**Contents:**

* Struktur von Arbeitspaket01

\* Weatherdateneinlesen

\* Darstellung und Analyse von Ergebnisse

* Kollektormodell in openmodelica erstellen

\* Icon und Model einstellen

\* Model von Solarkollektor einschreiben und Eingabeswerten importieren

\* Model von Weatherdaten einschreiben und Dateneinlesen

\* Hauptprogramme einschreiben und verbinden alle Komponenten

* **Um die Simulation von ‘Solarkolektor.mo‘ zu laufen**

**\* Herunterladen** **und Speichern**

**\* Arbeitsverzeichnis einzustellen**

**\* Simulation zu laufen**

* Ergebnisse von Kollektormodell als mat-file output
* Ergebnisse von openmodelica in Matlab zu analysieren

\* Reihenfolge und Name von Output-Varianten zu analysieren

\* Ergebnisse von openmodelica einlesen

\* Dateneingabe

\* Zuordnung der Ergebnisse

\* Solarertrag-Bildung von Monatswerten

* **Struktur von Arbeitspaket01**

Arbeitspaket 01 kann in zwei Teilen geteilt werden：Weatherdateneinlesen und

* **Weatherdateneinlesen**

**Germany\_Potsdam\_2019\_DWD.txt**

**Potsdam2019\_xls.xls**

**Solarkollektor.Datasource**

**Potsdam2019\_mat.mat**

* **Analyse und Darstellung von Ergebnissen**

**‘Solarkollektor.Hauptprogramm****\_res.mat’**

**in ‘Solarkollektor.Hauptprogramm\_output’**

**Solarkollektor.mo**

**Kollektormodell in openmodelica einstellen**

**Reihenfolge\_M01.mat**

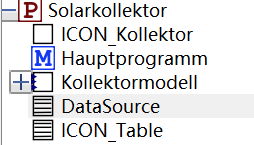
**OManalyse01\_mat.mat**

**Analyse von Ergebnissen**

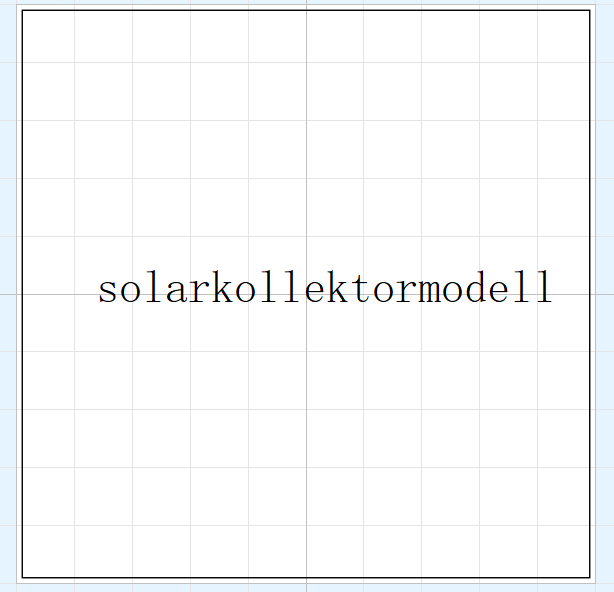
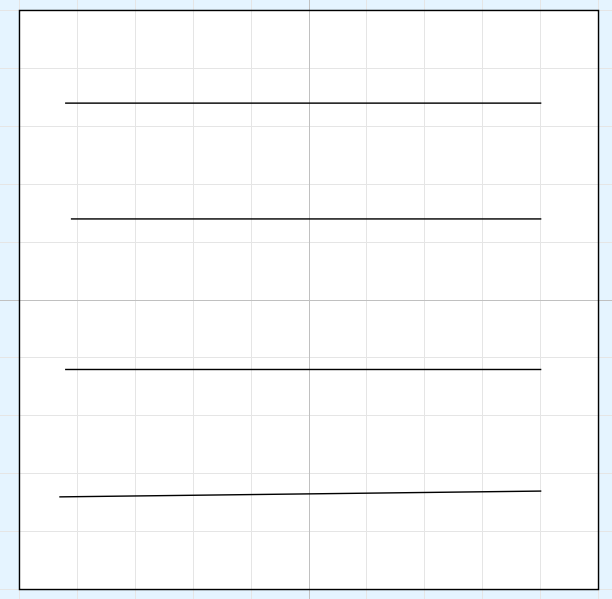
**Zuordnung von Output-Varianten**

* **Icon und Model einstellen**

**Zuerst ICON von Solarkollektormodell und Weatherdaten erstellen in Package Solarkollektor. ICON von Solarkollektormodell ist als ‘ICON\_Kollektor‘ und ICON von Weatherdaten ist als ‘ICON\_Table‘ bezeichnet. Danach das Model von Solarkollektor heißt ‚Kollektormodell‘ und das Model von Weatherdaten heißt ‚DataSouce‘ einstellen. Zuletzt das Model von Hauptprogramm einstellen.**

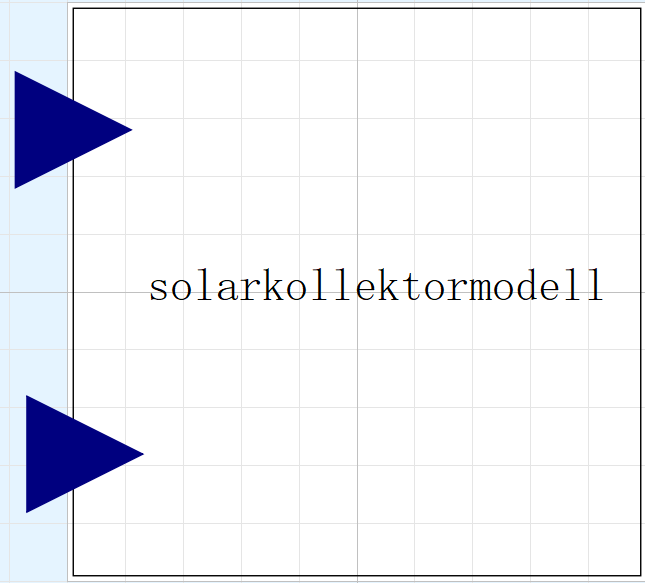


**ICON von Solarkollektormodell und Weatherdaten sind sowie folgendes:**

ICON von Solarkollektormodell ICON von Weatherdaten

**Bei Kollektormodel: Kuppeln das Kollektormodel mit Block ‘Input‘ . ‘Input‘ sind diese blaue Dreiecke. Diese ‘Input‘ sind ‘Gt‘, ‘Ta‘.**



* **Model von Solarkollektor einschreiben und verbinden mit Icon**

**Zuerst verbinden das Model mit ICON, danach beschreiben das model von Solarkollektor. Die Code für die Beschreibung ist sowie folgendes:**

model Kollektormodell

extends ICON\_Kollektor;**“Verbindung von model und ICON“**

**“Definition der Kollektorparameter ‘Types’,‘parameters‘, ‘Variables‘”**

//Types

type HeatLossCoeficient = Real(unit = "w/(m2\*K)");

type Temperature = Real(unit = "deg");

type HeatCapacity = Real(unit = "J/(m2\*K)");

type area = Real(unit = "m2");

type MassFlow = Real(unit = "kg/h");

//parameters

parameter area A\_g = 2.0 "gross area";

parameter Real eta = 0.8 "top efficiency";

parameter MassFlow m\_dot = 75.0 "mass flow";

parameter HeatLossCoeficient a\_1 = 3.5;

parameter HeatLossCoeficient a\_2 = 0.01;

parameter HeatCapacity a\_5 = 5000;

parameter Temperature T\_ci = 50 "collector entry temperatur";

//Varialbes

Real Tco "collector outlet temperature";

Real Tm "middle temperature";

Real cp "specificheat";

Real Qabs "absorbs amout of heat";

Real Quse "using energy";

Real Qvp "Lost Energy";

Real Qv "lost energy";

**“Verbindung das Kollektormodell mit Inputsblock von Eingabewerten von ‚‘GT‘,‘Ta‘”**

Modelica.Blocks.Interfaces.RealInput Gt annotation(

Placement(visible = true, transformation(origin = {-98, 56}, extent = {{-20, -20}, {20, 20}}, rotation = 0), iconTransformation(origin = {-98, 56}, extent = {{-20, -20}, {20, 20}}, rotation = 0)));

Modelica.Blocks.Interfaces.RealInput ta annotation(

Placement(visible = true, transformation(origin = {-94, -56}, extent = {{-20, -20}, {20, 20}}, rotation = 0), iconTransformation(origin = {-94, -56}, extent = {{-20, -20}, {20, 20}}, rotation = 0)));

**“Gleichungen beschreiben“**

equation

Qabs = eta\*Gt\*A\_g;

Qvp = (a\_1\*(Tm - ta) + a\_2\*(Tm - ta)\*(Tm - ta))\*A\_g;

Quse = m\_dot\* cp\*(Tco - T\_ci)/3.6;

der(Tm) = (Qabs - Qvp - Quse) / (A\_g \* a\_5);

Tm = (T\_ci + Tco) / 2;

cp = 4.21091532 - 1.85646462e-3 \* Tm + 3.09300765e-5 \* Tm \* Tm - 1.22631409e-7 \* Tm \* Tm \* Tm;

Qv=-Qvp;

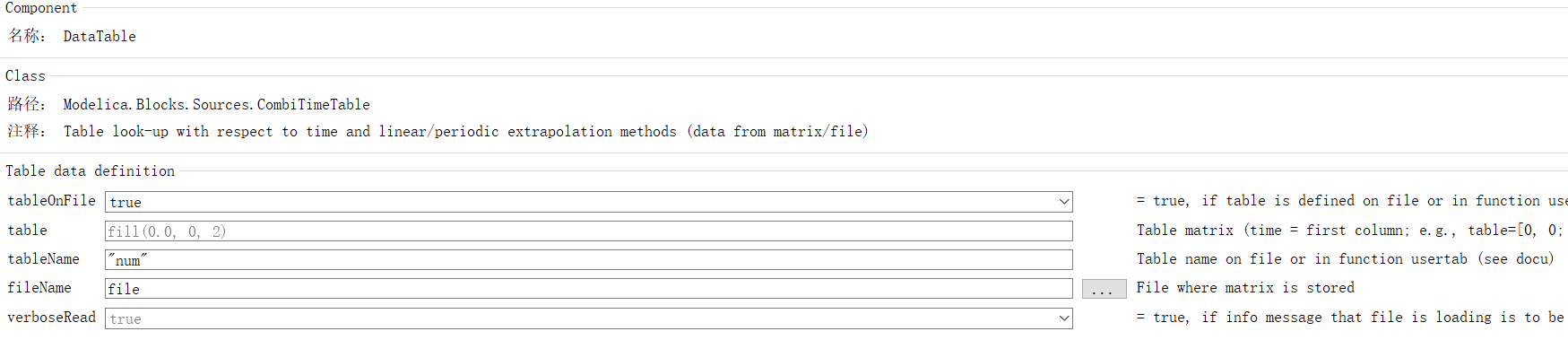
end Kollektormodell;

* **Model von Weatherdaten einschreiben und Dateneinlesen**

**Zuerst Stellen eine DataTable in das Ansichtschema von Model der Datasoure**



**Nennen wir die Filename von DataTable als ‚file‘ , und die TableName ist als ‚num‘ genannt.**



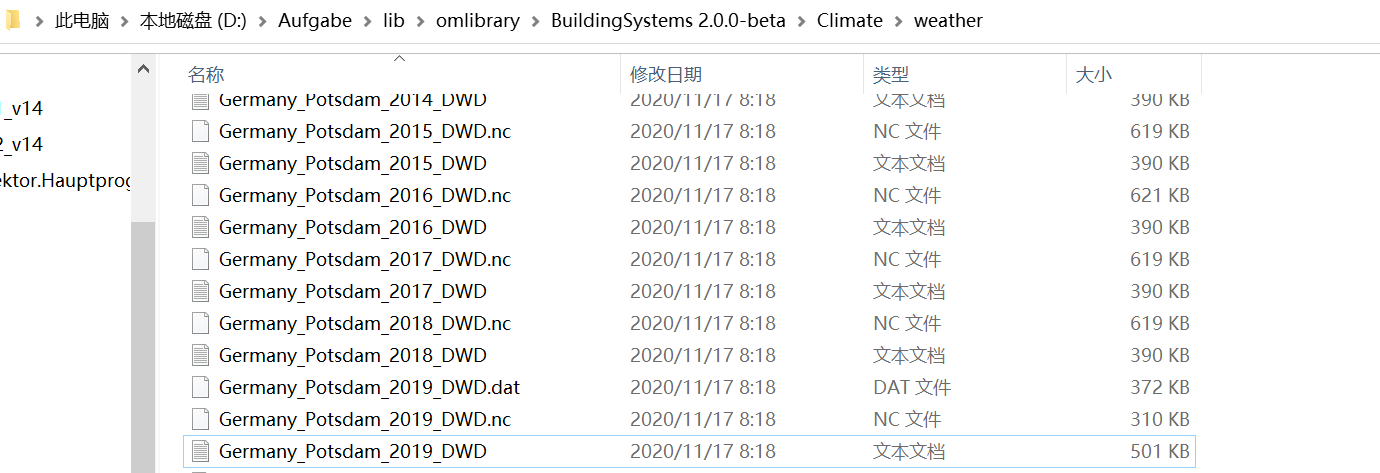
**‚file‘ bezeichnet die Weatherdatenquelle in Aktendeckel von Openmodelica Arbeitsverzeichnis ‘Modell01\_v15‘. Diese Datenquelle heißt** **‚Potsdam2019\_mat.mat‘ .**

**Weatherdatenquelle :**

**‚****Germany\_Potsdam\_2019\_DWD.txt’** **‚Potsdam2019\_xls.xlsx‘**

**‚Potsdam2019\_mat.mat‘**

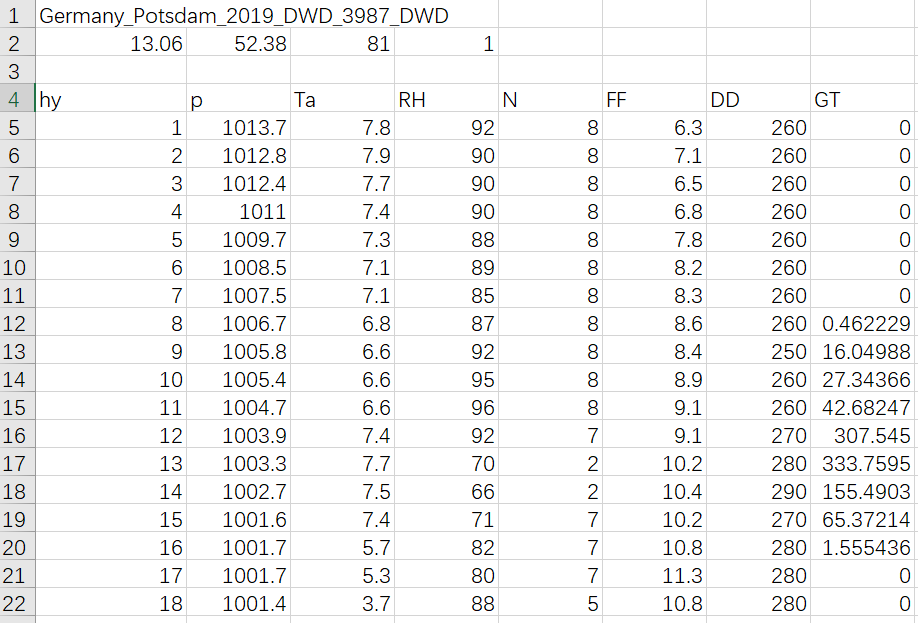
**‚Germany\_Potsdam\_2019\_DWD.txt’ bleibt in die Bibliothek von OpenModelica . Es zeigt die Weathersituation von Potsdam im Jahr 2019.**



**‚Potsdam2019\_xls.xlsx‘ ist von ‚Germany\_Potsdam\_2019\_DWD.txt’ umgewandelt.**

**In die erste Spalte von diesen Tabelle bezeichnet die Zeit in Uhr. Die zweite Spalte zeigt den Umgebungdruck. Die tertiäre Spalte bezeichnet die äußere Temperatur oder Lufttemperatur. Die vierte Spalte zeigt die relative Luftfeuchtigkeit. Die fünfte Spalte bezeichnet die Windrichtung. Die sechste Spalte zeigt die Windgeschwindigkeit. Die siebte Spalte zeigt die Wolkendeckekoffizient.**

**Die letzte Spalte zeigt die Globalstrahlung auf geneigte Fläche. GT kommt nicht aus ‚Potsdam2019\_xls.xlsx‘, sondern aus die Ergebnissedaten von GT in Modell02.**

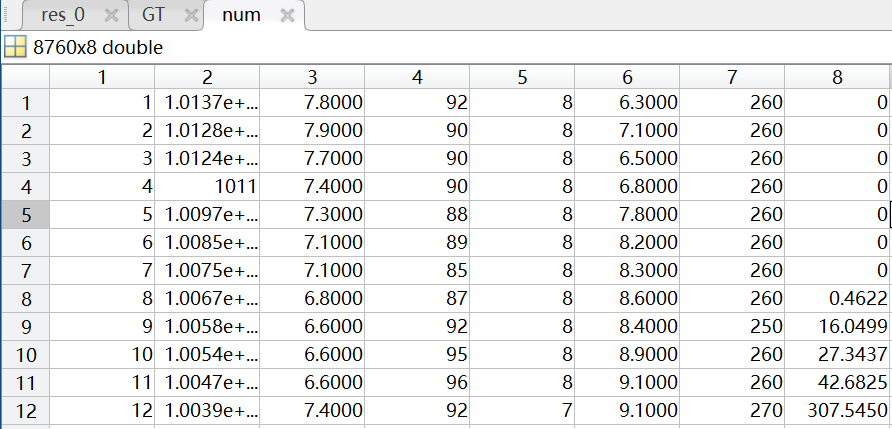


**‚Potsdam2019\_mat.mat‘ ist von ‚Potsdam2019\_xls.xlsx‘ umgewandelt.**

**Wir geben den folgenden Code in Kommandozeile von Matlab ein.**



**Danach bekommen wir die folgenden Tabelle, die TableName ist ‚num‘.**



**Danach beschreiben das Model von Weatherdaten. Um die Weatherdaten einzulesen, benutzt man die Datenquelle ‘****Potsdam2019\_mat.mat‘ und beschreibt file=** **‘modelica://Solarkollektor/ Potsdam2019\_mat.mat‘ .**

**‘modelica://Solarkollektor/‘ bedeutet, diese Datenquelle in Package ‘Solarkollektor‘ verwendet werden. Und die Datenquelle ‘Potsdam2019\_mat.mat‘ bezeichnet die Weatherdatenquelle in das OM-Arbeitsverzeichnis. Danach kann diese mat-Datei von weatherdaten in model ‘DataSource‘ und model ‘Hauptprogramme‘ eingeleset werden.**

model DataSource

extends ICON\_Table; **“Verbindung von model und ICON“**

import SI = Modelica.SIunits;

**“Einschreibung von Datenquelle”**

parameter String file = "modelica://Solarkollektor/Potsdam2019\_mat.mat";

**“Definition der Ausgabewerten von DataSouce“**

output SI.Irradiance G;

output Real Ta;

**“Verbindung das model mit DataTable”**

protected

Modelica.Blocks.Sources.CombiTimeTable DataTable( columns = 1:8, fileName = file, smoothness = Modelica.Blocks.Types.Smoothness.ContinuousDerivative, tableName = "num",tableOnFile = true, timeScale = 3600) annotation(

Placement(visible = true, transformation(origin = {-62, 62}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));

**“Beschreibung der Gleichung für die Ausgabewerten”**

equation

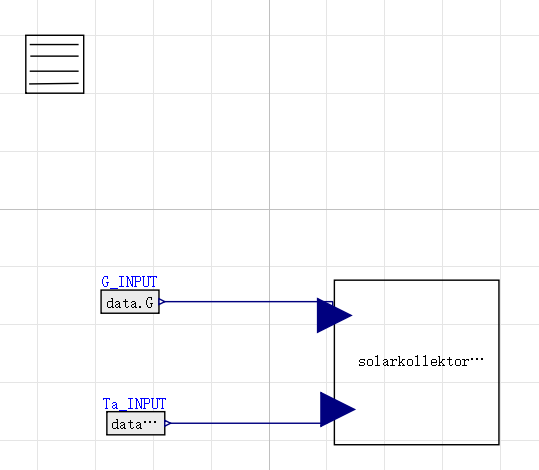
Ta = DataTable.y[3];

G= DataTable.y[8];

end DataSource;

* **Hauptprogramme einschreiben und verbinden alle Komponenten**

**Zuletzt beschreiben Hauptprogramme, verbinden das Model von Solarkollektor und das Model von Weatherdaten mit die Block‘Input‘.**



model Hauptprogramm

parameter String file\_weather = Modelica.Utilities.Files.loadResource("modelica://Solarkollektor/Potsdam2019\_mat.mat"); **“Beschreibung der Datenquelle”**

**“Verbindung mit allen Komponenten”**

Solarkollektor.Kollektormodell kollektormodell annotation(

Placement(visible = true, transformation(origin = {51, -53}, extent = {{-29, -29}, {29, 29}}, rotation = 0)));

Solarkollektor.DataSource data(file = file\_weather) annotation(

Placement(visible = true, transformation(origin = {-74, 50}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));

Modelica.Blocks.Sources.RealExpression Ta\_INPUT(y = data.Ta) annotation(

Placement(visible = true, transformation(origin = {-46, -74}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));

Modelica.Blocks.Sources.RealExpression G\_INPUT(y = data.G) annotation(

Placement(visible = true, transformation(origin = {-48, -32}, extent = {{-10, -10}, {10, 10}}, rotation = 0)));

equation

connect(Ta\_INPUT.y, kollektormodell.ta) annotation(

Line(points = {{-34, -74}, {20, -74}, {20, -70}, {24, -70}}, color = {0, 0, 127}));

connect(G\_INPUT.y, kollektormodell.Gt) annotation(

Line(points = {{-36, -32}, {22, -32}, {22, -36}}, color = {0, 0, 127}));

annotation(

experiment(StartTime = 0, StopTime = 3.1536e+07, Tolerance = 1e-06, Interval = 3600)); **“SimulationsZeit einstellen”**

end Hauptprogramm;

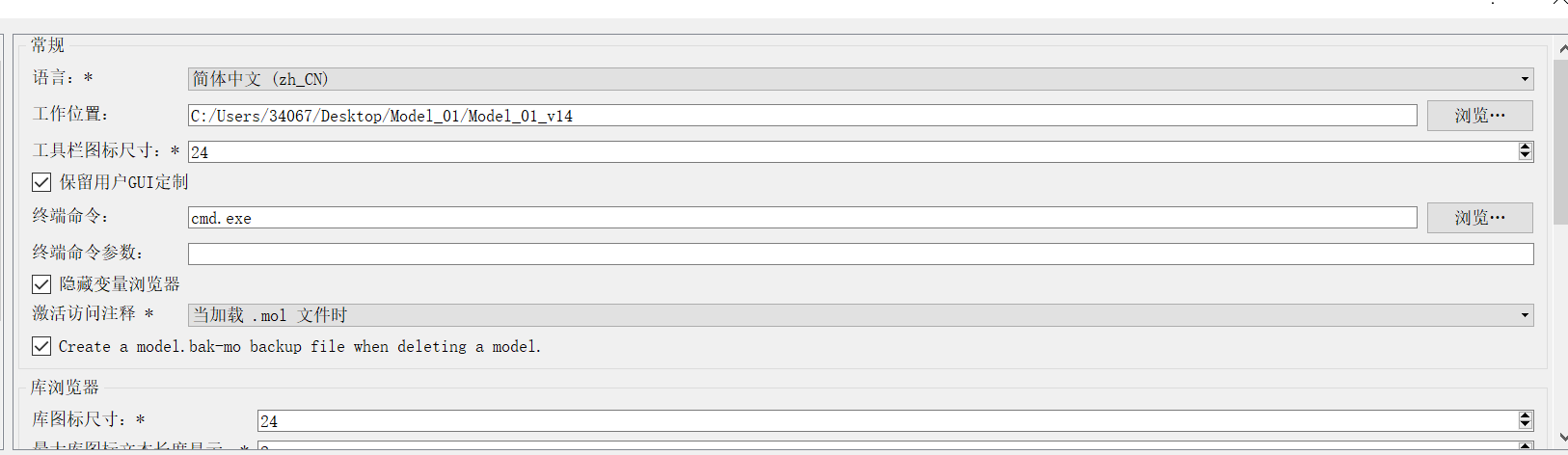
* **Um die Simulation zu laufen:**
* **Herunterladen und Speichern**

**Zuerst herunterladen und speichern Sie diese ‚Modell01\_v15.zip‘ an der Desktop oder andere Speicherverzeichnis, danach öffnen Sie diese Solarkollektor.mo.**

* **Arbeitsverzeichnis einzustellen**

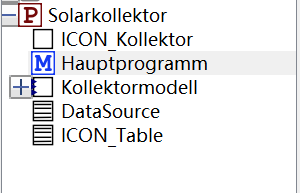
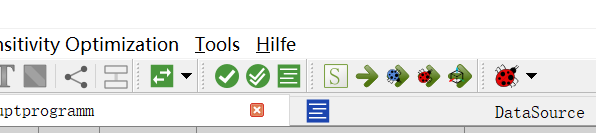
**Zweite klicken Sie ‘Tools‘, danach ‘Optionen‘ , um das Arbeitsverzeichnis einzustellen.**

**Dritte klicken Sie bitte ‘Durchführen‘ , und wählen Sie bitte Aktendeckel ‘Modell01\_v15‘ .**



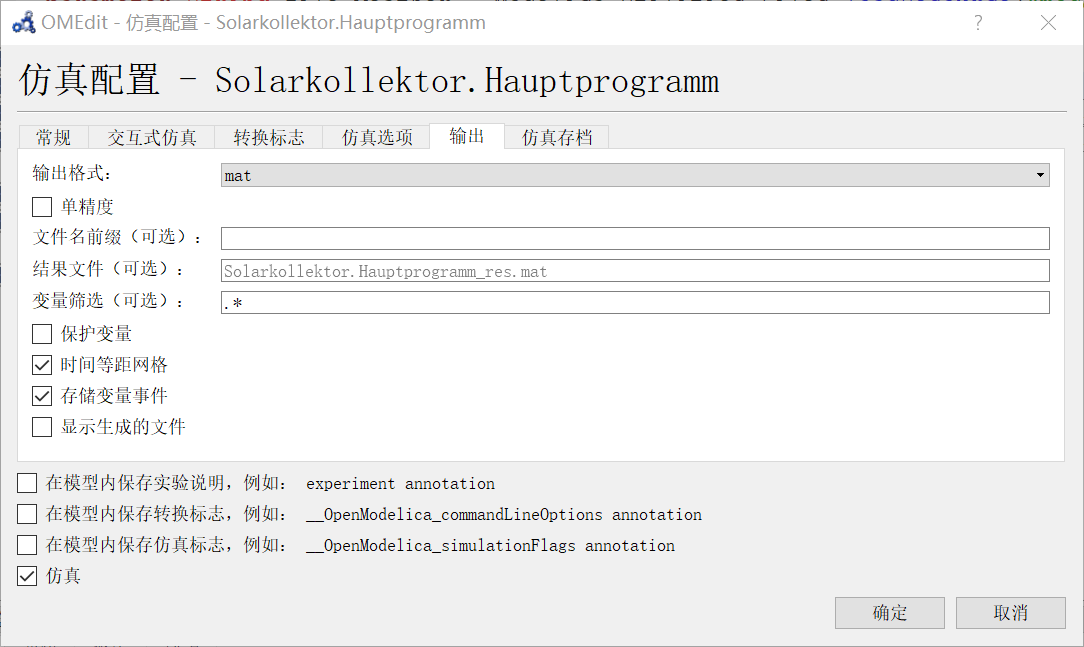
* **Simulation zu laufen**

**Zuletzt klicken ‘Hauptprogramme‘, danach ‘S‘ ( ‚S‘ bedeutet ‚Simulation‘ ), dann beginnen die Simulation.**

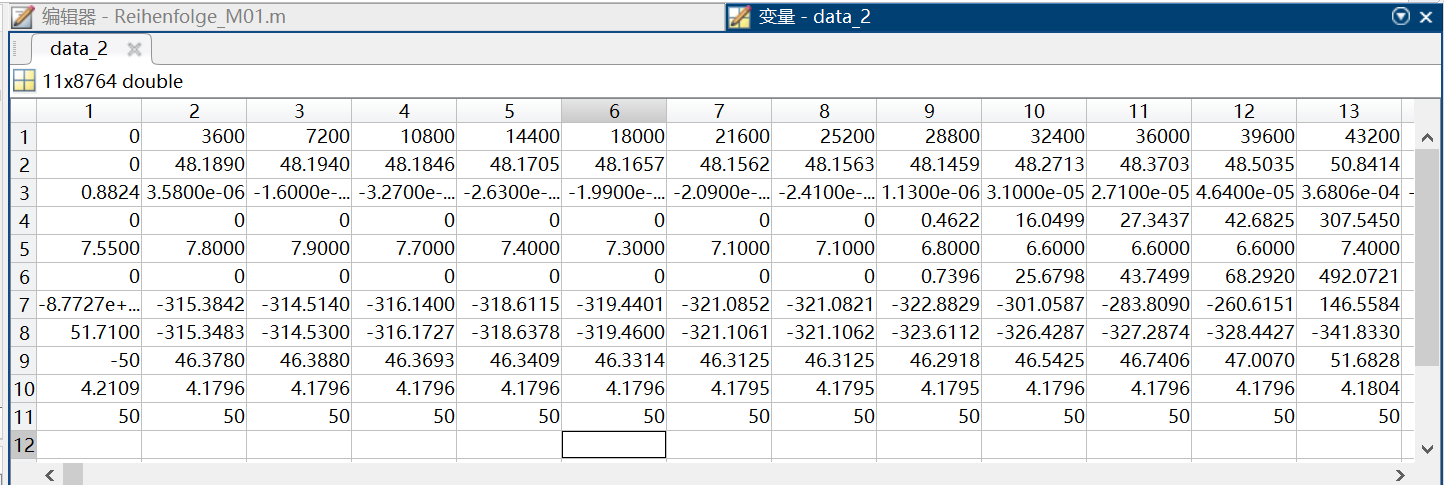
* **Ergebnisse von Kollektormodell als mat-file output**

**Klicken Sie bitte ‘S‘ und ‘Ausgebe‘. Hier habe ich die Mat-file in ‘Solarkollektor.Hauptprogramm\_output‘ von ‚Modell01\_v15‘ gespeichert, um die Ergebnissetabelle von Openmodelica in MatData einzuspeichern.**



* **Ergebnisse von Openmodelica in Matlab zu analysieren**
* **Reihenfolge und Name von Output-Varianten zu analysieren**

**Ausgegebene Varianten sind wie folgende Bild gezeigt.**



**Diese Varianten zeigen keine Namen.Und wir können nicht von diese Tabelle wissen, welche Varianten welche Reihenfolge haben. Um die Reihenfolge und Name von Output-Varianten zu analysieren benutzen wir die Matlab-Code ‚Reihenfolge\_M01,mat‘ . Die Matlab-Code sind sowie folgende:**

clc

clear all

%%

% load Sim\_out.mat % individueller Name derAusgabedatei, alle Var: .\*

load 'Solarkollektor.Hauptprogramm\_output/Solarkollektor.Hauptprogramm\_res.mat' % alternativ

%% Datenaufbereitung

[r,c] = size(data\_2);

%% Header

col1 = string(name(:,1:r)');

col2 = string(description(:,1:r)');

data = (data\_2);

res\_0 = table(col1,col2,data);

res\_0.Properties.VariableNames{1} = 'Bez\_var';

res\_0.Properties.VariableNames{2} = 'OM\_var';

**Dann bekommen wir die Reihenfolge und Namen von Varianten.**

* **Ergebnisse von Openmodelica einlesen:**

**Einschreiben die Matlab-Code um die Simulationsergebnissen von OM in Matlab zu analysieren. Diese Code sind in ‚OManalyse01\_mat.mat‘ gespeichert.**

clc

load Solarkollektor.Hauptprogramm\_res.mat **“Einlesen die Weatherdaten in MatData”**

data=data\_2;

%%

**”Definition des Parameters”**

**(****Reihenfolge Von Varianten ist durch ‚Reihenfolge\_M01.mat‘ zu bekommen)**

ts = data(1,:);

Tm = data(2,:);

derTm = data(3,:);

Ta = data(5,:);

G = data(4,:);

Qabs = data(6,:);

Quse = data(7,:);

Qver = data(8,:);

Tco = data(9,:);

* **Dateneingabe**

T\_ci=50;

th = ts/3600;

Quse(Quse<0)=0;

ctr=Quse./Quse;

ctr(isnan(ctr))=0;

* **Zuordnung der Ergebnisse**

figure(1)

plot(th,Tco.\*ctr,'.r');

hold on

plot(th,Tm.\*ctr,'.m')

plot(th,Ta,'.b');

hold off

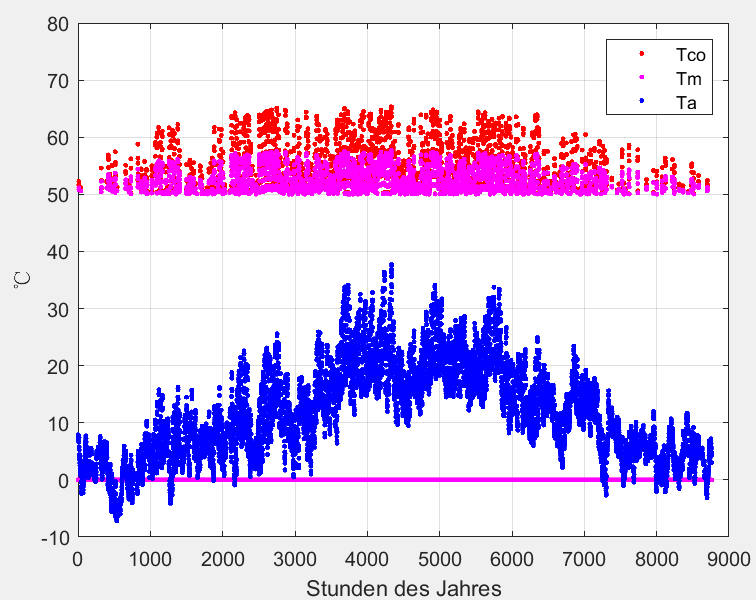
grid

legend('Tco','Tm','Ta')

xlabel('Stunden des Jahres')

ylabel('¡æ')

ylim([-10 80])



figure(2)

subplot(2,1,1)

plot(ts/3600 ,G.\*ctr ,'.-r');

hold on

plot(ts/3600 ,Qabs/2.\*ctr,'.-b');

hold off

legend('GT','Qabs/2')

grid

ylabel('w/m^2')

subplot(2,1,2)

plot(ts/3600 ,Qver/2.\*ctr,'.-r');

hold on

plot(ts/3600 ,Quse/2.\*ctr,'.-g');

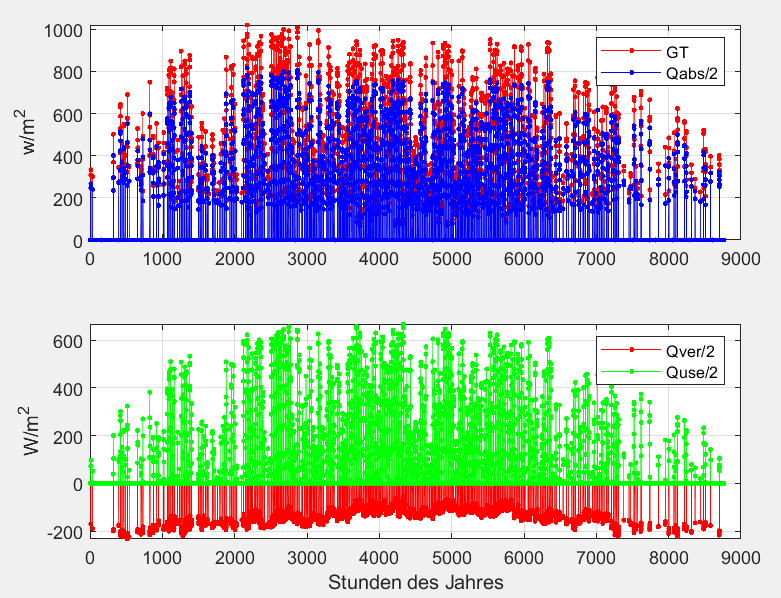
hold off

grid

legend('Qver/2','Quse/2')

xlabel('Stunden des Jahres')

ylabel('W/m^2')



* **Solarertrag-Bildung von Monatswerten**

Quse=Quse';

QuseD=sum(reshape(Quse(1:8760,1),[24,365]))./1000;

QuseM(1,1)=sum(QuseD(1:31));

QuseM(1,2)=sum(QuseD(32:59));

QuseM(1,3)=sum(QuseD(60:90));

QuseM(1,4)=sum(QuseD(91:120));

QuseM(1,5)=sum(QuseD(121:151));

QuseM(1,6)=sum(QuseD(152:181));

QuseM(1,7)=sum(QuseD(182:212));

QuseM(1,8)=sum(QuseD(213:243));

QuseM(1,9)=sum(QuseD(244:273));

QuseM(1,10)=sum(QuseD(274:304));

QuseM(1,11)=sum(QuseD(305:334));

QuseM(1,12)=sum(QuseD(335:365));

%%

figure(3)

title 'Monatssummen'

bar(QuseM,'grouped');

grid

legend(['T{ci}=',num2str(Tci),'¡æ'])

xlim([1 12])

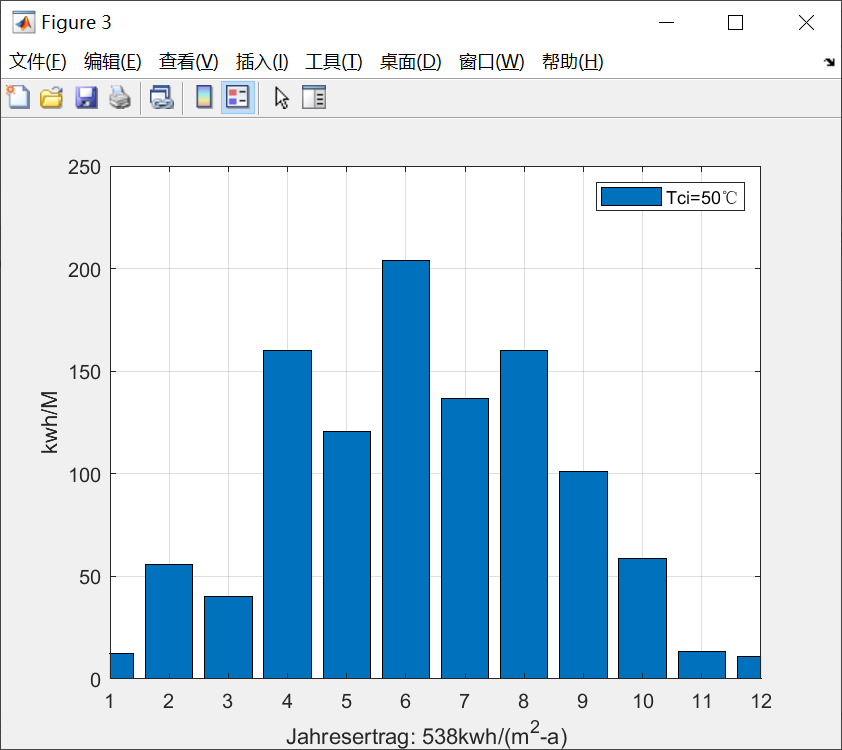
q\_sol=sum(QuseM)/2;

Tmm=mean(Tm);

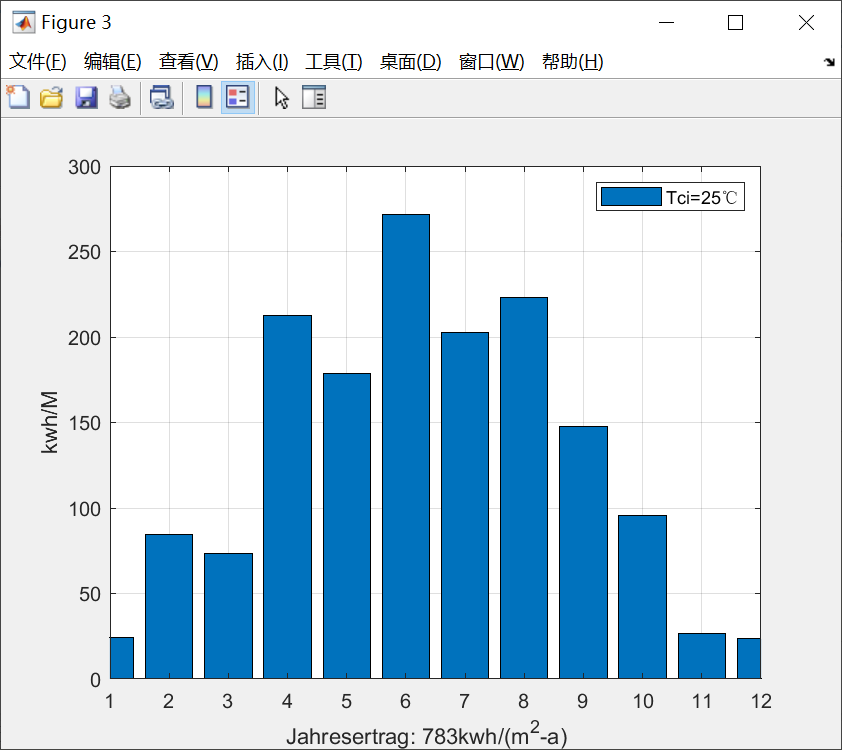
xlabel(['Jahresertrag: ',num2str(ceil(q\_sol)),...

'kwh/(m^2-a) Mitteltemp.: ',num2str(ceil(Tmm)),'¡æ'])

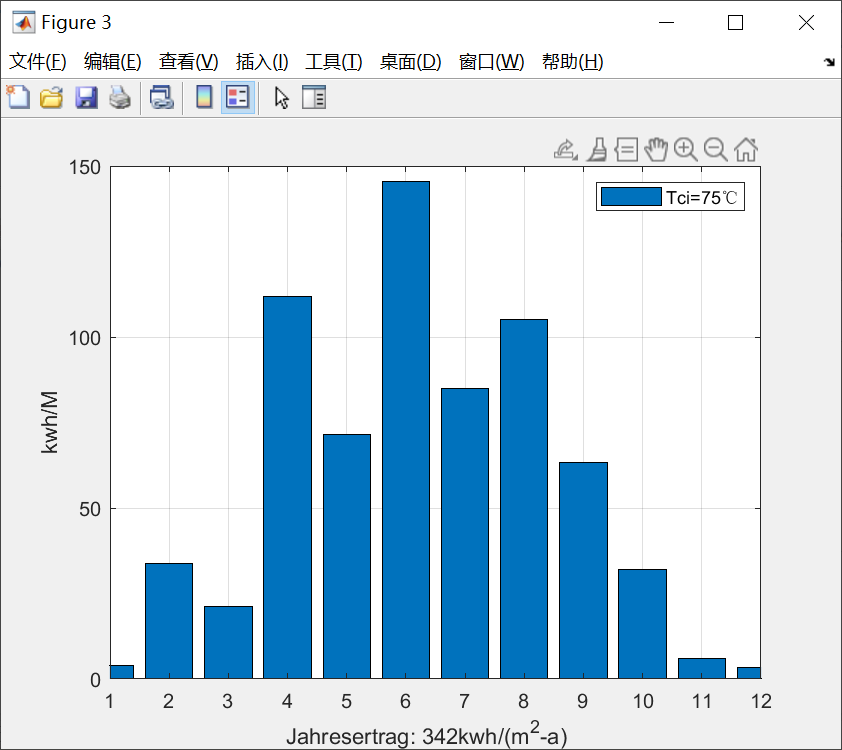
ylabel('kwh/M')



Wenn Tci=25 :



Wenn Tci=75 :



**Folgerung:**

**Je kleiner die Eintrittstemperatur von Kollektor ist, ist der Jahhresertrag je größer.**