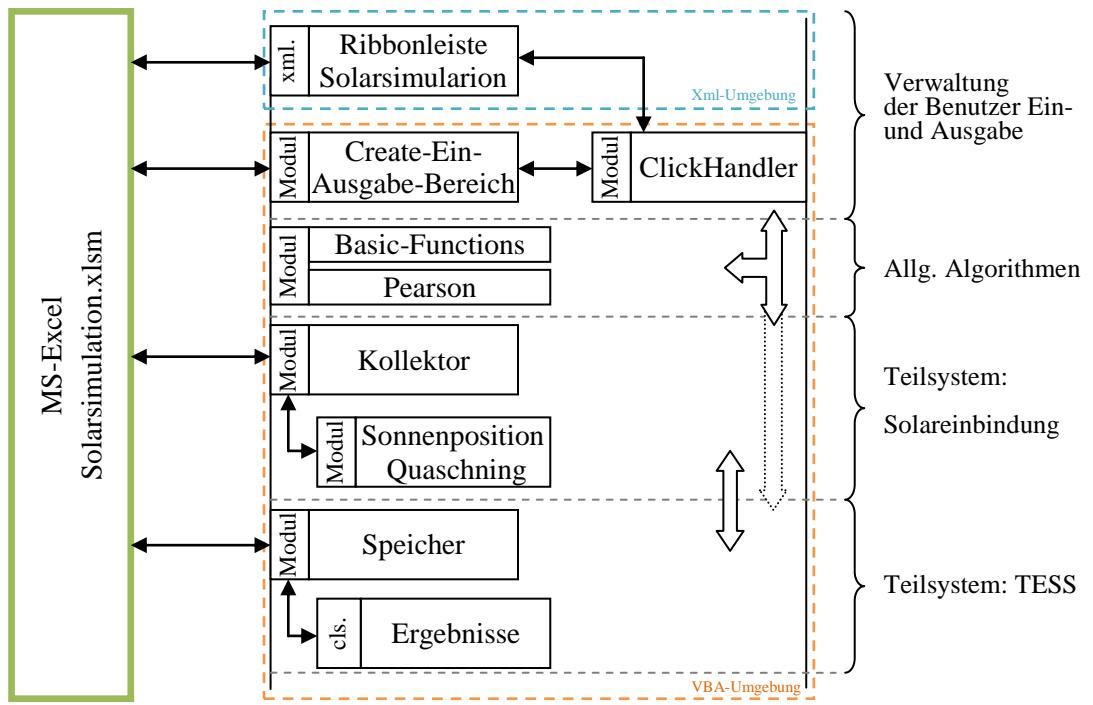


Abbildung 4.5.: Überblick des VBA-Programmes und der Entwicklungsumgebung

#### 4.3.1. Module und Klassen im Programmcode

**Click\_Handler** Dieses Modul verwaltet die Befehlseingabe aus dem Menüband<sup>7</sup> und initialisiert die weiteren Programm Routinen. Das Menüband ist als CustomUI.xml Datei im Solarsimulation.xlsm File enthalten<sup>8</sup>. Es wurde mit dem Programm Custom UI Editor for MS<sup>9</sup> geschrieben, welches als Hauptaufgabe ein Herausfiltern des richtigen \*.xml File aus der Excel Datei übernimmt.

**Create\_Ein\_Ausgabe\_Bereich** Hier geschieht unter Anweisung des Click\_Handlers die Erstellung der Tabellenblätter mit Eingabemaske für jedes angelegte Projekt.

**Basic\_Functions** Diese allgemeinen Algorithmen werden von beiden Teilsystemen verwendet und beinhalten Standardroutinen zur Formatierung der Datenausgabe, Critical Message Box Anzeigen, Statuszeilen Ausgabe, Datenblatt Löschtätigkeiten und andere oft verwendete Subs<sup>10</sup>.

**Pearson** ist ein Modul, welches im Speziellen vorgefertigte Subs zur array Bearbeitung beinhaltet und von beiden Teilsystemen verwendet. Diese Subs sind frei von der Internetseite siehe [10] bezogen.

<sup>7</sup> sprich Ribbonleiste Solarsimulation

<sup>8</sup> Eine Excel Datei ist im Grunde nichts anderes als eine ZIP Datei, man muss nur die Dateiendung verändern.

<sup>9</sup> Open Source Software D.1; Anleitung siehe [13]

<sup>10</sup> sind die im Modul enthaltenen, aufrufbaren Programmsequenzen

**Kollektor** Hier werden die Projektdaten des Teilsystems Solareinbindung kalkuliert. Die wichtigsten Subs dieses Moduls sind in [4.3.2](#) beschrieben. Zusätzlich ist für tiefere Einblicke in den Quellcode dieser im Anhang beigelgt, siehe Anhang [E](#). Damit der Quellcode des Modules Kollektor übersichtlicher wird, ist er in folgende Kapitel unterteilt:

1. Allgemeine Konstanten
2. Allgemeine Deklarationen
3. Allgemeine Funktionen und Subs
4. Initialisierung des Programmes
5. Hauptprogrammsequenzen
6. Visualisierende Programmsequenzen
7. Daten Ein- Ausgabe Erzeugung

**Sonnenposition \_ Quaschning** In diesem Modul sind nur die Sonnenstand Berechnungsmodelle 2 und 3 für Stundenwerte nach [15] welche nach dem Struktogramm [A.1](#) und [A.2](#) arbeiten, implementiert. Es dient in erster Linie zu Kontrollzwecken.

**Speicher** Das Teilsystem [TESS](#) wird mit diesem Modul in die Simulation implementiert, Näheres siehe [[7](#)].

**Ergebnisse** Dieses Klassenmodul enthält die Ergebnisse des Teilsystems [TESS](#), für weitere Informationen siehe ebenfalls [[7](#)].

### 4.3.2. Die wichtigsten Subs und Funktionen

Da der Sourcecode ausreichend kommentiert ist, folgt hier nur mehr eine zusammenfassende Auflistung der wichtigsten Subs und Funktionen aus den obigen Modulen [4.3.1](#). Diese Auflistung beschränkt sich auf das Teilsystem Solareinbindung.

**ClickHandler.Berechne \_ Handler** Nach anklicken der Schaltfläche **Berechne** wird unter anderem die Berechnungszeit gestoppt, weiters findet mit der **testSheets()** und **init** Funktion eine Überprüfung statt, ob das Projekt auch bereits erstellt wurde. Erst dann, wenn kein Fehler auftritt, werden die zwei Subs **Kollektor.Kollektor \_ Start** und **Speicher.BerechneSpeicher** initialisiert. Zuerst erfolgt die Berechnung des Kollektors und anschließend die des Speichers. Ein Struktogramm in Abbildung [4.6](#) soll die Verzweigungen des Sub **Kollektor.Kollektor \_ Start** anschaulich machen.

**Kollektor.Kollektor \_ Start** Dies Routine startet nur die weiteren Unterprogramme:

- **getEingabe**
- **getKlimadaten**
- **Globalstrahlung**
- **Datenzusammenfassung \_ für \_ Output**

**Kollektor.getEingabe** Schreibt die vom Benutzer im Tabellenblatt **Projektnname\_Eingabe** eingegebenen Parameter in interne Variablen. Dies werden im Modul Kollektor unter den Deklarationen als Privat<sup>11</sup> bereitgestellt.

**Kollektor.getKlimadaten** liest die gewählten Standort Daten wie Umgebungstemperatur, Globalstrahlung, Diffusstrahlung und Direktstrahlung aus, überprüft die Einheiten und rechnet diese falls nötig um, siehe [4.2.1](#). Zusätzlich erlaubt diese Routine eine unvollständige Eingabe des Standortes Vergleichsstandort, siehe [4.2.1](#). Diese vier Arrays<sup>12</sup> stehen als Public deklarierte Variablen im Modul Kollektor bereit.

**Kollektor.Globalstrahlung** beinhaltet die wichtigsten mathematischen Modelle für den Teil Solareinbindung. Beginnend mit der Erstellung des Zeitarrays, weiter mit der Berechnung des Sonnenstandes, Einstrahlung auf die horizontale Fläche bis zur Berechnung der Einstrahlung auf die geneigte Fläche mit optionaler Winkelfaktor Abminderung, werden all diese Sub Inhalte nach den mathematischen Modellen in Abschnitt [2.4](#) berechnet.

**Kollektor.Datenzusammenfassung\_fur\_Output** ist für die Bereitstellung der gesamten Daten zuständig. Das Sub kumuliert die Daten und stellt diese mit den richtigen Formatierungsrichtlinien dem Modul **Create\_Ein\_Ausgabe\_Bereich** bereit.

**Kollektor.Wirkungsgrad** berechnet den Wirkungsgrad und stellt diesen als Public deklaration des Moduls Kollektor bereit.

**Kollektor.Diag\_Start** ruft die Prozeduren der einzelnen Diagramme je nach Parameter auf. Jede Taste **Energie**, **Strahlungsvariation** und **Sonnenstand** des Menübandes hat also eine eigenen Parameter 1, 2, 3<sup>13</sup> mit welchem die bestimmten Subs der Diagramme angesprochen und ausgeführt werden. Näheres dazu im kommentierten VBA-Code des Simulationstool *SolarSimulation* Anhang [E.3](#) und [E.5](#).

**Kollektor.Diag\_Update** wird von der Prozedur **ClickHandler.UpdateDiag\_Handler** aufgerufen welcher mit dem Button **Update** des Menübandes angesprochen wird. Es kontrolliert, welche Diagramme vorhanden sind und gibt den Letztstand der Berechneten Daten an die Diagramme weiter, welche dann, falls nötig, neu dargestellt werden.

---

<sup>11</sup>Dim, Public, Privat dienen der Variablendeklaration, und legen fest, von wo die Variablen angesprochen werden können.

<sup>12</sup>Variablen Name der Arrays: dblT\_AussenKS, EghorKS, EdiffhorKS, EdirhorKS

<sup>13</sup>Parameter stimmen mit Aufzählungsreihenfolge zusammen

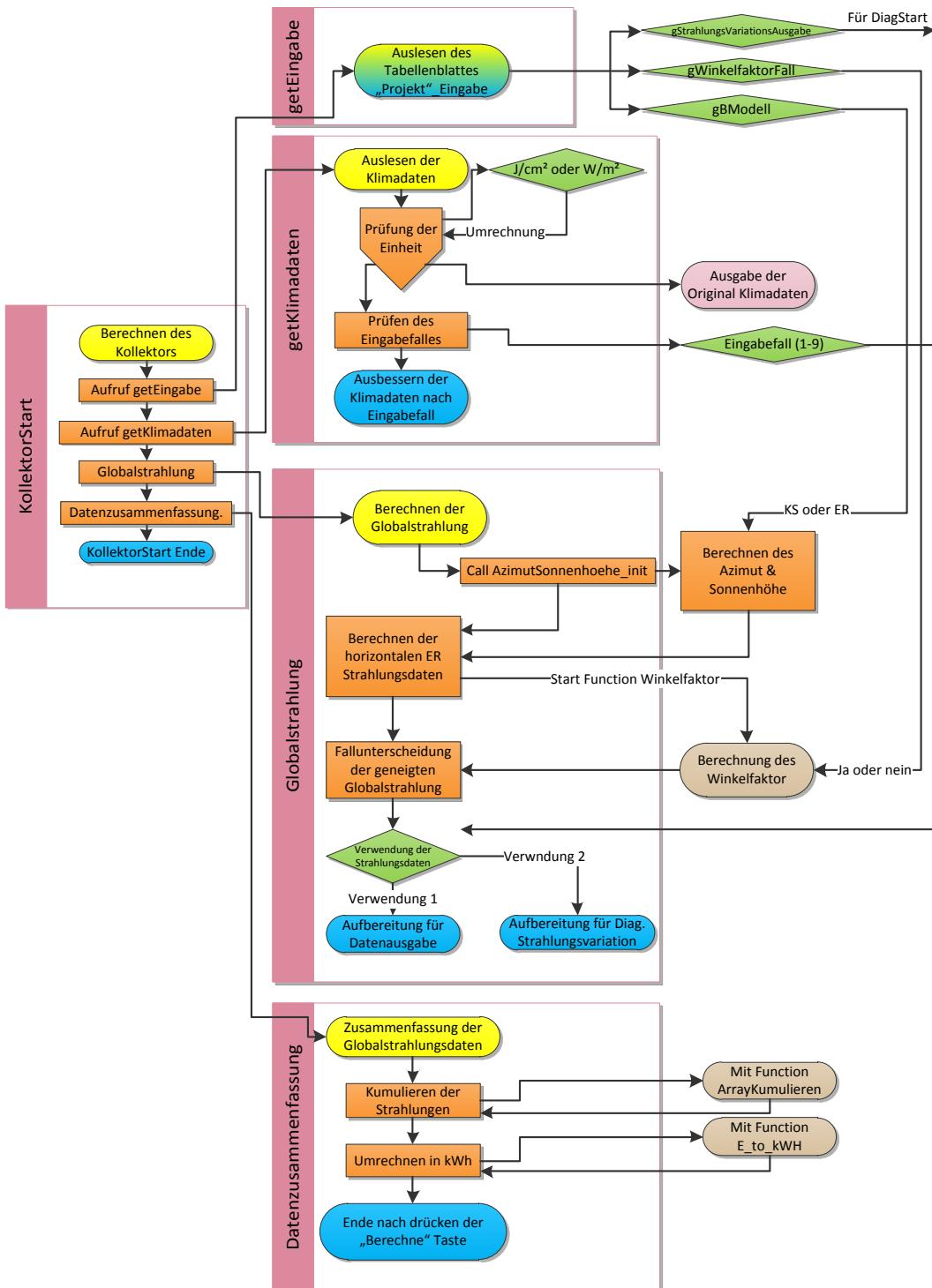


Abbildung 4.6.: Struktogramm des VBA-Programmes nach klicken auf **Berechne**

## 4.4. *MS* Excel Makro tauglich machen

Je nach Einstellung von *MS* Excel werden beim Aufruf von der Datei SolarSimulation.xlsx die Makros<sup>14</sup> mitgeladen oder nicht. Diese Sicherheitseinstellung ist im Normalfall auf Makros deaktivieren eingestellt<sup>15</sup>, da die Gefahr besteht in Makros Virus-Programme zu implementieren. Hiermit bestätige ich, dass ich im Originalfile auf beiliegender CD keine bösartigen Programmteile oder Viren programmiert habe. Um die Makroeinstellung zu aktivieren muss, je nach *MS* Office Version <sup>16</sup>, unter den „Excel Optionen / Vertrauensstellungscenter / Einstellungen für Vertrauensstellungscenter / Einstellungen für Makros“ der Punkt „Alle Makros aktivieren“ ausgewählt und das Kästchen „Zugriff auf das VBA-Projektmodell vertrauen“ angekreuzt werden. Durch Bestätigung mit „OK“ und einem Neustart von *MS* Excel sollte das Tool SolarSimulation gestartet werden.

## 4.5. Weitere Ausblicke zur Programmiersprache VBA

Nach Recherche will Microsoft die VBA Programmierung auflassen und die neue .net Programmiersprache forcieren. Da aber die VBA-Gemeinde eine sehr große Useranzahl besitzt, ist ein offizielles Supportende von Microsoft noch nicht ausgesprochen. In manchen Anwendungen wie Autocad MEP 2011 werden aber beispielsweise die Entwicklertools<sup>17</sup> nicht mehr mit installiert.

---

<sup>14</sup>das in VBA programmierte Script

<sup>15</sup>Mit dieser Einstellung ist keine Verwendung des Programmes SolarSimulation möglich.

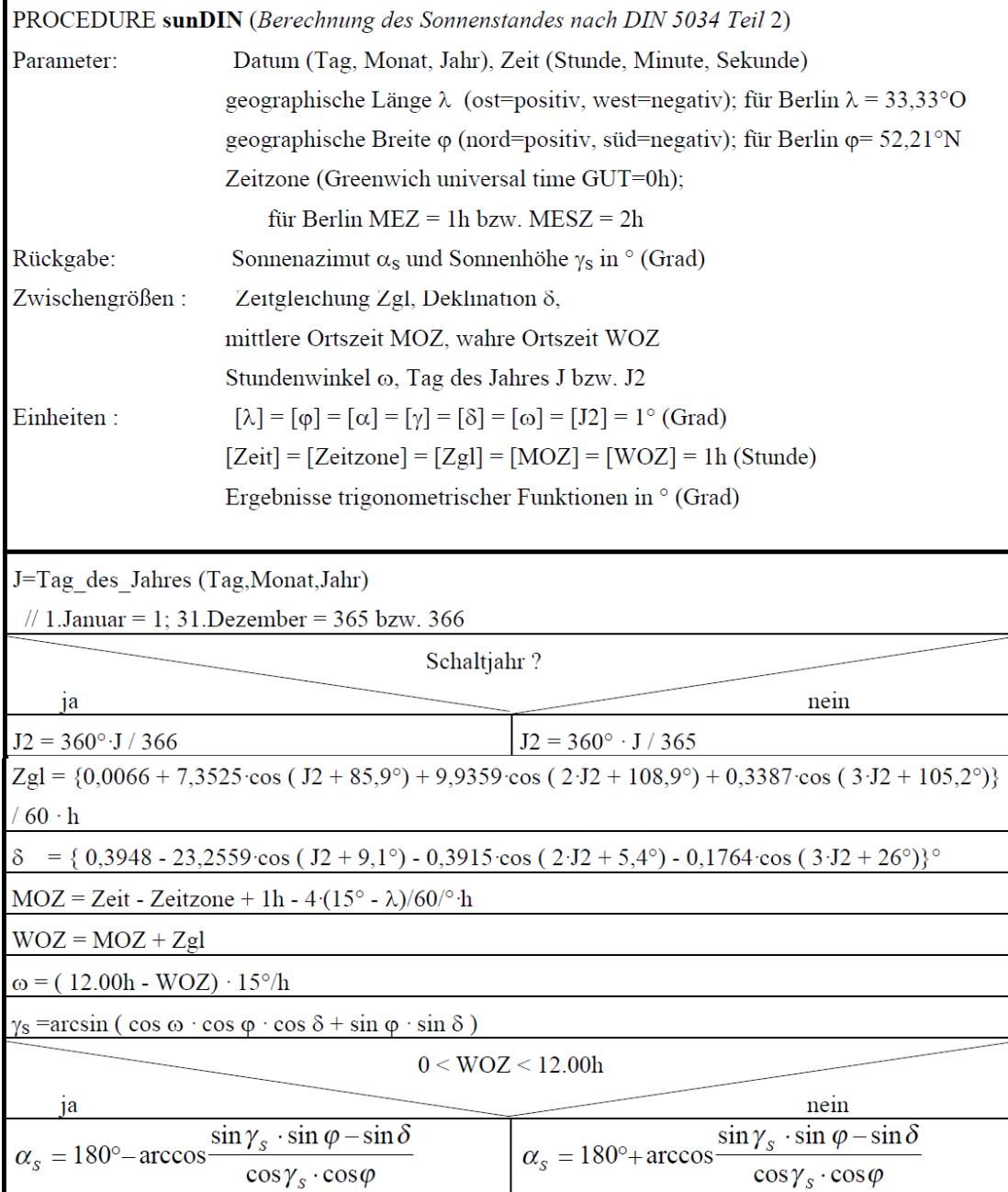
<sup>16</sup>siehe Anleitungen aus dem Internet

<sup>17</sup>so nennt sich die Umgebung in der die VBA Programmierung stattfindet

# Anhang A.

## Struktogramme

### A.1. Struktogramme DIN5034



## A.2. Struktogramme Sunae

**PROCEDURE SUNAE** (*Berechnung des Sonnenstandes nach dem Sunae-Algorithmus*)

Parameter:

- Datum (Tag, Monat, Jahr)
- Zeit (Stunde, Minute, Sekunde),
- geographische Länge  $\lambda$  (ost=positiv, west=negativ)  
für Berlin  $\lambda = 33,33^\circ\text{O}$
- geographische Breite  $\varphi$  (nord=positiv, süd=negativ)  
für Berlin  $\varphi = 52,21^\circ\text{N}$
- Zeitzone (Greenwich universal time GUT=0h)  
für Berlin MEZ = 1h bzw. MESZ = 2h

Rückgabe:

- Sonnenazimut  $\alpha_S$
- Sonnenhöhe  $\gamma_S$  in  $^\circ$  (Grad)

Zwischengrößen und Einheiten:

- $[\Delta J] = [\text{leap}] = 1\text{a}$  (Jahr)
- $[J] = [\text{Time}] = 1\text{d}$  (Tag)
- $[T] = 1\text{h}$  (Stunde)
- $[\theta] = [g] = [L] = [\varepsilon] = [\alpha] = [\delta] = [\text{ST}] = [S] = [H] = [\varphi_B] =$
- $[\text{ETS}_B] = [R_B] = \text{rad}$
- $[\text{ETS}] = [R] = [A] = 1^\circ$  (Grad)

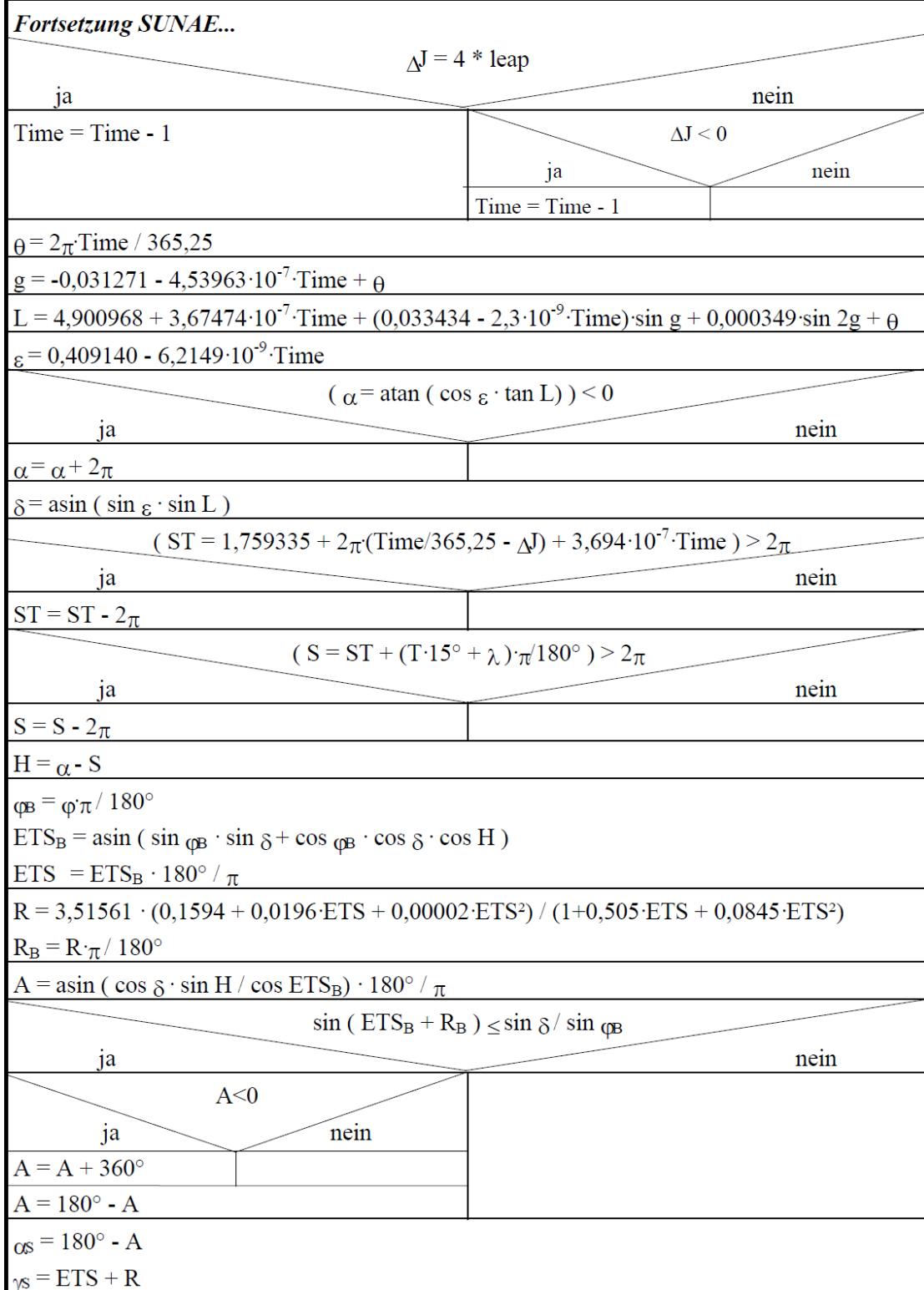
$\Delta J = \text{Jahr} - 1980;$        $\text{leap} = \Delta J \bmod 4;$

$J = \text{Tag\_des\_Jahres} (\text{Tag}, \text{Monat}, \text{Jahr});$  // 1.Januar = 1; 31.Dezember = 365 bzw. 366

$T = \text{Stunde} + (\text{Minute} + \text{Sekunde}/60) / 60 - \text{Zeitzone}$

$\text{Time} = \Delta J \cdot 365 + \text{leap} + J - 1 + T / 24$

*Fortsetzung SUNAE...*



# Anhang B.

## Tabellen

### B.1. Albedo

Untergrund	Albedo	Untergrund	Albedo
Gras(Juli, August)	0,25	Asphalt	0,15
Rasen	0,18...	Wälder	0,05...0,18
trockenes Gras	0,28...0,32	Heide- und Sandflächen	0,10...0,25
nicht bestellte Felder	0,26	Wasserflächen( $\gamma_s > 45^\circ$ )	0,05
nackter Boden	0,17	Wasserflächen( $\gamma_s > 30^\circ$ )	0,08
Schotter	0,18	Wasserflächen( $\gamma_s > 20^\circ$ )	0,12
Beton, verwittert	0,20	Wasserflächen( $\gamma_s > 10^\circ$ )	0,22
Beton, sauber	0,30	frische Schneedecke	0,80...0,90
Zement, sauber	0,55	alte Schneedecke	0,45...0,70

Albedo für unterschiedliche Umgebungen nach [16]

### B.2. Parametertabellen für Perez Algorithmus

$\epsilon$ -Klasse	1	2	3	4	5	6	7	8
$\epsilon$	1,000... 1,065	1,065... 1,230	1,230... 1,500	1,500... 1,950	1,950... 2,800	2,800... 4,500	4,500... 6,200	6,200... $\infty$
$F_{11}$	-0,008	0,130	0,330	0,568	0,873	1,132	1,060	0,678
$F_{12}$	0,588	0,683	0,0487	0,187	-0,392	-1,237	-1,600	-0,327
$F_{13}$	-0,062	-0,151	-0,221	-0,295	-0,362	-0,412	-0,359	-0,250
$F_{21}$	-0,060	-0,019	0,055	0,109	0,226	0,288	0,264	0,156
$F_{22}$	0,072	0,066	-0,064	-0,152	-0,462	-0,823	-1,127	-1,377
$F_{23}$	-0,022	-0,029	-0,026	-0,014	0,001	0,056	0,131	0,251

Konstanten zur Bestimmung des Horizontalhelligkeitsindex  $F_1$  und des Sonnenumgebungshelligkeitsindex  $F_2$  in Abhängigkeit  $\epsilon$  für Gl. 3.30 und 3.31 nach [16]

## Anhang C.

### Diagramme von Solarsimulation

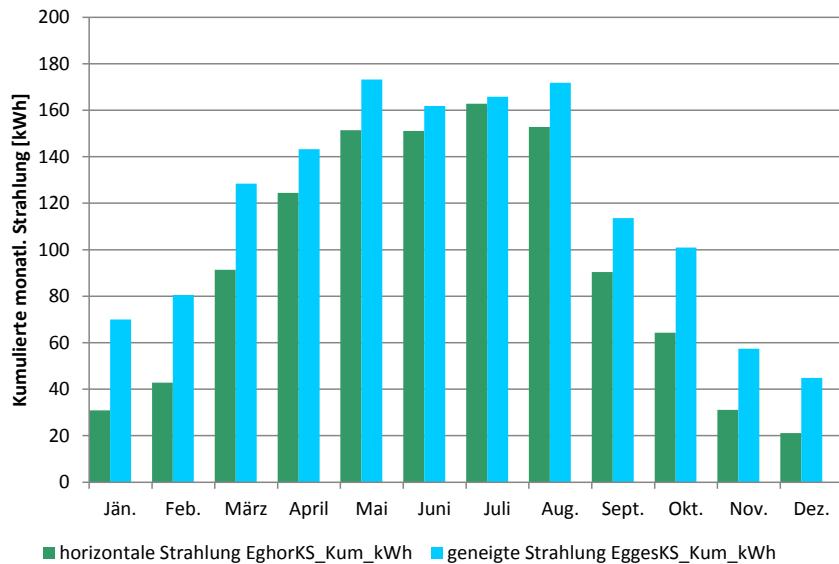


Abbildung C.1.: Gegenüberstellung horizontaler zu geneigter Globalstrahlung

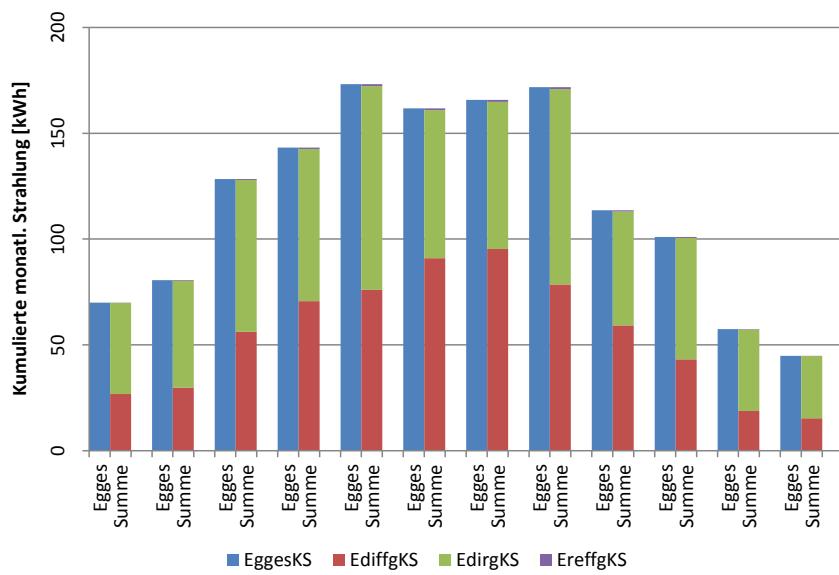


Abbildung C.2.: Aufteilung Direkt-, Reflexions- und Diffusstrahlung

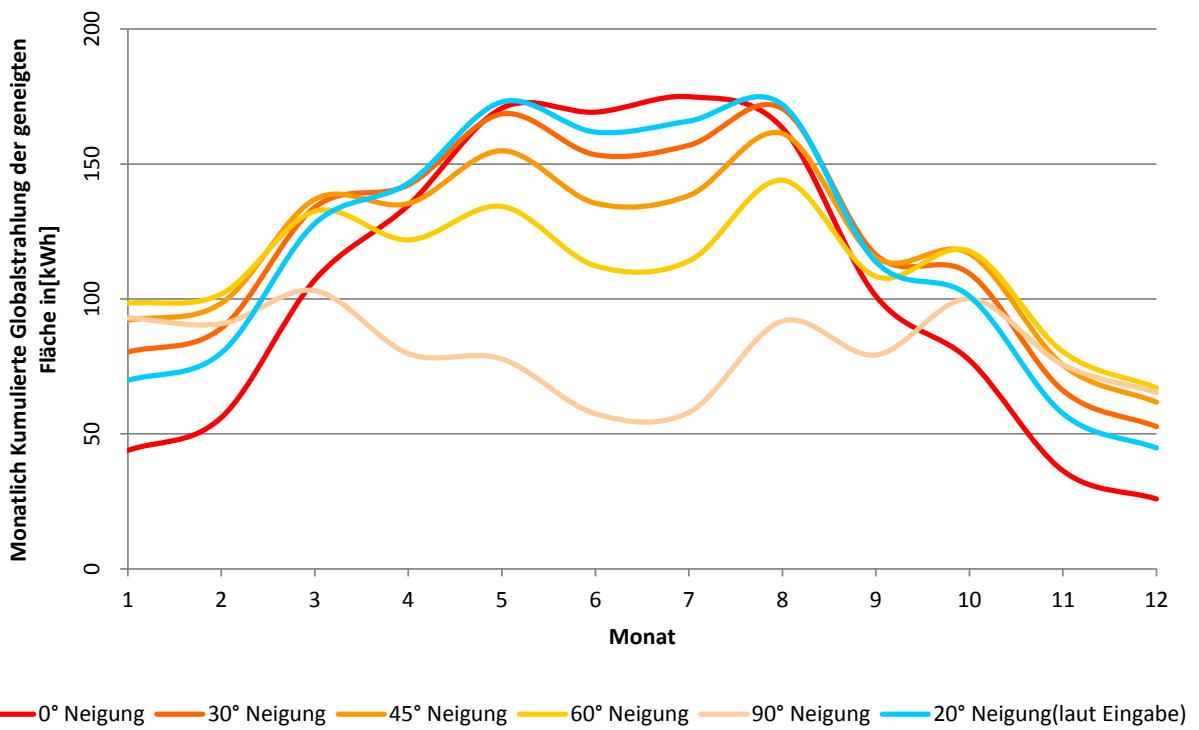


Abbildung C.3.: Jahresgang der Globalstrahlung auf geneigte Flächen

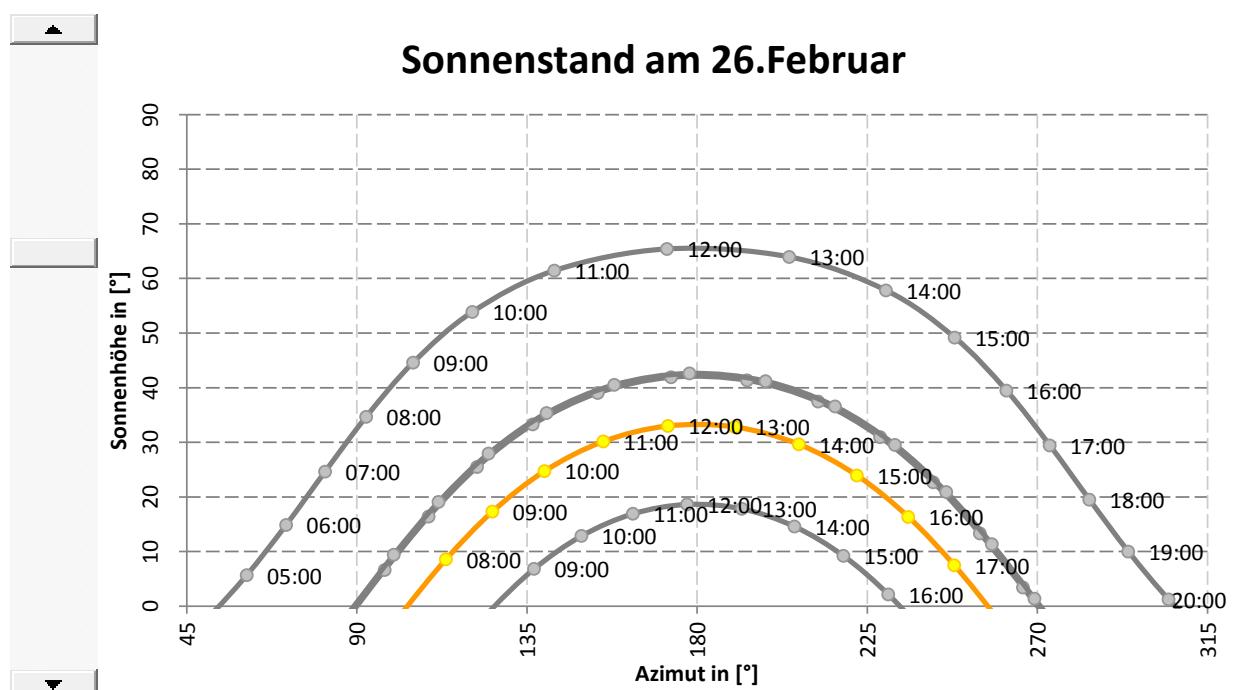
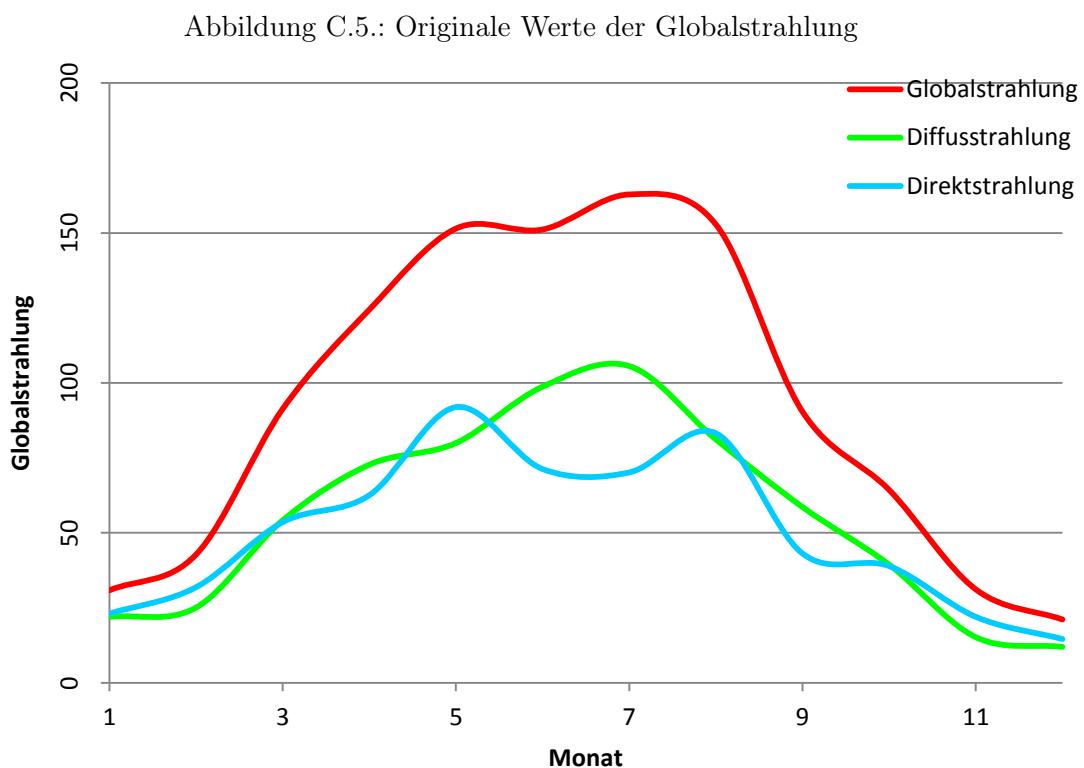
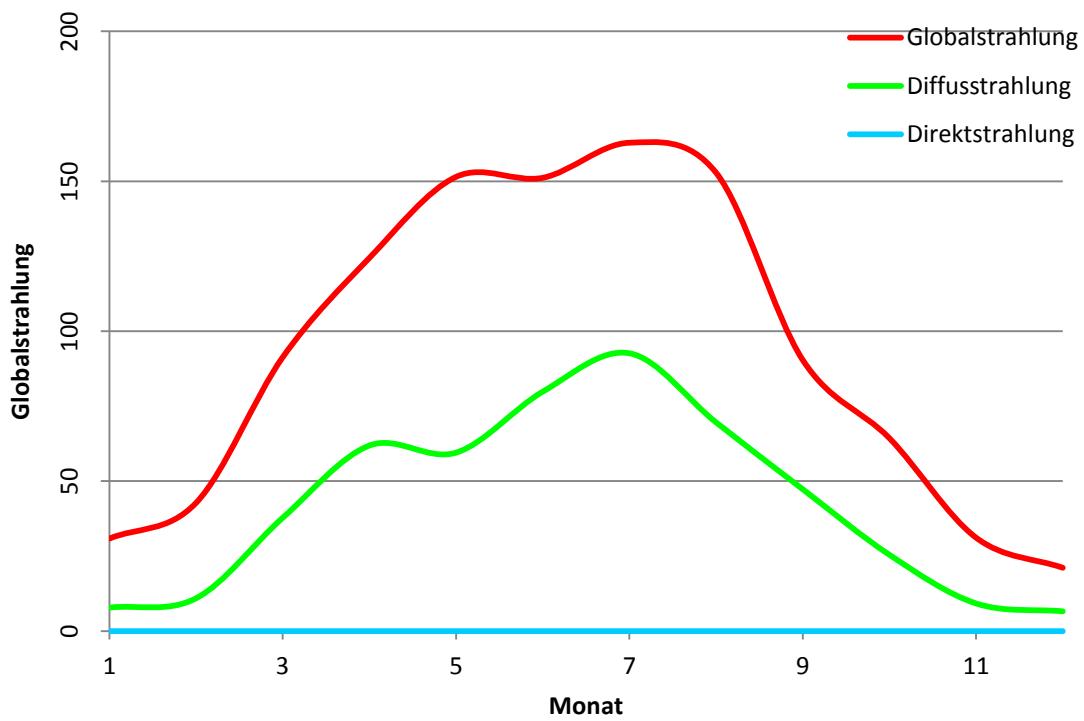


Abbildung C.4.: Sonnenstand mit Scrollbalken



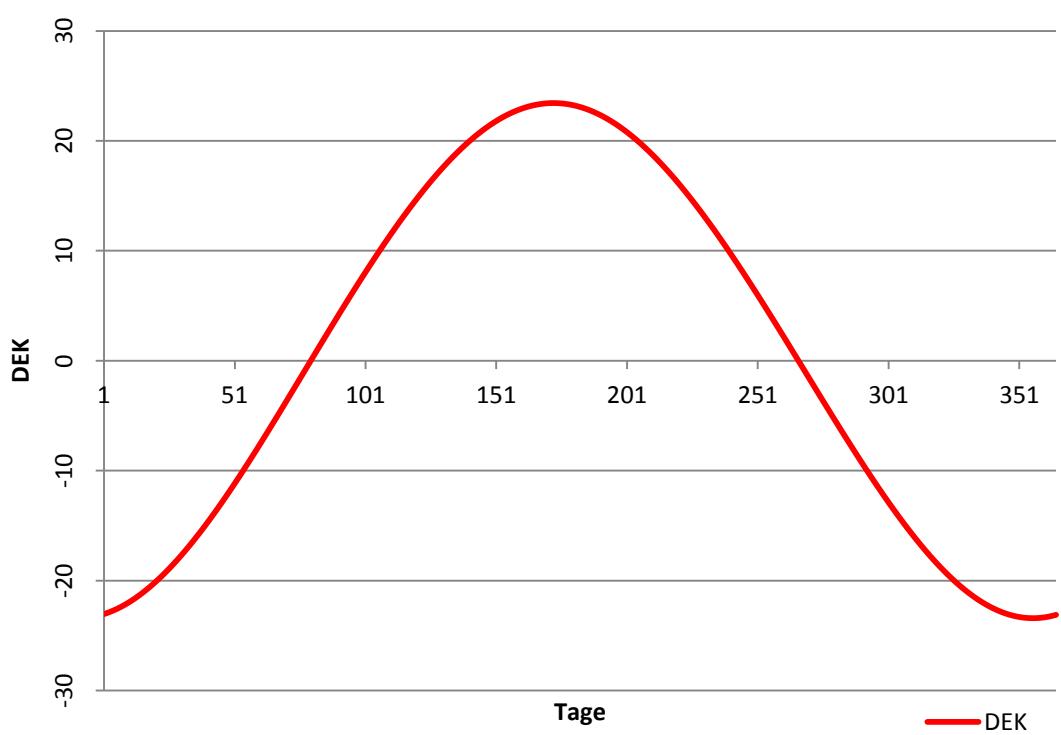


Abbildung C.7.: Sonnendeklination

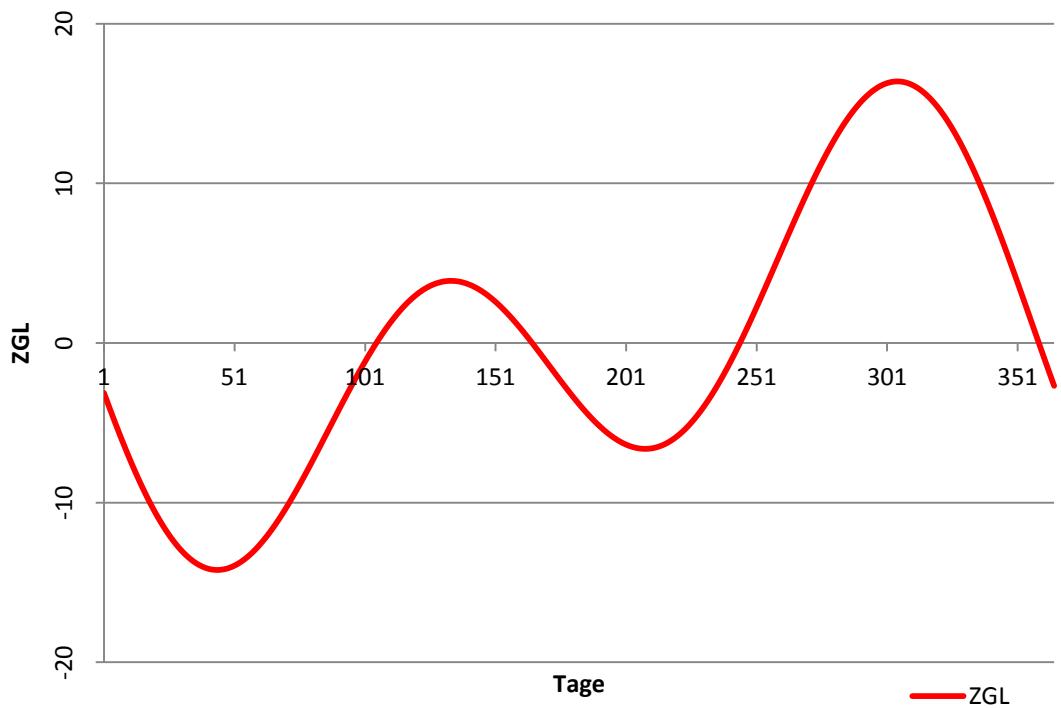


Abbildung C.8.: Zeitgleichung

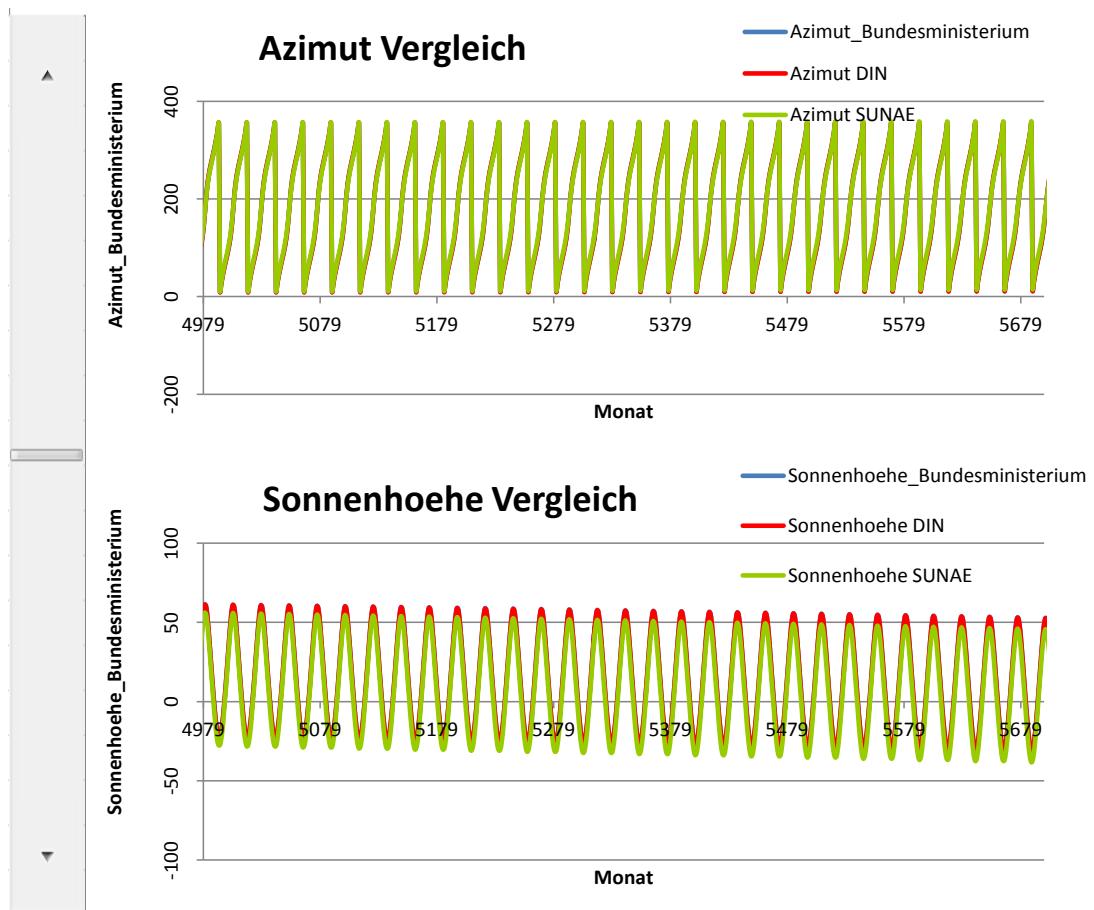
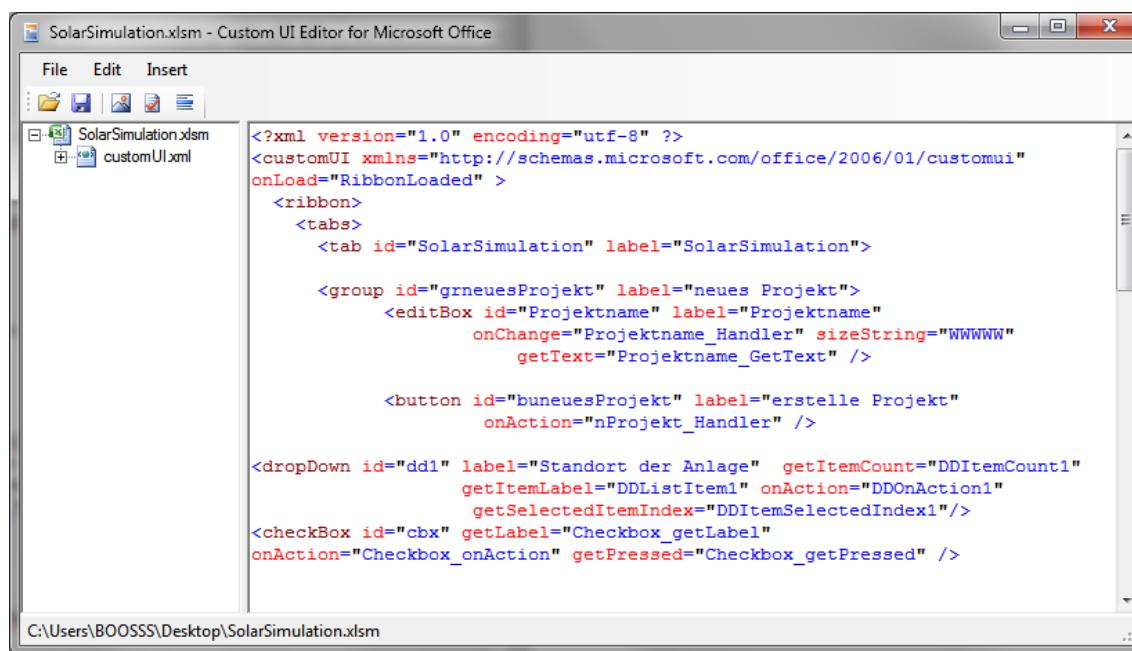


Abbildung C.9.: Vergleich der Berechnungsmodelle 1,2,3 des Sonnenstandes

# Anhang D.

## Bilder

### D.1. Allgemeine Abbildungen



The screenshot shows the Microsoft Custom UI Editor interface. The title bar reads "SolarSimulation.xlsxm - Custom UI Editor for Microsoft Office". The menu bar includes "File", "Edit", and "Insert". On the left, there's a file tree with "SolarSimulation.xlsxm" expanded, showing "customUI.xml". The main area displays the XML code for a ribbon tab:

```
<?xml version="1.0" encoding="utf-8" ?>
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
onLoad="RibbonLoaded" >
    <ribbon>
        <tabs>
            <tab id="SolarSimulation" label="SolarSimulation">
                <group id="grneuesProjekt" label="neues Projekt">
                    <editBox id="Projektname" label="Projektname"
                            onChange="Projektname_Handler" sizeString="WWWWWW"
                            getText="Projektname_GetText" />
                    <button id="buneuesProjekt" label="erstelle Projekt"
                            onAction="nProjekt_Handler" />
                </group>
                <dropDown id="dd1" label="Standort der Anlage" getItemCount="DDItemCount1"
                        getItemLabel="DDListItems" onAction="DDOnAction1"
                        getSelectedIndex="DDItemSelectedIndex1"/>
                <checkBox id="cbx" getLabel="Checkbox_getLabel"
                        onAction="Checkbox_onAction" getPressed="Checkbox_getPressed" />
            </tab>
        </tabs>
    </ribbon>
</customUI>
```

The status bar at the bottom shows the path "C:\Users\BOOSS\Desktop\SolarSimulation.xlsxm".

Programmoberfläche des Custom Ui Editors for [MS](#) in der, der XML-Code geschrieben wird. Download-Link siehe[13]

	A	B	C	D	E	F	G	H	I	J	K
<b>Test</b>											
1					TESS Grundgeometrie						
2					Durchmesser	1,50 m					
3	Standort	Kollektor	Linz SH263mBen.def	Klimasatz	Höhe	2,00 m					
4	GeLänge		14,29 *		Position KV1	1,50 m					
5	GeoBreite		48,30 *		Position KRL	6,00 m					
6	Albedo		0,20 -		Position KolVL	1,95 m					
7	Azimut des Kollektors		0,00 *		n Einlässe KolRL	7,00					
8	Höhenwinkel des Kollektors		60,00 *		n Einlässe HV1	5					
9	C0 optischer Wirkungsgrad		0,72 %		n Einlässe HV1	7					
10	C1 lineare Wärmeverlustkoeffizient		2,25 W/m <sup>2</sup> K		TESS-Weiters						
11	C2 quadratischer Wärmeverlustkoeffizient		0,01 W/m <sup>2</sup> K		Wärme Kapazität des Fluides	1004 J/kgK					
12	Winkelfaktor bei 50° Einfallwinkel		0,89 %		Dichte des Fluides	998,00 kg/m <sup>3</sup>					
13	Einfallwinkel		50,00 *		Dicke der Isolierung	0,30 m					
14	Kollektor Fläche		16,00 m <sup>2</sup>		Wärmeleitfähigkeit der Isolierung	0,00 W/mK					
15	Sonnenstand Berechnungsmodell (1,2,3)		1,00 -		Kessel						
16	Winkelfaktor Berücksichtigung (ja / nein)		ja -		Heizleistung des Kessels	10000 W					
17	Diagrammausgabe Strahlungsvariation (KS oder ER)		KS -		Trinkwarmwasser						
18					Temperatur Trinkwarmwasser	60 °C					
19					mittlere Verbrauch	100 Liter/Tag					
20					Spalte Heizleistung						
21					Außentemperatur A	10 °C					
22					Heizungsvorlauftemperatur A	60 °C					
23					Außentemperatur B	40 °C					
24					Heizungsvorlauftemperatur B	60 °C					
25					Heizungsrücklauftemperatur						
26					Verschiedenes						
27					Temperatur Zufluss	10 °C					
28					max erlaubte Temperatur	80 °C					
29					mittlere Temp. Aufstellung	20 °C					
30											
31											
32											
33											

Abbildung D.1.: Solarsimulation.xlsx Benutzeroberfläche

# Anhang E.

## Code des VBA Programmes Solarsimulation

Der komplette Source Code dieses Programmes liegt auf der beiliegenden CD, im Excel File **Solarsimulation.xlsm** und als **Source\_Code\_Solarsimulation.pdf** bei. Hier ist nur das Grundmodul dieser Arbeit

### E.1. 1. Kapitel Allgemeine Konstanten und 2. Allg. Deklarationen

```
1 Option Explicit
2
3 '######
4 '# 1. Allgemeine Konstanten
5 '######
6 Const pi = 3.14159265358979
7 '######
8 '# 2. Allgemeine Deklarationen
9 '######
10 Public arrStandortSuche() As Variant      'zur bestimmung des Standortes
     in Blatt Klimadaten
11 Public strAusgesuchterStandort As String   'zur bestimmung des Standortes
     in Blatt Klimadaten
12 Public EingabeFall As Integer            'Klimadaten Eingabe
13 '
14 '
15 'Deklarationen des Tabelleblattes "Projektname" _Eingabe
16 '
17 Public gAlbedo As Double                  'Variable von Blatt
     Eingabe => Albedo
18 Public gAzimutEbeneGrad As Double        'Variable von Blatt
     Eingabe => Azimut des Kollektors
19 Public gHohenwinkelEbeneGrad As Double   'Variable von Blatt
     Eingabe => Höhenwinkel des Kollektors
20 Public gAzimutEbeneBog As Double         'Azimut der Ebene im
     Bogenmaß
21 Public gHohenwinkelEbeneBog As Double    'Höhenwinkel des
     Kollektors im Bogenmaß
22 Private gC0 As Double                   'Variable von Blatt
     Eingabe => C0 optischer Wirkungsgrad
23 Private gC1 As Double                   'Variable von Blatt
     Eingabe => C1 lineare Wärmeverlustkoeffizient
24 Private gC2 As Double                   'Variable von Blatt
     Eingabe => C2 quadratischer Wärmeverlustkoeffizient
25 Private gWinkelfaktor50 As Double       'Variable von Blatt
     Eingabe => Winkelfaktor bei 50° Einfallwinkel
```

```

26  Private gEinfallwinkelGrad As Double           ' Variable von Blatt
27    Eingabe => Einfallwinkel
28  Public gKofFlaeche As Double                  ' Variable von Blatt
29    Eingabe => Kollektor Fläche
30  Public gBModell As Integer                   ' Variable von Blatt
31    Eingabe => Sonnenstand Berechnungsmodell (1,2,3)
32  Public gStrahlungsVariationsAusbabe As String ' Variable von Blatt
33    Eingabe => Diagrammausbabe Strahlungsvariation (KS oder ER)
34  Public gWinkelfaktorFALL As String          ' Variable von Blatt
35    Eingabe => Winkelfaktor Berücksichtigung (ja / nein)
36  ,
37  ' Deklarationen der Arrays für Sonnenstand Berechnungsmodell (1,2,3)
38  ,
39  Public arrAzimutGradBMBT() As Double          ,
40  Public arrSonnenHoeheGradBMBT() As Double
41  Public arrAzimutGradDIN() As Double
42  Public arrSonnenHoeheGradDIN() As Double
43  Public arrAzimutGradSUNAE() As Double
44  Public arrSonnenHoeheGradSUNAE() As Double
45  ,
46  ' Allgemeine Deklarationen
47  ,
48  Public arrStunde() As Integer                 ' Array von 1 bis 8670
49  Public arrTagesnummer() As Integer            ' Tage von 0 bis 364
50  Public arrDatum() As Integer                 ' Tage von 1 bis 365
51  Public intMEZ() As Integer                   ' Ortszeit t MEZ(
52    Mitteleuropäische Zeit)
53  Public arrAzimutGrad() As Double             ' Array des Sonnen Azimut in
54    Grad
55  Public arrAzimutBog() As Double              ' Array des Sonnen Azimut im
56    Bogenmaß
57  Public gDatum() As Date                    ' Array mit tt ,mm, jjjj , hh:
58    mm: ss wird
59  Public arrSonnenHoeheGrad() As Double         ' Array der Sonnenhöhe in
60    Grad
61  Public arrSonnenHoeheBog() As Double          ' Array der Sonnenhöhe im
62    Bogenmaß
63  ' Umrechnung der geographischen Konstanten von Grad ins Bogenmaß
64  Public dblGeoLaengeGradKlimadaten As Double
65  Public dblGeoBreiteGradKlimadaten As Double
66  Public dblGeoLaengeGrad As Double
67  Public dblGeoLaengeBog As Double
68  Public dblGeoBreiteGrad As Double
69  Public dblGeoBreiteBog As Double
70  Dim intStundenMax As Integer
71  ,
72  ' Deklarationen der Klimadaten
73  ,
74
75  Public dblT_AussenKS() As Variant            ' wird aus Klimasatz
76  Public EghorKS() As Variant
77    eingeleSEN
78  Public EdiffhorKS() As Variant              ' wird aus Klimasatz
79    eingeleSEN
80  Public EdirhorKS() As Variant               ' wird aus Klimasatz
81    eingeleSEN
82
83  Public EdirhorER() As Double
84  Public EdiffhorER() As Double                ' Ausgerechnet

```

```

73
74  Public thetagen() As Double
75  Public thetagenGrad() As Double
76  Public epsilonER_Grad() As Double
77  Public epsilonKS_Grad() As Double
78
79  Public EdiffgenER() As Double
80  Public EdiffgenKS() As Double
81  Public EdiffgenIsoER() As Double
82  Public EdiffgenIsoKS() As Double
83  Public EdiffgenAnIsoER() As Double
84  Public EdiffgenAnIsoKS() As Double
85  Public EdirgenER() As Double
86  Public EdirgenKS() As Double
87  Public Ereflgen() As Double
88  Public EggesER() As Double
89  Public EggesKS() As Double
90
91  Public E0hor() As Double
92
93  Public arrWinkelfaktorKollektor() As Double
94  Public arrWirkungsgradKollektor(1 To 8760) As Double
95
96  Dim zwischenfaktor() As Double
97  '
98  'Übergabevariablen an Modul Speicher für Ausgabe
99  '
100 Public mOffsetKollektorAusgabe As Integer 'letzte Spalte die Kollektor-
101           Modul in der Ausgabe benötigt, zur Übergabe an das Speicher-Modul
102 Public mOffsetAusgabeKum As Integer 'letzte Spalte die Kollektor-Modul
103           in der AusgabeKum benötigt, zur Übergabe an das Speicher-Modul
104  '
105  'Deklarationen von kumulierten Werten und Umrechnungen in kWh
106  '
107  'Dekl. der kumulierten Werte
108  'Horizontal
109  Public EghorKS_Kum(1 To 12) As Double
110
111  Public EdiffhorER_Kum(1 To 12) As Double
112  Public EdiffhorKS_Kum(1 To 12) As Double
113
114  Public EdirhorER_Kum(1 To 12) As Double
115  Public EdirhorKS_Kum(1 To 12) As Double
116  'Geneigt
117  Public EggesER_Kum(1 To 12) As Double
118  Public EggesKS_Kum(1 To 12) As Double
119
120  Public EdiffgenER_Kum(1 To 12) As Double      'Perez Modell Errechnet
121  Public EdiffgenKS_Kum(1 To 12) As Double      'Perez Modell Klimasatz
122
123  Public EdirgenER_Kum(1 To 12) As Double
124  Public EdirgenKS_Kum(1 To 12) As Double
125
126  'Dekl. der kumulierten kWh Werte
127  'Horizontal
128  Public EghorKS_Kum_kWh(1 To 12) As Double
129
130  Public EdiffhorER_Kum_kWh(1 To 12) As Double
131  Public EdiffhorKS_Kum_kWh(1 To 12) As Double

```

```

131
132  Public EdirhorER_Kum_kWh(1 To 12) As Double
133  Public EdirhorKS_Kum_kWh(1 To 12) As Double
134  'Geneigt
135  Public EggesKS_Kum_kWh(1 To 12) As Double
136  Public EggesER_Kum_kWh(1 To 12) As Double
137
138  Public EdirgenER_Kum_kWh(1 To 12) As Double
139  Public EdirgenKS_Kum_kWh(1 To 12) As Double
140
141
142  Public EdiffgenER_Kum_kWh(1 To 12) As Double 'Perez Modell Errechnet
143  Public EdiffgenKS_Kum_kWh(1 To 12) As Double 'Perez Modell Klimasatz
144
145  Public Erefftgen_kum_kWh(1 To 12) As Double
146  '
147  'Deklarationen von Debug Variablen
148  '
149  Public mdebug_showDiag As Boolean      'wird in KollektorStart definiert
     zeigt zusätzliche Diagramme zur Fehlersuche

```

Listing E.1: Modul Kollektor Kapitel 1 und 2

## E.2. 3. Kapitel Allgemeine Funktionen und Subs

```

1 Sub Kapitel_3()
2 '######
3 '## 3. Allgemeine Funktionen und Subs
4 '######
5 End Sub
6
7 Public Function ArcCos(x)
8     ArcCos = Atn(-x / Sqr(-x * x + 1)) + 2 * Atn(1)
9 End Function ' ArcCos
10 Public Function ArcSin(x)
11     ArcSin = Atn(x / Sqr(-x * x + 1))
12 End Function ' ArcSin
13 Public Function rad(grad) 'von Grad => Bogenmaß (Ausgabe Bogenmaß)
14     rad = grad * pi / 180
15 End Function
16 Public Function grad(rad) 'von Bogenmaß => Grad (Ausgabe Grad)
17     grad = rad * 180 / pi
18 End Function
19 Sub ArrayKumulieren(arrArray, arrArrayKum)
20     '
21     ' Erzeugt mit dem übergebenen arrArray kumulierte Arrays
22     ' und gibt diese in marrArrayKum zurück
23     '
24 'ReDim marrArrayKum(1 To 12) As Double
25 Dim i As Integer
26 Dim marrArrayKum(1 To 12) As Double
27
28 If NumberOfArrayDimensions(arrArray) = 1 Then
29     For i = 1 To 744      'Jänner
30         marrArrayKum(1) = marrArrayKum(1) + arrArray(i)
31     Next
32     For i = 745 To 1440   'Februar
33         marrArrayKum(2) = marrArrayKum(2) + arrArray(i)
34     Next
35     For i = 1441 To 2184  'März
36         marrArrayKum(3) = marrArrayKum(3) + arrArray(i)

```

```

37   Next
38     For i = 2185 To 2904 'April
39       marrArrayKum(4) = marrArrayKum(4) + arrArray(i)
40   Next
41     For i = 2905 To 3648 'Mai
42       marrArrayKum(5) = marrArrayKum(5) + arrArray(i)
43   Next
44     For i = 3649 To 4368 'Juni
45       marrArrayKum(6) = marrArrayKum(6) + arrArray(i)
46   Next
47     For i = 4369 To 5112 'Juli
48       marrArrayKum(7) = marrArrayKum(7) + arrArray(i)
49   Next
50     For i = 5113 To 5856 'August
51       marrArrayKum(8) = marrArrayKum(8) + arrArray(i)
52   Next
53     For i = 5857 To 6576 'September
54       marrArrayKum(9) = marrArrayKum(9) + arrArray(i)
55   Next
56     For i = 6577 To 7320 'Oktober
57       marrArrayKum(10) = marrArrayKum(10) + arrArray(i)
58   Next
59     For i = 7321 To 8040 'November
60       marrArrayKum(11) = marrArrayKum(11) + arrArray(i)
61   Next
62     For i = 8041 To 8760 'Dezember
63       marrArrayKum(12) = marrArrayKum(12) + arrArray(i)
64   Next
65 End If
66 If NumberOfArrayDimensions(arrArray) = 2 Then
67   If UBound(arrArray, 2) > 1 Then
68     MsgBox ("Übergebene Array hat mehr als eine Spalte")
69   End If
70   If UBound(arrArray, 2) = 1 Then
71     For i = 1 To 744 'Jänner
72       marrArrayKum(1) = marrArrayKum(1) + arrArray(i, 1)
73   Next
74     For i = 745 To 1440 'Februar
75       marrArrayKum(2) = marrArrayKum(2) + arrArray(i, 1)
76   Next
77     For i = 1441 To 2184 'März
78       marrArrayKum(3) = marrArrayKum(3) + arrArray(i, 1)
79   Next
80     For i = 2185 To 2904 'April
81       marrArrayKum(4) = marrArrayKum(4) + arrArray(i, 1)
82   Next
83     For i = 2905 To 3648 'Mai
84       marrArrayKum(5) = marrArrayKum(5) + arrArray(i, 1)
85   Next
86     For i = 3649 To 4368 'Juni
87       marrArrayKum(6) = marrArrayKum(6) + arrArray(i, 1)
88   Next
89     For i = 4369 To 5112 'Juli
90       marrArrayKum(7) = marrArrayKum(7) + arrArray(i, 1)
91   Next
92     For i = 5113 To 5856 'August
93       marrArrayKum(8) = marrArrayKum(8) + arrArray(i, 1)
94   Next
95     For i = 5857 To 6576 'September
96       marrArrayKum(9) = marrArrayKum(9) + arrArray(i, 1)
97   Next

```

```

98  For i = 6577 To 7320 'Oktober
99      marrArrayKum(10) = marrArrayKum(10) + arrArray(i, 1)
100     Next
101     For i = 7321 To 8040 'November
102         marrArrayKum(11) = marrArrayKum(11) + arrArray(i, 1)
103         Next
104         For i = 8041 To 8760 'Dezember
105             marrArrayKum(12) = marrArrayKum(12) + arrArray(i, 1)
106             Next
107             End If
108         End If
109
110     For i = 1 To 12
111         arrArrayKum(i) = marrArrayKum(i)
112     Next
113
114 End Sub
115 Sub arrDatum_to_strDatum(tag, strDatum)
116     '
117     ' Erzeugt einen String (Text) anhand einer Zahl von
118     ' 1 bis 365, Ausgabe: "Tag.Monat"
119     ' wird als strDatum zurückgegeben und muss dort
120     ' deklariert sein wo es aufgerufen wird
121     '
122     If tag <= 31 Then
123         strDatum = tag & ".Jänner"
124     ElseIf tag > 31 And tag <= 59 Then
125         strDatum = tag - 31 & ".Februar"
126     ElseIf tag > 59 And tag <= 90 Then
127         strDatum = tag - 59 & ".März"
128     ElseIf tag > 90 And tag <= 120 Then
129         strDatum = tag - 90 & ".April"
130     ElseIf tag > 120 And tag <= 151 Then
131         strDatum = tag - 120 & ".Mai"
132     ElseIf tag > 151 And tag <= 181 Then
133         strDatum = tag - 151 & ".Juni"
134     ElseIf tag > 181 And tag <= 212 Then
135         strDatum = tag - 181 & ".Juli"
136     ElseIf tag > 212 And tag <= 243 Then
137         strDatum = tag - 212 & ".August"
138     ElseIf tag > 243 And tag <= 273 Then
139         strDatum = tag - 243 & ".September"
140     ElseIf tag > 273 And tag <= 304 Then
141         strDatum = tag - 273 & ".Oktober"
142     ElseIf tag > 273 And tag <= 334 Then
143         strDatum = tag - 273 & ".November"
144     ElseIf tag > 334 And tag <= 365 Then
145         strDatum = tag - 334 & ".Dezember"
146     End If
147 End Sub
148
149 Sub E_to_kWH(arrArray_kum, arrArray_kum_kWh)
150     '
151     ' Umrechnung in kWh
152     '
153     Dim i As Integer
154     Dim dblStunde As Integer
155     dblStunde = 1
156     'Umrechnung in kWh
157     For i = 1 To 12

```

```

158     arrArray_kum_kWh(i) = (arrArray_kum(i) * gKofFlaeche * dblStunde) /
159     1000
160
161 Next
162
163 End Sub
164
165 Sub kum_kwh(arrArray , arrArray_kum_kWh)
166     '
167     ' Erzeugt aus den Arrays kumulierte monatliche Werte und rechnet diese in
168     ' kWh um
169
170 Dim arrArrayKum(1 To 12) As Double
171
172 Dim i As Integer
173 Dim parrArraykum(1 To 12) As Double
174 Dim parrArray_kum_kWh(1 To 12) As Double
175
176 If NumberOfArrayDimensions(arrArray) = 1 Then
177     For i = 1 To 744      'Jänner
178         parrArraykum(1) = parrArraykum(1) + arrArray(i)
179     Next
180     For i = 745 To 1440  'Februar
181         parrArraykum(2) = parrArraykum(2) + arrArray(i)
182     Next
183     For i = 1441 To 2184 'März
184         parrArraykum(3) = parrArraykum(3) + arrArray(i)
185     Next
186     For i = 2185 To 2904 'April
187         parrArraykum(4) = parrArraykum(4) + arrArray(i)
188     Next
189     For i = 2905 To 3648 'Mai
190         parrArraykum(5) = parrArraykum(5) + arrArray(i)
191     Next
192     For i = 3649 To 4368 'Juni
193         parrArraykum(6) = parrArraykum(6) + arrArray(i)
194     Next
195     For i = 4369 To 5112 'Juli
196         parrArraykum(7) = parrArraykum(7) + arrArray(i)
197     Next
198     For i = 5113 To 5856 'August
199         parrArraykum(8) = parrArraykum(8) + arrArray(i)
200     Next
201     For i = 5857 To 6576 'September
202         parrArraykum(9) = parrArraykum(9) + arrArray(i)
203     Next
204     For i = 6577 To 7320 'Oktober
205         parrArraykum(10) = parrArraykum(10) + arrArray(i)
206     Next
207     For i = 7321 To 8040 'November
208         parrArraykum(11) = parrArraykum(11) + arrArray(i)
209     Next
210 End If
211 If NumberOfArrayDimensions(arrArray) = 2 Then
212     If UBound(arrArray , 2) > 1 Then
213         MsgBox ("Übergebene Array hat mehr als eine Spalte")
214     End If
215     If UBound(arrArray , 2) = 1 Then
216         For i = 1 To 744      'Jänner

```

```

217     parrArraykum(1) = parrArraykum(1) + arrArray(i, 1)
218 Next
219     For i = 745 To 1440 'Februar
220         parrArraykum(2) = parrArraykum(2) + arrArray(i, 1)
221 Next
222     For i = 1441 To 2184 'März
223         parrArraykum(3) = parrArraykum(3) + arrArray(i, 1)
224 Next
225     For i = 2185 To 2904 'April
226         parrArraykum(4) = parrArraykum(4) + arrArray(i, 1)
227 Next
228     For i = 2905 To 3648 'Mai
229         parrArraykum(5) = parrArraykum(5) + arrArray(i, 1)
230 Next
231     For i = 3649 To 4368 'Juni
232         parrArraykum(6) = parrArraykum(6) + arrArray(i, 1)
233 Next
234     For i = 4369 To 5112 'Juli
235         parrArraykum(7) = parrArraykum(7) + arrArray(i, 1)
236 Next
237     For i = 5113 To 5856 'August
238         parrArraykum(8) = parrArraykum(8) + arrArray(i, 1)
239 Next
240     For i = 5857 To 6576 'September
241         parrArraykum(9) = parrArraykum(9) + arrArray(i, 1)
242 Next
243     For i = 6577 To 7320 'Oktober
244         parrArraykum(10) = parrArraykum(10) + arrArray(i, 1)
245 Next
246     For i = 7321 To 8040 'November
247         parrArraykum(11) = parrArraykum(11) + arrArray(i, 1)
248 Next
249     For i = 8041 To 8760 'Dezember
250         parrArraykum(12) = parrArraykum(12) + arrArray(i, 1)
251 Next
252     End If
253 End If
254 Dim dblStunde As Integer
255     dblStunde = 1
256     'Umrechnung in kWh
257     For i = 1 To 12
258         parrArray_kum_kWh(i) = (parrArraykum(i) * gKolFlaeche * dblStunde)
259             / 1000
260         arrArray_kum_kWh(i) = parrArray_kum_kWh(i)
261     Next
262 End Sub
263
264 ,
265 ' Gibt den Namen des Monats nach der Monatszahl 1-12 als Namen zurück
266
267 Function strMonat(intMonat)
268     Select Case intMonat
269         Case 1
270             strMonat = "Jän."
271         Case 2
272             strMonat = "Feb."
273         Case 3
274             strMonat = "März"
275         Case 4
276             strMonat = "April"

```

```

277 Case 5
278     strMonat = "Mai"
279 Case 6
280     strMonat = "Juni"
281 Case 7
282     strMonat = "Juli"
283 Case 8
284     strMonat = "Aug."
285 Case 9
286     strMonat = "Sept."
287 Case 10
288     strMonat = "Okt."
289 Case 11
290     strMonat = "Nov."
291 Case 12
292     strMonat = "Dez."
293 End Select
294 End Function
295 Sub FParameterPerez(g_epsilon, F_ER_oder_KS)
296 ##### Zwischenrechnung und Rückgabe der folgenden
297 ##### Parameter für Perez Modell
298 ##### 'F11=> F_ER_oder_KS(1,1)
299 ##### 'F12=> F_ER_oder_KS(1,2)
300 ##### 'F13=> F_ER_oder_KS(1,3)
301 ##### 'F21=> F_ER_oder_KS(2,1)
302 ##### 'F22=> F_ER_oder_KS(2,2)
303 ##### 'F23=> F_ER_oder_KS(2,3)
304 #####
305 #####
306 If g_epsilon > 1 And g_epsilon <= 1.065 Then
307     F_ER_oder_KS(1, 1) = -0.008
308     F_ER_oder_KS(1, 2) = 0.588
309     F_ER_oder_KS(1, 3) = -0.062
310     F_ER_oder_KS(2, 1) = -0.06
311     F_ER_oder_KS(2, 2) = 0.072
312     F_ER_oder_KS(2, 3) = -0.022
313 ElseIf g_epsilon > 1.065 And g_epsilon <= 1.23 Then
314     F_ER_oder_KS(1, 1) = 0.13
315     F_ER_oder_KS(1, 2) = 0.0682
316     F_ER_oder_KS(1, 3) = -0.151
317     F_ER_oder_KS(2, 1) = -0.019
318     F_ER_oder_KS(2, 2) = 0.066
319     F_ER_oder_KS(2, 3) = -0.029
320 ElseIf g_epsilon > 1.23 And g_epsilon <= 1.5 Then
321     F_ER_oder_KS(1, 1) = 0.33
322     F_ER_oder_KS(1, 2) = 0.487
323     F_ER_oder_KS(1, 3) = -0.221
324     F_ER_oder_KS(2, 1) = 0.55
325     F_ER_oder_KS(2, 2) = -0.064
326     F_ER_oder_KS(2, 3) = -0.026
327 ElseIf g_epsilon > 1.5 And g_epsilon <= 1.95 Then
328     F_ER_oder_KS(1, 1) = 0.568
329     F_ER_oder_KS(1, 2) = 0.187
330     F_ER_oder_KS(1, 3) = -0.295
331     F_ER_oder_KS(2, 1) = 0.109
332     F_ER_oder_KS(2, 2) = -0.152
333     F_ER_oder_KS(2, 3) = -0.014
334 ElseIf g_epsilon > 1.95 And g_epsilon <= 2.8 Then
335     F_ER_oder_KS(1, 1) = 0.873
336     F_ER_oder_KS(1, 2) = -0.392
337     F_ER_oder_KS(1, 3) = -0.362

```

```

338 F_ER_oder_KS(2, 1) = 0.226
339 F_ER_oder_KS(2, 2) = -0.462
340 F_ER_oder_KS(2, 3) = 0.001
341 ElseIf g_epsilon > 2.8 And g_epsilon <= 4.5 Then
342     F_ER_oder_KS(1, 1) = 1.132
343     F_ER_oder_KS(1, 2) = -1.237
344     F_ER_oder_KS(1, 3) = -0.412
345     F_ER_oder_KS(2, 1) = 0.288
346     F_ER_oder_KS(2, 2) = -0.832
347     F_ER_oder_KS(2, 3) = 0.056
348 ElseIf g_epsilon > 4.5 And g_epsilon <= 6.2 Then
349     F_ER_oder_KS(1, 1) = 1.06
350     F_ER_oder_KS(1, 2) = -1.6
351     F_ER_oder_KS(1, 3) = -0.359
352     F_ER_oder_KS(2, 1) = 0.264
353     F_ER_oder_KS(2, 2) = -1.127
354     F_ER_oder_KS(2, 3) = 0.131
355 ElseIf g_epsilon > 6.2 Then
356     F_ER_oder_KS(1, 1) = 0.678
357     F_ER_oder_KS(1, 2) = -0.327
358     F_ER_oder_KS(1, 3) = -0.25
359     F_ER_oder_KS(2, 1) = 0.156
360     F_ER_oder_KS(2, 2) = -1.377
361     F_ER_oder_KS(2, 3) = 0.251
362 End If
363
364 End Sub

```

Listing E.2: Modul Kollektor Kapitel 3

### E.3. 4. Kapitel Initialisierung des Programmes

```

1 Sub Kapitel_4()
2 '#####
3 '## 4. Initialisierung des Programmes
4 '##
5 End Sub
6
7
8
9 Sub KollektorStart()
10 '
11 ' Wird vom Button in der Ribbon Leiste aufgerufen
12 ' welcher bearbeitet wird von ClickHandler.berechneHaendler
13 ' soll den Programmdurchlauf des Kollektors starten und initialisieren
14 "
15 mdebug_showDiag = False           ' Aktiviert mit True weitere
16     Diagramme zur Fehlersuche
17 Call getEingabe                  ' Erfassung der Eingabedaten
18 Call getKlimadaten               ' Erfassung der Klimadaten
19 Call Globalstrahlung(gHoehenwinkelEbeneGrad, 1, EingabeFall)
20     Berechnung der Globalstrahlung
21 Call Datenzusammenfassung_fur_Output   ' Aufbereitung der Datensätze
22 End Sub
23 Sub Diag_Start(pDiagGruppe As Integer)
24 '
25 ' Wird vom Modul ClickHandler mit den Schaltflächen
26 ' in der Visualisierungs Werkzeugbox aufgerufen
27 ' Startet alle Diagramme bis auf Diagramm Original welches direkt vom Sub
28 ' get_Klimadaten ausgeführt wird um fälschliche Überschreibungen zu

```

```

28   ' verhindern .
29   ,
30 Select Case pDiagGruppe
31   Case 1 'Button Energie
32
33     Call diag_Klimadaten(2)           ' Fall 2 ist nach
34       Klimadaten-Anpassung
35     Call diag_DiffusVergleich      ' Darstellung der diffusen
36       Strahlungsansätze im Vergleich
37     Call diag_Geneigt_Horizont_GST ' Darstellung geneigter
38       gegen horizontale Strahlung
39     Call Diag_Clustered_Stacked_Column_Chart ' Aufteilung der direkten,
40       diffusen und reflektierter Strahlung
41
42   Case 2 'Button Strahlungsvariation
43     Call Diag_GlobalNeigung        ' Darstellung der
44       Strahlungsvariation anhand unterschiedlicher Kollektorwinkel
45
46   Case 3 'Button Sonnenstand
47     Call Diag_Sonnenstand_Scroll_Chart ' Darstellung des
48       Sonnenstandes
49     Call get_ZeitgleichungDeklination ' Darstellung der
50       Zeitgleichung und Deklination
51     Call Diag_ZOOM_Scroll_Balken    ' Darstellung Azimut/
52       Sonnenhöhen Vergleich
53
54 End Select
55 End Sub
56 Public Sub Diag_Update()
57   ,
58   ' Updatet alle vom Kollektor Modul erzeugten
59   ' Diagramme des aktuellen Projekts
59   ,
60
61 Dim funcNames(1 To 10) As String
62 Dim Names() As Variant
63 Dim i As Integer
64 Names = Array("Clustered", "JahresgangGlobalstrahlung", "Sonnenstand",
65   "HorGen_Strahlung", _
66     "Vergleich der Globalstrahlung auf geneigte Fläche mit
67     drei diffusen Ansätze (ER)", _
68     "Ausgebesserte Werte aus Klimadaten", _
69     "Original Klimadaten Werte ohne Einheitenberichtigung")
69   ' Abfrage ob berechnet '
70 If testberechnet() = False Then
71   Exit Sub
72 End If
73 On Error Resume Next
74 For i = 0 To UBound(Names)
75   Select Case Worksheets(strDiagramme).ChartObjects.Item(Names(i)).name
76     Case "Clustered": Call Diag_Clustered_Stacked_Column_Chart
77     Case "JahresgangGlobalstrahlung": Call Diag_GlobalNeigung
78     Case "Sonnenstand": Call Diag_Sonnenstand_Scroll_Chart
79     Case "HorGen_Strahlung": Call diag_Geneigt_Horizont_GST
80     Case "Vergleich der Globalstrahlung auf geneigte Fläche mit
81     drei diffusen Ansätze (ER)": Call diag_DiffusVergleich
82     Case "Ausgebesserte Werte aus Klimadaten": Call diag_Klimadaten
83       (2)
84     Case "Original Klimadaten Werte ohne Einheitenberichtigung":
85       Call diag_Klimadaten(3)
86   End Select
87   Next i
88 End Sub
89 Sub AzimutSonnenhoehe_init()

```

```

75 '
76 ' Wird vom Sub Globalstrahlung gestartet um je nach Eingabe
77 ' des Parameters gBModell =>(Sonnenstand Berechnungsmodell 1,2 oder 3)
78 ' die richtigen Werte für Azimut und Sonnenhöhe zu erhalten.
79 '
80
81 Select Case gBModell
82
83 Case 1      'Modell vom BMBT
84     Call Kollektor.AzimutSonnenhoehe(intStundenMax)
85
86 Case 2      'Modell DIN
87     Call Kollektor.get_ZeitArrays(intStundenMax)
88     Call Kollektor.AzimutSonnenhoehe_DIN_SUNAU(1)
89
90 Case 3      'Modell SUNAE
91     Call Kollektor.get_ZeitArrays(intStundenMax)
92     Call Kollektor.AzimutSonnenhoehe_DIN_SUNAU(2)
93 End Select
94 End Sub

```

Listing E.3: Modul Kollektor Kapitel 4

## E.4. 5. Kapitel Hauptprogrammsequenzen

```

1 Sub Kapitel_5()
2 ##### 5. Hauptprogrammsequenzen
3 #####
4
5 End Sub
6
7
8 Sub AzimutSonnenhoehe(pmaxStunden As Integer)
9 '
10 ' Erzeugt Azimut und Sonnenhöhe als Arrays VBA intern
11 ' Ausgabe in Grad und Bogenmaß
12 '
13
14 Dim picount As Integer
15
16 Dim pAzimutBog() As Double
17 Dim pAzimutGrad() As Double
18 Dim pSonnenHoeheBog() As Double
19 Dim pSonnenHoeheGrad() As Double
20
21 ReDim pAzimutBog(1 To pmaxStunden) As Double
22 ReDim pAzimutGrad(1 To pmaxStunden) As Double
23 ReDim pSonnenHoeheBog(1 To pmaxStunden) As Double
24 ReDim pSonnenHoeheGrad(1 To pmaxStunden) As Double
25 '
26 Dim pStundenwinkelBog As Double 'Stundenwinkel Sigma im Bogenmaß
27 Dim pStundenwinkelGrad As Double 'Stundenwinkel Sigma in Grad
28 Dim pEkliptikaleLaenge As Double
29 Dim pSonnenDeklinationBog As Double
30 Dim pWinkelabweichung As Double
31 Dim dblHw As Double 'Hilfswinkel in Bogenmaß umgerechnet
32 Dim dblZ As Double 'Zeitgleichung z in Stunden
33 Dim pdblz(1 To 8760) As Double
34
35 For picount = 1 To pmaxStunden
36     arrStunde(picount) = (picount)

```

```

37 arrTagesnummer(picount + 1) = (arrStunde(picount) \ 24)
38 arrDatum(picount) = arrTagesnummer(picount) + 1
39 '
40 intMEZ(picount) = arrStunde(picount) - 24 * (arrTagesnummer(picount))
41 '
42 dblHw = rad(360 / 365 * arrTagesnummer(picount))
43 '
44 dblZ = 0.008 * Cos(dblHw) - 0.122 * Sin(dblHw) - 0.052 * Cos(2 * dblHw)
45 - 0.157 * Sin(2 * dblHw) - 0.001 * Cos(3 * dblHw) - 0.005 * Sin(3 *
46 dblHw)
47 '
48 pdblz(picount) = dblZ * 180 / pi
49 pStundenwinkelGrad = 15 * (intMEZ(picount) + dblZ + 1 / 15 *
50 dblGeoLaengeGrad - 1)
51 '
52 pStundenwinkelBog = rad(pStundenwinkelGrad)
53 '
54 pEkliptikaleLaenge = 0.9863 * (arrTagesnummer(picount) - 2.8749) +
55 1.9137 * Sin(rad(0.9863 * (arrTagesnummer(picount) - 2.8749))) +
56 102.06
57 '
58 pSonnenDeklinationBog = (ArcSin(-0.3979 * Sin(rad(pEkliptikaleLaenge))))
59 '
60 pSonnenHoeheBog(picount) = ArcSin(Sin(pSonnenDeklinationBog) * Sin(
61 dblGeoBreiteBog) - Cos(pSonnenDeklinationBog) * Cos(dblGeoBreiteBog)
62 * Cos(pStundenwinkelBog))
63 pSonnenHoeheGrad(picount) = grad(pSonnenHoeheBog(picount))
64 '
65 '
66 'Formel zur Berechnung des Azimutwinkel Ausgabe in Grad
67 'Mit Berücksichtigung des arctan
68 pWinkelabweichung = Atn((-Cos(pSonnenDeklinationBog) * Sin(
69 pStundenwinkelBog)) / (-Cos(pSonnenDeklinationBog) * Sin(
70 dblGeoBreiteBog) * Cos(pStundenwinkelBog) - Sin(
71 pSonnenDeklinationBog) * Cos(dblGeoBreiteBog)))
72 '
73 If intMEZ(picount) > 0 And intMEZ(picount) < 12 And pWinkelabweichung >
74 0 Then
75 pAzimutBog(picount) = pi + pWinkelabweichung - pi
76 ElseIf intMEZ(picount) > 12 And intMEZ(picount) < 25 And
77 pWinkelabweichung < 0 Then
78 pAzimutBog(picount) = 2 * pi + pWinkelabweichung
79 Else
80 pAzimutBog(picount) = pi + pWinkelabweichung
81 End If
82 #####
83 'alte Berechnungsformel ohne Berücksichtigung des arctan Fehler
84 pAzimutBog(picount) = pi + Atn((-Cos(pSonnenDeklinationBog) * Sin(
85 pStundenwinkelBog)) / (-Cos(pSonnenDeklinationBog) * Sin(
86 dblGeoBreiteBog) * Cos(pStundenwinkelBog) - Sin(
87 pSonnenDeklinationBog) * Cos(dblGeoBreiteBog)))
88 #####
89 '
90 'Umrechnung von Bogenmaß in Grad
91 pAzimutGrad(picount) = grad(pAzimutBog(picount))
92 'Übergabe der internen Prozedurvariablen an das Modul

```

```

83  If gBModell = 1 Then 'in Fall 1 werden die Werte in die Arrays gepackt
84      mit denen die Hauptberechnung weiter verläuft
85      arrSonnenHoeheGrad(picount) = pSonnenHoeheGrad(picount)
86      arrSonnenHoeheBog(picount) = pSonnenHoeheBog(picount)
87      arrAzimutGrad(picount) = pAzimutGrad(picount)
88      arrAzimutBog(picount) = pAzimutBog(picount)
89
90      arrAzimutGradBMBT(picount) = pAzimutGrad(picount)           ,
91          zusätzlich die Arrays die zum Aufbau der Diagramme dienen
92      arrSonnenHoeheGradBMBT(picount) = pSonnenHoeheGrad(picount)
93
94 Else
95     arrAzimutGradBMBT(picount) = pAzimutGrad(picount)           ' im Fall
96         2 und 3 nur die BMT Arrays
97     arrSonnenHoeheGradBMBT(picount) = pSonnenHoeheGrad(picount)
98
99 End If
100 Next picount
101 '
102 ' Optionale Ausgabe der Zeitgleichung mit Parameter
103 ' Dient nur zur Kontrolle und kann False gesetzt werden
104 ' muss aber nicht im Programm mitlaufen
105 '
106
107 If pBMT = True Then
108
109     Dim pdblzT(1 To 356) As Double
110     Dim parrTagesnummer(1 To 356) As Integer
111
112     For picount = 1 To 356
113         pdblzT(picount) = pdblz(picount * 24)
114         parrTagesnummer(picount) = (picount)
115     Next
116
117     Call Kollektor.diag_1Datenreihe("ZGL nach BMT", "ZGL", "Tage", pdblzT
118         , parrTagesnummer, 15, 2915, 475, 305)
119 End If
120 End Sub
121 Sub get_ZeitArrays(pmaxStunden As Integer)
122
123     Dim picount As Integer
124     For picount = 1 To pmaxStunden
125         arrStunde(picount) = (picount)
126
127         arrTagesnummer(picount + 1) = (arrStunde(picount) \ 24)
128         arrDatum(picount) = arrTagesnummer(picount) + 1
129
130         intMEZ(picount) = arrStunde(picount) - 24 * (arrTagesnummer(picount
131             ))
132     Next
133 End Sub
134
135 Sub get_ZeitgleichungDeklination()
136
137     ' Erzeugt den Array Azimut und Sonnenhöhe nach DIN
138

```

```

139  Dim SonnenHoeheGrad(1 To 8760) As Double
140  Dim SonnenHoeheBog(1 To 8760) As Double
141  Dim azi(1 To 8760) As Double
142  Dim Zeitzone As Integer
143
144  Dim pdblZGL() As Double
145  Dim pdblZGLBog() As Double
146  Dim pdblDEK() As Double
147  Dim pdblDEKBog() As Double
148
149  Dim pdblMEZ As Double
150  Dim pdblMOZ As Double
151  Dim pdblWOZ As Double
152
153  Dim pintTag() As Double
154  Dim PintTagBog As Double
155  Dim pintTagGrad As Double
156  Dim Schaltjahr As Boolean
157  Dim pintStunden() As Integer
158  Dim pdblOmega As Double
159  Dim pdblOmegaBog As Double
160
161  Dim pintN As Integer 'wird 366 oder 365 gesetzt
162  Dim pintH As Integer 'wird 8760 oder 8784 gesetzt
163  Dim icount As Integer 'variabler counter
164
165  Schaltjahr = False
166  Zeitzone = 1
167
168  If Schaltjahr = False Then
169    ReDim pintTag(1 To 365) As Double
170    ReDim pdblZGL(1 To 365) As Double
171    ReDim pdblZGLBog(1 To 365) As Double
172    ReDim pdblDEK(1 To 365) As Double
173    ReDim pdblDEKBog(1 To 365) As Double
174    ReDim pintStunden(1 To 8760) As Integer
175    For icount = 1 To 365
176      pintTag(icount) = icount
177    Next
178    For icount = 1 To 8760
179      pintStunden(icount) = icount
180    Next
181    pintN = 365
182    pintH = 8760
183  Else
184    ReDim pintTag(1 To 366) As Double
185    ReDim pdblZGL(1 To 366) As Double
186    ReDim pdblZGLBog(1 To 366) As Double
187    ReDim pdblDEK(1 To 366) As Double
188    ReDim pdblDEKBog(1 To 366) As Double
189    ReDim pintStunden(1 To 8784) As Integer
190    For icount = 1 To 366
191      pintTag(icount) = pintTag(icount)
192    Next
193    For icount = 1 To 8784
194      pintStunden(icount) = icount
195    Next
196    pintN = 366
197    pintH = 8784
198  End If
199
```

```

200  For icount = 1 To pintN
201      pintTagGrad = 360 * pintTag(icount) / pintN
202      PintTagBog = rad(pintTagGrad)
203
204      ' pintTag(icount)=
205      pdblZGL(icount) = 0.0066 + 7.3525 * Cos(rad(pintTagGrad + 85.9)) +
206          9.9359 * Cos(rad(2 * pintTagGrad + 108.9)) + 0.3387 * Cos(rad(3 *
207          pintTagGrad + 105.2)) / 60
208      pdblZGLBog(icount) = rad(pdblZGL(icount))
209      pdblDEK(icount) = 0.3948 - 23.2559 * Cos(rad(pintTagGrad + 9.1)) -
210          0.3915 * Cos(rad(2 * pintTagGrad + 5.4)) - 0.1764 * Cos(rad(3 *
211          pintTagGrad + 26))
212      pdblDEKBog(icount) = rad(pdblDEK(icount))
213
214  Next
215
216
217  'Aufruf der Diagramme Zeitgleichung und Deklination
218  Call Kollektor.diag_1Datenreihe("Zeitgleichung", "ZGL", "Tage", pdblZGL,
219      pintTag, 10, 2910, 470, 300)
220  Call Kollektor.diag_1Datenreihe("Sonnendeklination", "DEK", "Tage",
221      pdblDEK, pintTag, 480, 2910, 470, 300)
222  Call AzimutSonnenhoehe(8760)
223  End Sub
224
225  Sub AzimutSonnenhoehe_DIN_SUNAU(Anwendung As Integer)
226      '
227      ' Anwendung:
228      ' 1 => DIN => Übergabe der Spezifische Werte (zb. arrAzimutGradDIN) an die
229      ' im Modul Deklarierten Hauptvariable(zb. arrAzimutGrad)
230      ' 2 => SUNAE => Übergabe der Spezifische Werte (zb. arrAzimutGradSUNAE) an
231      ' die im Modul Deklarierten Hauptvariable(zb. arrAzimutGrad)
232      ' 3 => DIN + SUNAU => nur an spezifische Modul Deklarationen übergeben
233      '
234  Dim pDatum() As Date
235  Dim ptag As Double
236  Dim pZeit As Double
237
238  Dim picount As Integer
239  Dim picountmax As Integer
240
241  Dim pAzi As Double
242  Dim pSunh As Double
243
244  Dim pAziDIN() As Double
245  Dim pAziSOLPOS() As Double
246  Dim pAziSUNAE() As Double
247
248  Dim pSunhDIN() As Double
249  Dim pSunhSOLPOS() As Double
250  Dim pSunhSUNAE() As Double
251
252  Dim pdiff As Double      'Differenz aus gewähltem Jahr und 1900
253  Dim pSJ As Double        'Anzahl Schaltjahre
254  Dim pNJ As Double        'Anzahl Normaljahre
255  Dim pSJ_hilf As Integer 'Schaltjahr Hilfsvariable
256  Dim pTagescount As Integer
257
258  Dim Jahreszahl As Integer
259  Dim Zeitzone As Double
260  ' Könnte man noch von dem Tabellenblatt Eingabe, Abfragen
261  ' ist aber nicht nötig
262  '

```

```

253 Jahreszahl = 1900
254 Zeitzone = 1           'Hier kann die Zeitzone verändert werden
255 '
256 'Prüfung: ob Schaltjahr
257 If (Jahreszahl Mod 4) = 0 And (Jahreszahl Mod 100) <> 0 Or _
258 ((Jahreszahl Mod 400) = 0) Then
259   'Schaltjahr
260   pSJ_hilf = 0
261   picountmax = 366 * 24
262   ReDim pDatum(1 To 366 * 24) As Date
263 Else
264   'kein Schaltjahr
265   pSJ_hilf = 1
266   picountmax = 365 * 24
267   ReDim pDatum(1 To 365 * 24) As Date
268 End If
269
270 ReDim pAziDIN(1 To picountmax) As Double
271 ReDim pAziSOLPOS(1 To picountmax) As Double
272 ReDim pAziSUNAE(1 To picountmax) As Double
273
274 ReDim pSunhDIN(1 To picountmax) As Double
275 ReDim pSunhSOLPOS(1 To picountmax) As Double
276 ReDim pSunhSUNAE(1 To picountmax) As Double
277
278 pdiff = Jahreszahl - 1900
279 pSJ = pdiff \ 4
280 pNJ = pdiff - pSJ
281 pTagescount = 1
282
283 For picount = 1 To picountmax
284   'Erzeugen des pDatum Arrays anhand des Jahres
285   'in der ganzen Zahlen steckt das Datum
286   'in den Zahlen nach dem Komma steckt die Uhrzeit
287   'Bsp.: 1,5 => wäre das Datum 01.01.1900 12:00:00 usw.
288 If pTagescount Mod 24 = 0 Then
289   pTagescount = pTagescount + 1
290 End If
291 'Erzeugung der Tage
292 ptag = (pSJ * 366) + (pNJ * 365) + pTagescount + pSJ_hilf
293
294 'Erzeugung der Zeit
295 pZeit = (picount / 24)
296 'Summe
297 pDatum(picount) = ptag + pZeit
298 gDatum(picount) = pDatum(picount)
299 'Ab hier erfolgt der Aufruf der Berechnung von DIN und SUNAE nach
299 Quaschning
300 Select Case Anwendung
301   Case 1
302     Call Sonnenposition_Quaschning.SunPositionDIN(pAzi, pSunh, pDatum(
302       picount), dblGeoBreiteGrad, dblGeoLaengeGrad, Zeitzone)
303     pAziDIN(picount) = pAzi
304     pSunhDIN(picount) = pSunh
305     pAzi = 0
306     pSunh = 0
307   Case 2
308     Call Sonnenposition_Quaschning.SunPositionSUNAE(pAzi, pSunh,
308       pDatum(picount), dblGeoBreiteGrad, dblGeoLaengeGrad, Zeitzone)
309     pAziSUNAE(picount) = pAzi
310     pSunhSUNAE(picount) = pSunh

```

```

311 pAzi = 0
312 pSunh = 0
313 Case 3
314 Call Sonnenposition_Quaschning.SunPositionDIN(pAzi, pSunh, pDatum(
315     picount), dblGeoBreiteGrad, dblGeoLaengeGrad, Zeitzone)
316 pAziDIN(picount) = pAzi
317 pSunhDIN(picount) = pSunh
318 pAzi = 0
319 pSunh = 0
320
321 Call Sonnenposition_Quaschning.SunPositionSUNAE(pAzi, pSunh, pDatum(
322     picount), dblGeoBreiteGrad, dblGeoLaengeGrad, Zeitzone)
323 pAziSUNAE(picount) = pAzi
324 pSunhSUNAE(picount) = pSunh
325 pAzi = 0
326 pSunh = 0
327 End Select
328 'Wäre der Aufruf der 4. Berechnungsvariante die aber nicht Lauffähig
329     ist.
330 'Call Sonnenposition_Quaschning.SunPositionSolPos(pAzi, pSunh, pDatum(
331     picount), dblGeoBreiteGrad, dblGeoLaengeGrad, Zeitzone)
332 'arraziSOLPOS(picount) = pAzi
333 'arrsunhSOLPOS(picount) = pSunh
334 'pAzi = 0
335 'pSunh = 0
336 Next
337 'Füllung der Arrays für die Weiterrechnung Globalstrahlung
338 Select Case Anwendung
339 Case 1
340     For picount = 1 To picountmax
341         arrAzimutGrad(picount) = pAziDIN(picount)
342         arrAzimutBog(picount) = rad(arrAzimutGrad(picount))
343         arrSonnenHoeheGrad(picount) = pSunhDIN(picount)
344         arrSonnenHoeheBog(picount) = rad(arrSonnenHoeheGrad(picount))
345     Next
346 Case 2
347     For picount = 1 To picountmax
348
349         arrAzimutGrad(picount) = pAziSUNAE(picount)
350         arrAzimutBog(picount) = rad(arrAzimutGrad(picount))
351         arrSonnenHoeheGrad(picount) = pSunhSUNAE(picount)
352         arrSonnenHoeheBog(picount) = rad(arrSonnenHoeheGrad(picount))
353     Next
354 Case 3
355     arrAzimutGradDIN = pAziDIN
356     arrSonnenHoeheGradDIN = pSunhDIN
357     arrAzimutGradSUNAE = pAziSUNAE
358     arrSonnenHoeheGradSUNAE = pSunhSUNAE
359 End Select
360
361 End Sub
362 Sub AzimutSonnenhoehe_Kontrolle(nscroll As Integer)
363
364 Call AzimutSonnenhoehe(intStundenMax)
365 Call AzimutSonnenhoehe_DIN_SUNAU(3)
366 Call Kollektor.diag_Vergleich("Azimut Vergleich", _
367             "Azimut_Bundesministerium", _
368             "Azimut DIN", _
369             "Azimut SUNAE", _
370             "Monat", _
371             arrAzimutGradBMBT, _

```

```

368 arrAzimutGradDIN, _
369 arrAzimutGradSUNAE, _
370 gDatum,
371 50, 3210, 900, 150, _
372 nscroll)
373 Call Kollektor.diag_Vergleich("Sonnenhoehe Vergleich", _
374 "Sonnenhoehe Bundesministerium", _
375 "Sonnenhoehe DIN", _
376 "Sonnenhoehe SUNAE", _
377 "Monat", _
378 arrSonnenHoeheGradBMBT, _
379 arrSonnenHoeheGradDIN, _
380 arrSonnenHoeheGradSUNAE, _
381 gDatum,
382 50, 3360, 900, 150, _
383 nscroll)
384
385 End Sub
386 Sub Globalstrahlung(KollektorWinkel, Verwendung As Integer, FallParameter,
387 Optional Egges_kum_kwh_Allgemein)
388 '##### Berechnung Globalstrahlung auf geineigte Fläche mit Aufteilung in
389 '### Direkter und Diffuser Strahlung
390 '### Erzeugung für ER und KS
391 '### je nach EingabeFall & KollektorWinkel erfolgt Ausgabe
392 '### Egges_kum_kwh_Allgemein
393 '#####
394
395
396
397 dblGeoLaengeBog = rad(dblGeoLaengeGrad)
398 dblGeoBreiteBog = rad(dblGeoBreiteGrad)
399 '
400 ' !! Fehlerprüfung !!
401 '
402 If dblGeoLaengeGrad = 0 Or dblGeoLaengeBog = 0 Then
403   MsgBox "GeoLänge wurde mit Null Berechnet oder Daten wurden nicht
404   übergeben"
405 End If
406 If dblGeoBreiteGrad = 0 Or dblGeoBreiteBog = 0 Then
407   MsgBox "Geobreite wurde mit Null Berechnet oder Daten wurden nicht
408   übergeben"
409 End If
410 intStundenMax = 8760      'Anzahl der Berechnungsstunden (1 Jahr)
411 '
412 '
413 ' Festlegung der Datenfeldgröße der Array Variablen der
414 ' Schleife für Stundenzahl Sonnen Höhe und Azimutwinkel
415 '
416 ReDim arrStunde(1 To intStundenMax) As Integer
417 ReDim arrTagesnummer(1 To intStundenMax + 1) As Integer
418 ReDim arrDatum(1 To intStundenMax + 1) As Integer
419 ReDim intMEZ(0 To intStundenMax) As Integer
420 ReDim arrSonnenHoeheGrad(1 To intStundenMax) As Double
421 ReDim arrSonnenHoeheBog(1 To intStundenMax) As Double
422 ReDim arrAzimutGrad(1 To intStundenMax) As Double
423 ReDim arrAzimutBog(1 To intStundenMax) As Double
424 ReDim E0hor(1 To intStundenMax) As Double
425 ReDim gDatum(1 To intStundenMax) As Date

```

```

426 ReDim arrAzimutGradBMBT(1 To intStundenMax) As Double
427 ReDim arrSonnenHoeheGradBMBT(1 To intStundenMax) As Double
428 ReDim arrWinkelfaktorKollektor(1 To intStundenMax) As Double
429
430 '
431 ' Schleife für Stundenzahl Sonnen Höhe und Azimutwinkel
432 '
433 Dim icount As Integer
434 '
435 ' Dim dblStundenwinkelBog As Double 'Stundenwinkel Sigma im Bogenmaß
436 ' Dim dblStundenwinkelGrad As Double 'Stundenwinkel Sigma in Grad
437 ' Dim dblEkliptikaleLaenge As Double
438 ' Dim dblSonnenDeklinationBog As Double
439 '
440 ' Aufruf des Sub für Stundenzahl Sonnen Höhe und Azimutwinkel
441 '
442 Call AzimutSonnenhoehe_init
443 '
444 ' Berechnung der Globalstrahlung
445 '
446 Dim epsilonER_GradA(1 To 8760) As Double
447 Dim epsilonKS_GradA(1 To 8760) As Double
448
449 Dim EdirhorER_A(1 To 8760) As Double
450 Dim EdiffhorER_A(1 To 8760) As Double
451
452 Dim EggesER_A(1 To 8760) As Double
453 Dim EggesKS_A(1 To 8760) As Double
454
455 Dim Erefflgen_A(1 To 8760) As Double
456
457 Dim EdirgenER_A(1 To 8760) As Double
458 Dim EdiffgenER_A(1 To 8760) As Double 'Perez Modell mit errechneten Werten
        (soll am genauesten sein und für die weitere Rechnung verwendet)
459 Dim EdiffgenIsoER_A(1 To 8760) As Double 'isotroper Ansatz mit errechneten Werten
460 Dim EdiffgenAnIsoER_A(1 To 8760) As Double 'anisotroper Ansatz mit Errechneten Werten
461
462 Dim EdirgenKS_A(1 To 8760) As Double
463 Dim EdiffgenKS_A(1 To 8760) As Double 'Perez Modell mit Klimasatz Werten (
        soll am genauesten sein und wird für die weitere Rechnung verwendet)
464 Dim EdiffgenIsoKS_A(1 To 8760) As Double 'isotroper Ansatz mit Klimasatz Werten
465 Dim EdiffgenAnIsoKS_A(1 To 8760) As Double 'anisotroper Ansatz mit Klimasatz Werten
466
467 Dim zwischenfaktor_A(1 To 8760) As Double
468 Dim pHoehenwinkelEbeneBog As Double
469
470 gAzimutEbeneBog = rad(gAzimutEbeneGrad)
471 pHoehenwinkelEbeneBog = rad(KollektorWinkel)
472
473 Dim maximum As Double
474 Dim kt As Double
475 Dim epsilonER As Double
476 Dim epsilonKS As Double
477 Dim deltaER As Double
478 Dim deltaKS As Double
479 Dim thetahor As Double
480 Dim g_epsilonER As Double

```

```

481 Dim g_epsilonKS As Double
482 Dim FER(1 To 2, 1 To 3) As Double
483 Dim FKS(1 To 2, 1 To 3) As Double
484 Dim F1ER As Double
485 Dim F1KS As Double
486 Dim F2ER As Double
487 Dim F2KS As Double
488 Dim a As Double
489 Dim B As Double
490 Dim maximumA As Double
491 Dim maximumB As Double
492 Dim ZER As Double
493 Dim ZKS As Double
494 Dim dblHilfsvariable As Double
495
496 Dim thetagenA(1 To 8760) As Double
497 Dim thetagenAGrad(1 To 8760) As Double
498 Dim y As Double
499 Dim pWinkelfaktor As Double
500 Dim pmaximum(1 To 8760) As Double
501 Dim pResult(1 To 8760) As Double
502 Dim pMittel_for_Edirgen(1 To 8760) As Double
503 Dim NewXs(1 To 8760) As Single
504 For icount = 1 To intStundenMax
505
506     thetagenA(icount) = ArcCos(-Cos(arrSonnenHoeheBog(icount)) * Sin(
507         pHoehenwinkelEbeneBog) * -
508             Cos(arrAzimutBog(icount) - gAzimutEbeneBog) + Sin(
509                 arrSonnenHoeheBog(icount)) * -
510                     Cos(pHoehenwinkelEbeneBog))
511     NewXs(icount) = icount
512     pMittel_for_Edirgen(icount) = (Cos(thetagenA(icount)) / Sin(
513         arrSonnenHoeheBog(icount)))
514     If pMittel_for_Edirgen(icount) < 0 Then pMittel_for_Edirgen(icount) = 0
515 Next
516 icount = 0
517 'Zwischenrechnung zur Spitzenunterdrückung in Abhängigkeit von
518     pMittel_for_Edirgen(icount)
519 'Lineare Regression des Arrays pmaximum um die Spizen der Direkten
520     Einstrahlung zu glätten
521     Dim Result As Variant
522     Result = Application.WorksheetFunction.Trend(pMittel_for_Edirgen,
523         , NewXs, True)
524
525 For icount = 1 To intStundenMax
526     If Result(icount) < 0 Then Result(icount) = 0
527     thetagenAGrad(icount) = grad(thetagenA(icount))
528
529     dblHilfsvariable = (Cos(thetagenA(icount)) / Sin(arrSonnenHoeheBog(
530         icount)))
531
532     If Sin(arrSonnenHoeheBog(icount)) = 0 Then
533         MsgBox ("Divison durch Null bei thetagenA " & "icount= " &
534             icount & "Sin(arrSonnenHoeheBog(icount))=" & Sin(arrSonnenHoeheBog(
535                 icount)))
536     End If
537     maximum = 0
538     If dblHilfsvariable < 0 Then maximum = 0
539     If dblHilfsvariable > 0 Then maximum = dblHilfsvariable
540     If dblHilfsvariable > (Result(icount) + 2) Then
541         'Der Faktor zwei ist ein abgeschätzter Wert

```

```

534     maximum = Result(icount)
535 End If
536     'pmaximum(icount) wird nur zur Dartstellung im Diagramm erzeugt
537     'und beinhaltet die verwendeten maximum faktoren zur Edirgen bewertung
538     pmaximum(icount) = maximum
539
540 ,
541 ' Berechnung des Winkelkoeffizienten für die Einwirkung
542 ' der Reflexion der Direkt Strahlung auf die geneigte
543 ' Fläche
544 ,
545
546 pWinkelkoeffizient = Winkelkoeffizient(thetaGenA(icount), icount)
547 ,
548
549 ' Diff.- Dir. Strahlungs Aufteilung
550 ' deren Wetterdaten nur aus Globalstrahlung besteht
551 ' wenn kt kleiner 0 wird, wird Ediffhor größer Eghor
552 ' dies wird in der Funktion Ediffhor berücksichtigt
553 ,
554 If EghorKS(icount, 1) < 0 Then                                'And arrSonnenHoeheBog(icount)
555     > 0.017
556     E0hor(icount) = 1367 * Sin(arrSonnenHoeheBog(icount))
557     kt = EghorKS(icount, 1) / E0hor(icount)
558
559     'Normaler Klarheitsindex KT
560     dblHilfsvariable = EghorKS(icount, 1)
561     EdiffhorER_A(icount) = Ediffhor(kt, dblHilfsvariable,
562                                     arrSonnenHoeheBog(icount))
563
564     'Isotroper Ansatz nach Perez für Strahlung durch Bodenreflexion
565     Erefflgen_A(icount) = EghorKS(icount, 1) * gAlbedo * (1 - Cos(
566         pHoehenwinkelEbeneBog)) / 2
567
568     thetahor = pi / 2 - arrSonnenHoeheBog(icount)
569     'Um Direktstrahlung Positiv zu halten wird ab einem winkel von 1°
570     ' Edirgen, hor nicht berechnet
571     If arrSonnenHoeheBog(icount) > 0.017 Then      '0,014rad sind 1°
572     ,
573     ' Direktstrahlung auf horizontale Fläche
574     ,
575     EdirhorER_A(icount) = EghorKS(icount, 1) - EdiffhorER_A(icount)
576     'Direkte horizontale Strahlung
577
578     ,
579     ' Direktstrahlung auf geneigte Fläche
580     ,
581     EdirgenER_A(icount) = EdirhorER_A(icount) * maximum          ,
582     ' Direkte geneigte Strahlung ER
583     EdirgenKS_A(icount) = EdirhorKS(icount, 1) * maximum          ,
584     ' Direkte geneigte Strahlung KS
585 End If
586     epsilonER = (((EdiffhorER_A(icount) + EdirhorER_A(icount)) * (1 / Sin(
587         arrSonnenHoeheBog(icount)))) /_
588             EdiffhorER_A(icount)) + 1.041 * (thetahor ^ 3)) / (1 +
589             1.041 * (thetahor ^ 3))
590
591     ' Auswertung nach EingabeFall der hier als FallParameter bezeichnet
592
593
594     Select Case FallParameter
595     Case 1, 5

```

```

586      ' 1: bekannter Standort (Wien, Wien hohe Warte,...) und Glob-, Diff-
587      - , Direktstrahlung gegeben
588      ' 5: Vergleichsstandort , globalstrahlung bekannt , diff- bekannt ,
589      Dir- bekannt
590
591      ' Für den Sonderfall wenn in Abend und Morgenstunden nur die Glob-
592      - . auf horizont.Fl. gegeben ist .
593      ' If EdirhorKS(icount , 1) = 0 And EghorKS(icount , 1) <> 0 Or
594      EdirhorKS(icount , 1) = EghorKS(icount , 1) Then
595      '     EdiffhorKS(icount , 1) = EdiffhorER(icount)
596      '     EdirhorKS(icount , 1) = EghorKS(icount , 1) - EdiffhorKS(
597      icount , 1)
598      ' End If
599      If EdiffhorKS(icount , 1) = 0 Then
600          epsilonKS = 0
601      Else
602          epsilonKS = ((( EdiffhorKS(icount , 1) + EdirhorKS(icount , 1)
603          * (1 / Sin(arrSonnenHoeheBog(icount)))) /
604          EdiffhorKS(icount , 1)) + 1.041 * (thetahor ^ 3))
605          / (1 + 1.041 * (thetahor ^ 3))
606      End If
607      Case 2
608          ' 2: Vergleichsstandort , Globalstrahlung bekannt , Diff- &
609          Direktstrahlung nicht gegeben
610          ' In diesem Fall wurde keine direkte und diffuse Strahlung im
611          Klimadatensatz
612          ' eingegeben, darum wird epsilonKS 0 um eine Division durch
613          Null zu verhindern .
614          epsilonKS = 0
615          ' Strahlung für ER (Errechnete Werte)
616          If gStrahlungsVariationsAusgabe <> "ER" Then
617              MsgBox "Falscher Eingabeparameter der Diagrammausgabe
618                  Strahlungsvariation , für Eingabefall 2 ist nur die
619                  Eingabe ER zulässig . Die Berechnung wird unterbrochen !"
620              ' Da sich die KS Datensätze auf Edir,hor und Ediff,hor
621              beziehen würden , welche
622              ' in diesem Eingabefall nicht vorhanden sind .
623              Error (4)
624              Exit Sub
625          End If
626          Case 3
627              ' 3: Vergleichsstandort , globalstrahlung bekannt , diff- Fehlt ,
628              Dir- bekannt
629
630              ' If EdirhorKS(icount , 1) = 0 And EghorKS(icount , 1) <> 0 Then
631              ' Für den Sonderfall wenn in Abend und Morgenstunden nur die
632              Glob-. auf horizont.Fl. gegeben ist .
633
634              ' EdiffhorKS(icount , 1) = EdiffhorER_A(icount)
635              ' EdirhorKS(icount , 1) = EghorKS(icount , 1) - EdiffhorKS(
636              icount , 1)
637              ' End If
638              If EdiffhorKS(icount , 1) = 0 Then
639                  epsilonKS = 0
640              Else
641                  epsilonKS = ((( EdiffhorKS(icount , 1) + EdirhorKS(icount , 1)
642                  * (1 / Sin(arrSonnenHoeheBog(icount)))) /
643                  EdiffhorKS(icount , 1)) + 1.041 * (thetahor ^ 3)) /
644                  (1 + 1.041 * (thetahor ^ 3))
645              End If
646          Case 4

```

```

629      ' 4: Vergleichsstandort , Globalstrahlung bekannt , diff – bekannt ,
630      ' dir – fehlt
631      ' Falls die diffuse Strahlung 0 ist aber die Globale nicht dann
632      ' liegt ein
633      ' Fehler/Falschmessung des Klimadatensatzes vor der hier
634      ' ausgebessert wird
635      If EdiffhorKS(icount , 1) = 0 And EghorKS(icount , 1) <> 0 Then
636          EdiffhorKS(icount , 1) = EghorKS(icount , 1)
637      End If
638      epsilonKS = (((EdiffhorKS(icount , 1) + EdirhorKS(icount , 1) *
639          (1 / Sin(arrSonnenHoeheBog(icount)))) / _
640          EdiffhorKS(icount , 1)) + 1.041 * (thetahor ^ 3)) /
641          (1 + 1.041 * (thetahor ^ 3))
642
643      Case 6
644          ' 6: Vergleichsstandort , diff – bekannt , Dir – bekannt
645          epsilonKS = (((EdiffhorKS(icount , 1) + EdirhorKS(icount , 1) *
646              (1 / Sin(arrSonnenHoeheBog(icount)))) / _
647              EdiffhorKS(icount , 1)) + 1.041 * (thetahor ^ 3)) /
648              (1 + 1.041 * (thetahor ^ 3))
649      End Select
650
651      g_epsilonER = grad(epsilonER)
652      g_epsilonKS = grad(epsilonKS)
653
654      epsilonER_GradA(icount) = g_epsilonER
655      epsilonKS_GradA(icount) = g_epsilonKS
656
657      Call FParameterPerez(g_epsilonER , FER)
658      Call FParameterPerez(g_epsilonKS , FKS)
659
660      deltaER = (EdiffhorER_A(icount) / 1367) * (1 / Sin(arrSonnenHoeheBog(
661          icount)))
662      deltaKS = (EdiffhorKS(icount , 1) / 1367) * (1 / Sin(arrSonnenHoeheBog(
663          icount)))
664      ' Horizontalhelligkeitsindex
665      F1ER = FER(1 , 1) + FER(1 , 2) * deltaER + FER(1 , 3) * thetahor
666      F1KS = FKS(1 , 1) + FKS(1 , 2) * deltaKS + FKS(1 , 3) * thetahor
667      ' Sonnenumgebungshelligkeitsindex
668      F2ER = FER(2 , 1) + FER(2 , 2) * deltaER + FER(2 , 3) * thetahor
669      F2KS = FKS(2 , 1) + FKS(2 , 2) * deltaKS + FKS(2 , 3) * thetahor
670
671      If 0 > (Cos(thetagenA(icount))) Then maximumA = 0
672      If 0 < (Cos(thetagenA(icount))) Then maximumA = (Cos(thetagenA(icount
673          )))
674      a = maximumA
675
676      If 0.087 > (Sin(arrSonnenHoeheBog(icount))) Then maximumB = 0.087
677      If 0.087 < (Sin(arrSonnenHoeheBog(icount))) Then maximumB = (Sin(
678          arrSonnenHoeheBog(icount)))
679      B = maximumB
680
681      ' Einfluss des Winkelfaktors auf die dierekt geneigte Strahlung
682
683      If gWinkelfaktorFALL = "ja" Then           ' Berechnung erfolgt mit
684          ' Einfluss des Winkelfaktors
685          EdirgenER_A(icount) = EdirgenER_A(icount) * pWinkelfaktor
686          EdirgenKS_A(icount) = EdirgenKS_A(icount) * pWinkelfaktor

```

```

677 ElseIf gWinkelfaktorFALL = "nein" Then      'Berechnung erfolgt ohne
678     ' Einfluss des Winkelfaktors
679     EdirgenER_A(icount) = EdirgenER_A(icount)
680     EdirgenKS_A(icount) = EdirgenKS_A(icount)
681 Else
682     MsgBox "Falscher Eingabeparameter der Winkelfaktor
683         Berücksichtigung, es ist nur ja oder nein zulässig. Die
684         Berechnung wird unterbrochen!"
685     Error (4)          'wird Ausgelöst um Programm abzubrechen
686     Exit Sub
687 End If
688 '
689 ' Berechnung der Strahlungsmodelle auf geneigte Fläche der ER
690 ' Datensätze
691 '
692 'ER... Errechnet
693 ' isotroper Ansatz
694 EdiffgenIsoER_A(icount) = EdiffhorER_A(icount) * (1 / 2) * (1 + Cos(
695     pHoehenwinkelEbeneBog))
696 '
697 ' anisotroper Ansatz nach Klucher
698 ZER = 1 - (EdiffhorER_A(icount) / EghorKS(icount, 1)) ^ 2
699 EdiffgenAnIsoER_A(icount) = EdiffhorER_A(icount) * (1 / 2) * (1 + Cos(
700     (pHoehenwinkelEbeneBog)) *
701     (1 + ZER * ((Sin(pHoehenwinkelEbeneBog / 2)) ^ 3)) * (1 + ZER * ((Cos(
702     (thetagenA(icount))) ^ 2) *
703     ((Cos(arrSonnenHoeheBog(icount))) ^ 3)))
704 '
705 'Ansatz nach Perez <= derzeit am genauesten
706 If arrSonnenHoeheBog(icount) < 0.017 Then
707     EdiffgenER_A(icount) = (EdiffgenIsoER_A(icount) +
708     EdiffgenAnIsoER_A(icount)) / 2
709 Else
710     EdiffgenER_A(icount) = EdiffhorER_A(icount) * (((1 + Cos(
711         pHoehenwinkelEbeneBog)) *
712         (1 - F1ER) / 2) + a / B * F1ER + F2ER * Sin(pHoehenwinkelEbeneBog))
713     If EdiffgenER_A(icount) < 0 Then
714         EdiffgenER_A(icount) = (EdiffgenIsoER_A(icount) +
715         EdiffgenAnIsoER_A(icount)) / 2
716     End If
717 End If
718 '
719 'Gesamtstrahlung auf geneigte Fläche
720 If EdirgenER_A(icount) < 0 Then EdirgenER_A(icount) = 0
721 EggesER_A(icount) = EdiffgenER_A(icount) + Erefftgen_A(icount) +
722     EdirgenER_A(icount)
723 If EggesER_A(icount) < 0 Or EggesER_A(icount) > 1700 Then
724     If mdebug_showDiag = True Then
725         Debug.Print "Kontrolle EggesER_A" & icount
726     End If
727 End If
728 '
729 ' Berechnung der Strahlungsmodelle auf geneigte Fläche der KS
730 ' Datensätze
731 '
732 'KS... Klimasatz
733 ' isotroper Ansatz
734 EdiffgenIsoKS_A(icount) = EdiffhorKS(icount, 1) * (1 / 2) * (1 + Cos(
735     (pHoehenwinkelEbeneBog)))
736 ' anisotroper Ansatz nach Klucher

```

```

724 ZKS = 1 - (EdiffhorKS(icount, 1) / EghorKS(icount, 1)) ^ 2
725 EdiffgenAnIsoKS_A(icount) = EdiffhorKS(icount, 1) * (1 / 2) * (1 +
726   Cos(pHoehenwinkelEbeneBog)) * _  

727   (1 + ZKS * ((Sin(pHoehenwinkelEbeneBog / 2)) ^ 3)) * (1 + ZKS * ((Cos(
728     (thetagenA(icount))) ^ 2) *  

729     ((Cos(arrSonnenHoeheBog(icount))) ^ 3)))
730   'Ansatz nach Perez <= derzeit am genauesten
731   If arrSonnenHoeheBog(icount) < 0.017 Then
732     EdiffgenKS_A(icount) = (EdiffgenIsoKS_A(icount) +
733       EdiffgenAnIsoKS_A(icount)) / 2
734   Else
735     EdiffgenKS_A(icount) = EdiffhorKS(icount, 1) * (((1 + Cos(
736       pHoehenwinkelEbeneBog)) * (1 - F1KS) / 2) _  

737       + a / B * F1KS + F2KS * Sin(pHoehenwinkelEbeneBog))
738   If EdiffgenKS_A(icount) < 0 Then
739     EdiffgenKS_A(icount) = (EdiffgenIsoKS_A(icount) +
740       EdiffgenAnIsoKS_A(icount)) / 2
741   End If
742 End If
743
744 'Gesamtstrahlung auf geneigte Fläche
745 If FallParameter = 2 Then 'EingabeFall 2=> Vergleichsstandort, nur
746   Globalstrahlung bekannt
747   EggesKS_A(icount) = 0
748 Else
749   If EdirgenKS_A(icount) < 0 Then EdirgenKS_A(icount) = 0
750   EggesKS_A(icount) = EdiffgenKS_A(icount) + Erefflgen_A(icount) +
751     EdirgenKS_A(icount)
752 End If
753 If EggesKS_A(icount) < 0 Or EggesKS_A(icount) > 1700 Then
754   If mdebug_showDiag = True Then
755     Debug.Print "Kontrolle EggesKS_A" & icount
756   End If
757 End If
758 End If 'Schleifen Ende von: 'If EghorKS(icount, 1) < 0 Then
759 Next
760 'Call Kollektor.Diag_Test("Vergleich Winkel", "thetagenAGrad", "
761   arrSonnenHoeheBog", thetagenAGrad, arrSonnenHoeheBog, 10, 3510, 940,
762   300)
763
764 If mdebug_showDiag = True Then
765   Call Kollektor.Diag_Test("Vergleich maximum", "pmaximum", "Result",
766     pmaximum, Result, 10, 3810, 940, 300)
767 End If
768 #####
769 ##### Für die Fallunterscheidung und Aufteilung
770 #####
771 Select Case Verwendung
772 Case 1
773   'Übergabe der berechneten Daten von Prozedurebene an die in
774     Modulebene deklarierten Public Variablen
775   '
776   ' Redimensionieren der Deklarationen
777   '
778   ReDim thetagen(1 To intStundenMax) As Double
779   ReDim thetagenGrad(1 To intStundenMax) As Double
780   ReDim epsilonER_Grad(1 To intStundenMax) As Double
781   ReDim epsilonKS_Grad(1 To intStundenMax) As Double
782
783   ReDim EdirhorER(1 To intStundenMax) As Double

```

```

774 ReDim EdiffhorER(1 To intStundenMax) As Double
775
776 ReDim EggesER(1 To intStundenMax) As Double
777 ReDim EggesKS(1 To intStundenMax) As Double
778
779 ReDim Erefflgen(1 To intStundenMax) As Double
780
781 ReDim EdirgenER(1 To intStundenMax) As Double
782 ReDim EdiffgenER(1 To intStundenMax) As Double 'Perez Modell mir
783   Errechneten Werten (soll am genauesten sein und _
784   für die weitere
785   Rechnung
786   verwendet werden
787 ReDim EdiffgenIsoER(1 To intStundenMax) As Double 'isotroper ansatz
788   mit Errechneten Werten
789 ReDim EdiffgenAnIsoER(1 To intStundenMax) As Double 'anisotroper
790   ansatz mit Errechneten Werten
791
792 ReDim EdirgenKS(1 To intStundenMax) As Double
793 ReDim EdiffgenKS(1 To intStundenMax) As Double 'Perez Modell mir
794   Klimasatz Werten (soll am genauesten sein und _
795   für die weitere
796   Rechnung
797   verwendet werden
798 ReDim EdiffgenIsoKS(1 To intStundenMax) As Double 'isotroper ansatz
799   mit Klimasatz Werten
800 ReDim EdiffgenAnIsoKS(1 To intStundenMax) As Double 'anisotroper
801   ansatz mit Klimasatz Werten
802
803 ReDim zwischenfaktor(1 To intStundenMax) As Double
804
805 thetagen = thetagenA
806 thetagenGrad = thetagenAGrad
807
808 epsilonER_Grad = epsilonER_GradA
809 epsilonKS_Grad = epsilonKS_GradA
810
811 EdirhorER = EdirhorER_A
812 EdiffhorER = EdiffhorER_A
813
814 EdiffgenER = EdiffgenER_A
815 EdiffgenIsoER = EdiffgenIsoER_A
816 EdiffgenAnIsoER = EdiffgenAnIsoER_A
817
818 EdiffgenKS = EdiffgenKS_A
819 EdiffgenIsoKS = EdiffgenIsoKS_A
820 EdiffgenAnIsoKS = EdiffgenAnIsoKS_A
821
822 If mdebug_showDiag = True Then
823   Call Diag_Test("Strahlungsvergleich der Datensätze ER & KS auf
824     geneigten Kollektor", "EggesER", "EggesKS", EggesER, EggesKS,
825     10, 3510, 940, 300)
826 End If

```

```

823 zwischenfaktor = zwischenfaktor_A
824
825 Case 2
826     ' Vorbereiten der Daten für das Diagramm Strahlungsvariation
827     Dim EggesKumER(1 To 12) As Double
828     Dim EggesKumKS(1 To 12) As Double
829     Dim EggesKumERkWh(1 To 12) As Double
830     Dim EggesKumKSskWh(1 To 12) As Double
831
832     ' Für die Berechnung mit einem Kollektorwinkel von 0 Grad wird die
833     ' Strahlung auf die
834     ' horizontale Fläche herangezogen
835     If KollektorWinkel = 0 Then
836         For icount = 1 To intStundenMax
837             EggesKS_A(icount) = EghorKS(icount, 1)
838             EggesER_A(icount) = EghorKS(icount, 1)
839         Next
840     End If
841
842     Call ArrayKumulieren(EggesKS_A, EggesKumKS)
843     Call ArrayKumulieren(EggesER_A, EggesKumER)
844
845     If gStrahlungsVariationsAusgabe = "KS" Then
846         Call E_to_kWH(EggesKumKS, Egges_kum_kwh_Allgemein)
847     ElseIf gStrahlungsVariationsAusgabe = "ER" Then
848         Call E_to_kWH(EggesKumER, Egges_kum_kwh_Allgemein)
849     Else
850         MsgBox "Die Eingabe Diagrammausgabe Strahlungsvariation ist
851         Fehlerhaft" &
852         "Bitte geben Sie ER oder KS ein."
853         Error (4)
854     End If
855
856 End Select
857 End Sub
858
859 Function Wirkungsgrad(ByVal micount As Integer, ByVal TSpRL As Double,
860 , ByVal TSpVL As Double) As Double
861
862     ' Berechnung des Wirkungsgrades mit der Eingangsvariable der
863     ' mittleren Fluidtemperatur 'aus dem Modul Speicher.
864
865     Dim dblT_Koll As Double
866     Dim pSonneninstrahlung As Double      ' Deklaration zur übernahme der zu
867     ' berechneten Strahlung, je nach Datensatz ER/KS
868     dblT_Koll = (TSpVL + TSpRL + 10) / 2
869
870     ' Einlesen Sonneninstrahlung je nach Datensatz ER oder KS
871     If gStrahlungsVariationsAusgabe = "KS" Then
872         pSonneninstrahlung = Kollektor.EggesKS(micount)
873     ElseIf gStrahlungsVariationsAusgabe = "ER" Then
874         pSonneninstrahlung = Kollektor.EggesER(micount)
875     End If
876     If pSonneninstrahlung <= 10 Then arrWirkungsgradKollektor(micount) = 0
877     If pSonneninstrahlung > 10 Then arrWirkungsgradKollektor(micount) = (
878         gC0 - gC1 * ((dblT_Koll - dblT_AussenKS(micount, 1)) -
879         / pSonneninstrahlung) - gC2 * ((dblT_Koll - dblT_AussenKS(micount, 1))
880             ^ 2 / pSonneninstrahlung))
881
882     If arrWirkungsgradKollektor(micount) < 0 Or arrWirkungsgradKollektor(
883         micount) > 1 Then
884         arrWirkungsgradKollektor(micount) = 0

```

```

877     End If
878     Wirkungsgrad = arrWirkungsgradKollektor(micount)
879 End Function
880
881 Function Winkelfaktor(ByRef pthetagen As Double, picount As Integer)
882 '
883 ' Berechnet den Winkelfaktor
884 ' Wird vom Sub Globalstrahlung initialisiert,
885 ' der ReDim Befehl der deklaration arrWinkelfaktorKollektor(picount)
886 ' sowie die weitere Verwendung findet ebenfalls dort statt.
887 '
888
889
890 Dim dblEinfallwinkelBog As Double
891 Dim pzwischenfaktor As Double
892 dblEinfallwinkelBog = rad(gEinfallwinkelGrad)
893
894
895 If gEinfallwinkelGrad = 50 Then
896     ' Zwischenfaktor ist der Therm erster Ordnung b0, der Quadratische Therm
897     ' b1 wird vernachlässigt
898     pzwischenfaktor = (1 - (1 - gWinkelfaktor50)) / ((1 / (Cos(
899         dblEinfallwinkelBog))) - 1)
900     arrWinkelfaktorKollektor(picount) = 1 - pzwischenfaktor * ((1 / Cos(
901         pthetagen)) - 1)
902     ' Ausbesserung der Werte unter 1% und über 100% auf null
903     If arrWinkelfaktorKollektor(picount) < 0 Or arrWinkelfaktorKollektor(
904         picount) > 1 Then
905         arrWinkelfaktorKollektor(picount) = 0
906     End If
907 Else
908     Call CriticalMsgbox("Der Einfallwinkel ist in bezug auf den Winkelfaktor
909         auszuwählen. Es wird momentan nicht mit 50° gerechnet", "
910         Eingabekontrolle")
911     ' pzwischenfaktor ist der Therm erster Ordnung b0, der Quadratische
912     ' Therm b1 wird vernachlässigt
913     pzwischenfaktor = (1 - (1 - gWinkelfaktor50)) / ((1 / (Cos(
914         dblEinfallwinkelBog))) - 1)
915     arrWinkelfaktorKollektor(picount) = 1 - pzwischenfaktor * ((1 / Cos(
916         pthetagen)) - 1)
917     If arrWinkelfaktorKollektor(picount) < 0 Or arrWinkelfaktorKollektor(
917         picount) > 1 Then
918         arrWinkelfaktorKollektor(picount) = 0
919     End If
920 End If
921 Winkelfaktor = arrWinkelfaktorKollektor(picount)
922
923
924 End Function 'Winkelfaktor
925
926 Public Function Ediffhor(ByRef kt As Double, ByRef Eghor As Double, ByRef
927     SonnenHoeheBog As Double)
928     'Normaler Klarheitsindex KT
929     If kt <= 0 Then Ediffhor = Eghor
930     If kt > 0 And kt <= 0.3 Then Ediffhor = Eghor * (1.02 - 0.254 * kt +
931         0.0123 * Sin(SonnenHoeheBog))
932     If kt > 0.3 And kt < 0.78 Then Ediffhor = Eghor * (1.4 - 1.749 * kt +
933         0.177 * Sin(SonnenHoeheBog))
934     If kt >= 0.78 Then Ediffhor = Eghor * (0.486 - 0.182 * Sin(
935         SonnenHoeheBog))
936     If Ediffhor > Eghor Then Ediffhor = Eghor

```

923 End Function

Listing E.4: Modul Kollektor Kapitel 5

## E.5. 6. Kapitel Visualisierende Programmsequenzen

```
1 Sub Kapitel_6()
2 '#####
3 '## 6. Visualisierende Programmsequenzen
4 '#####
5 End Sub
6 Sub diag_1Datenreihe(strTitel As String, strDiagDatenreihe1 As String,
7     strXAchseTitel As String, _
8         array1, arrayX, _
9             x As Integer, y As Integer, _
10            B As Integer, H As Integer)
11 Dim strDiagObjektName As String
12
13 Dim strDiagWorksheet As String
14
15 'Namen der Datenreihe für Legende
16 strDiagObjektName = strTitel
17
18 strDiagWorksheet = strDiagramme
19 'Übergabe der Arrays
20 Dim varx As Variant
21
22 'Erzeugen des Diagramms
23
24 On Error GoTo ErrorHandler
25     Application.ScreenUpdating = False
26
27
28     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName)
29         .Chart
30             .SeriesCollection(1).Values = array1
31             .Legend.Delete
32             .Legend.Refresh
33             .Refresh
34     End With
35
36     Application.ScreenUpdating = True
37     Exit Sub
38
39 ErrorHandler:
40     '' Falls Diagramm nicht existiert ''
41     If Err.Number = -2147024809 Then
42         Worksheets(strDiagWorksheet).ChartObjects.Add(x, y, B, H).name =
43             strDiagObjektName
44         'Worksheets(strDiagWorksheet).ChartObjects.Add(10, 810, 940, 500).
45             name = strDiagObjektName
46     End If
47
48     '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
49     If Err.Number = 1004 Then
50         For Each varx In Worksheets(strDiagWorksheet).ChartObjects.Item(
51             strDiagObjektName).Chart.SeriesCollection
52             varx.Delete
53         Next varx
54     End If
```

```

51
52
53    '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
54 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
55     Chart
56         .HasTitle = True
57         .ChartTitle.text = strTitel
58         .ChartType = xlXYScatterSmoothNoMarkers
59         .HasLegend = True
60         .Legend.IncludeInLayout = False
61         .Legend.Position = xlCorner
62         .Axes(xlValue).TickLabels.Orientation = xlUpward
63         .Axes(xlCategory, xlPrimary).HasTitle = True
64         .Axes(xlCategory, xlPrimary).AxisTitle.Characters.text =
65             strX AchseTitel
66         .Axes(xlCategory).MinimumScale = LBound(arrayX)
67         .Axes(xlCategory).MaximumScale = UBound(arrayX)
68     With .Axes(xlValue)
69         .HasTitle = True
70         .AxisTitle.text = strDiagDatenreihe1
71     End With
72 End With
73
74 varx = 1
75 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
76     Chart.SeriesCollection
77     With .NewSeries
78         .name = strDiagDatenreihe1
79         .Values = array1
80         .XValues = arrayX
81         .Border.ColorIndex = 3
82
83     End With
84 End With
85 'Debug.Print "Sub: diag_1Datenreihe" & Err.Number, Err.Description
86 End Sub
87 Sub diag_2Datenreihe(strTitel As String, strDiagDatenreihe1 As String,
88                     strDiagDatenreihe2 As String, strX AchseTitel As String, _
89                     array1, array2, arrayX, _
90                     x As Integer, y As Integer, _
91                     B As Integer, H As Integer)
92
93 Dim strDiagObjektName As String
94
95 Dim strDiagWorksheet As String
96
97 'Namen der Datenreihe für Legende
98 strDiagObjektName = strTitel
99
100 strDiagWorksheet = strDiagramme
101 'Übergabe der Arrays
102 Dim varx As Variant
103
104 'Erzeugen des Diagramms
105
106 On Error GoTo ErrorHandler
107 Application.ScreenUpdating = False
108
109 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
110     Chart

```

```

107     . SeriesCollection(1). Values = array1
108     . SeriesCollection(2). Values = array2
109     . Legend .Delete
110     . Legend .Refresh
111     . Refresh
112 End With
113
114 Application . ScreenUpdating = True
115 Exit Sub
116
117 ErrorHandler:
118   '' Falls Diagramm nicht existiert ''
119 If Err .Number = -2147024809 Then
120   Worksheets(strDiagWorksheet) . ChartObjects .Add(x, y, B, H) .name =
121     strDiagObjektName
122   ' Worksheets(strDiagWorksheet) . ChartObjects .Add(10, 810, 940, 500) .
123     name = strDiagObjektName
124 End If
125
126   '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
127 If Err .Number = 1004 Then
128   For Each varx In Worksheets(strDiagWorksheet) . ChartObjects .Item(
129     strDiagObjektName) .Chart .SeriesCollection
130     varx .Delete
131   Next varx
132 End If
133
134   '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
135 With Worksheets(strDiagWorksheet) . ChartObjects .Item(strDiagObjektName) .
136   Chart
137     . HasTitle = True
138     . ChartTitle .text = strTitel
139     . ChartType = xlXYScatterSmoothNoMarkers
140     . HasLegend = True
141     . Legend .IncludeInLayout = False
142     . Legend .Position = xlCorner
143     . Axes(xlValue) .TickLabels .Orientation = xlUpward
144     . Axes(xlCategory, xlPrimary) .HasTitle = True
145     . Axes(xlCategory, xlPrimary) .AxisTitle .Characters .text =
146       strXAchseTitel
147     . Axes(xlCategory) .MinimumScale = LBound(arrayX)
148     . Axes(xlCategory) .MaximumScale = UBound(arrayX)
149
150   ' Achsentitel der Y-Achse
151   With .Axes(xlValue)
152     . HasTitle = True
153     . AxisTitle .text = strDiagDatenreihe1
154   End With
155 End With
156
157 varx = 1
158 With Worksheets(strDiagWorksheet) . ChartObjects .Item(strDiagObjektName) .
159   Chart .SeriesCollection
160   With .NewSeries
161     .name = strDiagDatenreihe1
162     .Values = array1
163     .XValues = arrayX
164     .Border .ColorIndex = 3
165
166   End With

```

```

162 End With
163 varx = varx + 1
164 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
    Chart.SeriesCollection
165     With .NewSeries
166         .name = strDiagDatenreihe2
167         .Values = array2
168         .XValues = arrayX
169         .Border.ColorIndex = 7
170
171     End With
172 End With
173 'Debug.Print "diag_2Datenreihe" & Err.Number, Err.Description
174 End Sub
175
176 Sub diag_3Datenreihe(strTitel As String, strDiagDatenreihe1 As String,
    strDiagDatenreihe2 As String, strDiagDatenreihe3 As String,
    strXAchseTitel As String,
    array1, array2, array3, arrayX, _
    x As Integer, y As Integer, _
    B As Integer, H As Integer)
177
178
179
180
181     Dim strDiagObjektName As String
182
183     Dim strDiagWorksheet As String
184
185     'Namen der Datenreihe für Legende
186     strDiagObjektName = strTitel
187
188     strDiagWorksheet = strDiagramme
189     'Übergabe der Arrays
190     Dim varx As Variant
191
192     'Erzeugen des Diagramms
193
194     On Error GoTo ErrorHandler
195     Application.ScreenUpdating = False
196
197
198     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
        Chart
199         .SeriesCollection(1).Values = array1
200         .SeriesCollection(2).Values = array2
201         .SeriesCollection(3).Values = array3
202         .Legend.Delete
203         .Legend.Refresh
204         .Refresh
205     End With
206
207     Application.ScreenUpdating = True
208     Exit Sub
209
210     ErrorHandler:
211     'Falls Diagramm nicht existiert '
212     If Err.Number = -2147024809 Then
213         Worksheets(strDiagWorksheet).ChartObjects.Add(x, y, B, H).name =
            strDiagObjektName
214     End If
215
216     'Falls Diagramm zwar existiert aber eine Datenreihe fehlt '
217     If Err.Number = 1004 Then

```

```

218     For Each varx In Worksheets(strDiagWorksheet).ChartObjects.Item(
219         strDiagObjektName).Chart.SeriesCollection
220         varx.Delete
221     Next varx
222 End If
223
224     '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
225     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
226         Chart
227             .HasTitle = True
228             .ChartTitle.text = strTitel
229             .ChartType = xlXYScatterSmoothNoMarkers
230             .HasLegend = True
231             .Legend.IncludeInLayout = False
232             .Legend.Position = xlCorner
233             .Axes(xlValue).TickLabels.Orientation = xlUpward
234             .Axes(xlCategory, xlPrimary).HasTitle = True
235             .Axes(xlCategory, xlPrimary).AxisTitle.Characters.text =
236                 strX AchseTitel
237             .Axes(xlCategory).MinimumScale = LBound(arrayX)
238             .Axes(xlCategory).MaximumScale = UBound(arrayX)
239             With .Axes(xlValue)
240                 .HasTitle = True
241                 .AxisTitle.text = strDiagDatenreihe1
242             End With
243         End With
244
245     varx = 1
246     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
247         Chart.SeriesCollection
248         With .NewSeries
249             .name = strDiagDatenreihe1
250             .Values = array1
251             .XValues = arrayX
252             .Border.ColorIndex = 3
253         End With
254     End With
255     varx = varx + 1
256     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
257         Chart.SeriesCollection
258         With .NewSeries
259             .name = strDiagDatenreihe2
260             .Values = array2
261             .XValues = arrayX
262             .Border.ColorIndex = 4
263         End With
264     End With
265     varx = varx + 1
266     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
267         Chart.SeriesCollection
268         With .NewSeries
269             .name = strDiagDatenreihe3
270             .Values = array3
271             .XValues = arrayX
272             .Border.ColorIndex = 33
273         End With
274     End With
275
276     'Debug.Print "diag_3Datenreihe" & Err.Number, Err.Description
277 End Sub
278 Sub diag_3Datenreihe_yScaliert(strTitel As String, strDiagDatenreihe1 As
279     String, strDiagDatenreihe2 As String, strDiagDatenreihe3 As String,

```

```

    strXAchseTitel As String, _
272        array1, array2, array3, arrayX, _
273        x As Integer, y As Integer, _
274        B As Integer, H As Integer, max_Y_Achsen_Skalierung
275        As Double)

276 Dim strDiagObjektName As String
277
278 Dim strDiagWorksheet As String
279
280 'Namen der Datenreihe für Legende
281 strDiagObjektName = strTitel
282
283 strDiagWorksheet = strDiagramme
284 'Übergabe der Arrays
285 Dim varx As Variant
286
287 'Erzeugen des Diagramms
288
289 On Error GoTo ErrorHandler
290 Application.ScreenUpdating = False
291
292 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
293     Chart
294         .SeriesCollection(1).Values = array1
295         .SeriesCollection(2).Values = array2
296         .SeriesCollection(3).Values = array3
297         .Legend.Delete
298         .Legend.Refresh
299         .Refresh
300
301 End With
302 Application.ScreenUpdating = True
303 Exit Sub
304
305 ErrorHandler:
306     '' Falls Diagramm nicht existiert ''
307     If Err.Number = -2147024809 Then
308         Worksheets(strDiagWorksheet).ChartObjects.Add(x, y, B, H).name =
309             strDiagObjektName
310         'Worksheets(strDiagWorksheet).ChartObjects.Add(10, 810, 940, 500).
311         name = strDiagObjektName
312     End If
313
314     '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
315     If Err.Number = 1004 Then
316         For Each varx In Worksheets(strDiagWorksheet).ChartObjects.Item(
317             strDiagObjektName).Chart.SeriesCollection
318             varx.Delete
319         Next varx
320     End If
321
322     '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
323     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
324         Chart
325             .HasTitle = True
326             .ChartTitle.text = strTitel
327             .ChartType = xlXYScatterSmoothNoMarkers
328             .HasLegend = True
329             .Legend.IncludeInLayout = False
330             .Legend.Position = xlCorner

```

```

326     .Axes(xlValue).TickLabels.Orientation = xlUpward
327     .Axes(xlCategory, xlPrimary).HasTitle = True
328     .Axes(xlCategory, xlPrimary).AxisTitle.Characters.text =
329         strX AchseTitel
330     .Axes(xlCategory).MinimumScale = LBound(arrayX)
331     .Axes(xlCategory).MaximumScale = UBound(arrayX)
332     .Axes(xlValue).MaximumScale = max_Y_Achsen_Skalierung
333     With .Axes(xlValue)
334         .HasTitle = True
335         .AxisTitle.text = strDiagDatenreihe1
336     End With
337 End With
338
339 varx = 1
340 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
341     Chart.SeriesCollection
342     With .NewSeries
343         .name = strDiagDatenreihe1
344         .Values = array1
345         .XValues = arrayX
346         .Border.ColorIndex = 3
347     End With
348 End With
349 varx = varx + 1
350 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
351     Chart.SeriesCollection
352     With .NewSeries
353         .name = strDiagDatenreihe2
354         .Values = array2
355         .XValues = arrayX
356         .Border.ColorIndex = 4
357     End With
358 End With
359 varx = varx + 1
360 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
361     Chart.SeriesCollection
362     With .NewSeries
363         .name = strDiagDatenreihe3
364         .Values = array3
365         .XValues = arrayX
366         .Border.ColorIndex = 33
367     End With
368 End With
369 'Debug.Print "diag_3Datenreihe_yScaliert" & Err.Number, Err.Description
370 End Sub
371 Public Sub Diag_Test(strTitel As String, strDiagDatenreihe1 As String,
372     strDiagDatenreihe2 As String, _
373         array1, array2, _
374             x_pos As Integer, y_pos As Integer, _
375                 B As Integer, H As Integer)
376     'mit AxisGroup 2 kann eine zweite Achse ander Datenreihen geschrieben
377     'werden
378     Dim strDiagObjektName As String
379     Dim strDiagWorksheet As String
380
381     'Namen der Datenreihe für Legende
382     strDiagObjektName = strTitel
383
384     strDiagWorksheet = strDiagramme

```

```

380   ' Erzeugt oder bzw. aktualisiert falls vorhanden, das Diagramm
381   ' DiagEnergieAufteilung
382   '' Ergebnisse des aktuellen Projektes einlesen
383   'Dim mE_Sonneneinstrahlung As Variant, mE_Sonneneinstrahlung_kum As
384   ' Variant
385   Dim i As Integer, j As Variant, x As Variant
386   'mE_Sonneneinstrahlung = colErgebnis.Item(strProjekt).arrSonne
387   'mE_Sonneneinstrahlung_kum = colErgebnis.Item(strProjekt).
388   arrMonatEkumEinstrahlungGes
389 On Error GoTo ErrorHandler
390 Application.ScreenUpdating = False
391
392 If Worksheets(strDiagramme).ChartObjects.Item(strDiagObjektName).Chart.
393   SeriesCollection.Count <> 1 Then
394   Error 1004
395 End If
396 With Worksheets(strDiagramme).ChartObjects.Item(strDiagObjektName).
397   Chart
398   .SeriesCollection(1).Values = array1
399   .SeriesCollection(2).Values = array2
400 End With
401 Application.ScreenUpdating = True
402 Exit Sub
403
404 ErrorHandler:
405
406 Select Case Err.Number
407   '' Falls Diagramm nicht existiert ''
408   Case -2147024809
409     Worksheets(strDiagramme).ChartObjects.Add(x_pos, y_pos, B, H).
410     name = strDiagObjektName
411
412   '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
413   Case 1004, 7
414     For Each x In Worksheets(strDiagramme).ChartObjects.Item(
415       strDiagObjektName).Chart.SeriesCollection
416       x.Delete
417     Next x
418   End Select
419 Select Case Err.Number
420   Case -2147024809, 1004, 7
421     With Worksheets(strDiagramme).ChartObjects.Item(strDiagObjektName)
422       With .Chart
423         .ChartType = xlLine
424         With .SeriesCollection
425           With .NewSeries
426             .name = strDiagDatenreihe1
427             .Values = array1
428           End With
429           With .NewSeries
430             .name = strDiagDatenreihe2
431             .Values = array2
432             .AxisGroup = 2
433           End With
434         End With
435       End With

```

```

434     . SetElement ( msoElementSecondaryCategoryAxisShow )
435   End With
436 End With
437
438 Case Else
439   Debug. Print "DiagSonneneinstrahlung", Err. Number, Err. Description
440 End Select
441
442 Application. ScreenUpdating = True
443
444 End Sub
445
446
447 Sub diag_Vergleich(strTitel As String, strDiagDatenreihe1 As String,
448   strDiagDatenreihe2 As String, strDiagDatenreihe3 As String,
449   strXAchseTitel As String, _
450   array1, array2, array3, arrayX, _
451   x As Integer, y As Integer, _
452   B As Integer, H As Integer, nscroll As Integer)
453 '#####
454 '# Kann drei Datenreihen vergleichen, allerdings wird
455 '# die Achse mit dem Scrollbalken verschoben
456 '#####
457
458 Dim strDiagObjektName As String
459
460 Dim strDiagWorksheet As String
461
462 'Namen der Datenreihe für Legende
463 strDiagObjektName = strTitel
464
465 strDiagWorksheet = strDiagramme
466 'Übergabe der Arrays
467 Dim varx As Variant
468
469 'Erzeugen des Diagramms
470
471 On Error GoTo ErrorHandler
472 Application. ScreenUpdating = False
473
474 With Worksheets(strDiagWorksheet). ChartObjects. Item(strDiagObjektName) .
475   Chart
476     . SeriesCollection(1). Values = array1
477     . SeriesCollection(2). Values = array2
478     . SeriesCollection(3). Values = array3
479     . Legend. Delete
480     . Legend. Refresh
481     . Refresh
482
483 End With
484 Application. ScreenUpdating = True
485 Exit Sub
486
487 ErrorHandler:
488   ' Falls Diagramm nicht existiert '
489   If Err. Number = -2147024809 Then
490     Worksheets(strDiagWorksheet). ChartObjects. Add(x, y, B, H). name =
491       strDiagObjektName
492     'Worksheets(strDiagWorksheet). ChartObjects.Add(10, 810, 940, 500).
493     name = strDiagObjektName
494   End If
495

```

```

490   '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
491   If Err.Number = 1004 Then
492     For Each varx In Worksheets(strDiagWorksheet).ChartObjects.Item(
493       strDiagObjektName).Chart.SeriesCollection
494       varx.Delete
495     Next varx
496   End If
497   '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
498   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
499     Chart
500       .HasTitle = True
501       .ChartTitle.text = strTitel
502       .ChartType = xlXYScatterSmoothNoMarkers
503       .HasLegend = True
504       .Legend.IncludeInLayout = False
505       .Legend.Position = xlCorner
506       .Axes(xlValue).TickLabels.Orientation = xlUpward
507       .Axes(xlCategory, xlPrimary).HasTitle = True
508       .Axes(xlCategory, xlPrimary).AxisTitle.Characters.text =
509         strX AchseTitel
510       .Axes(xlCategory).MinimumScale = nscroll - 720
511       .Axes(xlCategory).MaximumScale = nscroll
512     With .Axes(xlValue)
513       .HasTitle = True
514       .AxisTitle.text = strDiagDatenreihe1
515     End With
516   End With
517   varx = 1
518   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
519     Chart.SeriesCollection
520     With .NewSeries
521       .name = strDiagDatenreihe1
522       .Values = array1
523       .XValues = arrayX
524     End With
525   End With
526   varx = varx + 1
527   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
528     Chart.SeriesCollection
529     With .NewSeries
530       .name = strDiagDatenreihe2
531       .Values = array2
532       .XValues = arrayX
533       .Border.ColorIndex = 3
534     End With
535   End With
536   varx = varx + 1
537   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
538     Chart.SeriesCollection
539     With .NewSeries
540       .name = strDiagDatenreihe3
541       .Values = array3
542       .XValues = arrayX
543       .Border.ColorIndex = 43
544     End With
545   End With
546   'Debug.Print "diag_Vergleich" & Err.Number, Err.Description
547 End Sub
548 Sub Diag_ZOOM_Scroll_Balken()

```

```

545 ##### Erstellt falls nicht vorhanden einen Scrollbalken
546 ##### und erzeugt anschließend mit Call Diag_Sonnenstand_Chart(.value),
547 ##### das dazugehörige Diagramm. Wobei ".value" der Rückgabewert
548 ##### des Scrollbalkens ist.
549 ##### Ausgabe findet in strprojekt_Diagramme statt
550 ##### update, bzw. erzeugung des Sonnenstand-Diagramm mit Scrollbar
551 #####
552
553 ' update, bzw. erzeugung des Sonnenstand-Diagramm mit Scrollbar
554
555 On Error GoTo ErrorHandler
556
557 Normal:
558     With Worksheets(strDiagramme).Shapes.Range(Array(strProjekt & "2
559         TageZoom")).Item(strProjekt & "2TageZoom").OLEFormat.Object
560             .max = 8760
561             Call AzimutSonnenhoehe_Kontrolle(.value)
562     End With
563     Exit Sub
564
565 ErrorHandler:
566
567     If Err.Number = 1004 Then
568         With Worksheets(strDiagramme).ScrollBars.Add(10, 3210, 40, 300)
569             .value = 720
570             .Min = 1
571             .max = 8760
572             .SmallChange = 720
573             .LargeChange = 720
574             .LinkedCell = ""
575             .Display3DShading = True
576             .name = strProjekt & "2TageZoom"
577             .OnAction = "Diag_ZOOM_Scroll_Balken"
578     End With
579     GoTo Normal
580
581     Else
582         Debug.Print "Sonnenstand Vergleich: " & Err.Number, Err.Description
583     End If
584
585     End Sub
586
587 Sub diag_DiffusVergleich()
588 ##### Es werden zwei Diagramme erzeugt, einmal für KS und einmal für ER,
589 ##### wobei die drei DiffusenStrahlungs Ansätze auf die geneigte Fläche
590 ##### verglichen werden sollen
591
592 Dim picount As Integer
593 Dim pXArray(1 To 12) As Integer
594 Dim pmaxYScal(1 To 6) As Double
595 Dim pmax As Double
596
597 Dim pEdiffgenIsoER_kum_KWH(1 To 12) As Double
598 Dim pEdiffgenAnIsoER_kum_KWH(1 To 12) As Double
599 Dim pEdiffgenIsoKS_kum_KWH(1 To 12) As Double
600 Dim pEdiffgenAnIsoKS_kum_KWH(1 To 12) As Double
601
602 Dim pEggenIsoER(1 To 12) As Double
603 Dim pEggenAnIsoER(1 To 12) As Double
604 Dim pEggenIsoKS(1 To 12) As Double
605 Dim pEggenAnIsoKS(1 To 12) As Double

```

```

605 'kumuliert und rechnet in kwh um
606 Call kum_kwh(EdiffgenIsoER , pEdiffgenIsoER_kum_KWH)
607 Call kum_kwh(EdiffgenAnIsoER , pEdiffgenAnIsoER_kum_KWH)
608 Call kum_kwh(EdiffgenIsoKS , pEdiffgenIsoKS_kum_KWH)
609 Call kum_kwh(EdiffgenAnIsoKS , pEdiffgenAnIsoKS_kum_KWH)
610
611 For picount = 1 To 12
612   'Array für A Achse
613   pXArray(picount) = picount
614   'ER
615   pEggenIsoER(picount) = Erefflgen_kum_kWh(picount) + EdirgenER_Kum_kWh(
616     picount) + pEdiffgenIsoER_kum_KWH(picount)
617   pEggenAnIsoER(picount) = Erefflgen_kum_kWh(picount) + EdirgenER_Kum_kWh(
618     picount) + pEdiffgenAnIsoER_kum_KWH(picount)
619   'KS
620   pEggenIsoKS(picount) = Erefflgen_kum_kWh(picount) + EdirgenKS_Kum_kWh(
621     picount) + pEdiffgenIsoKS_kum_KWH(picount)
622   pEggenAnIsoKS(picount) = Erefflgen_kum_kWh(picount) + EdirgenKS_Kum_kWh(
623     picount) + pEdiffgenAnIsoKS_kum_KWH(picount)
624
625 Next
626 'generiert den Maximalen Ausschlag für .Axes(xlValue).MaximumScale abfrage
627   zum besseren vergleich
628 pmaxYScal(1) = BasicFunctions.MaxValOfdblArray(pEggenIsoER)
629 pmaxYScal(2) = BasicFunctions.MaxValOfdblArray(pEggenAnIsoER)
630 pmaxYScal(3) = BasicFunctions.MaxValOfdblArray(EggesER_Kum_kWh)
631 pmaxYScal(4) = BasicFunctions.MaxValOfdblArray(EggesKS_Kum_kWh)
632 pmaxYScal(5) = BasicFunctions.MaxValOfdblArray(pEggenIsoKS)
633 pmaxYScal(6) = BasicFunctions.MaxValOfdblArray(pEggenAnIsoKS)
634
635 pmax = Round(BasicFunctions.MaxValOfdblArray(pmaxYScal) + 20, 0)
636
637 Call Kollektor.diag_3Datenreihe_yScaliert("Vergleich der Globalstrahlung
638   auf geneigte Fläche mit drei diffusen Ansätze (ER)", _
639   "EggenER Perez Modell", "EggenER isotropes
640   Model", "EggenER anisotropes Modell",
641   "Monat", EggesER_Kum_kWh, pEggenIsoER,
642   pEggenAnIsoER, pXArray, _
643   10, 1310, 470, 300, pmax)
644 Call Kollektor.diag_3Datenreihe_yScaliert("Vergleich der Globalstrahlung
645   auf geneigte Fläche mit drei diffusen Ansätze(KS)", _
646   "EggenKS Perez Modell", "EggenKS isotropes
647   Model", "EggenKS anisotropes Modell",
648   "Monat", EggesKS_Kum_kWh, pEggenIsoKS,
649   pEggenAnIsoKS, pXArray, _
650   480, 1310, 470, 300, pmax)
651
652 End Sub
653
654 Sub diag_Klimadaten(Fall As Integer)
655   '#####
656   '## Erzeugt ein Diagramme welches die Eingabe vom
657   '## Klimadatenkatalog kumuliert und in kwh umrechnet.
658   '#####
659
660 Dim picount As Integer
661 Dim pXArray(1 To 12) As Integer
662
663 Dim pEghor_kum_KWH(1 To 12) As Double
664 Dim pEdiffhor_kum_KWH(1 To 12) As Double
665 Dim pEdirhor_kum_KWH(1 To 12) As Double

```

```

653 'kumuliert und rechnet in kwh um
654 If EingabeFall = 2 And Fall = 2 Then
655   'kommt zur Anwendung wenn Eingabefall 2 vorliegt (Horizontale
656   'Globalstrahlung bekannt, Diff- & Direktstrahlung leer)
657   Call kum_kwh(EghorKS, pEghor_kum_KWH)
658   Call kum_kwh(EdiffhorER, pEdiffhor_kum_KWH)
659   Call kum_kwh(EdirhorER, pEdirhor_kum_KWH)
660 Else
661   Call kum_kwh(EghorKS, pEghor_kum_KWH)
662   Call kum_kwh(EdiffhorKS, pEdiffhor_kum_KWH)
663   Call kum_kwh(EdirhorKS, pEdirhor_kum_KWH)
664 End If
665
666 For picount = 1 To 12
667   'Array für A Achse
668   pXArray(picount) = picount
669 Next
670 Select Case Fall
671
672 Case 1
673   Call Kollektor.diag_3Datenreihe("Original Werte aus Klimadaten", _
674     "Globalstrahlung", "Diffusstrahlung", "_
675     Direktstrahlung", _
676     "Monat", pEghor_kum_KWH,
677     pEdiffhor_kum_KWH, pEdirhor_kum_KWH
678     , pXArray, _
679     10, 1010, 470, 300)
680
681 Case 2
682   Call Kollektor.diag_3Datenreihe("Ausgebesserte Werte aus Klimadaten", _
683     "Globalstrahlung", "Diffusstrahlung", "_
684     Direktstrahlung", _
685     "Monat", pEghor_kum_KWH,
686     pEdiffhor_kum_KWH, pEdirhor_kum_KWH
687     , pXArray, _
688     480, 1010, 470, 300)
689
690 Case 3           'muss auskommentiert werden in get_Klimadaten
691   Call Kollektor.diag_3Datenreihe("Original Klimadaten Werte ohne
692     Einheitenberichtigung", _
693     "Globalstrahlung", "Diffusstrahlung", "_
694     Direktstrahlung", _
695     "Monat", pEghor_kum_KWH,
696     pEdiffhor_kum_KWH, pEdirhor_kum_KWH
697     , pXArray, _
698     950, 1010, 470, 300)
699
700 Sub Saeulendiagramm_Aufteilung(arrWerte_fuer_GanzeSaeule, _
701   arrWerte_fuer_gestapelteSaeule1, _
702   arrWerte_fuer_gestapelteSaeule2, _
703   arrWerte_fuer_gestapelteSaeule3, _
704   GanzeSaeule, _
705   VerbundeneSaeule1, _
706   VerbundeneSaeule2, _
707   VerbundeneSaeule3)
708   #####
709   '### Erstellt aus den Arrays: arrWerte_fuer_GanzeSaeule,
710   '###                                         arrWerte_fuer_gestapelteSaeule1,
711   '###                                         arrWerte_fuer_gestapelteSaeule2,
712   '###                                         arrWerte_fuer_gestapelteSaeule3,

```

```

702 ##### die für die Clustered Stacked Column Chart benötigten ,
703 ##### umsortierten Arrays .
704 ##### Rückgabe Arrays: GanzeSaeule ,
705 ##### VerbundeneSaeule1 ,
706 ##### VerbundeneSaeule2 ;
707 ##### Eingabe Arrays dürfen nur die Länge 12 haben (für ein Jahr)
708 ##### Ausgabe Arrays erhalten die Länge 36
709 ##########
710
711 Dim i As Integer
712 Dim m As Integer
713 Dim k As Integer
714
715 'Ganze Säule
716 m = 1
717 k = 1
718 For i = 1 To 36
719   If i = k Then
720     GanzeSaeule(i) = arrWerte_fuer_GanzeSaeule(m)
721     m = m + 1
722     k = k + 3
723   End If
724 Next
725 'Verbundene Säule
726 m = 1
727 k = 2
728 For i = 1 To 36
729   If i = k Then
730     VerbundeneSaeule1(i) = arrWerte_fuer_gestapelteSaeule1(m)
731     VerbundeneSaeule2(i) = arrWerte_fuer_gestapelteSaeule2(m)
732     VerbundeneSaeule3(i) = arrWerte_fuer_gestapelteSaeule3(m)
733     m = m + 1
734     k = k + 3
735   End If
736 Next
737 End Sub
738
739 Sub Diag_Clustered_Stacked_Column_Chart()
740 ##########
741 ##### Erstellt ein Clustered_Stacked_Column_Chart
742 #####(Gruppiertes und Gestapeltes Säulendiagramm)
743 #####
744
745 'arrA ,arrB ,arrC sind Arrays die eine andere Aufteilung der Eingegebenen
    zurückgeben
746 Dim arrA(1 To 36) As Variant
747 Dim arrB(1 To 36) As Variant
748 Dim arrC(1 To 36) As Variant
749 Dim arrD(1 To 36) As Variant
750
751
752 'Erefflgen_kum_kWh
753 Dim strDiagObjektName As String
754 Dim strDiagDatenreihe1 As String
755 Dim strDiagDatenreihe2 As String
756 Dim strDiagDatenreihe3 As String
757 Dim strDiagDatenreihe4 As String
758 Dim strDiagWorksheet As String
759 Dim strDiagTitel As String
760
761 'Namendes Objektes der Datenreihe für Legende

```

```

762 strDiagObjektName = "Clustered"
763 strDiagWorksheet = strDiagramme
764
765 Select Case EingabeFall
766 Case 1, 3, 4, 5 ' bekannter Standort (Wien, Wien hohe Warte,...) und Glob
    - , Diff-, Direktstrahlung gegeben
767     Call Saeulendiagramm_Aufteilung(EggesKS_Kum_kWh, EdiffgenKS_Kum_kWh,
        EdirgenKS_Kum_kWh, Erefftgen_kum_kWh, -
        arrA, arrB, arrC, arrD)
768     strDiagDatenreihe1 = "EggesKS"
769     strDiagDatenreihe2 = "EdiffgKS"
770     strDiagDatenreihe3 = "EdirgKS"
771     strDiagDatenreihe4 = "EreffgKS"
772     strDiagTitel = "Aufteilung der Direkten und Diffusen Strahlung (KS)"
773 Case 2 ' Vergleichsstandort , nur Globalstrahlung bekannt
774     Call Saeulendiagramm_Aufteilung(EggesER_Kum_kWh, EdiffgenER_Kum_kWh,
        EdirgenER_Kum_kWh, Erefftgen_kum_kWh, -
        arrA, arrB, arrC, arrD)
775     strDiagDatenreihe1 = "EggesER"
776     strDiagDatenreihe2 = "EdiffgER"
777     strDiagDatenreihe3 = "EdirgER"
778     strDiagDatenreihe4 = "EreffgER"
779     strDiagTitel = "Aufteilung der Direkten und Diffusen Strahlung (ER)"
780 End Select
781
782
783 'Übergabe der Arrays
784 Dim x As Variant
785 Dim array1 As Variant
786 Dim array2 As Variant
787 Dim array3 As Variant
788 Dim array4 As Variant
789 array1 = arrA
790 array2 = arrB
791 array3 = arrC
792 array4 = arrD
793
794 ' Erzeugung der X-Achse
795 Dim arrayX(1 To 36) As Variant
796
797
798 Dim i As Integer
799 Dim m As Integer
800 Dim k As Integer
801 k = 1
802 m = 2
803 For i = 1 To 36
804     If i = k Then
805         arrayX(i) = "Egges"
806         k = k + 3
807     ElseIf i = m Then
808         arrayX(i) = "Summe"
809         m = m + 3
810     End If
811 Next
812 'Erzeugen des Diagramms
813 On Error GoTo ErrorHandler
814     Application.ScreenUpdating = False
815     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName)
816         .Chart
817         .ChartTitle.text = strDiagTitel
818         .SeriesCollection(1).Values = array1
819         .SeriesCollection(1).name = strDiagDatenreihe1

```

```

819     . SeriesCollection(2).Values = array2
820     . SeriesCollection(2).name = strDiagDatenreihe2
821     . SeriesCollection(3).Values = array3
822     . SeriesCollection(3).name = strDiagDatenreihe3
823     . SeriesCollection(4).Values = array4
824     . SeriesCollection(4).name = strDiagDatenreihe4
825   End With
826   Application.ScreenUpdating = True
827   Exit Sub
828
829 ErrorHandler:
830   '' Falls Diagramm nicht existiert ''
831   If Err.Number = -2147024809 Then
832     Worksheets(strDiagWorksheet).ChartObjects.Add(480, 2110, 470, 300).
833     name = strDiagObjektName
834   End If
835   '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
836   If Err.Number = 1004 Then
837     For Each x In Worksheets(strDiagWorksheet).ChartObjects.Item(
838       strDiagObjektName).Chart.SeriesCollection
839       x.Delete
840     Next x
841   End If
842   '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
843   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
844     Chart
845       .ChartGroups(1).GapWidth = 0
846       .HasTitle = True
847       .ChartTitle.text = strDiagTitel
848       .ChartType = xlColumnStacked
849       .HasLegend = True
850       .Legend.Position = xlBottom
851       .Axes(xlValue).TickLabels.Orientation = xlUpward
852       With .Axes(xlValue)
853         .HasTitle = True
854         .AxisTitle.text = "Kumulierte monatl. Strahlung [kWh]"
855       End With
856     End With
857   x = 1
858   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
859     Chart.SeriesCollection
860     With .NewSeries
861       .name = strDiagDatenreihe1
862       .Values = array1
863       .XValues = arrayX
864     End With
865   End With
866   x = x + 1
867   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
868     Chart.SeriesCollection
869     With .NewSeries
870       .name = strDiagDatenreihe2
871       .Values = array2
872       .XValues = arrayX
873     End With
874   End With
875   x = x + 1
876   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
877     Chart.SeriesCollection
878     With .NewSeries
879       .name = strDiagDatenreihe3

```

```

874     .Values = array3
875     .XValues = arrayX
876   End With
877 End With
878 x = x + 1
879 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
880   Chart.SeriesCollection
881   With .NewSeries
882     .name = strDiagDatenreihe4
883     .Values = array4
884     .XValues = arrayX
885   End With
886 End With
887 End Sub
888 Sub Diag_Sonnenstand_Scroll_Chart()
889 '#####
890 '## Erstellt falls nicht vorhanden einen Scrollbalken
891 '## und erzeugt anschließend mit Call Diag_Sonnenstand_Chart(.value),
892 '## das dazugehörige Diagramm. Wobei ".value" der Rückgabewert
893 '## des Scrollbalkens ist.
894 '## Ausgabe findet in strDiagramme statt
895 '#####
896 ' update , bzw. erzeugung des Sonnenstanddiagrammes mit scrollbar
897
898 On Error GoTo ErrorHandler
899
900 Normal:
901 Application.ScreenUpdating = False
902   With Worksheets(strDiagramme).Shapes.Range(Array(strProjekt &
903     "TagesScrollbar")).Item(strProjekt & "TagesScrollbar").OLEFormat.
904     Object
905       .max = 365
906       Call Diag_Sonnenstand_Chart(.value)
907     End With
908
909 Application.ScreenUpdating = True
910 Exit Sub
911 ErrorHandler:
912
913 If Err.Number = 1004 Then
914   With Worksheets(strDiagramme).ScrollBars.Add(10, 2410, 40, 500)
915     .value = 1
916     .Min = 1
917     .max = 365
918     .SmallChange = 1
919     .LargeChange = 10
920     .LinkedCell = ""
921     .Display3DShading = True
922     .name = strProjekt & "TagesScrollbar"
923     .OnAction = "Diag_Sonnenstand_Scroll_Chart"
924   End With
925   GoTo Normal
926 Else
927   Debug.Print "Diag_Sonnenstand_Scroll_Chart: ", Err.Number, Err.
928     Description
929 End If
930 End Sub
931 Sub Diag_Sonnenstand_Chart(nscroll)
932 '#####

```

```

931 ##### Erstellt ein Sonnenstandsdiagramm Typ:
932 ##### Punkte mit interpolierter Linie
933 ##### Eingabeparameter vom Scrollbutton nScroll
934 ##### Application.ScreenUpdating = False
935 Application.ScreenUpdating = False
936 'Datenauswahl durch Scrollbutton-Parameter
937 Dim i As Integer
938 Dim m As Integer
939 Dim n As Integer
940 Dim k As Integer
941 Dim strDatumsangabe As Variant
942 'für nScroll Werte
943 Dim arrAzimut24(1 To 24) As Double
944 Dim arrSonnenHoehe24(1 To 24) As Double
945 'für Frühling
946 Dim arrAzimut24Fr(1 To 24) As Double
947 Dim arrSonnenHoehe24Fr(1 To 24) As Double
948 'für Sommer
949 Dim arrAzimut24So(1 To 24) As Double
950 Dim arrSonnenHoehe24So(1 To 24) As Double
951 'für Herbst
952 Dim arrAzimut24He(1 To 24) As Double
953 Dim arrSonnenHoehe24He(1 To 24) As Double
954 'für Winter
955 Dim arrAzimut24Wi(1 To 24) As Double
956 Dim arrSonnenHoehe24Wi(1 To 24) As Double
957
958 m = 1
959
960 Call arrDatum_to_strDatum(nscroll, strDatumsangabe)
961
962 ' Schleife erzeugt aus den Arrays
963 'arrAzimutGrad(i) und arrSonnenHoeheGrad
964 'ein Array mit der Länge von 24,
965 'dh.: 24 stündlichen Tageswerte in Abhängigkeit von nScroll.
966
967 For i = 1 To 8760
968     If i > (nscroll * 24) - 24 And i < (nscroll * 24) Or i = (nscroll * 24)
969         Then
970             arrAzimut24(m) = arrAzimutGrad(i)
971             arrSonnenHoehe24(m) = arrSonnenHoeheGrad(i)
972             m = m + 1
973     End If
974 Next
975
976 For n = 1 To 24
977     arrAzimut24Fr(n) = arrAzimutGrad(1921 + n - 1)
978     arrSonnenHoehe24Fr(n) = arrSonnenHoeheGrad(1921 + n - 1)
979 Next
980 For n = 1 To 24
981     arrAzimut24So(n) = arrAzimutGrad(4129 + n - 1)
982     arrSonnenHoehe24So(n) = arrSonnenHoeheGrad(4129 + n - 1)
983 Next
984 For n = 1 To 24
985     arrAzimut24He(n) = arrAzimutGrad(6361 + n - 1)
986     arrSonnenHoehe24He(n) = arrSonnenHoeheGrad(6361 + n - 1)
987 Next
988 For n = 1 To 24
989     arrAzimut24Wi(n) = arrAzimutGrad(8521 + n - 1)
990     arrSonnenHoehe24Wi(n) = arrSonnenHoeheGrad(8521 + n - 1)
991 Next

```

```

991
992
993
994 'arrA ,arrB ,arrC sind Arrays die eine andere Aufteilung der Eingegebenen
995     zurückgeben
996 Dim arrA(1 To 36) As Variant
997 Dim arrB(1 To 36) As Variant
998 Dim arrC(1 To 36) As Variant
999
1000 'die ersten drei Arrays sind die , die im Diagramm stehen sollen
1001 'Call Saeulendiagramm_Aufteilung(EggesKS_Kum_kWh, EdiffgenKS_Kum_kWh,
1002     EdirgenKS_Kum_kWh, arrA, arrB, arrC)
1003
1004 Dim strDiagObjektName As String
1005 Dim strDiagDatehreihe1 As String
1006 Dim strDiagDatenreihe2 As String
1007 Dim strDiagWorksheet As String
1008
1009 'Namen der Datenreihe für Legende
1010 strDiagObjektName = "Sonnenstand"
1011 strDiagDatehreihe1 = "Azimut"
1012 strDiagDatenreihe2 = "Sonnenhöhe"
1013
1014 strDiagWorksheet = strDiagramme
1015 'Übergabe der Arrays
1016 Dim x As Variant
1017 Dim array1 As Variant
1018 Dim array2 As Variant
1019 Dim array3 As Variant
1020
1021 array1 = arrAzimut24
1022 array2 = arrSonnenHoehe24
1023
1024 ' Erzeugung der X–Achse
1025 Dim arrayX(1 To 36) As Variant
1026 Dim icount As Integer
1027
1028 For icount = 1 To 24
1029     arrayX(icount) = icount
1030
1031 Next
1032 Application.ScreenUpdating = True
1033 'Erzeugen des Diagramms
1034
1035 On Error GoTo ErrorHandler
1036
1037 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1038     Chart
1039     .ChartTitle.text = "Sonnenstand am " & strDatumsangabe
1040     With .SeriesCollection(1)
1041         .Values = array2
1042         .XValues = array1
1043     End With
1044 End With
1045
1046 Application.ScreenUpdating = True
1047 Exit Sub
1048

```

```

1049 ErrorHandler:
1050   '' Falls Diagramm nicht existiert ''
1051   If Err.Number = -2147024809 Then
1052     Worksheets(strDiagWorksheet).ChartObjects.Add(50, 2410, 900, 500).
1053     name = strDiagObjektName
1054   End If
1055
1056   '' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
1057   If Err.Number = 1004 Then
1058     For Each x In Worksheets(strDiagWorksheet).ChartObjects.Item(
1059       strDiagObjektName).Chart.SeriesCollection
1060       x.Delete
1061     Next x
1062   End If
1063
1064   '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
1065   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1066     Chart
1067       'Diagrammtitel
1068       .HasTitle = True
1069       .ChartTitle.text = "Sonnenstand am " & strDatumsangabe
1070       'Achsentitel der Y-Achse
1071       .Axes(xlValue).HasTitle = True
1072       .Axes(xlValue).AxisTitle.text = "Sonnenhöhe in [°]"
1073       'Achsentitel der X-Achse
1074       .Axes(xlCategory).HasTitle = True
1075       .Axes(xlCategory).AxisTitle.text = "Azimut in [°]"
1076       'Chart Type
1077       .ChartType = xlXYScatterSmooth
1078       'Legende
1079       .HasLegend = False
1080       'Hauptachsen Formatierung
1081       .Axes(xlValue).HasMajorGridlines = True
1082       .Axes(xlValue).MajorGridlines.Border.Color = RGB(127, 127, 127)
1083       .Axes(xlValue).MajorGridlines.Border.LineStyle = xlDash
1084       .Axes(xlCategory).HasMajorGridlines = True
1085       .Axes(xlCategory).MajorGridlines.Border.Color = RGB(204, 204, 204)
1086       .Axes(xlCategory).MajorGridlines.Border.LineStyle = xlDash
1087       .Axes(xlValue).TickLabels.Orientation = xlUpward
1088       .Axes(xlCategory).TickLabels.Orientation = xlUpward
1089       .Axes(xlCategory).MinimumScale = 45
1090       .Axes(xlCategory).MaximumScale = 315
1091       .Axes(xlCategory).MajorUnit = 45
1092       .Axes(xlValue).MinimumScale = 0
1093       .Axes(xlValue).MaximumScale = 90
1094       .Axes(xlValue).MajorUnit = 10
1095     End With
1096     Dim ptag As Variant
1097     x = 1
1098     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1099       Chart.SeriesCollection
1100       'Tag vom Scrollbutton
1101       With .NewSeries
1102         .name = strDiagDatereihe1
1103         .Values = array2
1104         .XValues = array1
1105         .Border.ColorIndex = 45
1106         .MarkerStyle = xlMarkerStyleCircle
1107         .MarkerBackgroundColorIndex = 6
1108         .MarkerForegroundColorIndex = 44
1109       For n = 1 To 24

```

```

1106     If n < 10 Then
1107         ptag = "0" & n & ":00"
1108     Else
1109         ptag = n & ":00"
1110     End If
1111     . Points(n) . ApplyDataLabels Type:=xlDataLabelsShowLabel ,
1112         AutoText:=True
1113     . Points(n) . DataLabel . text = ptag
1114 Next
1115 End With
1116 'Frühling
1117 With . NewSeries
1118     . name = strDiagDatereihe1
1119     . Values = arrSonnenHoehe24Fr
1120     . XValues = arrAzimut24Fr
1121     . Border . ColorIndex = 16
1122     . MarkerStyle = xlMarkerStyleCircle
1123     . MarkerBackgroundColorIndex = 15
1124     . MarkerForegroundColorIndex = 48
1125 End With
1126 'Sommer
1127 With . NewSeries
1128     . name = strDiagDatereihe1
1129     . Values = arrSonnenHoehe24So
1130     . XValues = arrAzimut24So
1131     . Border . ColorIndex = 16
1132     . MarkerStyle = xlMarkerStyleCircle
1133     . MarkerBackgroundColorIndex = 15
1134     . MarkerForegroundColorIndex = 48
1135 For n = 1 To 24
1136     If n < 10 Then
1137         ptag = "0" & n & ":00"
1138     Else
1139         ptag = n & ":00"
1140     End If
1141     . Points(n) . ApplyDataLabels Type:=xlDataLabelsShowLabel ,
1142         AutoText:=True
1143     . Points(n) . DataLabel . text = ptag
1144 Next
1145 End With
1146 'Herbst
1147 With . NewSeries
1148     . name = strDiagDatereihe1
1149     . Values = arrSonnenHoehe24He
1150     . XValues = arrAzimut24He
1151     . Border . ColorIndex = 16
1152     . MarkerStyle = xlMarkerStyleCircle
1153     . MarkerBackgroundColorIndex = 15
1154     . MarkerForegroundColorIndex = 48
1155 End With
1156 'Winter
1157 With . NewSeries
1158     . name = strDiagDatereihe1
1159     . Values = arrSonnenHoehe24Wi
1160     . XValues = arrAzimut24Wi
1161     . Border . ColorIndex = 16
1162     . MarkerStyle = xlMarkerStyleCircle
1163     . MarkerBackgroundColorIndex = 15
1164     . MarkerForegroundColorIndex = 48
1165 For n = 1 To 24
1166     If n < 10 Then

```

```

1165         ptag = "0" & n & ":00"
1166     Else
1167         ptag = n & ":00"
1168     End If
1169     . Points(n). ApplyDataLabels Type:=xlDataLabelsShowLabel,
1170         AutoText:=True
1171     . Points(n). DataLabel.text = ptag
1172     Next
1173 End With
1174 'Debug.Print "Fehler in Diag_Sonnenstand_Chart" & Err.Number, Err.
1175     Description
1176 End Sub
1177
1178 Sub Diag_GlobalNeigung()
1179
1180 '
1181 ' Zur Erstellung der Strahlungsvariation auf verschiedenen geneigten
1182 ' Flächen werden hier temporäre Variablen erzeugt mit denen anschließend
1183 ' das Sub Globalstrahlung mehrmals durchlaufen wird.
1184 ' Die Temporär gespeicherten Daten werden anschließend als Diagramm erzeugt
1185 '
1186 Dim alpha0 As Integer
1187 Dim alpha30 As Integer
1188 Dim alpha45 As Integer
1189 Dim alpha60 As Integer
1190 Dim alpha90 As Integer
1191
1192 Dim Egges_0(1 To 12) As Variant
1193 Dim Egges_30(1 To 12) As Variant
1194 Dim Egges_45(1 To 12) As Variant
1195 Dim Egges_60(1 To 12) As Variant
1196 Dim Egges_90(1 To 12) As Variant
1197 Dim Egges_gHoehenwinkelEbeneGrad(1 To 12) As Variant
1198
1199 alpha0 = 0
1200 alpha30 = 30
1201 alpha45 = 45
1202 alpha60 = 60
1203 alpha90 = 90
1204
1205 Call Globalstrahlung(alpha0, 2, EingabeFall, Egges_0)
1206 Call Globalstrahlung(alpha30, 2, EingabeFall, Egges_30)
1207 Call Globalstrahlung(alpha45, 2, EingabeFall, Egges_45)
1208 Call Globalstrahlung(alpha60, 2, EingabeFall, Egges_60)
1209 Call Globalstrahlung(alpha90, 2, EingabeFall, Egges_90)
1210 Call Globalstrahlung(gHoehenwinkelEbeneGrad, 2, EingabeFall,
1211     Egges_gHoehenwinkelEbeneGrad)
1212
1213 Dim i As Integer
1214 Dim m As Integer
1215 Dim k As Integer
1216 Dim arrAzimut24(1 To 24) As Double
1217 Dim arrSonnenHoehe24(1 To 24) As Double
1218 Dim strDatumsangabe As Variant
1219
1220
1221 Dim strDiagObjektName As String

```

```

1222 Dim strDiagDatenreihe1 As String
1223 Dim strDiagDatenreihe2 As String
1224 Dim strDiagDatenreihe3 As String
1225 Dim strDiagDatenreihe4 As String
1226 Dim strDiagDatenreihe5 As String
1227 Dim strDiagDatenreihe6 As String
1228 Dim strDiagWorksheet As String
1229
1230 'Namen der Datenreihe für Legende
1231 strDiagObjektName = "JahresgangGlobalstrahlung"
1232 strDiagDatenreihe1 = "0° Neigung"
1233 strDiagDatenreihe2 = "30° Neigung"
1234 strDiagDatenreihe3 = "45° Neigung"
1235 strDiagDatenreihe4 = "60° Neigung"
1236 strDiagDatenreihe5 = "90° Neigung"
1237 strDiagDatenreihe6 = gHoehenwinkelEbeneGrad & "° Neigung(laut Eingabe)"
1238 strDiagWorksheet = strDiagramme
1239 'Übergabe der Arrays
1240 Dim x As Variant
1241 Dim array1 As Variant
1242 Dim array2 As Variant
1243 Dim array3 As Variant
1244 Dim array4 As Variant
1245 Dim array5 As Variant
1246 Dim array6 As Variant
1247 array1 = Egges_0
1248 array2 = Egges_30
1249 array3 = Egges_45
1250 array4 = Egges_60
1251 array5 = Egges_90
1252 array6 = Egges_gHoehenwinkelEbeneGrad
1253 'array2 = arrSonnenHoehe24
1254
1255
1256 ' Erzeugung der X-Achse
1257 Dim arrayX(1 To 12) As Variant
1258 Dim icount As Integer
1259 For icount = 1 To 12
1260     arrayX(icount) = strMonat(icount)
1261 Next
1262
1263 'Erzeugen des Diagramms
1264
1265 On Error GoTo ErrorHandler
1266 Application.ScreenUpdating = False
1267
1268
1269 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1270     Chart
1271         .SeriesCollection(1).Values = array1
1272         .SeriesCollection(2).Values = array2
1273         .SeriesCollection(3).Values = array3
1274         .SeriesCollection(4).Values = array4
1275         .SeriesCollection(5).Values = array5
1276         .SeriesCollection(6).Values = array6
1277         .Legend.Delete
1278         .Legend.Refresh
1279     End With
1280
1281 Application.ScreenUpdating = True

```

```

1282 Exit Sub
1283
1284 ErrorHandler:
1285   ' Falls Diagramm nicht existiert ''
1286   If Err.Number = -2147024809 Then
1287     Worksheets(strDiagWorksheet).ChartObjects.Add(10, 1610, 940, 500).
1288     name = strDiagObjektName
1289   End If
1290   ' Falls Diagramm zwar existiert aber eine Datenreihe fehlt ''
1291   If Err.Number = 1004 Then
1292     For Each x In Worksheets(strDiagWorksheet).ChartObjects.Item(
1293       strDiagObjektName).Chart.SeriesCollection
1294       x.Delete
1295     Next x
1296   End If
1297   ' wird bei einem Fehler auf jedenfall ausgefuehrt ''
1298   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1299     Chart
1300       .HasTitle = True
1301       .ChartTitle.text = "Jahresgang der Globalstrahlung auf geneigte
1302         Fläche"
1303       .ChartType = xlXYScatterSmoothNoMarkers
1304       .HasLegend = True
1305       .Legend.IncludeInLayout = False
1306       .Legend.Position = xlCorner
1307       .Axes(xlValue).TickLabels.Orientation = xlUpward
1308       .Axes(xlCategory, xlPrimary).HasTitle = True
1309       .Axes(xlCategory, xlPrimary).AxisTitle.Characters.text = "Monat"
1310       .Axes(xlCategory).MinimumScale = 1
1311       .Axes(xlCategory).MaximumScale = 12
1312       .Axes(xlCategory).MajorUnit = 1
1313     With .Axes(xlValue)
1314       .HasTitle = True
1315       .AxisTitle.text = "Monatlich Kumulierte Globalstrahlung der
1316         geneigten Fläche in [kWh]"
1317     End With
1318   End With
1319   x = 1
1320   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1321     Chart.SeriesCollection
1322     With .NewSeries
1323       .name = strDiagDatenreihe1
1324       .Values = array1
1325       .XValues = arrayX
1326       .Border.ColorIndex = 3
1327     End With
1328   End With
1329   x = x + 1
1330   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1331     Chart.SeriesCollection
1332     With .NewSeries
1333       .name = strDiagDatenreihe2
1334       .Values = array2
1335       .XValues = arrayX
1336       .Border.ColorIndex = 46
1337     End With
1338   End With
1339   x = x + 1
1340   With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1341     Chart.SeriesCollection
1342     With .NewSeries

```

```

1335     .name = strDiagDatenreihe3
1336     .Values = array3
1337     .XValues = arrayX
1338     .Border.ColorIndex = 45
1339   End With
1340 End With
1341   x = x + 1
1342 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1343   Chart.SeriesCollection
1344   With .NewSeries
1345     .name = strDiagDatenreihe4
1346     .Values = array4
1347     .XValues = arrayX
1348     .Border.ColorIndex = 44
1349   End With
1350 End With
1351   x = x + 1
1352 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1353   Chart.SeriesCollection
1354   With .NewSeries
1355     .name = strDiagDatenreihe5
1356     .Values = array5
1357     .XValues = arrayX
1358     .Border.ColorIndex = 40
1359   End With
1360 End With
1361   x = x + 1
1362 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1363   Chart.SeriesCollection
1364   With .NewSeries
1365     .name = strDiagDatenreihe6
1366     .Values = array6
1367     .XValues = arrayX
1368     .Border.ColorIndex = 33
1369   End With
1370 End With
1371 Sub diag_Geneigt_Horizont_GST()
1372 '#####
1373 '## Erstellt ein Clustered_Stacked Column Chart
1374 '##(Gruppiertes und Gestapeltes Säulendiagramm)
1375 '#####
1376 'arrA ,arrB ,arrC sind Arrays die eine andere Aufteilung der Eingegebenen
1377 'zurückgeben
1378 Dim arrA(1 To 36) As Variant
1379 Dim arrB(1 To 36) As Variant
1380 Dim arrC(1 To 36) As Variant
1381 'die ersten drei Arrays sind die , die im Diagramm stehen sollen
1382 'Call Saeulendiagramm_Aufteilung(EggesKS_Kum_kWh, EdiffgenKS_Kum_kWh,
1383 'EdingenKS_Kum_kWh, arrA , arrB , arrC)
1384 Dim strDiagObjektName As String
1385 Dim strDiagDatenreihe1 As String
1386 Dim strDiagDatenreihe2 As String
1387 Dim strDiagDatenreihe3 As String
1388 Dim strDiagWorksheet As String
1389 'Namendes Objektes der Datenreihe für Legende

```

```

1391 strDiagObjektName = "HorGen_Strahlung"
1392 strDiagDatenreihe1 = "horizontale Strahlung EghorKS_Kum_kWh"
1393 strDiagDatenreihe2 = "geneigte Strahlung EggesKS_Kum_kWh"
1394 strDiagDatenreihe3 = "leer"
1395 strDiagWorksheet = strDiagramme
1396 'Übergabe der Arrays
1397 Dim x As Variant
1398 Dim array1 As Variant
1399 Dim array2 As Variant
1400 Dim array3 As Variant
1401
1402 Select Case EingabeFall
1403     Case 1, 3, 4, 5      ' bekannter Standort (Wien, Wien hohe Warte,...) und
1404         Glob-, Diff-, Direktstrahlung gegeben
1405         strDiagDatenreihe1 = "horizontale Strahlung EghorKS_Kum_kWh"
1406         strDiagDatenreihe2 = "geneigte Strahlung EggesKS_Kum_kWh"
1407         strDiagDatenreihe3 = "leer"
1408
1409         array1 = EghorKS_Kum_kWh
1410         array2 = EggesKS_Kum_kWh
1411         array3 = ""
1412     Case 2      ' Vergleichsstandort, nur Globalstrahlung bekannt
1413
1414         strDiagDatenreihe1 = "horizontale Strahlung EghorKS_Kum_kWh"
1415         strDiagDatenreihe2 = "geneigte Strahlung EggesER_Kum_kWh"
1416         strDiagDatenreihe3 = "leer"
1417
1418         array1 = EghorKS_Kum_kWh
1419         array2 = EggesER_Kum_kWh
1420         array3 = ""
1421 End Select
1422 ' Erzeugung der X-Achse
1423 Dim arrayX(1 To 12) As Variant
1424
1425 Dim i As Integer
1426
1427 For i = 1 To 12
1428     arrayX(i) = strMonat(i)
1429 Next
1430 'Erzeugen des Diagramms
1431 On Error GoTo ErrorHandler
1432 Application.ScreenUpdating = False
1433
1434
1435 With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName).
1436     Chart
1437         .SeriesCollection(1).Values = array1
1438         .SeriesCollection(1).name = strDiagDatenreihe1
1439         .SeriesCollection(2).Values = array2
1440         .SeriesCollection(2).name = strDiagDatenreihe2
1441         .SeriesCollection(3).Values = array3
1442 End With
1443 Application.ScreenUpdating = True
1444 Exit Sub
1445
1446 ErrorHandler:
1447     ' Falls Diagramm nicht existiert ''
1448 If Err.Number = -2147024809 Or Err.Number = 1004 Then

```

```

1449     Worksheets(strDiagWorksheet).ChartObjects.Add(10, 2110, 470, 300).
1450         name = strDiagObjektName
1451     '' wird bei einem Fehler auf jedenfall ausgefuehrt ''
1452     With Worksheets(strDiagWorksheet).ChartObjects.Item(strDiagObjektName)
1453         With .Chart
1454             .ChartType = xlColumnClustered
1455             .ChartGroups(1).GapWidth = 0
1456             .HasTitle = True
1457             .ChartTitle.text = "Horizontal- vs. geneigter Strahlung"
1458             .HasLegend = True
1459             .Legend.Position = xlBottom
1460             .PlotArea.Interior.ColorIndex = 2
1461         With .SeriesCollection
1462             With .NewSeries
1463                 .name = strDiagDatenreihe1
1464                 .Values = array1
1465                 .XValues = arrayX
1466                 .Interior.ColorIndex = 50
1467             End With
1468             With .NewSeries
1469                 .name = strDiagDatenreihe2
1470                 .Values = array2
1471                 .XValues = arrayX
1472                 .Interior.ColorIndex = 33
1473             End With
1474             With .NewSeries
1475                 .name = ""
1476                 .Values = array2
1477                 .XValues = arrayX
1478                 .Interior.ColorIndex = 4
1479                 .Format.Fill.Transparency = 1
1480             End With
1481         With .Axes(xlValue)
1482             .HasTitle = True
1483             .AxisTitle.text = "Kumulierte monatl. Strahlung [kWh]"
1484         End With
1485     End With
1486 End With
1487 Else
1488     Debug.Print "funcESpDiag", Err.Number, Err.Description
1489 End If
1490 Application.ScreenUpdating = True
1491 End Sub

```

Listing E.5: Modul Kollektor Kapitel 6

## E.6. 7. Kapitel Daten Ein- Ausgabe Erzeugung

```

1 Sub Kapitel_7()
2 '### 7. Daten Ein- Ausgabe Erzeugung
3 '### 7. Daten Ein- Ausgabe Erzeugung
4 '### 7. Daten Ein- Ausgabe Erzeugung
5 End Sub
6
7
8 Sub print_Dropdown_update(Titel As String, strAnlagenIndex As Integer)
9 'Private Farb Parameter
10 Dim colorParameterListeKollektorPrivat As Variant
11 Dim colorparameterWerteKollektorPrivat As Variant
12 Dim colorparameterAuswahlKollektorPrivat As Variant

```

```

13
14 colorParameterListeKollektorPrivat = RGB(255, 150, 60)
15 colorparameterWerteKollektorPrivat = RGB(230, 150, 60)
16 colorparameterAuswahlKollektorPrivat = RGB(240, 150, 60)
17
18 ' Privates Auslesen des Projekt String
19 Dim strProjektName As String
20 Dim strEingabeName As String
21     strProjektName = Evaluate(Names.Item("Projektname").RefersTo)
22     strEingabeName = strProjektName & "_Eingabe"
23
24 Dim strAnlageStandort As String
25 Dim arrStandortListe As Variant
26     arrStandortListe = Array("Wien Hohe Warte SH198m",
27                             "Wien Innere Stadt SH171m",
28                             "Klagenfurt SH450m",
29                             "Bregenz SH424m",
30                             "Eisenstadt SH184m",
31                             "Lienz SH659m",
32                             "Linz SH263m",
33                             "St. Pölten SH270m",
34                             "Innsbruck Uni SH578m",
35                             "Grossenzersdorf SH153m",
36                             "Graz Uni SH366m",
37                             "Vergleichsstandort")
38
39     strAnlageStandort = arrStandortListe(strAnlagenIndex)
40 ' Ausgabe je nach Status der Checkbox
41 Dim strbenutzerdef As String
42     strbenutzerdef = strAnlageStandort & "Ben.def."
43 Dim checkBox As Boolean
44     checkBox = Evaluate(Names.Item("CheckBoxBenutzer").RefersTo)
45
46 If checkBox = False Then
47     With Worksheets(strEingabeName).Select
48         With Worksheets(strEingabeName)
49             .Cells(3, 2).Value = strbenutzerdef
50             .Cells(4, 2).Value = dblGeoLaengeGradKlimadaten
51             .Cells(5, 2).Value = dblGeoBreiteGradKlimadaten
52             .Range(Cells(3, 2), Cells(5, 2)).EntireColumn.AutoFit
53             .Range(Cells(3, 2), Cells(5, 2)).Interior.Color =
54                 colorparameterAuswahlKollektorPrivat
55         End With
56     End With
57 Else
58     With Worksheets(strEingabeName).Select
59         With Worksheets(strEingabeName)
60             .Range(Cells(3, 2), Cells(5, 2)).ClearContents
61             .Cells(3, 2).Value = strAnlageStandort
62             .Range(Cells(3, 2), Cells(5, 2)).EntireColumn.AutoFit
63             .Range(Cells(3, 2), Cells(3, 2)).Interior.Color =
64                 colorparameterAuswahlKollektorPrivat
65             .Range(Cells(4, 2), Cells(5, 2)).Interior.Color =
66                 colorparameterWerteKollektorPrivat
67         End With
68     End With
69 End If
70 End Sub      'print_Dropdown_update
71
72 Sub print_to_strEingabe(checkBox As Boolean)
73     'Private Farb Parameter

```

```

71  Dim colorParameterListeKollektorPrivat As Variant
72  Dim colorparameterWerteKollektorPrivat As Variant
73  Dim colorparameterAuswahlKollektorPrivat As Variant
74
75      colorParameterListeKollektorPrivat = RGB(255, 150, 60)
76      colorparameterWerteKollektorPrivat = RGB(230, 150, 60)
77      colorparameterAuswahlKollektorPrivat = RGB(240, 150, 60)
78
79  'Privates Auslesen des Projekt String
80  Dim strProjektName As String
81  Dim strEingabeName As String
82      strProjektName = Evaluate(Names.Item("Projektname").RefersTo)
83      strEingabeName = strProjektName & "_Eingabe"
84
85  Dim strAnlageStandort As String
86  Dim arrStandortListe As Variant
87      arrStandortListe = Array("Wien Hohe Warte SH198m",
88                                "Wien Innere Stadt SH171m",
89                                "Klagenfurt SH450m",
90                                "Bregenz SH424m",
91                                "Eisenstadt SH184m",
92                                "Lienz SH659m",
93                                "Linz SH263m",
94                                "St. Pölten SH270m",
95                                "Innsbruck Uni SH578m",
96                                "Grossenzersdorf SH153m",
97                                "Graz Uni SH366m",
98                                "Vergleichsstandort")
99
100
101     strAnlageStandort = arrStandortListe(Evaluate(Names.Item("StandortIndex").RefersTo))
102
103  'Ausgabe je nach Status der Checkbox
104  Dim strbenutzerdef As String
105      strbenutzerdef = strAnlageStandort & "Ben.def."
106
107  If checkBox = False Then
108      With Worksheets(strEingabeName).Select
109          With Worksheets(strEingabeName)
110              .Cells(3, 2).value = strbenutzerdef
111              .Cells(4, 2).value = dblGeoLaengeGradKlimadaten
112              .Cells(5, 2).value = dblGeoBreiteGradKlimadaten
113              .Range(Cells(3, 2), Cells(5, 2)).EntireColumn.AutoFit
114              .Range(Cells(3, 2), Cells(5, 2)).Interior.Color =
115                  colorparameterAuswahlKollektorPrivat
116          End With
117      End With
118  Else
119      'BereichBenutzerdef.ClearContents
120      With Worksheets(strEingabeName).Select
121          With Worksheets(strEingabeName)
122              .Range(Cells(3, 2), Cells(5, 2)).ClearContents
123              .Cells(3, 2).value = strAnlageStandort
124              .Range(Cells(3, 2), Cells(5, 2)).EntireColumn.AutoFit
125              .Range(Cells(3, 2), Cells(3, 2)).Interior.Color =
126                  colorparameterAuswahlKollektorPrivat
127              .Range(Cells(4, 2), Cells(5, 2)).Interior.Color =
128                  colorparameterWerteKollektorPrivat
129          End With
130      End With

```

```

128     End If
129 End Sub      'print_to_strEingabe
130
131
132 Sub getKlimadaten()
133 '#####
134 '## Sub Holt sich mit "StandortIndex" den Index des Dropdown
135 '## Feldes aus der Ribbon Umgebung.
136 '## Mit dem Index wird der Namen strAusgesuchterStandort
137 '## des Ortes erzeugt.
138 '#####
139
140 Dim i As Integer
141 Dim m As Integer
142 Dim aktivierterCount As Integer
143 Dim intBlattIndex As Variant
144
145 Dim intRelOffsetAussenTemp As Integer
146 Dim intRelOffsetEghorKS As Integer
147 Dim intRelOffsetEdiffhorKS As Integer
148 Dim intRelOffsetEdirhorKS As Integer
149 Dim intRelOffsetGeoLaenge As Integer
150 Dim intRelOffsetGeoBreite As Integer
151 Dim intRelOffsetEinheit As Integer
152
153 Dim strEinheit As String
154 Dim strGSCheck As String
155 Dim strHSCheck As String
156 Dim strDSCheck As String
157
158
159 Dim strProjektName As String
160 Dim strAusgabeName As String
161     strProjektName = Evaluate(Names.Item("Projektname").RefersTo)
162     strAusgabeName = strProjektName & "_Ausgabe"
163
164 arrStandort = Array("Wien Hohe Warte SH198m", _
165                     "Wien Innere Stadt SH171m", _
166                     "Klagenfurt SH450m", _
167                     "Bregenz SH424m", _
168                     "Eisenstadt SH184m", _
169                     "Lienz SH659m", _
170                     "Linz SH263m", _
171                     "St. Pölten SH270m", _
172                     "Innsbruck Uni SH578m", _
173                     "Grossenzersdorf SH153m", _
174                     "Graz Uni SH366m", _
175                     "Vergleichsstandort")
176
177
178 arrStandortSuche() = Sheets("klimadaten").Range("A3:FL3").value
179 strAusgesuchterStandort = arrStandort(Evaluate(Names.Item("StandortIndex"
180     ") .RefersTo)))
181
182 For i = 1 To 168
183     If arrStandortSuche(1, i) = strAusgesuchterStandort Then
184         intBlattIndex = i
185     Next
186
187     intRelOffsetAussenTemp = 3
188     intRelOffsetEghorKS = 5

```

```

187 intRelOffsetEdiffhorKS = 6
188 intRelOffsetEdirhorKS = 7
189 intRelOffsetGeoLaenge = 1
190 intRelOffsetGeoBreite = 3
191 intRelOffsetEinheit = 5
192
193 With Worksheets("klimadaten")
194     dblT_AussenKS() = .Range(.Cells(12, intBlattIndex +
195         intRelOffsetAussenTemp), .Cells(8772, intBlattIndex +
196         intRelOffsetAussenTemp)).value
197
198     EghorKS() = .Range(.Cells(12, intBlattIndex + intRelOffsetEghorKS), .
199         Cells(8772, intBlattIndex +
200         intRelOffsetEghorKS)).value
201     EdiffhorKS() = .Range(.Cells(12, intBlattIndex +
202         intRelOffsetEdiffhorKS), .Cells(8772, intBlattIndex +
203         intRelOffsetEdiffhorKS)).value
204     EdirhorKS() = .Range(.Cells(12, intBlattIndex + intRelOffsetEdirhorKS
205         ), .Cells(8772, intBlattIndex +
206         intRelOffsetEdirhorKS)).value
207
208     dblGeoLaengeGradKlimadaten = .Cells(7, intBlattIndex +
209         intRelOffsetGeoLaenge)
210     dblGeoBreiteGradKlimadaten = .Cells(7, intBlattIndex +
211         intRelOffsetGeoBreite)
212
213     strEinheit = .Cells(10, intBlattIndex + intRelOffsetEinheit)
214
215     strGSCheck = .Cells(12, intBlattIndex + 5)      'GS... Globalstrahlung
216     strHSCheck = .Cells(12, intBlattIndex + 6)      'HS...
217     Himmelsschrahlung = Diffusestrahlung
218     strDSCheck = .Cells(12, intBlattIndex + 7)      'DS... Direktstrahlung
219
220 End With
221 #####
222 '### Falls EghorKS, EdiffhorKS & EdirhorKS in J/cm^2 wird es automatisch
223 umgerechnet
224 #####
225 Dim Antwort As Variant
226     'Call diag_Klimadaten(3)
227
228 If strEinheit = "J/cm2" Or strEinheit = "j/cm2" Or strEinheit = "J/cm
229     ^2" Or strEinheit = "j/cm^2" Then
230     For m = 1 To UBound(EghorKS, 1)
231         EghorKS(m, 1) = EghorKS(m, 1) / (3600 * 10 ^ (-4))
232         EdiffhorKS(m, 1) = EdiffhorKS(m, 1) / (3600 * 10 ^ (-4))
233         EdirhorKS(m, 1) = EdirhorKS(m, 1) / (3600 * 10 ^ (-4))
234     Next
235     'Debug.Print "Achtung der Klimadatensatz wird in J/cm2 gerechnet,
236     'es findet eine Umrechnung statt"
237 ElseIf strEinheit = "W/m2" Or strEinheit = "w/m2" Or strEinheit = "W/
238     m^2" Or strEinheit = "w/m^2" Then
239     'Debug.Print "Achtung der Klimadatensatz wird in W/m2 gerechnet,
240     'es findet keine Umrechnung statt"
241 Else
242     Antwort = CriticalMsgbox("Falsche oder keine Eingabe der Einheiten im
243         Klimadatensatz, " _
244             & vbCrLf & "es ist nur W/m^2
245             oder J/cm^2 zulässig." &
246             vbCrLf _
```

```

232                                     & "Berechnung wird Abgebrochen  

233                                     !" & vbCrLf  

234                                     & "siehe: Klimadaten unter " &  

235                                     strAusgesuchterStandort ,  

236                                     "Eingabekontrolle" , vbCritical  

237                                     + vbOKOnly)  

238 If Antwort = 1 Then  

239     Call SchreibeStatusleiste  

240         ("Projekt wurde aufgrund eines Fehlers der Einheiten im  

241         Klimadatensatz nicht berechnet")  

242     Exit Sub  

243 End If  

244 End If  

245  

246 Call diag_Klimadaten(1)  

247 '##### Fallunterscheidung beim Vergleisstandort  

248 '### Fallunterscheidung beim Vergleisstandort  

249  

250 EingabeFall = 1  

251 If strAusgesuchterStandort = "Vergleichsstandort" Then  

252  

253 If strGSCheck = "" Then          'wenn leer dann  

254     If strHSCheck <> "" And strDSCheck <> "" Then  

255         'Schleife  

256         'Eghor= Ediff + Edir  

257         For m = 1 To UBound(EdirhorKS, 1)  

258             EghorKS(m, 1) = EdiffhorKS(m, 1) + EdirhorKS(m, 1)  

259         Next  

260         EingabeFall = 6  

261     ElseIf strHSCheck = "" And strDSCheck <> "" Then  

262         EingabeFall = 8  

263         MsgBox ("Eingabefall" & EingabeFall & " wurde noch nicht  

264             berücksichtigt")  

265     ElseIf strHSCheck <> "" And strDSCheck = "" Then  

266         EingabeFall = 9  

267         MsgBox ("Eingabefall" & EingabeFall & " wurde noch nicht  

268             berücksichtigt")  

269     ElseIf strHSCheck = "" And strDSCheck = "" Then  

270         EingabeFall = 7  

271         MsgBox ("Eingabefall" & EingabeFall & " wurde noch nicht  

272             berücksichtigt")  

273     End If  

274     ElseIf strGSCheck <> "" Then          'wenn voll dann  

275         If strHSCheck <> "" And strDSCheck <> "" Then  

276             EingabeFall = 5  

277             'Alle Arrays sind gefüllt  

278  

279             ElseIf strHSCheck = "" And strDSCheck <> "" Then  

280                 For m = 1 To UBound(EghorKS, 1)  

281                     If EdirhorKS(m, 1) <> 0 Then          '0|0  

282                         EdiffhorKS(m, 1) = EghorKS(m, 1) - EdirhorKS(m, 1)  

283                     Else 'Der Fall ergibt sich wenn Glob-. gegeben ist aber keine  

284                         Direkt-. mehr da ist.  

285                         'Es wird im sub Globalstrahlung eine Aufteilung von diff  

286                         -. & Direkt-. unternommen!  

287                         EdiffhorKS(m, 1) = 0  

288                     End If  

289                 Next  

290                 EingabeFall = 3

```

```

283 ElseIf strHSCheck <> "" And strDSCheck = "" Then
284     For m = 1 To UBound(EghorKS, 1)
285         EdirhorKS(m, 1) = EghorKS(m, 1) - EdiffhorKS(m, 1)
286     Next
287     EingabeFall = 4
288 ElseIf strHSCheck = "" And strDSCheck = "" Then
289     'Die beiden Terme werden errechnet
290     EingabeFall = 2
291 End If
292 End If
293 End If
294 End Sub
295 Sub getGeodaten()
296     '#####
297     '## Sub sucht sich mit "StandortIndex" den Index des Dropdown
298     '## Feldes aus der Ribbon Umgebung.
299     '## Mit dem Index wird der Namen strAusgesuchterStandort
300     '## des Ortes erzeugt.
301     '## somit können GeoLänge- und Breite ausgelesen werden.
302     '#####
303
304 Dim i As Integer
305 Dim m As Integer
306 Dim aktivierterCount As Integer
307 Dim intBlattIndex As Variant
308
309 Dim intRelOffsetGeoLaenge As Integer
310 Dim intRelOffsetGeoBreite As Integer
311
312 Dim strProjektName As String
313 Dim strAusgabeName As String
314     strProjektName = Evaluate(Names.Item("Projektname").RefersTo)
315     strAusgabeName = strProjektName & "_Ausgabe"
316
317 arrStandort = Array("Wien Hohe Warte SH198m",
318                     "Wien Innere Stadt SH171m",
319                     "Klagenfurt SH450m",
320                     "Bregenz SH424m",
321                     "Eisenstadt SH184m",
322                     "Lienz SH659m",
323                     "Linz SH263m",
324                     "St. Pölten SH270m",
325                     "Innsbruck Uni SH578m",
326                     "Grossenzersdorf SH153m",
327                     "Graz Uni SH366m",
328                     "Vergleichsstandort")
329
330
331 arrStandortSuche() = Sheets("klimadaten").Range("A3:FL3").value
332 strAusgesuchterStandort = arrStandort(Evaluate(Names.Item("StandortIndex"
333             ").RefersTo)))
334
335 For i = 1 To 168
336     If arrStandortSuche(1, i) = strAusgesuchterStandort Then
337         intBlattIndex = i
338     Next
339     intRelOffsetGeoLaenge = 1
340     intRelOffsetGeoBreite = 3
341
342     With Worksheets("klimadaten")

```

```

341     dblGeoLaengeGradKlimadaten = .Cells(7, intBlattIndex +
342         intRelOffsetGeoLaenge)
343     dblGeoBreiteGradKlimadaten = .Cells(7, intBlattIndex +
344         intRelOffsetGeoBreite)
344     End With
345
346
347 Sub getEingabe()
348 'Auslesen der Eingetragenen Werte aus dem strEingabebereich
349 Dim strEingabebereich As String
350 strEingabebereich = Range( Cells(intReiheObenParameterListe + 1,
351     intSpalteParameterListeKollektor), _
352     Cells(intReiheUntenParameterListeKollektor + 1,
353     intSpalteParameterListeKollektor + 1)).Address
354
355 dblGeoLaengeGrad = WorksheetFunction.VLookup("GeoLänge", Worksheets(
356     strEingabe).Range(strEingabebereich), 2, 0)
357 dblGeoBreiteGrad = WorksheetFunction.VLookup("GeoBreite", Worksheets(
358     strEingabe).Range(strEingabebereich), 2, 0)
359 gAlbedo = WorksheetFunction.VLookup("Albedo", Worksheets(strEingabe).
360     Range(strEingabebereich), 2, 0)
361 gAzimutEbeneGrad = WorksheetFunction.VLookup("Azimut des Kollektors",
362     Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
363 gHoehenwinkelEbeneGrad = WorksheetFunction.VLookup("Höhenwinkel des
364     Kollektors", Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
365 gC0 = WorksheetFunction.VLookup("C0 optischer Wirkungsgrad", Worksheets(
366     strEingabe).Range(strEingabebereich), 2, 0)
367 gC1 = WorksheetFunction.VLookup("C1 lineare Wärmeverlustkoeffizient",
368     Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
369 gC2 = WorksheetFunction.VLookup("C2 quadratischer Wärmeverlustkoeffizient
370     ", Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
371 gWinkelfaktor50 = WorksheetFunction.VLookup("Winkelfaktor bei 50°
372     Einfallwinkel", Worksheets(strEingabe).Range(strEingabebereich), 2,
373     0)
374 gEinfallwinkelGrad = WorksheetFunction.VLookup("Einfallwinkel",
375     Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
376 gKolFlaeche = WorksheetFunction.VLookup("Kollektor Fläche", Worksheets(
377     strEingabe).Range(strEingabebereich), 2, 0)
378 gBModell = WorksheetFunction.VLookup("Sonnenstand Berechnungsmodell
379     (1,2,3)", Worksheets(strEingabe).Range(strEingabebereich), 2, 0)
380 gWinkelfaktorFALL = WorksheetFunction.VLookup("Winkelfaktor
381     Berücksichtigung (ja / nein)", Worksheets(strEingabe).Range(
382     strEingabebereich), 2, 0)
383 gStrahlungsVariationsAusgabe = WorksheetFunction.VLookup("Diagrammausgabe
384     Strahlungsvariation (KS oder ER)", Worksheets(strEingabe).Range(
385     strEingabebereich), 2, 0)
386
387 End Sub
388 Sub Datenzusammenfassung_fur_Output()
389 '######
390 '# In dem Sub werden die Umrechnungen von
391 '# W zu kWh und kumulierten werte erstellt
392 '##
393 '# Deklarationen stehen im Kollektor Modul
394 '######
395
396
397
398 'Werte Kumulieren
399   'Horizontal
400     'Gesamt Horizontal

```

```

381 Call ArrayKumulieren(EghorKS, EghorKS_Kum)
382
383 'Diffuse
384 Call ArrayKumulieren(EdiffhorER, EdiffhorER_Kum)
385 Call ArrayKumulieren(EdiffhorKS, EdiffhorKS_Kum)
386 'Direkte
387 Call ArrayKumulieren(EdirhorER, EdirhorER_Kum)
388 Call ArrayKumulieren(EdirhorKS, EdirhorKS_Kum)
389
390 'Geneigt
391   'Gesamt Horizontal
392     Call ArrayKumulieren(EggesER, EggesER_Kum)
393     Call ArrayKumulieren(EggesKS, EggesKS_Kum)
394   'Diffuse
395     Call ArrayKumulieren(EdiffgenER, EdiffgenER_Kum)
396     Call ArrayKumulieren(EdiffgenKS, EdiffgenKS_Kum)
397   'Direkte
398     Call ArrayKumulieren(EdirgenER, EdirgenER_Kum)
399     Call ArrayKumulieren(EdirgenKS, EdirgenKS_Kum)
400
401 'Umrechnung von W in kWh
402   'Horizontal
403     'Gesamt Horizontal
404       Call E_to_kWh(EghorKS_Kum, EghorKS_Kum_kWh)
405
406     'Diffuse
407       Call E_to_kWh(EdiffhorKS_Kum, EdiffhorKS_Kum_kWh)
408       Call E_to_kWh(EdiffhorER_Kum, EdiffhorER_Kum_kWh)
409     'Direkte
410       Call E_to_kWh(EdirhorKS_Kum, EdirhorKS_Kum_kWh)
411       Call E_to_kWh(EdirhorER_Kum, EdirhorER_Kum_kWh)
412
413   'Geneigt
414     'Gesamt Horizontal
415       Call E_to_kWh(EggesKS_Kum, EggesKS_Kum_kWh)
416       Call E_to_kWh(EggesER_Kum, EggesER_Kum_kWh)
417     'Diffuse
418       Call E_to_kWh(EdiffgenKS_Kum, EdiffgenKS_Kum_kWh)
419       Call E_to_kWh(EdiffgenER_Kum, EdiffgenER_Kum_kWh)
420     'Direkte
421       Call E_to_kWh(EdirgenKS_Kum, EdirgenKS_Kum_kWh)
422       Call E_to_kWh(EdirgenER_Kum, EdirgenER_Kum_kWh)
423
424
425   'Kumuliert und in kWh
426     Call kum_kwh(Kollektor.Erefflgen, Erefftgen_kum_kWh)
427
428 End Sub
429
430
431 Sub Output_Gesamt_to_ProjektAusgabe()
432
433 Dim strAusgabeName As String
434 strAusgabeName = strProjekt & "_Ausgabe"
435
436
437 Dim strAnlageStandort As String
438 Dim arrStandortListe As Variant
439 arrStandortListe = Array("Wien Hohe Warte SH198m", _
440                           "Wien Innere Stadt SH171m", _
441                           "Klagenfurt SH450m", _
```

```

442         "Bregenz SH424m" ,_
443         "Eisenstadt SH184m" ,_
444         "Lienz SH659m" ,_
445         "Linz SH263m" ,_
446         "St. Pölten SH270m" ,_
447         "Innsbruck Uni SH578m" ,_
448         "Grossenzersdorf SH153m" ,_
449         "Graz Uni SH366m" ,_
450         "Vergleichsstandort")
451
452 strAnlageStandort = arrStandortListe(Evaluate(Names.Item("StandortIndex") .
453     RefersTo))
454
455 Dim intRowOffset As Integer
456 intRowOffset = 2
457
458 Dim strFarbe1 As String
459 Dim strFarbeKS As String
460 Dim strFarbeKS2 As String
461 Dim strFarbeKS3 As String
462
463 Dim strFarbeER As String
464 Dim strFarbeER2 As String
465 Dim strFarbeER3 As String
466
467 strFarbe1 = RGB(202, 255, 112)
468
469 strFarbeKS = RGB(255, 165, 0)
470 strFarbeKS2 = RGB(255, 80, 30)
471 strFarbeKS3 = RGB(255, 165, 30)
472
473 strFarbeER = RGB(84, 255, 159)
474 strFarbeER2 = RGB(0, 255, 127)
475 strFarbeER3 = RGB(0, 255, 80)
476 Dim n As Integer
477
478 With Worksheets(strAusgabeName).Select
479     n = 1
480     Call WriteArrayFormatiert(arrStunde, intRowOffset, n, "Jahresstunde [h]")
481     n = n + 1
482     Call WriteArrayFormatiert(arrTagesnummer, intRowOffset, n, "Tag [Tag]")
483     n = n + 1
484     Call WriteArrayFormatiertDat(arrDatum, intRowOffset, n, "Datum [Tag.
485         Monat]")
486     n = n + 1
487     Call WriteArrayFormatiert(intMEZ, intRowOffset, n, "Ortszeit t MEZ (
488         Mitteleuropäische Zeit) [h]")
489     n = n + 1
490     Call WriteArrayFormatiert(arrAzimutGrad, intRowOffset, n, "Azimut der
491         Sonne [°]")
492     n = n + 1
493     Call WriteArrayFormatiert(arrSonnenHoeheGrad, intRowOffset, n, "
494         Sonnenhoehe [°]")
495     n = n + 1
496     Call WriteArrayFormatiert(E0hor, intRowOffset, n, "E0hor
497         extraterrestrische Bestrahlungsstärke auf Hor.Fl. [W/m^2]")
498     n = n + 1
499     Call WriteArrayFormatiert(epsilonKS_Grad, intRowOffset, n, "
500         epsilonKS_Grad Himmelsklarheitsindex KS [°]", strFarbe1)
501     n = n + 1

```

```

495  Call WriteArrayFormatiert(EghorKS, intRowOffset, n, "EghorKS Glob.str.
496    auf hor.fl. aus KS [W/m^2]", strFarbeKS2)
497  n = n + 1
498  Call WriteArrayFormatiert(EdiffhorKS, intRowOffset, n, "EdiffhorKS
499    diffuse Str. auf hor.Fl. aus KS [W/m^2]", strFarbeKS)
500  n = n + 1
501  Call WriteArrayFormatiert(EdirhorKS, intRowOffset, n, "EdirhorKS Dir.
502    Str. hor.Fl. aus KS [W/m^2]", strFarbeKS)
503  n = n + 1
504  Call WriteArrayFormatiert(EggesKS, intRowOffset, n, "EggesKS Glob.str.
505    auf gen.fl. aus KS [W/m^2]", strFarbeKS2)
506  n = n + 1
507  Call WriteArrayFormatiert(EdiffgenKS, intRowOffset, n, "EdiffgenKS
508    diffuse Str. auf gen.Fl. aus KS Perez M. [W/m^2]", strFarbeKS)
509  n = n + 1
510  Call WriteArrayFormatiert(EdiffgenIsoKS, intRowOffset, n, "EdiffgenIsoKS
511    Diff. Str. gen.Fl. aus KS isot. A. [W/m^2]", strFarbeKS)
512  n = n + 1
513  Call WriteArrayFormatiert(EdiffgenAnIsoKS, intRowOffset, n, "
514    EdiffgenAnIsoKS Glob.str. auf gen.fl. aus KS anisot. A. [W/m^2]",
515    strFarbeKS)
516  n = n + 1
517  Call WriteArrayFormatiert(EdirgenKS, intRowOffset, n, "EdirgenKS dir.
518    Str. auf gen.Fl. aus KS [W/m^2]", strFarbeKS)
519  n = n + 1
520  Call WriteArrayFormatiert(Erefflgen, intRowOffset, n, "Erefflgen
521    Bodenreflexion. Str. gen.Fl. isot.Anatz [W/m^2]", strFarbeKS)
522  n = n + 1
523  Call WriteArrayFormatiert(epsilonER_Grad, intRowOffset, n, "
524    epsilonER_Grad Himmelsklarheitsindex ER [°]", strFarbe1)
525  n = n + 1
526  Call WriteArrayFormatiert(EdiffhorER, intRowOffset, n, "EdiffhorER
527    diffuse Str. auf hor.Fl. aus ER [W/m^2]", strFarbeER)
528  n = n + 1
529  Call WriteArrayFormatiert(EdirhorER, intRowOffset, n, "EdirhorER Dir.
530    Str. hor.Fl. aus ER [W/m^2]", strFarbeER)
531  n = n + 1
532  Call WriteArrayFormatiert(EggesER, intRowOffset, n, "EggesER Glob.str.
533    auf gen.fl. aus ER [W/m^2]", strFarbeER2)
534  n = n + 1
535  Call WriteArrayFormatiert(EdiffgenER, intRowOffset, n, "EdiffgenER
536    diffuse Str. auf gen.Fl. aus ER Perez M. [W/m^2]", strFarbeER)
537  n = n + 1
538  Call WriteArrayFormatiert(EdiffgenIsoER, intRowOffset, n, "EdiffgenIsoER
539    Diff. Str. gen.Fl. aus ER isot. A. [W/m^2]", strFarbeER)
540  n = n + 1
541  Call WriteArrayFormatiert(EdiffgenAnIsoER, intRowOffset, n, "
542    EdiffgenAnIsoER Glob.str. auf gen.fl. aus ER anisot. A. [W/m^2]",
543    strFarbeER)
544  n = n + 1
545  Call WriteArrayFormatiert(EdirgenER, intRowOffset, n, "EdirgenER dir.
546    Str. auf gen.Fl. aus ER [W/m^2]", strFarbeER)
547  n = n + 1
548  Call WriteArrayFormatiert(Erefflgen, intRowOffset, n, "Erefflgen
549    Bodenreflexion. Str. gen.Fl. isot.Anatz [W/m^2]", strFarbeER)
550  n = n + 1
551  Call WriteArrayFormatiertProz(arrWirkungsgradKollektor, intRowOffset, n,
552    "Wirkungsgrad [%]")
553  n = n + 1
554  Call WriteArrayFormatiertProz(arrWinkelfaktorKollektor, intRowOffset, n,
555    "Winkelfaktor [%]")

```

```

534     mOffsetAusgabeKum = n + 1
535 
536 End With
537 mOffsetKollektorAusgabe = n
538 
539 With Worksheets(strAusgabeName).Range(Cells(intRowOffset - 1, 1), Cells(
540     intRowOffset - 1, n))
541     .Merge
542     .value = Evaluate(Names.Item("Projektnname").RefersTo) & " Standort: " &
543         strAnlageStandort
544     .HorizontalAlignment = xlCenter
545     .Interior.Color = RGB(70, 150, 255)
546     .Font.Size = 14
547 End With
548 
549 
550 
551 
552 Sub Output_to_AusgabeKum()
553 
554 '######
555 'wir in sub Datenzusammenfassung_fur_Output() erzeugt
556 ,
557 ''Werte Kkummulieren
558 ,
559 '    Dim EggesER_Kum(1 To 12) As Variant
560 '    Dim EggesKS_Kum(1 To 12) As Variant
561 ,
562 '    Call ArrayKumulieren(EggesER, EggesER_Kum)
563 '    Call ArrayKumulieren(EggesKS, EggesKS_Kum)
564 ''Debug.Print EggesER_Kum
565 ,
566 ''Umrechnung in kWh
567 '    Dim EggesKS_Kum_kWh(1 To 12) As Variant
568 '    Dim EggesER_Kum_kWh(1 To 12) As Variant
569 ,
570 ,
571 '    Call E_to_kWh(EggesKS_Kum, EggesKS_Kum_kWh)
572 '    Call E_to_kWh(EggesER_Kum, EggesER_Kum_kWh)
573 ,
574 '    'Call ArrayKumulieren(EghorKS, EghorKS_Kum)
575 '######
576 
577 'Privates Auslesen des Projekt String
578 'Dim strProjektName As String
579 Dim strAusgabeName As String
580 'strProjektName = Evaluate(Names.Item("Projektnname").RefersTo)
581 strAusgabeName = strProjekt & "_AusgabeKum"
582 'Debug.Print strAusgabeName
583 Dim strAnlageStandort As String
584 Dim arrStandortListe As Variant
585 arrStandortListe = Array("Wien Hohe Warte SH198m", _
586                             "Wien Innere Stadt SH171m", _
587                             "Klagenfurt SH450m", _
588                             "Bregenz SH424m", _
589                             "Eisenstadt SH184m", _
590                             "Lienz SH659m", _
591                             "Linz SH263m", _
592                             "St. Pölten SH270m", _

```

```

593         "Innsbruck Uni SH578m", _
594         "Grossenzersdorf SH153m", _
595         "Graz Uni SH366m", _
596         "Vergleichsstandort")
597 strAnlageStandort = arrStandortListe(Evaluate(Names.Item("StandortIndex") .
      RefersTo))
598
599 Dim intRowOffset As Integer
600 intRowOffset = 2
601
602 Dim strFarbe1 As String
603 Dim strFarbe2 As String
604 Dim strFarbeKS As String
605 Dim strFarbeKS2 As String
606 Dim strFarbeKS3 As String
607
608 Dim strFarbeER As String
609 Dim strFarbeER2 As String
610 Dim strFarbeER3 As String
611
612 strFarbe1 = RGB(202, 255, 112)
613 strFarbe2 = RGB(102, 119, 112)
614 strFarbeKS = RGB(255, 165, 0)
615 strFarbeKS2 = RGB(255, 80, 30)
616 strFarbeKS3 = RGB(255, 165, 30)
617
618 strFarbeER = RGB(84, 255, 159)
619 strFarbeER2 = RGB(0, 255, 127)
620 strFarbeER3 = RGB(0, 255, 80)
621
622 Dim n As Integer
623
624 With Worksheets(strAusgabeName).Select
625     n = 1
626     Call WriteArrayFormatiert(EggesER_Kum_kWh, intRowOffset, n, "
          EggesER_Kum_kWH [kwh]", strFarbe1)
627     n = n + 1
628     Call WriteArrayFormatiert(EggesKS_Kum_kWh, intRowOffset, n, "
          EggesKS_Kum_kWH [kwh]", strFarbe2)
629     n = n + 1
630     Call WriteArrayFormatiert(EdiffgenER_Kum_kWh, intRowOffset, n, "
          EdiffgenER_Kum_kWH [kwh]", strFarbe1)
631     n = n + 1
632     Call WriteArrayFormatiert(EdiffgenKS_Kum_kWh, intRowOffset, n, "
          EdiffgenKS_Kum_kWH [kwh]", strFarbe2)
633     n = n + 1
634     Call WriteArrayFormatiert(EdirgenER_Kum_kWh, intRowOffset, n, "
          EdirgenER_Kum_kWH [kwh]", strFarbe1)
635     n = n + 1
636     Call WriteArrayFormatiert(EdirgenKS_Kum_kWh, intRowOffset, n, "
          EdirgenKS_Kum_kWH [kwh]", strFarbe2)
637 '######neu
638     n = n + 1
639     Call WriteArrayFormatiert(EghorKS_Kum_kWh, intRowOffset, n, "Global
          horizontal KS [kwh]", strFarbeKS)
640     n = n + 1
641     Call WriteArrayFormatiert(EdiffhorKS_Kum_kWh, intRowOffset, n, "Diffuse
          horizontal KS [kwh]", strFarbeKS)
642     n = n + 1
643     Call WriteArrayFormatiert(EdirhorKS_Kum_kWh, intRowOffset, n, "Direkte
          horizontal KS [kwh]", strFarbeKS)

```

```

644 '######
645     n = n + 2
646     Call WriteArrayFormatiert(EggesKS_Kum_kWh, intRowOffset, n, "Global
647         geneigte KS [kwh]", strFarbeKS)
648     n = n + 1
649     Call WriteArrayFormatiert(EdiffgenKS_Kum_kWh, intRowOffset, n, "Diffuse
650         geneigte KS [kwh]", strFarbeKS)
651     n = n + 1
652     Call WriteArrayFormatiert(EdirgenKS_Kum_kWh, intRowOffset, n, "Direkte
653         geneigte KS [kwh]", strFarbeKS)
654     n = n + 1
655     Call WriteArrayFormatiert(Erefflgen_kum_kWh, intRowOffset, n, "Reflekt
656         [kwh]", strFarbeKS)
657 '######
658     n = n + 2
659     Call WriteArrayFormatiert(EghorKS_Kum_kWh, intRowOffset, n, "Global
660         horizontal KS [kwh]", strFarbeKS)
661     n = n + 1
662     Call WriteArrayFormatiert(EdiffhorER_Kum_kWh, intRowOffset, n, "Diffuse
663         horizontal [kwh]", strFarbeER)
664     n = n + 1
665     Call WriteArrayFormatiert(EdirhorER_Kum_kWh, intRowOffset, n, "Direkte
666         horizontal [kwh]", strFarbeER)
667 '######
668     n = n + 2
669     Call WriteArrayFormatiert(EggesER_Kum_kWh, intRowOffset, n, "Global
670         geneigte [kwh]", strFarbeER)
671     n = n + 1
672     Call WriteArrayFormatiert(EdiffgenER_Kum_kWh, intRowOffset, n, "Diffuse
673         geneigte [kwh]", strFarbeER)
674     n = n + 1
675     Call WriteArrayFormatiert(EdirgenER_Kum_kWh, intRowOffset, n, "Direkte
676         geneigte [kwh]", strFarbeER)
677     n = n + 1
678     Call WriteArrayFormatiert(Erefflgen_kum_kWh, intRowOffset, n, "DReflekt
679         [kwh]", strFarbeER)
680 End With
681 mOffsetAusgabeKum = n
682
683 With Worksheets(strAusgabeName).Range(Cells(intRowOffset - 1, 1), Cells(
684     intRowOffset - 1, n))
685     .Merge
686     .Value = Evaluate(Names.Item("Projektname").RefersTo) & " Standort: " &
687     strAnlageStandort
688     .HorizontalAlignment = xlCenter
689     .Interior.Color = RGB(70, 150, 255)
690     .Font.Size = 14
691 End With
692 End Sub
693
694 Sub TE_markieren()
695     '
696     ' TE_markieren Makro
697     '
698     Range("FC12:FC8771").Select
699
700 End Sub
701 Sub GS_markieren()
702     '
703     ' GS_markieren Makro
704     '

```

```

692     Range( "FE12:FE8771" ).Select
693
694 End Sub
695 Sub HS_markieren()
696 ,
697 ' HS_markieren Makro
698 ,
699     Range( "FF12:FF8771" ).Select
700
701 End Sub
702 Sub DS_markieren()
703 ,
704 ' DS_markieren Makro
705 ,
706     Range( "FG12:FG8771" ).Select
707
708 End Sub
709
710 Sub CopyPaste1_markieren()
711 ,
712 ' CopyPaste1_markieren Makro
713 ,
714     Range( "FI12:FI8771" ).Select
715
716 End Sub
717
718 Sub CopyPaste2_markieren()
719 ,
720 ' CopyPaste2_markieren Makro
721 ,
722     Range( "FJ12:FJ8771" ).Select
723
724 End Sub
725 Sub CopyPaste3_markieren()
726 ,
727 ' CopyPaste3_markieren Makro
728 ,
729     Range( "FK12:FK8771" ).Select
730
731 End Sub
732 Sub BestimmteSpalte_markieren()
733 ' BestimmteSpalte_markieren Makro
734 Dim Spalte As String
735 Spalte = InputBox("Bestimme Spalte Markieren", "Bitte geben sie die Spalte
    ein die markiert werden soll!")
736 If Spalte = "" Then
737     Exit Sub
738 End If
739 If IsNumeric(Spalte) = True Then
740     Exit Sub
741 End If
742     Range(Spalte & "12:" & Spalte & "8771").Select
743 End Sub

```

Listing E.6: Modul Kollektor Kapitel 7

# Abkürzungsverzeichnis

**VBA** Visual Basic for Applications

**Excel** Microsoft Excel

**TES** Thermischer Energie Speicher

**TESS** Thermischer Energie Schichten Speicher

**BMT** Bundesministerium für Bauten und Technik

**MS** Microsoft

**KS** Klimadatensatz

**ER** Errechnet

**EHF** Einstrahlungsmodell auf die horizontale Fläche

**EGF** Einstrahlungsmodell auf die geneigte Fläche

# Abbildungsverzeichnis

2.1. Schematische Einteilung der Solaranlage in die Teilsysteme . . . . .	3
2.2. Skizze des Verlaufs der Sonnenbahn . . . . .	4
2.3. Azimut . . . . .	5
2.4. Sonnenhöhe . . . . .	5
2.5. Schematische Darstellung der Sonneneinstrahlung auf einen Kollektor . . . . .	6
2.6. Schematische Darstellung der Globalstrahlung . . . . .	7
2.7. Energiefluss durch den Kollektor mit optischen und thermischen Verlusten[2]	8
3.1. Prozentuelle Aufteilung diffuser Strahlung . . . . .	12
3.2. Darstellung des Einfall- und Moduleinfallswinkel . . . . .	13
3.3. Winkelfaktor $K_{\theta_{gen}}$ eines Solarkollektors in Abhängigkeit vom Einfallwinkel	18
3.4. Wirkungsgradkurven $\eta_{Kollektor}$ eines Solarkollektors in Abhängigkeit vom $\Delta T$ und Einstrahlung auf den Kollektor . . . . .	19
3.5. Wirkungsgradverlauf $\eta_{Kollektor}$ eines Solarkollektors mit Bereichseinteilung	19
4.1. Benutzeroberfläche im MS Excel Solarsimulation . . . . .	21
4.2. Tabellenblatt nach Programmstart . . . . .	21
4.3. Tabellenblatt Klimadaten - Vergleichsstandort . . . . .	22
4.4. $\alpha_E$ ...Azimut des Kollektors, $\gamma_E$ ...Höhe des Kollektors, $\alpha_S$ ...Azimut der Sonne, $\gamma_S$ ...Sonnenhöhe . . . . .	25
4.5. Überblick des VBA-Programmes und der Entwicklungsumgebung . . . . .	26
4.6. Struktogramm des VBA-Programmes nach klicken auf <b>Berechne</b> . . . . .	29
C.1. Gegenüberstellung horizontaler zu geneigter Globalstrahlung . . . . .	35
C.2. Aufteilung Direkt-, Reflexions- und Diffusstrahlung . . . . .	35
C.3. Jahresgang der Globalstrahlung auf geneigte Flächen . . . . .	36
C.4. Sonnenstand mit Scrollbalken . . . . .	36
C.5. Originale Werte der Globalstrahlung . . . . .	37
C.6. Verbesserte Werte der Globalstrahlung . . . . .	37
C.7. Sonnendeklination . . . . .	38
C.8. Zeitgleichung . . . . .	38
C.9. Vergleich der Berechnungsmodelle 1,2,3 des Sonnenstandes . . . . .	39
D.1. Solarsimulation.xlsx Benutzeroberfläche . . . . .	41

# Literaturverzeichnis

- [1] DIETRICH WERTZ: *Solerstrahlungsdaten verschiedener „Osterreichischer Standorte, bereitgestellt vom Institut für Energietechnik und Thermodynamik*
- [2] DIETRICH WERTZ: *Seminararbeit: Beschreibung einer Modernen Solaranlage.* Jan 2006
- [3] EUROPÄISCHES KOMITEE FÜR NORMUNG: *EN ISO DIN 9488 - Sonnenenergie Vokabular*, Okt 1999
- [4] FABIENNE LANINI: *Division of Global Radiation into Direct Radiation and Diffuse Radiation*, University of Bern, Diss., 2010
- [5] MALTE CHRISTIAN THOMA: *Optimierte Betriebsführung von Niederspannungsnetzen mit einem hohen Anteil an dezentraler Erzeugung*, EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE ZÜRICH, Diss., 2007
- [6] MANFRED KLEEMANN, MICHAEL MELISS: *Regenerative Energiequellen*. Springer Verlag, 1993
- [7] MARTIN ORTBAUER: *Modellierung eines Wärmeschichtspeichers mit Solareinbindung in EXCEL/VBA* Untertitel: *Teil thermischer Schichtspeicher*, Okt 2011
- [8] NORMENAUSSCHUSS LICHTTECHNIK (FNL) IM DEUTSCHEN INSTITUT FÜR NORMUNGEN E. V.: *DIN 5034 Teil 2 - Tageslicht in Innenräumen* , Feb 1985
- [9] NORMENAUSSCHUSS HEIZ- UND RAUMLUFTTECHNIK (NHRS) IM DIN: *DIN EN 12975 Thermische Solaranlagen und ihre Bauteile - Kollektoren - Teil 2: Prüfverfahren*, Jun 2006
- [10] PEARSON SOFTWARE CONSULTING: *Your Source For Excel Development*. <http://www.cpearson.com/Excel/MainPage.aspx>
- [11] REINDL, D.T.; BECKMAN, W.A., DUFFIE, J.A: *Diffuse Fraction Correlations*. Tagungsband ISES Solar World Conference 1989, S. 2082-2086
- [12] ROBERT WALRAVEN: *Calculation the position of the sun.* 1978. – pp. 393-397.
- [13] RON DE BRUIN: *Change the Ribbon in Excel 2007 or Excel 2010* . <http://www.rondebruin.nl/ribbon.htm>
- [14] VIKTOR WESSELAK, THOMAS SCHABBACH: *Regenerative Energietechnik*. Springer Verlag, Berlin Heidelberg, 2009. – ISBN 978-3-540-95881-9
- [15] VOLKER QUASCHNING: *Beispiel Sonnenstand* <http://www.f1.htw-berlin.de/studiengang/ut/downloads/unterlagen/M3UT-MoSi/index.html>
- [16] VOLKER QUASCHNING: *Simulation der Abschattungsverluste bei solarelektrischen Systemen*. 1996. – ISBN 3-89574-191-4

- [17] VOLKER QUASCHNING: *Regenerative Energiesysteme* . Hanser Verlag München, 2008 ( 5. Auflage). – ISBN 978-3-446-40973-6
- [18] WALTER HEINDL, ALFRED SIGMUND, KLAUS KREC: *Klimadatenkatalog - Bundesministerium für Bauten und Technik Staatlicher Hochbau Allgemeines, Grundlagen(theorie), Sonneneinstrahlungsdaten, Allgemeine Ergänzungen zur Lufttemperatur, Näheres zur Lufttemperatur für 10 Standorte Österreichs.* A-1010 Wien : Leykam Universitätsbuchdruckerei
- [19] WIKIPEDIA: *Milankovic-Zyklen*. <http://de.wikipedia.org/wiki/Milankovi%C4%87-Zyklen>