A review of PV, solar thermal, and PV/thermal collector models in TRNSYS

A Report of IEA SHC - Task 35

PV/Thermal Solar Systems

Report DB1

November 13, 2009

Mike Collins

Nov 2009

SOLAR HEATING & COOLING PROGRAMME
INTERNATIONAL ENERGY AGENCY

A review of PV, solar thermal, and PV/thermal collector models in TRNSYS

by

Mike Collins

A technical report of Subtask B
Report DB1

Mike Collins
Dept. of Mechanical Engineering
University of Waterloo
Waterloo, Ontario
Canada
N2L 3G1
Phone: 1.519.888.4567x33655
E-mail: mcollins@uwaterloo.ca

Nov 2009

Contents

IEA Solar Heating and Cooling Programme

The *International Energy Agency* (IEA) is an autonomous body within the framework of the Organization for Economic Co-operation and Development (OECD) based in Paris. Established in 1974 after the first "oil shock," the IEA is committed to carrying out a comprehensive program of energy cooperation among its members and the Commission of the European Communities.

The IEA provides a legal framework, through IEA Implementing Agreements such as the *Solar Heating and Cooling Agreement*, for international collaboration in energy technology research and development (R&D) and deployment. This IEA experience has proved that such collaboration contributes significantly to faster technological progress, while reducing costs; to eliminating technological risks and duplication of efforts; and to creating numerous other benefits, such as swifter expansion of the knowledge base and easier harmonization of standards.

The *Solar Heating and Cooling Programme* was one of the first IEA Implementing Agreements to be established. Since 1977, its members have been collaborating to advance active solar and passive solar and their application in buildings and other areas, such as agriculture and industry. Current members are:

Australia	Finland	Portugal
Austria	France	Spain
Belgium	Italy	Sweden
Canada	Mexico	Switzerland
Denmark	Netherlands	United States
European Commission	New Zealand	
Germany	Norway	

A total of 44 Tasks have been initiated, 33 of which have been completed. Each Task is managed by an Operating Agent from one of the participating countries. Overall control of the program rests with an Executive Committee comprised of one representative from each contracting party to the Implementing Agreement. In addition to the Task work, a number of special activities—Memorandum of Understanding with solar thermal trade organizations, statistics collection and analysis, conferences and workshops—have been undertaken.

To find Solar Heating and Cooling Programme publications and learn more about the Programme visit
www.iea-shc.org or contact the SHC Executive Secretary, Pamela Murphy, e-mail: pmurphy@kmgrp.net.

Nov 2009

The Tasks of the IEA Solar Heating and Cooling Programme, both underway and completed are as follows:

Current Tasks & Working Group:

Task 35	*PV/Thermal Solar Systems*
Task 36	*Solar Resource Knowledge Management*
Task 37	*Advanced Housing Renovation with Solar & Conservation*
Task 38	*Solar Thermal Cooling and Air Conditioning*
Task 39	*Polymeric Materials for Solar Thermal Applications*
Task 40	*Net Zero Energy Solar Buildings*
Task 42	*Compact Solar Thermal Energy Storage*
Working Group	*Daylight Research Group*

Completed Tasks:

Task 1	*Investigation of the Performance of Solar Heating and Cooling Systems*
Task 2	*Coordination of Solar Heating and Cooling R&D*
Task 3	*Performance Testing of Solar Collectors*
Task 4	*Development of an Insolation Handbook and Instrument Package*
Task 5	*Use of Existing Meteorological Information for Solar Energy Application*
Task 6	*Performance of Solar Systems Using Evacuated Collectors*
Task 7	*Central Solar Heating Plants with Seasonal Storage*
Task 8	*Passive and Hybrid Solar Low Energy Buildings*
Task 9	*Solar Radiation and Pyranometry Studies*
Task 10	*Solar Materials R&D*
Task 11	*Passive and Hybrid Solar Commercial Buildings*
Task 12	*Building Energy Analysis and Design Tools for Solar Applications*
Task 13	*Advance Solar Low Energy Buildings*
Task 14	*Advance Active Solar Energy Systems*
Task 16	*Photovoltaics in Buildings*
Task 17	*Measuring and Modeling Spectral Radiation*
Task 18	*Advanced Glazing and Associated Materials for Solar and Building Applications*
Task 19	*Solar Air Systems*
Task 20	*Solar Energy in Building Renovation*
Task 21	*Daylight in Buildings*
Task 23	*Optimization of Solar Energy Use in Large Buildings*
Task 22	*Building Energy Analysis Tools*
Task 24	*Solar Procurement*
Task 25	*Solar Assisted Air Conditioning of Buildings*
Task 26	*Solar Combisystems*
Task 28	*Solar Sustainable Housing*
Task 27	*Performance of Solar Facade Components*
Task 29	*Solar Crop Drying*
Task 31	*Daylighting Buildings in the 21st Century*
Task 32	*Advanced Storage Concepts for Solar and Low Energy Buildings*
Task 33	*Solar Heat for Industrial Processes*
Task 34	*Testing and Validation of Building Energy Simulation Tools*

Completed Working Groups:

CSHPSS, ISOLDE, Materials in Solar Thermal Collectors, and the Evaluation of Task 13 Houses

Nov 2009

IEA SHC Task 35 PV/Thermal Solar Systems

Objective

The objectives of the Task are to catalyze the development and market introduction of high quality and commercial competitive PV/Thermal Solar Systems and to increase general understanding and contribute to internationally accepted standards on performance, testing, monitoring and commercial characteristics of PV/Thermal Solar Systems in the building sector.

The Task is organized in 5 subtasks:

- Subtask A: Market and Commercialization of PV/T
- Subtask B: Energy Analysis and Modeling
- Subtask C: Product and System Development, Tests and Evaluation
- Subtask D: Demonstration Projects
- Subtask E: Dissemination

Organisation

IEA SHC Task 35 "PV/Thermal Solar Systems" is a three year Task initiated by the International Energy Agency (IEA) Solar Heating and Cooling (SHC) Programme in January 2005.
The Danish Energy Authority, acting through Mr. Henrik Sørensen, Esbensen Consulting Engineers A/S, Denmark, is designated as Operating Agent for the Task.

Task 35 is a so-called "minimum-level" collaboration task with IEA PVPS (Photovoltaic Power Systems Programme). At this level, experts selected by the PVPS Executive Committee participate in experts meetings of the Task managed by the SHC Executive Committee. The Task is fully defined and managed by the SHC Executive Committee with appropriate input from the PVPS Executive Committee. In this project Israel participated as a PVPS country member.

The official participants in the Task are listed in the table below:

Country	**Organization**	**Person**
Canada	Dept. of Mechanical Engineering, University of Waterloo, Waterloo, Ontario, Canada	Mike Collins
Denmark	Esbensen Consulting Engineers A/S	Henrik Sørensen
	Solar Energy Center, Danish Technological Institute	Ivan Katic
Israel	Millennium Electric	Ami Elazari
Sweden	Lund Technical University	Björn Karlsson Johan Nilsson Bengt Perers
The Netherlands	ECN (Energy Research Centre of the Netherlands)	Wim van Helden Herbert Zondag Marco Bakker

Apart from the above mentioned a number of manufacturers, universities, and research institutes from the countries Germany, Greece, Hong Kong, Italy, South Korea, Thailand, and Spain have been involved in the work.

Visit the Task 35 website: http://www.iea-shc.org/task35 for more details on activities and results.

INTRODUCTION

This document has been prepared in support of IEA Task 35: PV/Thermal Solar Systems – Subtask B: Energy Analysis and Modelling. Specifically, it reports on the availability of Photovoltaic and Solar Thermal system models. Special attention is paid to models of collectors that include both PV and Solar Thermal in the same absorber.

This report is intended as a reference to those who intend to develop new models.

EXISTING SYSTEM MODELS

An extensive search was undertaken to identify relevant solar thermal, solar electric, and PV/T system models. TRNSYS [1] was found to be the predominant source of models, and as such, it was decided by the Subtask to focus exclusively on that platform. The results have been laid out in the following sections.

The underlying theory behind each of these models can be complex. In most cases, each is the result of an individual Graduate research project, and is only fully described in the associated Thesis. The TRNSYS user manual [1] provides significant detail regarding the theory and capabilities of most of the models, and input and output parameters. Those not available with the TRNSYS package can likely found on the TRNSYS users web site: Thermal Energy System Specialists (TESS). Alternatively, Duffie and Beckman [2] provide theoretical detail of most of the solar thermal components. The capabilities of each model is described in point form.

Models programmed via other means were found. They have been included for reference, but will not be considered in detail.

Full code listing has been provided for the PV/T models as Appendix A.

A) PV/T Models

TRNSYS Models

TYPE 50: PV-Thermal Collector [3]

- A theoretical model of a generic collector (air or water) with integrated PV.
- Can be glazed/unglazed with non-transpired absorber.
- User choice of 8 operational modes based on known information
    1. Collector loss and cover transmission required
    2. Cover transmission required
    3. Collector loss and angular dependent cover transmission required
    4. Collector loss and cover transmission calculated internally
    5. Concentrating collector. Collector loss and cover transmission required. PV output is free-floating
    6. Concentrating collector. Detailed collector loss and cover transmission required. PV output is free-floating
    7. Concentrating collector. Collector loss and cover transmission required. PV output is set
    8. Concentrating collector. Detailed collector loss and cover transmission required. PV output is set

TYPE 555: PV-Thermal Collector [4]

- A theoretical model of an unglazed air collector with integrated PV.

TYPE 56(Mode): PV-Thermal Collector [4]

- A theoretical model of a generic collector (air or water) with integrated PV.
- Can be glazed/unglazed with non-transpired absorber.
- User choice of 6 operational modes based on known information
    1. Mode 0: Unglazed water. General heat transfer model
    2. Mode 3: Unglazed water. Detailed heat transfer model
    3. Mode 6: Glazed air. General heat transfer model
    4. Mode 7: Glazed air. Detailed heat transfer model
    5. Mode 8: Unglazed air. General heat transfer model
    6. Mode 9: Unglazed air. Detailed heat transfer model

Many of the TYPE50 models contain serious errors which limit their usability. In particular, a ‘floating point (division by zero) error’ is often encountered which prevents completion of a simulation.

The TYPE555 and 56(Mode) models are updated (and more reliable) versions of the TYPE50 models. Unfortunately, they are more specific in their applicability in that they have been adapted for building integrated systems.

Given that the TYPE50 models are not reliable, and that the TYPE555 and TYPE56(Mode) models are more specific than required. Subtask B of IEA SHC Task 35 has produced reliable generic PV/Thermal collector models for use in TRNSYS. The models (called the TYPE 201, 250, and 251 models) are available from the Task website as part of a downloadable package. The installation and use of these models, is given in a report included with that package [5].

It is further noted that these new models were used to help develop procedures for characterizing and monitoring PV/Thermal systems. Details of these procedures can be found in IEA SHC Task 35, Report DB-2 [6].

Other Models

Kalogirou [7] - The paper models a glazed PV/T water heater (Available from solar-net.teipat.gr/Kalogirou.pdf
.
Coventry [8] - The paper models a concentrating PV/T water heater (Available from solar.anu.edu.au/pages/pdfs/simulation_solar_2002.pdf)

Raghuraman [9] – Two separate one-dimensional analyses have been developed for the prediction of the thermal and electrical performance of both liquid and air flat-plate, photovoltaic/thermal (PV/T) collectors. The results of the analyses are compared with test measurements, and there from design recommendations are made to maximize the total energy extracted from the collectors.

Garg and Adhikari [10] - The paper describes modelling of Concentrating PV/T collectors.

B) Photovoltaic Component Models

TRNSYS Models

TYPE 94: Photovoltaic Array
- A theoretical model of a generic PV cell.
- Cell characteristics are required.

Other Electrical Elements
1. TYPE 47: Battery
2. TYPE 48: Inverter/Regulator
3. PV Array, Maximum Power Point Tracker, Charge Controllers, Battery [11]

C) Solar Thermal Component Models

TRNSYS Models

TYPE 1: Flat Plate Solar Collector

- A theoretical model of a generic flat plate collector (air or water).
- The results from a standard efficiency versus ΔT/G test is required as input (modeled as quadratic equation).
- Inlet, average, or outlet fluid temperatures can be used.
- User choice of 5 optical models
    1. normal incidence only
    2. 1 axis incidence angle modifier
    3. 1 axis incidence angle modifier
    4. properties based on cover materials
    5. bi-axial incidence angle modifier
- Modified versions found in literature include non-linear solar collector characterization [3], and a second order incidence angle modifier [4]
- User choice of 5 optical models

TYPE 71: Unglazed Transpired Collector System [12][1]

- A theoretical model of a transpired air collector.

TYPE 72: Performance Map Solar Collector

- A theoretical model of a generic flat plate collector (air or water).
- The results from an efficiency versus ΔT/G, wind speed, and radiation are required (modeled as quadratic equation).
- Inlet, average, or outlet fluid temperatures can be used.
- User choice of 5 optical models
    1. normal incidence only
    2. 1 axis incidence angle modifier
    3. 1 axis incidence angle modifier
    4. properties based on cover materials
    5. bi-axial incidence angle modifier

TYPE 73: Theoretical Flat Plate Solar Collector

- A theoretical model of a generic flat plate collector (air or water).
- No measured performance input is required.
- Inlet, average, or outlet fluid temperatures can be used.

TYPE 74: Compound Parabolic Concentrating Collector

- A theoretical model of a generic CPC collector (air or water).
- No measured performance input is required.

TYPE 132: Unglazed Collector [13]

- A theoretical model of a generic flat plate collector (air or water).

[1] This is not the Model 71 that is supplied with TRNSYS. It is a model that must be downloaded from the TRNSYS website.

TYPE 186: Serpentine Collector [14]

- A theoretical model of a generic flat plate collector (air or water).

Other Collector Models

1. TYPE 45:Thermosiphon Collector with Integral Storage
2. TYPE 71: Evacuated Tube Solar Collector

Solar Storage Elements

1. TYPE 4: Stratified Fluid Storage Tank
2. TYPE 10: Rock Bed Thermal Storage
3. TYPE 38: Algebraic Tank (Stratified)
4. TYPE 39: Variable Volume Tank
5. TYPE 60: Detailed Fluid Storage with Heaters
6. TYPE 74[2]: Stratified Fluid Storage with Internal Heat Exchanger [8]

Solar Controllers

1. TYPE 2: ON/Off Differential
2. TYPE 40: Microprocessor

Other Models

Griffith and Ellis [15] – The paper demonstrates EnergyPlus's capability to model PV or Thermal Systems. (Available from www.nrel.gov/docs/fy04osti/36275.pdf)

ESP-r - [16] – Supposedly has capability, but I was unable to find anything definitive.

RetScreen – Separate Excel based tools for PV and Solar Thermal [17].

MODEL INPUTS

A list of parameters, inputs, and outputs is provided via the TRNSYS software for each of the models it contains. For each of the models noted in the previous section, that list has been provided (Appendix B). If existing TRNSYS models are to be adapted as the norm, than these variables will be the ones of interest.

Attempting to compile all of the possible variables into a comprehensive list is difficult. The number of system components, for both PV and Solar Thermal systems, is large and varied in nature. Therefore, to facilitate model / experiment comparison, testing groups should refer to Appendix B and instrument accordingly. It is noted that as future tasks demand the development of new models or refinement of existing models, new parameters may be introduced that need to be monitored.

[2] TYPE 74 has already been used. I'm uncertain why the repeat has occurred at this point.

REFERENCES

[1] Solar Energy Laboratory (2000) "TRNSYS Reference Manual", University of Wisconsin-Madison

[2] Duffie, J.A., and Beckman, W.A. (1991) "Solar Engineering of Thermal Processes: 2nd Ed.", John Wiley and Sons, Inc.

[3] Evans, D.L., Facinelli, W.A., and Otterbein, R.T. (1978) "Combined Photovoltaic / Thermal System Studies" DOE Technical Report Dep. NTIS, PC A09/MF A01, 183 pages.

[4] Jay Burch, U.S. Department of Energy Solar Buildings Technologies and TESS, Inc (Thermal Energy System Specialists).

[5] Collins, M., and Delisle, V., (2008) "Instructions for Using the Downloadable Model Package in TRNSYS", IEA Report for IEA SHC - Task 35.

[6] Collins, M. (2008) "Recommended Standard for Characterization and Monitoring of PV/Thermal Systems", IEA Report for IEA SHC - Task 35, Report DB-2.

[7] Kalogirou, S.A (2005) "Design and Performance of Hybrid PV/T Solar Water Heaters", Intensive Programme: ICT Tools in PV-Systems Engineering: Teaching and Learning, on CD-ROM

[8] Coventry, J. (2002) "Simulation of a Concentrating PV/Thermal Collector Using TRNSYS", ANZSES Solar Energy Conference, Australia

[9] Raghuraman, P. (1981) "Analytical Predictions of Liquid and Air Photovoltaic/Thermal , Flat-Plate Collector Performance", Journal of Solar Energy Engineering, Vol 103, Issue 4.

[10] Garg, H.P., and Adhikari (1999) "Performance Analysis of a Combined Photovoltaic/Thermal (PV/T) Collector with Integrated CPC Troughs", ISES

[11] Eckstein, Jurgen (?) "Detailed Modelling of Photovoltaic System Components", M.S. Thesis, Solar Energy Laboratory, University of Wisconsin at Madison

[12] Summers, David N. (1995) "Thermal Simulation and Economic Assessment of Unglazed Transpired Collectors", M.S. Thesis in Mechanical Engineering, University of Wisconsin-Madison.

[13] Perers, B., and Bales, C. (2003) "A Solar Collector Model for TRNSYS Simulation and System Testing", IEA Report for IEA SHC - Task 26.

[14] http://sel.me.wisc.edu/trnsys

[15] Griffith, B.T. and Ellis, P.G. (2004) "Photovoltaic and Solar Thermal Modeling with the EnergyPlus Calculation Engine", World Renewable Energy Congress VIII and Expo, Denver, Colorado.

[16] http://www.esru.strath.ac.uk/

[17] http://www.retscreen.net/ang/menu.php

Appendix A

Code Listing for PV/T TRNSYS Models

Type 50

TYPE 50a-PV/T Flat Plate Collector (Constant losses)
TYPE 50b- PV/T Flat Plate Collector (Losses=f(temperature, wind, geometry))
TYPE 50c- PV/T Flat Plate Collector (Angular dependence of transmittance)
TYPE 50d- PV/T Flat Plate Collector (Losses=f(temperature, wind, geometry) and t=f(angle))
TYPE 50e- PV/T Concentrating collectors (Constant Losses- No cell operating voltage)
TYPE 50f- PV/T Concentrating collectors (Top Loss-f(wind, T) No cell operating voltage)
TYPE 50g- PV/T Concentrating collectors (Constant Losses- Cell operating V is input)
TYPE 50h- PV/T Concentrating collectors (Top Loss-f(wind, T) Cell operating voltage is input)

```
  SUBROUTINE TYPE50 (TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C*****************************************************************************
C.  THIS COMPONENT SIMULATES THE THERMAL PERFORMANCE OF A
C.  FLAT-PLATE SOLAR COLLECTOR USING THE MODEL DEVELOPED BY
C.  HOTTEL, WHILLIER, AND BLISS.
C   LAST MODIFIED 3/93 - JWT
C*****************************************************************************

C.
C.  HR     - TOTAL RADIATION INCIDENT ON THE TILTED COLLECTOR SURFACE
C.  S      - SOLAR ENERGY ABSORBED BY THE SURFACE OF THE ABSORBER
C.         - FOR THE CASE OF THE CONCENTRATING PV/THERMAL COLLECTOR
C.  SINC   - SOLAR ENERGY INCIDENT ON THE ABSORBER
C.  QU     - THE USEFUL ENERGY COLLECTION RATE PER UNIT AREA
C.  A      - COLLECTOR AREA
C.  FP     - (F-PRIME)  COLLECTOR GEOMETRY EFFICIENCY FACTOR
C.  UL     - OVERALL ENERGY LOSS COEFFICIENT
C.  TA     - AMBIENT TEMPERATURE
C.  TIN    - INLET FLUID TEMPERATURE
C.  TOUT   - OUTLET FLUID TEMPERATURE
C.  TM     - MEAN FLUID TEMPERATURE
C.  FLWRT  - COLLECTOR FLUID FLOWRATE
C.  CPF    - THERMAL CAPACITANCE OF THE COLLECTOR FLUID
C.  TAUALF - THE PRODUCT OF THE TRANSMITTANCE OF THE GLASS
C.  NOTE THAT DIFFUSE RADIATION IS TREATED AS IF IT STRIKES THE
C.  COLLECTOR SURFACE AT 60 DEGREES.
C.          AND THE ABSORPTANCE OF THE COLLECTOR PLATE SURFACE.
C.
C.
C.  THIS PROGRAM HAS EIGHT MODES OF OPERATION AS DETERMINED BY
C.  THE VALUE OF MODE.
C.     IF MODE=1,5,7 UL AND TAUALF ARE CONSTANTS
C.     IF MODE=2,4,6,8  UL IS CALCULATED AS A FUNCTION OF*
C.
C.  NG     - THE NUMBER OF GLASS COVERS
C.  EP     - THE THERMAL EMITTANCE OF THE COLLECTOR PLATE SURFACE
C.  UBE    - THE CONTRIBUTION TO UL DUE TO BOTTOM AND EDGE
C.          LOSSES (KJ/HR-M2-C)
C.  ANGLE  - THE TILT OF THE COLLECTOR WITH RESPECT TO HORIZONTAL
C.  WIND   -  THE WINDSPEED (M/SEC)
C.
C.     IF MODE=3,4  TAUALF IS CALCULATED AS A FUNCTION OF*
C.
C.  THETA1 - THE ANGLE OF INCIDENCE OF RADIATION ON THE COLLECTOR
C.  ALF    - THE ABSORPTANCE OF THE COLLECTOR PLATE SURFACE (CONSTANT)
C.  XKL    - PRODUCT OF THE EXTINCTION COEFFICIENT AND THE
C.          THICKNESS OF EACH GLASS COVER
C.  REFIND - THE REFRACTIVE INDEX OF THE GLASS
C.  HBT    - THE INSTANTANEOUS BEAM RADIATION ON THE COLLECTOR SURFACE
C.  HDT    - THE INSTANTANEOUS DIFFUSE RADIATION ON THE COLLECTOR SURFACE
C.
C.      IF MODE=5,6,7,8  A COMBINED (PHOTOVOLTAIC-THERMAL)
C.          COLLECTOR IS SIMULATED
C.          MODE=5,6 ASSUMES CELLS ARE OPERATED AT PEAK POWER
C.          MODE=7,8 ASSUMES THE VOLTAGE ON THE ARRAY IS FIXED EXTERNAL
C.   AR      -APERTURE AREA TO ABSORBER AREA(GEOMETRICAL CONCENTRATION
C.           RATIO) FOR USE IN CONCENTRATING COLLECTORS
C.   FE       APPROPRIATE FIN EFFICIENCY
C.   UB      -BACK LOSS COEFFICIENT
C.   CB      -THERMAL CONDUCTANCE BETWEEN CELLS AND ABSORBER
C.   UF      -FILM COEFFICIENT BETWEEN FLUID AND ABSORBER
C.   UT      -TOP LOSS COEFFICIENT FOR MODES 5 AND 7
C.
C.     MODES 1 THRU 4 HAVE BEEN MODIFIED TO ALLOW SIMULATION OF
C.        FLAT PLATE COMBINED COLLECTORS AS DEVELOPED BY
C.        FLORSCHUETZ (SHARING THE SUN JOINT SOLAR CONFERENCE
C.        PROCEEDINGS, VOL. 6, P.79-92, WINNIPEG (1976)
C.   THE FOLLOWING  PARAMETERS ARE REQUIRED IN ADDITION
C.        TO THE ORIGINAL ONES REQUIRED IN MODES 1 THRU 4
C.   BR      -TEMPERATURE COEFFICIENT OF THE CELLS
C.   TR      -A REFERENCE EFFICIENCY WHERE THE CELL EFFICIENCY IS
C.          KNOWN--- THE CELL EFFICIENCY IS ENTERED AS XIN(5,6,7,8)
C.           IN MODE 1,2,3,4 RESPECTIVELY
C.   CELLPF   -THE RATIO OF CELL AREA TO ABSORBER AREA

      DOUBLE PRECISION XIN,OUT
      REAL
     +   IC,        K1,        K2
      INTEGER*4 INFO,INFO7
      DIMENSION
     +   INFO(15),     OUT(20),     PAR(25),     XIN(10)
      COMMON /LUNITS/
     +   LUR,        LUW,        IFORM,     LUK
      DIMENSION
     +   ATR(3),      BTR(3),      CTR(3),      TAU040(3)
      DATA EG/0.88/,PI/3.1415927/,SB/5.678E-08/
      DATA ATR/-2.9868,-1.4214,-0.74816/
      DATA BTR/-3.7360,-5.7356,-6.5262/
      DATA CTR/4.3541,5.7723,6.3769/
      DATA TAU040/0.92,0.845,0.785/
      DATA IUNIT/0/,REFIND/1.526/,NPP/0/,AR/1./,UBE/0./
```

```
      INFO7=INFO(7)
      IF (INFO(7).GE.0) GO TO 1
C    REMOVED BECAUSE OF POSSIBLE TOLERANCE PROBLEMS
C      INFO(9)=0
C   TEMPORARILY ADD ANOTHER OUTPUT,FR
      FR=0.
      INFO(6)=20
C     INFO(6) =12
C  ******************D*C**********
      MODE=PAR(1)
      NP=INFO(4)
      IF (MODE.EQ.1) CALL TYPECK (1,INFO,5,NP,0)
      IF (MODE.EQ.2) CALL TYPECK (1,INFO,6,NP,0)
      IF (MODE.EQ.3) CALL TYPECK (1,INFO,7,NP,0)
      IF (MODE.EQ.4) CALL TYPECK (1,INFO,8,NP,0)
      IF (MODE.EQ.5) CALL TYPECK (1,INFO,4,NP,0)
      IF (MODE.EQ.6) CALL TYPECK (1,INFO,6,NP,0)
      IF (MODE.EQ.7) CALL TYPECK (1,INFO,5,NP,0)
      IF (MODE.EQ.8) CALL TYPECK (1,INFO,7,NP,0)
      IF (MODE.GE.1.AND.MODE.LE.8) GO TO 1
      CALL TYPECK (4,INFO,0,0,0)
      RETURN 1
    1 CONTINUE
      ITER=0
      IF (INFO(1).EQ.IUNIT) GO TO 10
      IUNIT=INFO(1)
      MODE=PAR(1)
      A=PAR(2)
      FP=PAR(3)
C  PAR(3) IS REDEFINED BELOW IF MODE.GE.5
      CPF=PAR(4)
      ALF=PAR(5)
      GO TO (2,3,4,5,6,6,6,6), MODE
    2 UL=PAR(6)
      TAU=PAR(7)
      BR=PAR(8)
      TR=PAR(9)
      CELLPF=PAR(10)
      TAUALF=TAU*ALF
      GO TO 10
    3 XNG=PAR(6)
      EP=PAR(7)
      UBE=PAR(8)
      ANGLE=PAR(9)
      TAU=PAR(10)
      BR=PAR(11)
      TR=PAR(12)
      CELLPF=PAR(13)
      NG=XNG
      TAUALF=TAU*ALF
      GO TO 10
    4 XNG=PAR(6)
      UL=PAR(7)
      XKL=PAR(8)
      BR=PAR(9)
      TR=PAR(10)
      CELLPF=PAR(11)
      NG=XNG
      GO TO 9
    5 XNG=PAR(6)
      EP=PAR(7)
      UBE=PAR(8)
      ANGLE=PAR(9)
      XKL=PAR(10)
      BR=PAR(11)
      TR=PAR(12)
      CELLPF=PAR(13)
      UL=0.
      NG=XNG
      GO TO 9
    6 AR=PAR(3)
      IF (MODE.GE.7) NPP=1
      FE=PAR(6)
      UB=PAR(7)
      CB=PAR(8)
      UF=PAR(9)
      UF=UF*(1.+FE)
      IF (CB.LT.1.E-5) CALL TYPECK (4,INFO,0,0,0)
      TAU=PAR(10)
      TAUALF=TAU*ALF
      GO TO (10,10,9,9,7,8,7,8), MODE
    7 UT=PAR(11)
      LUIN=-1
      IF(INFO(4).EQ.12)LUIN=PAR(12)
      GO TO 10
    8 XNG=PAR(11)
      EP=PAR(12)
      LUIN=-1
      IF(INFO(4).EQ.13) LUIN=PAR(13)
      GO TO 10
    9 CONTINUE
      TAU60=EXP(-1.21453*XNG*XKL)*(1.-EXP((ATR(NG)+(BTR(NG)+CTR(NG)*0.5)
     +*0.5)*0.5))
C.  TAU60 IS THE TRANSMITTANCE OF THE GLASS COVER SYSTEM AT AN
C.  INCIDENCE ANGLE OF 60 DEGREES, AS ASSUMED FOR DIFFUSE RADIATION
C.  SEE HOTTEL AND WOERTZ
   10 CONTINUE
      TIN=XIN(1)
      FLWRT=XIN(2)
      TA=XIN(3)
      IF (MODE.GE.5) GO TO 11
      TC=TR+1./BR
```

```
      BA=1./(TC-TA)
   11 GO TO (12,14,15,20,16,17,18,19), MODE
   12 CONTINUE
      UL=PAR(6)
      HR=XIN(4)
      ETAR=XIN(5)*CELLPF
      ETAA=ETAR*(1.-BR*(TA-TR))
      UL=UL-TAU*HR*ETAA*BA
      GO TO 29
   14 HR=XIN(4)
      WIND=XIN(5)
      ETAR=XIN(6)*CELLPF
      ETAA=ETAR*(1.-BR*(TA-TR))
      GO TO 24
   15 HBT=XIN(4)
      HDT=XIN(5)
      UL=PAR(6)
      THETA1=XIN(6)
      ETAR=XIN(7)*CELLPF
      ETAA=ETAR*(1.-BR*(TA-TR))
      GO TO 21
C
   16 HR=XIN(4)
      ICT=1
      GO TO 26
   17 HR=XIN(4)
      WIND=XIN(5)
      TILT=XIN(6)
      GO TO 24
   18 HR=XIN(4)
      V=XIN(5)
      ICT=1
      GO TO 26
   19 HR=XIN(4)
      WIND=XIN(5)
      V=XIN(6)
      TILT=XIN(7)
      GO TO 24
   20 HBT=XIN(4)
      HDT=XIN(5)
      THETA1=XIN(6)
      WIND=XIN(7)
      ETAR=XIN(8)*CELLPF
      ETAA=ETAR*(1.-BR*(TA-TR))
   21 CONTINUE
C.
      HR=HBT+HDT
      TAU=0.0
      IF (THETA1.GT.85.) GO TO 23
      IF (HR.LE.1.0E-10) GO TO 23
      THETA1=THETA1*2.*PI/360.0
      COSTH1=COS(THETA1)
      THETA2=ASIN(SIN(THETA1)/REFIND)
      COSTH2=COS(THETA2)
      TAU=TAU040(NG)
      IF (COSTH1.GE.0.766) GO TO 22
      TAU=1.0-EXP((ATR(NG)+(BTR(NG)+CTR(NG)*COSTH1)*COSTH1)*COSTH1)
   22 CONTINUE
      TAU=HBT/HR*TAU*EXP(-XNG*XKL/COSTH2)
      TAU=TAU+HDT/HR*TAU60
C.
C.
   23 CONTINUE
      TAUALF=ALF*TAU
      UL=UL-TAU*HR*ETAA*BA
C.
      GO TO (29,24,29,24,26,24,26,24), MODE
   24 CONTINUE
      ICT=0
      HWIND=5.7+3.8*WIND
      TM=TIN
   25 CONTINUE
      IF (ITER.EQ.0) ICT=ICT+1
      IF (ICT.GT.2) GO TO 45
      TMC=TM+273.15
      TAC=TA+273.15
      IF (TMC.LE.TAC) TMC=TAC+1.0
      F=(1.0-0.04*HWIND+5.0E-04*HWIND*HWIND)*(1.0+0.091*XNG)
      IF (XNG.LT..5) F=1.
      C=365.9*(1.0-0.00883*ANGLE+0.0001298*ANGLE*ANGLE)
      STF1=C/TMC*((TMC-TAC)/(XNG+F))**0.33
      STF1=XNG/STF1+1.0/HWIND
      STF1=1.0/STF1
      STF2=1.0/(EP+0.05*XNG*(1.0-EP))+(2.*XNG+F-1.)/EG-XNG
      STF2=SB*(TMC*TMC+TAC*TAC)*(TMC+TAC)/STF2
      UL=(STF1+STF2)*3.6+UBE
      IF (MODE.GE.5) GO TO 26
      UL=UL-TAU*HR*ETAA*BA
C  UT WILL BE SET EQUAL TO  UL FOR MODES 6 & 8
C.
C.  UL IS CALCULATED USING THE RELATION OF KLEIN
C.
      GO TO 29
   26 S=HR*TAUALF*AR
      SINC=S/ALF
      IF (MODE.EQ.6.OR.MODE.EQ.8) UT=UL
      DENOM=UT*(CB+UF)+CB*UF
      K1=CB/DENOM
      IF (ICT.NE.1) GO TO 27
      TP=K1*(S-UT*(TIN-TA))+TIN
      TCELLR=UF*(TP-TIN)/CB+TP
      IF (FLWRT.LT.1.E-5) TCELLR=(S+TA*(UT+CB*UB/(CB+UB)))/(UT+CB-CB**2/
```

```
   +(CB+UB))
    TCELL=TCELLR
 27 CONTINUE
C
    TEMP=TCELL
    IF (MODE.EQ.5.OR.MODE.EQ.6) TEMP=TCELLR
    CALL SOLCEL (TCELLR,TEMP,SINC,PR,BETA,IC,V,ITER,NPP,INFO7,A,AR,
   +MODE,LUIN,*28)
    CALL LINKCK('TYPE50','SOLCEL',1,99)
 28 K2=1.+K1*PR*BETA*(CB+UF)/CB
    IF (S.LT.1.E-5) PRS=0.
    IF (S.GE.1.E-5) PRS=PR/S
    FP=UF*K1*(1.-PRS*(1.+BETA*(TA-TCELLR)))/K2
    ULAB=UF*K1*(UT+PR*BETA)/(K2*FP)
    UL=ULAB/AR
    ULO=UT/AR
 29 CONTINUE
    IF (FLWRT-1.E-5) 34,34,30
 30 CONTINUE
    IF ((FP*UL*A/(FLWRT*CPF)).LT.0.01) GO TO 31
    FR=FLWRT*CPF*(1.0-EXP(-FP*UL*A/(FLWRT*CPF)))/(A*UL)
    GO TO 32
 31 CONTINUE
    FR=FP
 32 HRT=HR*(1.-ETAA/ALF)
    IF (MODE.GE.5) HRT=HR
    QU=FR*(HRT*TAUALF-UL*(TIN-TA))
    QE=TAU*HR*ETAA*(1.-BA*(FR*(TIN-TA)+(TAUALF*HRT/UL)*(1.-FR)))
    IF (MODE.LE.4) GO TO 33
    QE=PR*(1.+BETA*(TCELL-TCELLR))/AR
 33 CONTINUE
    TOUT=QU/FLWRT*A/CPF+TIN
    GO TO 36
 34 QU=0.0
    IF (MODE.LE.4) GO TO 35
    QE=PR/AR
    TCELLR=(S-PR+TA*(UT+CB*UB/(CB+UB)))/(UT+CB-CB**2/(CB+UB))
    TCELL=TCELLR
    TEMP=TCELL
    TOUT=TCELL*(1.-UB/(UB+CB))+TA*UB/(UB+CB)
    FLWRT=0.
    GO TO 39
 35 CONTINUE
    ULO=UL+TAU*HR*ETAA*BA
    QE=TAU*HR*ETAA*(ULO-TAUALF*HR*BA)/UL
    FLWRT=0.0
    TOUT=(TAUALF*HR-QE)/ULO+TA
 36 CONTINUE
    TM=(TIN+TOUT)/2.0
    IF (MODE.GE.5) GO TO 38
    TCELL=TM
    IF (HR.LE.1.E-5.AND.MODE.LE.4) GO TO 44
    IF (ETAR.LT.1.E-5) TCELL=TC
    IF (ETAR.LT.1.E-5) GO TO 37
    TCELL=TC-(QE/(TAU*HR*ETAR))*(TC-TR)
 37 IF (MODE.LE.4) GO TO 44
 38 TP=QU*AR/UF+TIN
    TCELLR=UF*(TP-TIN)/CB+TP
    TPR=QU*AR/UF+TM
    TCELL=UF*(TPR-TM)/CB+TPR
    TM=TCELL
    TEMP=TCELL
    IF (MODE.EQ.5.OR.MODE.EQ.6) TEMP=TCELLR
 39 CONTINUE
    PRSAVE=PR
    BETAS=BETA
    ITER=1
    CALL SOLCEL (TCELLR,TEMP,SINC,PR,BETA,IC,V,ITER,NPP,INFO7,A,AR,
   +MODE,LUIN,*391)
    CALL LINKCK('TYPE50','SOLCEL',1,99)
391   IF (PR.LT.1.E-5.AND.PRSAVE.LT.1.E-5) GO TO 40
    IF (PR.LT.1.E-5) GO TO 42
    IF (ABS((PR-PRSAVE)/PR).GT.0.05) GO TO 42
 40 IF (BETA.LT.1.E-5.AND.BETAS.LT.1.E-5) GO TO 41
    IF (BETA.LT.1.E-5) GO TO 42
    IF (ABS((BETA-BETAS)/BETA).GT.0.05) GO TO 42
 41 ITER=0
 42 CONTINUE
    IF (MODE.EQ.5.OR.MODE.EQ.7) GO TO 43
    IF (ITER.GT.0) GO TO 25
    GO TO 44
 43 CONTINUE
    ICT=ICT+1
    IF (ITER.GT.0) GO TO 28
    IF (ICT.LE.2) GO TO 28
 44 CONTINUE
    GO TO (45,25,45,25,45,25,45,25), MODE
 45 CONTINUE
    OUT(1)=TOUT
    OUT(2)=FLWRT
    OUT(3)=QU*A
    OUT(4)=UL
    OUT(5)=TAUALF
    OUT(6)=QE*A
    OUT(7)=TCELL
    OUT(8)=ULO
    OUT(9)=0.
    OUT(10)=0.
    OUT(11)=0.
    OUT(12)=0.
C******INSERT FR AS OUT(13) FOR TESTING****DC
    OUT(13)=FR
```

```
      OUT(14)=PR
C****************************************
      GO TO (47,47,47,47,46,46,46,46), MODE
   46 OUT(4)=ULO
      OUT(8)=UL
      OUT(9)=V
      OUT(11)=TCELLR
      OUT(12)=ITER
      IF (V.LT.1.E-5) GO TO 48
      OUT(10)=QE*A/(V*3.6)
   47 CONTINUE
      RETURN 1
   48 OUT(10)=0.
      GO TO 47
      END
```

TYPE 555- Unglazed air PV/T flat plate collector

*The code is not available

TYPE 560- Unglazed PV/T collector with fluid stream passing through tubes bonded to an absorber plate located beneath the PV cells

```
  SUBROUTINE TYPE560(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C    THIS SUBROUTINE MODELS AN UNGLAZED PV/THERMAL COLLECTOR.  IN THIS VERSION, THE
WORKING FLUID IS WATER
C    CARRIED IN TUBES BONDED TO A PLATE.  THE PV MATERIAL IS ADHERED TO THE TOP OF THIS
PLATE.  THE PV SYSTEM
C    IS ASSUMED TO BE WORKING UNDER THE MAXIMUM POWER POINT ASSUMPTION.
C
C    THIS MODEL IS BASED ON A DERIVATION BY JEFF THORNTON OF THERMAL ENERGY SYSTEM
SPECIALISTS FROM THE
C    STANDARD TUBE-FIN SOLAR COLLECTOR ALGORITHMS PRESENTED BY DUFFIE AND
BECKMAN IN THE CLASSIC "SOLAR
C    ENGINEERING OF THERMAL PROCESSES' - SPECIFICALLY CHAPTER SIX.
C
C    LAST MODIFIED:
C       FEBRUARY 2004 - JWT - INITIAL PROGRAMMING
C       MODIFIED APRIL 2004 - FIXED THE PV POWER EQUATION TO INCLUDE XKAT AND TAU-ALPHA
FACTORS
C
C-----------------------------------------------------------------------------------------------------------------------
```



```
C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
            USE TrnsysConstants
            USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED BY THE MULTI-DLL VERSION OF TRNSYS
     !DEC$ATTRIBUTES DLLEXPORT :: TYPE560
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
     IMPLICIT NONE
            DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

     INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL
     CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    USER DECLARATIONS
     PARAMETER (NP=20,NI=14,NOUT=19,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
     DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT),T(ND),
           1  DTDT(ND)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
     DOUBLE PRECISION TAUALF,ANGLE_INC,B0,RDCONV,LENGTH,WIDTH,
           1  THICK_ABSORBER,K_ABSORBER,DIA_TUBE,WIDTH_BOND,THICK_BOND,
    1  K_BOND,R_ADHESIVE,R_BACK,CP_FLUID,REFLECTANCE,EMISSIVITY,
           1  T_FLUID_IN,FLOW_IN,T_AMB,T_SKY,T_BACK,GT,GH,GD,REFL_GROUND,
           1  SLOPE,H_CONV_T,H_CONV_B,H_FLUID,AREA,W,T_FLUID_OUT,QU,POWER,
           1  EFF_PV,EFF_THERMAL,FR,T_PLATE_MEAN,T_FLUID_MEAN,XKAT,
    1  Q_TOP_CONV,Q_TOP_RAD,Q_BACK,Q_BASE,Q_FIN,Q_ABS,FPRIME,UL,
           1  TAUALPHAN,EFFSKY,EFFGND,COSSLP,FSKY,FGND,GDSKY,GDGND,XKATB,
           1  XKATDS,XKATDG,T_REF,GT_REF,EFF_PV_REF,EFF_CORR_T,EFF_CORR_I,
           1  R_T,R_B,T_PV,T_PV_OLD,FACTOR_T,FACTOR_I,H_RAD,H_RADIATION,S,
           1  T_PV_MEAN,B,J_FACTOR,M,N,U,R_BOND,Z,PI,EPSILON,SIGMA,THETA,
           1  QU_PRIME,T_BASE_MEAN,T_FIN_MEAN,T_PV_MEAN_BASE,
           1  T_PV_MEAN_FIN
           INTEGER N_TUBES,ICOUNT
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
     DATA RDCONV/0.017453292/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    FUNCTIONS
     TAUALF(ANGLE_INC)=1.-B0*(1./DMAX1(0.5,DCOS(ANGLE_INC*RDCONV))-1.)
    .         - (1.-B0)*(DMAX1(60.,ANGLE_INC)-60.)/30.
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
     TIME0=getSimulationStartTime()
     TFINAL=getSimulationStopTime()
     DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
     IF(INFO(7).EQ.-2) THEN
             INFO(12)=16
             RETURN 1
           ENDIF
C-----------------------------------------------------------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
                  RETURN 1
               ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
      IF(INFO(13).GT.0) THEN
                  RETURN 1
               ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C       RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
         IUNIT=INFO(1)
                  ITYPE=INFO(2)

C       SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
         INFO(6)=NOUT
         INFO(9)=1
                  INFO(10)=0

C       CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED
                  CALL TYPECK(1,INFO,NI,NP,ND)

C       SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
         DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','IR1','IR1','IR1',
               1               'DM1','DG1','DG1','HT1','HT1','HT1'/
         DATA OCHECK/'TE1','MF1','PW1','PW1','DM1','DM1','DM1','TE1',
               1               'TE1','DM1','PW1','PW1','PW1','PW1','HT1','DM1',
     1               'HT1','PW1','PW1'/

C       CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
         CALL RCHECK(INFO,YCHECK,OCHECK)

C       RETURN TO THE CALLING PROGRAM
         RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C       SET THE UNIT NUMBER FOR FUTURE CALLS
         IUNIT=INFO(1)
         ITYPE=INFO(2)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
         LENGTH=PAR(1)
                  WIDTH=PAR(2)
                  THICK_ABSORBER=PAR(3)
                  K_ABSORBER=PAR(4)
                  N_TUBES=JFIX(PAR(5)+0.5)
                  DIA_TUBE=PAR(6)
                  WIDTH_BOND=PAR(7)
                  THICK_BOND=PAR(8)
                  K_BOND=PAR(9)
                  R_ADHESIVE=PAR(10)
                  R_BACK=PAR(11)
                  CP_FLUID=PAR(12)
         REFLECTANCE=PAR(13)
                  EMISSIVITY=PAR(14)
         B0=PAR(15)
         T_REF=PAR(16)
         GT_REF=PAR(17)
         EFF_PV_REF=PAR(18)
         EFF_CORR_T=PAR(19)
         EFF_CORR_I=PAR(20)

C       CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
         IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
         IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
         IF(THICK_ABSORBER.LE.0.) CALL TYPECK(-4,INFO,0,3,0)
         IF(K_ABSORBER.LE.0.) CALL TYPECK(-4,INFO,0,4,0)
         IF(N_TUBES.LT.1) CALL TYPECK(-4,INFO,0,5,0)
         IF(DIA_TUBE.LE.0.) CALL TYPECK(-4,INFO,0,6,0)
                  IF(DBLE(N_TUBES)*DIA_TUBE.GT.WIDTH) CALL TYPECK(-4,INFO,0,6,0)
         IF(R_ADHESIVE.LE.0.) CALL TYPECK(-4,INFO,0,10,0)
         IF(R_BACK.LE.0.) CALL TYPECK(-4,INFO,0,11,0)
         IF(CP_FLUID.LE.0.) CALL TYPECK(-4,INFO,0,12,0)
         IF(REFLECTANCE.LT.0.) CALL TYPECK(-4,INFO,0,13,0)
         IF(REFLECTANCE.GT.1.) CALL TYPECK(-4,INFO,0,13,0)
         IF(EMISSIVITY.LT.0.) CALL TYPECK(-4,INFO,0,14,0)
         IF(EMISSIVITY.GT.1.) CALL TYPECK(-4,INFO,0,14,0)
         IF(B0.LT.0.) CALL TYPECK(-4,INFO,0,15,0)
         IF(B0.GT.1.) CALL TYPECK(-4,INFO,0,15,0)
         IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,17,0)
         IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,18,0)
```

```
      IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,18,0)

C      PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
      OUT(1)=XIN(1)
      OUT(2:7)=0.
      OUT(8)=XIN(1)
      OUT(9)=XIN(1)
             OUT(10:19)=0.

C      RETURN TO THE CALLING PROGRAM
      RETURN 1

    ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
    IF(INFO(1).NE.IUNIT) THEN

C      RESET THE UNIT NUMBER
             IUNIT=INFO(1)
             ITYPE=INFO(2)

C      READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
      LENGTH=PAR(1)
             WIDTH=PAR(2)
             THICK_ABSORBER=PAR(3)
             K_ABSORBER=PAR(4)
             N_TUBES=JFIX(PAR(5)+0.5)
             DIA_TUBE=PAR(6)
             WIDTH_BOND=PAR(7)
             THICK_BOND=PAR(8)
             K_BOND=PAR(9)
             R_ADHESIVE=PAR(10)
             R_BACK=PAR(11)
             CP_FLUID=PAR(12)
      REFLECTANCE=PAR(13)
             EMISSIVITY=PAR(14)
      B0=PAR(15)
      T_REF=PAR(16)
      GT_REF=PAR(17)
      EFF_PV_REF=PAR(18)
      EFF_CORR_T=PAR(19)
      EFF_CORR_I=PAR(20)
    ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
      T_FLUID_IN=XIN(1)
             FLOW_IN=XIN(2)
             T_AMB=XIN(3)
             T_SKY=XIN(4)
             T_BACK=XIN(5)
      GT=XIN(6)
      GH=XIN(7)
      GD=XIN(8)
      REFL_GROUND=XIN(9)
      ANGLE_INC=XIN(10)
      SLOPE=XIN(11)
      H_CONV_T=XIN(12)
      H_CONV_B=XIN(13)
      H_FLUID=XIN(14)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    CHECK THE INPUTS FOR PROBLEMS
      IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
      IF(GT.LT.0.) CALL TYPECK(-3,INFO,6,0,0)
      IF(GH.LT.0.) CALL TYPECK(-3,INFO,7,0,0)
      IF(GD.LT.0.) CALL TYPECK(-3,INFO,8,0,0)
      IF(REFL_GROUND.LT.0.) CALL TYPECK(-3,INFO,9,0,0)
      IF(REFL_GROUND.GT.1.) CALL TYPECK(-3,INFO,9,0,0)
      IF(H_CONV_T.LE.0.) CALL TYPECK(-3,INFO,12,0,0)
      IF(H_CONV_B.LE.0.) CALL TYPECK(-3,INFO,13,0,0)
      IF(H_FLUID.LE.0.) CALL TYPECK(-3,INFO,14,0,0)
             IF(ErrorFound()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C    SET PI
      PI=4*DATAN(1.D0)

C    CALCULATE THE AREA OF THE COLLECTOR
      AREA=LENGTH*WIDTH

C    CALCULATE THE TUBE-TO-TUBE DISTANCE
      W=WIDTH/DBLE(N_TUBES)

C    SET THE TRANSMITTANCE AT NORMAL INCIDENCE
      TAUALPHAN=1.-REFLECTANCE

C    DETERMINE INCIDENCE ANGLE MODIFIER
      IF(GT .GT. 0. .AND. ANGLE_INC .LT. 90.) THEN
```

```
C      USE RELATIONS OF BRANDEMUEHL FOR EFFECTIVE INCIDENCE ANGLES FOR DIFFUSE
       EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
       EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
       COSSLP=DCOS(SLOPE*RDCONV)
       FSKY=(1.+COSSLP)/2.
       FGND=(1.-COSSLP)/2.
       GDSKY=FSKY*GD
       GDGND=REFL_GROUND*FGND*GH

C      SET THE INCIDENCE ANGLE MODIFIERS
       XKATB=TAUALF(ANGLE_INC)
       XKATDS=TAUALF(EFFSKY)
       XKATDG=TAUALF(EFFGND)
       XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
       IF(XKAT.LE.0.) XKAT=0.
           ELSE
       XKAT=0.
      ENDIF

C   SET THE RESISTANCE FROM THE PV SURFACE TO THE ABSORBER
    R_T=R_ADHESIVE

C   SET THE RESISTANCE FROM THE ABSORBER TO THE BACK
    R_B=R_BACK+1./H_CONV_B

C   SEE IF THE DEVICE HAS FLUID FLOW
    IF(FLOW_IN.LE.0.) THEN

C      GUESS A PV SURFACE TEMPERATURE
       T_PV=T_AMB+1.
              T_PV_OLD=T_PV
              ICOUNT=1

C      CALCULATE THE PV CELL EFFICIENCY
10      FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
       FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
       EFF_PV=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))

C      GET THE RADIATION HEAT TRANSFER COEFFICIENT
       H_RAD=H_RADIATION(T_PV,T_SKY,EMISSIVITY)

C      SET THE ABSORBED SOLAR ENERGY
       S=TAUALPHAN*XKAT*GT*(1.-EFF_PV)

C      GET THE NEXT GUESS FOR THE PV TEMPERATURE FROM THE PV SURFACE ENERGY BALANCE
       T_PV=(S*(R_T+R_B)+H_RAD*T_SKY*(R_T+R_B)+T_BACK+H_CONV_T*
           1       (R_T+R_B)*T_AMB)/(1.+H_RAD*(R_T+R_B)+H_CONV_T*(R_T+R_B))

C      CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
       IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
           1      (ICOUNT.LT.1000)) THEN
              T_PV_OLD=T_PV
         ICOUNT=ICOUNT+1
         GOTO 10
       ENDIF

C      SET THE ABSORBER PLATE TEMPERATURE FROM AN ENERGY BALANCE ON THE ABSORBER
PLATE
       T_PV_MEAN=T_PV
            T_PLATE_MEAN=(R_T*T_BACK+R_B*T_PV_MEAN)/(R_T+R_B)

C      SET THE OUTLET FLUID CONDITIONS
       T_FLUID_MEAN=T_PLATE_MEAN
            T_FLUID_OUT=T_FLUID_MEAN

C      SET THE COLLECTOR USEFUL ENERGY GAIN TERMS AND THE THERMAL EFFICIENCY
       QU=0.
       Q_BASE=0.
       Q_FIN=0.
            EFF_THERMAL=0.

C      CALCULATE THE POWER OUTPUT FROM THE CELL EFFICIENCY
       POWER=TAUALPHAN*XKAT*EFF_PV*AREA*GT

C      SET SOME COLLECTOR CONSTANTS
       FR=0.

C      CALCULATE THE TOP CONVECTIVE LOSSES
       Q_TOP_CONV=AREA*H_CONV_T*(T_PV_MEAN-T_AMB)

C      CALCULATE THE TOP RADIATIVE LOSSES
       Q_TOP_RAD=AREA*H_RAD*(T_PV_MEAN-T_SKY)

C      CALCULATE THE BACK CONVECTIVE LOSSES
       Q_BACK=AREA*(T_PLATE_MEAN-T_BACK)/R_B

C      CALCULATE THE ABSORBED SOLAR RADIATION
            Q_ABS=AREA*GT*(1-EFF_PV)*TAUALPHAN*XKAT

C      CALCULATE THE COLLECTOR OVERALL LOSS COEFFICIENT
       IF(T_PLATE_MEAN.EQ.T_AMB) THEN
              UL=9999.
            ELSE
              UL=S/(T_PLATE_MEAN-T_AMB)
            ENDIF

           ELSE

C      CALCULATE THE TUBE-TO-PLATE BOND RESISTANCE
       IF((K_BOND.LE.0.).OR.(WIDTH_BOND.LE.0.)) THEN
              R_BOND=0.
```

```
            ELSE
         R_BOND=THICK_BOND/K_BOND/WIDTH_BOND
            ENDIF

C     GUESS A PV SURFACE TEMPERATURE
      T_PV=T_AMB+1.
            T_PV_OLD=T_PV
            ICOUNT=1

C     CALCULATE THE PV CELL EFFICIENCY
20     FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
      FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
      EFF_PV=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))

C     GET THE RADIATION HEAT TRANSFER COEFFICIENT
      H_RAD=H_RADIATION(T_PV,T_SKY,EMISSIVITY)

C     SET THE ABSORBED SOLAR ENERGY
      S=TAUALPHAN*XKAT*GT*(1.-EFF_PV)

C     SET SOME VARIABLES WE'LL NEED THROUGHOUT THE MODEL
      FPRIME=1./(H_RAD*R_T+H_CONV_T*R_T+1.)
      B=S+H_RAD*T_SKY+H_CONV_T*T_AMB+T_BACK/R_B/FPRIME
      J_FACTOR=1./R_T/FPRIME+1./R_B/FPRIME-1./R_T
      M=(FPRIME*J_FACTOR/K_ABSORBER/THICK_ABSORBER)**0.5
            N=2.*K_ABSORBER*THICK_ABSORBER*M*DTANH(M*(W-DIA_TUBE)/2.)
            U=DIA_TUBE*FPRIME*(H_RAD+H_CONV_T+1./R_B/FPRIME)
            Z=1./H_FLUID/PI/DIA_TUBE+R_BOND
            EPSILON=DIA_TUBE*FPRIME*B+N*B/J_FACTOR
            SIGMA=-U-N
            THETA=1+Z*U+N*Z

C     THE DERIVATION OF THIS MODEL IS EXPLAINED IN GREAT DETAIL IN THE ACCOMPANYING
TECHNICAL
C     DESCRIPTION MANUAL: TYPE560.PDF

C     SET THE FLUID OUTLET TEMPERATURE
      T_FLUID_OUT=(T_FLUID_IN+EPSILON/SIGMA)*DEXP(N_TUBES*SIGMA*
     1     LENGTH/THETA/FLOW_IN/CP_FLUID)-EPSILON/SIGMA

C     WITH THE OUTLET TEMPERATURE KNOWN, CALCULATE THE USEFUL ENERGY GAIN
      QU=FLOW_IN*CP_FLUID*(T_FLUID_OUT-T_FLUID_IN)
            QU_PRIME=QU/N_TUBES/LENGTH

C     SET THE MEAN FLUID TEMPERATURE IN THE Y-DIRECTION
      T_FLUID_MEAN=(T_FLUID_IN+EPSILON/SIGMA)*DEXP(N_TUBES*SIGMA*
     1     LENGTH/THETA/FLOW_IN/CP_FLUID)/(LENGTH*N_TUBES*SIGMA/THETA/
  1     FLOW_IN/CP_FLUID)-(T_FLUID_IN+EPSILON/SIGMA)/(LENGTH*N_TUBES
  1    *SIGMA/THETA/FLOW_IN/CP_FLUID)-EPSILON/SIGMA

C     WITH THE MEAN FLUID TEMPERATURE KNOWN, CALCULATE THE MEAN BASE
TEMPERATURE
      T_BASE_MEAN=T_FLUID_MEAN+QU_PRIME*Z

C     WITH THE MEAN BASE TEMPERATURE KNOWN, CALCULATE THE MEAN TEMPERATURE FOR
THE FIN ALONG THE X-AXIS
      T_FIN_MEAN=B/J_FACTOR+2.*(T_BASE_MEAN-B/J_FACTOR)*
     1     DTANH(M*(W-DIA_TUBE)/2.)/M/(W-DIA_TUBE)

C     NOW WE CAN CALCULATE THE MEAN ABSORBER TEMPERATURE
      T_PLATE_MEAN=(DIA_TUBE*T_BASE_MEAN+(W-DIA_TUBE)*T_FIN_MEAN)/W

C     WE CAN ALSO CALCULATE THE MEAN TEMPERATURE OF THE PV SURFACE ABOVE THE BASE
AND FIN
      T_PV_MEAN_BASE=R_T*FPRIME*(S+H_RAD*T_SKY+H_CONV_T*T_AMB+
     1     T_BASE_MEAN/R_T)
      T_PV_MEAN_FIN=R_T*FPRIME*(S+H_RAD*T_SKY+H_CONV_T*T_AMB+
     1     T_FIN_MEAN/R_T)

C     NOW WE CAN CALCULATE THE MEAN PV SURFACE TEMPERATURE
      T_PV_MEAN=(DIA_TUBE*T_PV_MEAN_BASE+(W-DIA_TUBE)*T_PV_MEAN_FIN)
     1     /W
            T_PV=T_PV_MEAN

C     CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
      IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
     1     (ICOUNT.LT.1000)) THEN
            T_PV_OLD=T_PV
        ICOUNT=ICOUNT+1
        GOTO 20
      ENDIF

C     CALCULATE THE ENERGY TO THE FLUID FROM THE BASE OF THE FIN
      Q_BASE=FPRIME*N_TUBES*DIA_TUBE*(S+H_RAD*(T_SKY-T_BASE_MEAN)+
     1     H_CONV_T*(T_AMB-T_BASE_MEAN)+1./R_B/FPRIME*(T_BACK-
  1     T_BASE_MEAN))

C     CALCULATE THE ENERGY TO THE FLUID FROM THE FIN
      Q_FIN=2.*K_ABSORBER*THICK_ABSORBER*M*(B/J_FACTOR-T_BASE_MEAN)*
     1     DTANH(M*(W-DIA_TUBE)/2.)/(W-DIA_TUBE)*N_TUBES*(W-DIA_TUBE)

C     CALCULATE THE THERMAL EFFICIENCY
      IF(GT.GT.0.) THEN
        EFF_THERMAL=QU/AREA/GT
            ELSE
              EFF_THERMAL=0.
      ENDIF

C     CALCULATE THE POWER OUTPUT FROM THE CELL EFFICIENCY
      POWER=TAUALPHAN*XKAT*EFF_PV*AREA*GT
```

```
C      CALCULATE THE TOP CONVECTIVE LOSSES
       Q_TOP_CONV=AREA*H_CONV_T*(T_PV_MEAN-T_AMB)

C      CALCULATE THE TOP RADIATIVE LOSSES
       Q_TOP_RAD=AREA*H_RAD*(T_PV_MEAN-T_SKY)

C      CALCULATE THE BACK CONVECTIVE LOSSES
       Q_BACK=AREA*(T_PLATE_MEAN-T_BACK)/R_B

C      CALCULATE THE ABSORBED SOLAR RADIATION
            Q_ABS=AREA*GT*(1-EFF_PV)*TAUALPHAN*XKAT

C      NOW SOLVE FOR THE FICTIONAL UL FROM QU=AREA*(S-UL(T_PLATE-T_AMB))
       IF(T_PLATE_MEAN.EQ.T_AMB) THEN
              UL=9999.
            ELSE
              UL=(S-QU/AREA)/(T_PLATE_MEAN-T_AMB)
            ENDIF

C      NOW CALCULATE THE COLLECTOR HEAT REMOVAL FACTOR FR
       IF((S-UL*(T_FLUID_IN-T_AMB)).EQ.0.) THEN
              FR=0.
            ELSE
         FR=QU/AREA/(S-UL*(T_FLUID_IN-T_AMB))
            ENDIF

          ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE OUTPUTS FROM THIS MODEL IN SEQUENTIAL ORDER AND GET OUT
     OUT(1)=T_FLUID_OUT
     OUT(2)=FLOW_IN
     OUT(3)=QU
     OUT(4)=POWER
     OUT(5)=EFF_PV
          OUT(6)=EFF_THERMAL
     OUT(7)=FR
     OUT(8)=T_PV_MEAN
     OUT(9)=T_FLUID_MEAN
          OUT(10)=XKAT
          OUT(11)=Q_TOP_CONV
          OUT(12)=Q_TOP_RAD
          OUT(13)=Q_BACK
          OUT(14)=Q_ABS
          OUT(15)=UL
          OUT(16)=FR*TAUALPHAN
          OUT(17)=FR*UL
          OUT(18)=Q_BASE
          OUT(19)=Q_FIN

             RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
     RETURN 1
     END
C-----------------------------------------------------------------------------------------------------------------------
```

TYPE 563- Unglazed PV/T collector with fluid stream passing through tubes bonded to an absorber plate located beneath the PV cells (considering conduction between the back of the collector and the roof)

```
      SUBROUTINE TYPE563(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C    THIS SUBROUTINE MODELS AN UNGLAZED PV/THERMAL COLLECTOR.  IN THIS VERSION, THE
WORKING FLUID IS WATER
C    CARRIED IN TUBES BONDED TO A PLATE.  THE PV MATERIAL IS ADHERED TO THE TOP OF THIS
PLATE.  THE PV SYSTEM
C    IS ASSUMED TO BE WORKING UNDER THE MAXIMUM POWER POINT ASSUMPTION.
C
C    THIS MODEL IS BASED ON A DERIVATION BY JEFF THORNTON OF THERMAL ENERGY SYSTEM
SPECIALISTS FROM THE
C    STANDARD TUBE-FIN SOLAR COLLECTOR ALGORITHMS PRESENTED BY DUFFIE AND
BECKMAN IN THE CLASSIC "SOLAR
C    ENGINEERING OF THERMAL PROCESSES' - SPECIFICALLY CHAPTER SIX.
C
C    LAST MODIFIED:
C       FEBRUARY 2004 - JWT - INITIAL PROGRAMMING
C       MODIFIED APRIL 2004 - FIXED THE PV POWER EQUATION TO INCLUDE XKAT AND TAU-ALPHA
FACTORS
C
C-----------------------------------------------------------------------------------------------------------------------
```



```
C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
              USE TrnsysConstants
              USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED BY THE MULTI-DLL VERSION OF TRNSYS
      !DEC$ATTRIBUTES DLLEXPORT :: TYPE563
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
      IMPLICIT NONE
              DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

      INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL
      CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    USER DECLARATIONS
      PARAMETER (NP=21,NI=13,NOUT=20,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
      DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT),T(ND),
              1  DTDT(ND)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
      DOUBLE PRECISION TAUALF,ANGLE_INC,B0,RDCONV,LENGTH,WIDTH,
              1  THICK_ABSORBER,K_ABSORBER,DIA_TUBE,WIDTH_BOND,THICK_BOND,
     1  K_BOND,R_ADHESIVE,R_BACK,CP_FLUID,REFLECTANCE,EMISSIVITY,
              1  T_FLUID_IN,FLOW_IN,T_AMB,T_SKY,T_INSIDE,GT,GH,GD,REFL_GROUND,
              1  SLOPE,H_CONV_T,H_FLUID,AREA,W,T_FLUID_OUT,QU,POWER,
              1  EFF_PV,EFF_THERMAL,FR,T_PLATE_MEAN,T_FLUID_MEAN,XKAT,T_BACK,
     1  Q_TOP_CONV,Q_TOP_RAD,Q_BACK,Q_BASE,Q_FIN,Q_ABS,FPRIME,UL,
              1  TAUALPHAN,EFFSKY,EFFGND,COSSLP,FSKY,FGND,GDSKY,GDGND,XKATB,
              1  XKATDS,XKATDG,T_REF,GT_REF,EFF_PV_REF,EFF_CORR_T,EFF_CORR_I,
              1  R_T,R_B,T_PV,T_PV_OLD,FACTOR_T,FACTOR_I,H_RAD,H_RADIATION,S,
              1  T_PV_MEAN,B,J_FACTOR,M,N,U,R_BOND,Z,PI,EPSILON,SIGMA,THETA,
              1  QU_PRIME,T_BASE_MEAN,T_FIN_MEAN,T_PV_MEAN_BASE,
              1  T_PV_MEAN_FIN,U_ROOF
              INTEGER N_TUBES,ICOUNT
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
      DATA RDCONV/0.017453292/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    FUNCTIONS
      TAUALF(ANGLE_INC)=1.-B0*(1./DMAX1(0.5,DCOS(ANGLE_INC*RDCONV))-1.)
     .          - (1.-B0)*(DMAX1(60.,ANGLE_INC)-60.)/30.
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
      TIME0=getSimulationStartTime()
      TFINAL=getSimulationStopTime()
      DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
      IF(INFO(7).EQ.-2) THEN
                 INFO(12)=16
                 RETURN 1
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
```

```
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
                  RETURN 1
               ENDIF
C------------------------------------------------------------------------------------------------------------------------

C------------------------------------------------------------------------------------------------------------------------
C    PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
      IF(INFO(13).GT.0) THEN
                  RETURN 1
               ENDIF
C------------------------------------------------------------------------------------------------------------------------

C------------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C       RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
         IUNIT=INFO(1)
                  ITYPE=INFO(2)

C       SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
         INFO(6)=NOUT
         INFO(9)=1
                  INFO(10)=0

C       CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED
                  CALL TYPECK(1,INFO,NI,NP,ND)

C       SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
         DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','IR1','IR1','IR1',
               1          'DM1','DG1','DG1','HT1','HT1'/
         DATA OCHECK/'TE1','MF1','PW1','PW1','DM1','DM1','DM1','TE1',
               1          'TE1','DM1','PW1','PW1','PW1','PW1','HT1','DM1',
      1          'HT1','PW1','PW1','TE1'/

C       CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
         CALL RCHECK(INFO,YCHECK,OCHECK)

C       RETURN TO THE CALLING PROGRAM
         RETURN 1

      ENDIF
C------------------------------------------------------------------------------------------------------------------------

C------------------------------------------------------------------------------------------------------------------------
C    DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C       SET THE UNIT NUMBER FOR FUTURE CALLS
         IUNIT=INFO(1)
         ITYPE=INFO(2)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
         LENGTH=PAR(1)
                  WIDTH=PAR(2)
                  THICK_ABSORBER=PAR(3)
                  K_ABSORBER=PAR(4)
                  N_TUBES=JFIX(PAR(5)+0.5)
                  DIA_TUBE=PAR(6)
                  WIDTH_BOND=PAR(7)
                  THICK_BOND=PAR(8)
                  K_BOND=PAR(9)
                  R_ADHESIVE=PAR(10)
                  R_BACK=PAR(11)
                  U_ROOF=PAR(12)
                  CP_FLUID=PAR(13)
         REFLECTANCE=PAR(14)
                  EMISSIVITY=PAR(15)
         B0=PAR(16)
         T_REF=PAR(17)
         GT_REF=PAR(18)
         EFF_PV_REF=PAR(19)
         EFF_CORR_T=PAR(20)
         EFF_CORR_I=PAR(21)

C       CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
         IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
         IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
         IF(THICK_ABSORBER.LE.0.) CALL TYPECK(-4,INFO,0,3,0)
         IF(K_ABSORBER.LE.0.) CALL TYPECK(-4,INFO,0,4,0)
         IF(N_TUBES.LT.1) CALL TYPECK(-4,INFO,0,5,0)
         IF(DIA_TUBE.LE.0.) CALL TYPECK(-4,INFO,0,6,0)
                  IF(DBLE(N_TUBES)*DIA_TUBE.GT.WIDTH) CALL TYPECK(-4,INFO,0,6,0)
         IF(R_ADHESIVE.LE.0.) CALL TYPECK(-4,INFO,0,10,0)
         IF(R_BACK.LE.0.) CALL TYPECK(-4,INFO,0,11,0)
         IF(U_ROOF.LE.0.) CALL TYPECK(-4,INFO,0,12,0)
         IF(CP_FLUID.LE.0.) CALL TYPECK(-4,INFO,0,13,0)
         IF(REFLECTANCE.LT.0.) CALL TYPECK(-4,INFO,0,14,0)
         IF(REFLECTANCE.GT.1.) CALL TYPECK(-4,INFO,0,14,0)
         IF(EMISSIVITY.LT.0.) CALL TYPECK(-4,INFO,0,15,0)
         IF(EMISSIVITY.GT.1.) CALL TYPECK(-4,INFO,0,15,0)
         IF(B0.LT.0.) CALL TYPECK(-4,INFO,0,16,0)
         IF(B0.GT.1.) CALL TYPECK(-4,INFO,0,16,0)
         IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,18,0)
```

```
      IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,19,0)
      IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,19,0)

C     PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
      OUT(1)=XIN(1)
      OUT(2:7)=0.
      OUT(8)=XIN(1)
      OUT(9)=XIN(1)
            OUT(10:19)=0.
            OUT(20)=XIN(5)

C     RETURN TO THE CALLING PROGRAM
      RETURN 1

    ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
    IF(INFO(1).NE.IUNIT) THEN

C     RESET THE UNIT NUMBER
            IUNIT=INFO(1)
            ITYPE=INFO(2)

C     READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
      LENGTH=PAR(1)
            WIDTH=PAR(2)
            THICK_ABSORBER=PAR(3)
            K_ABSORBER=PAR(4)
            N_TUBES=JFIX(PAR(5)+0.5)
            DIA_TUBE=PAR(6)
            WIDTH_BOND=PAR(7)
            THICK_BOND=PAR(8)
            K_BOND=PAR(9)
            R_ADHESIVE=PAR(10)
            R_BACK=PAR(11)
            U_ROOF=PAR(12)
            CP_FLUID=PAR(13)
      REFLECTANCE=PAR(14)
            EMISSIVITY=PAR(15)
      B0=PAR(16)
      T_REF=PAR(17)
      GT_REF=PAR(18)
      EFF_PV_REF=PAR(19)
      EFF_CORR_T=PAR(20)
       EFF_CORR_I=PAR(21)
     ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
     T_FLUID_IN=XIN(1)
            FLOW_IN=XIN(2)
            T_AMB=XIN(3)
            T_SKY=XIN(4)
            T_INSIDE=XIN(5)
     GT=XIN(6)
     GH=XIN(7)
     GD=XIN(8)
     REFL_GROUND=XIN(9)
     ANGLE_INC=XIN(10)
     SLOPE=XIN(11)
     H_CONV_T=XIN(12)
     H_FLUID=XIN(13)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   CHECK THE INPUTS FOR PROBLEMS
     IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
     IF(GT.LT.0.) CALL TYPECK(-3,INFO,6,0,0)
     IF(GH.LT.0.) CALL TYPECK(-3,INFO,7,0,0)
     IF(GD.LT.0.) CALL TYPECK(-3,INFO,8,0,0)
     IF(REFL_GROUND.LT.0.) CALL TYPECK(-3,INFO,9,0,0)
     IF(REFL_GROUND.GT.1.) CALL TYPECK(-3,INFO,9,0,0)
     IF(H_CONV_T.LE.0.) CALL TYPECK(-3,INFO,12,0,0)
     IF(H_FLUID.LE.0.) CALL TYPECK(-3,INFO,13,0,0)
            IF(ERRORFOUND()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C   SET PI
     PI=4*DATAN(1.D0)

C   CALCULATE THE AREA OF THE COLLECTOR
     AREA=LENGTH*WIDTH

C   CALCULATE THE TUBE-TO-TUBE DISTANCE
     W=WIDTH/DBLE(N_TUBES)

C   SET THE TRANSMITTANCE AT NORMAL INCIDENCE
     TAUALPHAN=1.-REFLECTANCE

C   DETERMINE INCIDENCE ANGLE MODIFIER
```

```
      IF(GT .GT. 0. .AND. ANGLE_INC .LT. 90.) THEN

C       USE RELATIONS OF BRANDEMUEHL FOR EFFECTIVE INCIDENCE ANGLES FOR DIFFUSE
        EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
        EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
        COSSLP=DCOS(SLOPE*RDCONV)
        FSKY=(1.+COSSLP)/2.
        FGND=(1.-COSSLP)/2.
        GDSKY=FSKY*GD
        GDGND=REFL_GROUND*FGND*GH

C       SET THE INCIDENCE ANGLE MODIFIERS
        XKATB=TAUALF(ANGLE_INC)
        XKATDS=TAUALF(EFFSKY)
        XKATDG=TAUALF(EFFGND)
        XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
        IF(XKAT.LE.0.) XKAT=0.
            ELSE
        XKAT=0.
      ENDIF

C     SET THE RESISTANCE FROM THE PV SURFACE TO THE ABSORBER
      R_T=R_ADHESIVE

C     SET THE RESISTANCE FROM THE ABSORBER TO THE BACK SURFACE (ZONE AIR/ROOF
INTERFACE)
      R_B=R_BACK+1./U_ROOF

C     SEE IF THE DEVICE HAS FLUID FLOW
      IF(FLOW_IN.LE.0.) THEN

C       GUESS A PV SURFACE TEMPERATURE
        T_PV=T_AMB+1.
             T_PV_OLD=T_PV
             ICOUNT=1

C       CALCULATE THE PV CELL EFFICIENCY
10      FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
        FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
        EFF_PV=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))

C       GET THE RADIATION HEAT TRANSFER COEFFICIENT
        H_RAD=H_RADIATION(T_PV,T_SKY,EMISSIVITY)

C       SET THE ABSORBED SOLAR ENERGY
        S=TAUALPHAN*XKAT*GT*(1.-EFF_PV)

C       GET THE NEXT GUESS FOR THE PV TEMPERATURE FROM THE PV SURFACE ENERGY BALANCE
        T_PV=(S*(R_T+R_B)+H_RAD*T_SKY*(R_T+R_B)+T_INSIDE+H_CONV_T*
     1      (R_T+R_B)*T_AMB)/(1.+H_RAD*(R_T+R_B)+H_CONV_T*(R_T+R_B))

C       CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
        IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
     1     (ICOUNT.LT.1000)) THEN
             T_PV_OLD=T_PV
          ICOUNT=ICOUNT+1
          GOTO 10
        ENDIF

C       SET THE ABSORBER PLATE TEMPERATURE FROM AN ENERGY BALANCE ON THE ABSORBER
PLATE
        T_PV_MEAN=T_PV
             T_PLATE_MEAN=(R_T*T_INSIDE+R_B*T_PV_MEAN)/(R_T+R_B)

C       SET THE OUTLET FLUID CONDITIONS
        T_FLUID_MEAN=T_PLATE_MEAN
             T_FLUID_OUT=T_FLUID_MEAN

C       SET THE COLLECTOR USEFUL ENERGY GAIN TERMS AND THE THERMAL EFFICIENCY
        QU=0.
        Q_BASE=0.
        Q_FIN=0.
             EFF_THERMAL=0.

C       CALCULATE THE POWER OUTPUT FROM THE CELL EFFICIENCY
        POWER=TAUALPHAN*XKAT*EFF_PV*AREA*GT

C       SET SOME COLLECTOR CONSTANTS
        FR=0.

C       CALCULATE THE TOP CONVECTIVE LOSSES
        Q_TOP_CONV=AREA*H_CONV_T*(T_PV_MEAN-T_AMB)

C       CALCULATE THE TOP RADIATIVE LOSSES
        Q_TOP_RAD=AREA*H_RAD*(T_PV_MEAN-T_SKY)

C       CALCULATE THE BACK CONVECTIVE LOSSES
        Q_BACK=AREA*(T_PLATE_MEAN-T_INSIDE)/R_B

C       CALCULATE THE ABSORBED SOLAR RADIATION
             Q_ABS=AREA*GT*(1-EFF_PV)*TAUALPHAN*XKAT

C       CALCULATE THE TEMPERATURE OF THE INTERFACE BETWEEN THE COLLECTOR AND THE
ROOF
        T_BACK=T_PLATE_MEAN-Q_BACK*R_BACK/AREA

C       CALCULATE THE COLLECTOR OVERALL LOSS COEFFICIENT
        IF(T_PLATE_MEAN.EQ.T_AMB) THEN
               UL=9999.
             ELSE
               UL=S/(T_PLATE_MEAN-T_AMB)
             ENDIF
```

```
            ELSE

C     CALCULATE THE TUBE-TO-PLATE BOND RESISTANCE
      IF((K_BOND.LE.0.).OR.(WIDTH_BOND.LE.0.)) THEN
                R_BOND=0.
              ELSE
        R_BOND=THICK_BOND/K_BOND/WIDTH_BOND
              ENDIF

C     GUESS A PV SURFACE TEMPERATURE
      T_PV=T_AMB+1.
              T_PV_OLD=T_PV
              ICOUNT=1

C     CALCULATE THE PV CELL EFFICIENCY
20     FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
      FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
      EFF_PV=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))

C     GET THE RADIATION HEAT TRANSFER COEFFICIENT
      H_RAD=H_RADIATION(T_PV,T_SKY,EMISSIVITY)

C     SET THE ABSORBED SOLAR ENERGY
      S=TAUALPHAN*XKAT*GT*(1.-EFF_PV)

C     SET SOME VARIABLES WE'LL NEED THROUGHOUT THE MODEL
      FPRIME=1./(H_RAD*R_T+H_CONV_T*R_T+1.)
      B=S+H_RAD*T_SKY+H_CONV_T*T_AMB+T_INSIDE/R_B/FPRIME
      J_FACTOR=1./R_T/FPRIME+1./R_B/FPRIME-1./R_T
      M=(FPRIME*J_FACTOR/K_ABSORBER/THICK_ABSORBER)**0.5
              N=2.*K_ABSORBER*THICK_ABSORBER*M*DTANH(M*(W-DIA_TUBE)/2.)
              U=DIA_TUBE*FPRIME*(H_RAD+H_CONV_T+1./R_B/FPRIME)
              Z=1./H_FLUID/PI/DIA_TUBE+R_BOND
              EPSILON=DIA_TUBE*FPRIME*B+N*B/J_FACTOR
              SIGMA=-U-N
              THETA=1+Z*U+N*Z

C     THE DERIVATION OF THIS MODEL IS EXPLAINED IN GREAT DETAIL IN THE ACCOMPANYING
TECHNICAL
C     DESCRIPTION MANUAL: TYPE563.PDF

C     SET THE FLUID OUTLET TEMPERATURE
      T_FLUID_OUT=(T_FLUID_IN+EPSILON/SIGMA)*DEXP(N_TUBES*SIGMA*
           1     LENGTH/THETA/FLOW_IN/CP_FLUID)-EPSILON/SIGMA

C     WITH THE OUTLET TEMPERATURE KNOWN, CALCULATE THE USEFUL ENERGY GAIN
      QU=FLOW_IN*CP_FLUID*(T_FLUID_OUT-T_FLUID_IN)
              QU_PRIME=QU/N_TUBES/LENGTH

C     SET THE MEAN FLUID TEMPERATURE IN THE Y-DIRECTION
      T_FLUID_MEAN=(T_FLUID_IN+EPSILON/SIGMA)*DEXP(N_TUBES*SIGMA*
           1     LENGTH/THETA/FLOW_IN/CP_FLUID)/(LENGTH*N_TUBES*SIGMA/THETA/
   1     FLOW_IN/CP_FLUID)-(T_FLUID_IN+EPSILON/SIGMA)/(LENGTH*N_TUBES
   1    *SIGMA/THETA/FLOW_IN/CP_FLUID)-EPSILON/SIGMA

C     WITH THE MEAN FLUID TEMPERATURE KNOWN, CALCULATE THE MEAN BASE
TEMPERATURE
      T_BASE_MEAN=T_FLUID_MEAN+QU_PRIME*Z

C     WITH THE MEAN BASE TEMPERATURE KNOWN, CALCULATE THE MEAN TEMPERATURE FOR
THE FIN ALONG THE X-AXIS
      T_FIN_MEAN=B/J_FACTOR+2.*(T_BASE_MEAN-B/J_FACTOR)*
           1     DTANH(M*(W-DIA_TUBE)/2.)/M/(W-DIA_TUBE)

C     NOW WE CAN CALCULATE THE MEAN ABSORBER TEMPERATURE
      T_PLATE_MEAN=(DIA_TUBE*T_BASE_MEAN+(W-DIA_TUBE)*T_FIN_MEAN)/W

C     WE CAN ALSO CALCULATE THE MEAN TEMPERATURE OF THE PV SURFACE ABOVE THE BASE
AND FIN
      T_PV_MEAN_BASE=R_T*FPRIME*(S+H_RAD*T_SKY+H_CONV_T*T_AMB+
           1     T_BASE_MEAN/R_T)
      T_PV_MEAN_FIN=R_T*FPRIME*(S+H_RAD*T_SKY+H_CONV_T*T_AMB+
           1     T_FIN_MEAN/R_T)

C     NOW WE CAN CALCULATE THE MEAN PV SURFACE TEMPERATURE
      T_PV_MEAN=(DIA_TUBE*T_PV_MEAN_BASE+(W-DIA_TUBE)*T_PV_MEAN_FIN)
           1     /W
               T_PV=T_PV_MEAN

C     CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
      IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
           1     (ICOUNT.LT.1000)) THEN
                T_PV_OLD=T_PV
        ICOUNT=ICOUNT+1
        GOTO 20
      ENDIF

C     CALCULATE THE ENERGY TO THE FLUID FROM THE BASE OF THE FIN
      Q_BASE=FPRIME*N_TUBES*DIA_TUBE*(S+H_RAD*(T_SKY-T_BASE_MEAN)+
           1     H_CONV_T*(T_AMB-T_BASE_MEAN)+1./R_B/FPRIME*(T_INSIDE-
   1     T_BASE_MEAN))

C     CALCULATE THE ENERGY TO THE FLUID FROM THE FIN
      Q_FIN=2.*K_ABSORBER*THICK_ABSORBER*M*(B/J_FACTOR-T_BASE_MEAN)*
           1     DTANH(M*(W-DIA_TUBE)/2.)/(W-DIA_TUBE)*N_TUBES*(W-DIA_TUBE)

C     CALCULATE THE THERMAL EFFICIENCY
      IF(GT.GT.0.) THEN
        EFF_THERMAL=QU/AREA/GT
               ELSE
                EFF_THERMAL=0.
```

```
      ENDIF

C     CALCULATE THE POWER OUTPUT FROM THE CELL EFFICIENCY
      POWER=TAUALPHAN*XKAT*EFF_PV*AREA*GT

C     CALCULATE THE TOP CONVECTIVE LOSSES
      Q_TOP_CONV=AREA*H_CONV_T*(T_PV_MEAN-T_AMB)

C     CALCULATE THE TOP RADIATIVE LOSSES
      Q_TOP_RAD=AREA*H_RAD*(T_PV_MEAN-T_SKY)

C     CALCULATE THE BACK CONVECTIVE LOSSES
      Q_BACK=AREA*(T_PLATE_MEAN-T_INSIDE)/R_B

C     CALCULATE THE ABSORBED SOLAR RADIATION
            Q_ABS=AREA*GT*(1-EFF_PV)*TAUALPHAN*XKAT

C     CALCULATE THE TEMPERATURE OF THE INTERFACE BETWEEN THE COLLECTOR AND THE
ROOF
      T_BACK=T_PLATE_MEAN-Q_BACK*R_BACK/AREA

C     NOW SOLVE FOR THE FICTIONAL UL FROM QU=AREA*(S-UL(T_PLATE-T_AMB))
      IF(T_PLATE_MEAN.EQ.T_AMB) THEN
             UL=9999.
            ELSE
             UL=(S-QU/AREA)/(T_PLATE_MEAN-T_AMB)
            ENDIF

C     NOW CALCULATE THE COLLECTOR HEAT REMOVAL FACTOR FR
      IF((S-UL*(T_FLUID_IN-T_AMB)).EQ.0.) THEN
             FR=0.
            ELSE
       FR=QU/AREA/(S-UL*(T_FLUID_IN-T_AMB))
            ENDIF

           ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   SET THE OUTPUTS FROM THIS MODEL IN SEQUENTIAL ORDER AND GET OUT
     OUT(1)=T_FLUID_OUT
     OUT(2)=FLOW_IN
     OUT(3)=QU
     OUT(4)=POWER
     OUT(5)=EFF_PV
           OUT(6)=EFF_THERMAL
     OUT(7)=FR
     OUT(8)=T_PV_MEAN
     OUT(9)=T_FLUID_MEAN
           OUT(10)=XKAT
           OUT(11)=Q_TOP_CONV
           OUT(12)=Q_TOP_RAD
           OUT(13)=Q_BACK
           OUT(14)=Q_ABS
           OUT(15)=UL
           OUT(16)=FR*TAUALPHAN
           OUT(17)=FR*UL
           OUT(18)=Q_BASE
           OUT(19)=Q_FIN
           OUT(20)=T_BACK

           RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
     RETURN 1
     END
C-----------------------------------------------------------------------------------------------------------------------
```

TYPE 566-Building integrated photovoltaic system (glazed, air)

```
      SUBROUTINE TYPE566(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C     THIS COMPONENT MODELS A BUILDING-INTEGRATED PHOTOVOLATIC SYSTEM.  IN THIS MODEL THERE IS ASSUMED TO BE
C     A SINGLE GLASS COVER OVER THE PV MATERIAL.  THE BACK OF THE BIPV SYSTEM INTERACTS WITH MODELS WHERE THE
C     BACK-SIDE AIR TEMPERATURE AND BACK-SIDE RADIATIVE SURFACE TEMPERATURE ARE KNOWN. THE CONVECTION
C     CALCULATIONS ARE FROM CORRELATIONS PROVIDED BY "INTRODUCTION TO HEAT TRANSFER" BY INCROPERA AND DEWITT.
C
C     LAST MODIFIED:
C        APRIL 2004 - JEFF THORNTON - ORIGINAL CODING
C-----------------------------------------------------------------------------------------------------------------------
```



```
C-----------------------------------------------------------------------------------------------------------------------
      !Export this subroutine for its use in external DLLs.
      !DEC$ATTRIBUTES DLLEXPORT :: TYPE566
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
              USE TrnsysConstants
              USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
      IMPLICIT NONE
              DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

      INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL,NPAR,NIN
      CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED PARAMETERS FOR THE SIZING OF THE ARRAYS
      PARAMETER (NP=21,NI=16,NOUT=18,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
      DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
      DOUBLE PRECISION RDCONV,ANGLE_INC,TAUALF,B0,
              1   AREA,FLOW_IN,T_AMB,GT,GH,GDH,REFL_GROUND,SLOPE,EMISS_COVER,
     1  TAUALPHAN,EFFSKY,EFFGND,COSSLP,FSKY,FGND,GDSKY,GDGND,XKATB,
     1  XKATDS,F_PRIME,XKATDG,T_FLUID_MEAN,T_SKY,H_CONV_T,X(2),
     1  H_CONV_B,H_FLUID,QU,EFF_THERMAL,S,R_1,R_2,R_3,Q_TOP_CONV,N,
     1  Q_TOP_RAD,Q_ABS,H_RADIATION,EMISS_1,EMISS_2,EMISS_BACK,T_ZONE,
     1  T_ZONE_RAD,Q_BACK_CONV,Q_BACK_RAD,T_1,T_2,T_3,H_RAD_12,
     1  H_RAD_GRAY,G_PRIME,M,AA,CC,K_COVER,THICK_COVER,RI_COVER,
     1  EXT_COVER,ABS_PLATE,EFF_PV,Y(1),EFF_PV_REF,T_REF,GT_REF,
     1  EFF_CORR_T,EFF_CORR_I,T_FLUID_IN,KL_COVER,RHO_DIFFUSE,
     1  TAU_ALPHA,COSSLOPE,XKAT,T_COVER,T_PV,T_PV_OLD,FACTOR_T,
     1  FACTOR_I,PV_EFFICIENCY,POWER,H_RAD_T,H_RAD_B,H_PRIME,J,
     1  T_FLUID_OUT,AIRPROPS(5),PI,LENGTH,WIDTH,THICK_CHANNEL,P_ATM,
     1  P_KPA,DIAMETER,T_PROPS_K,RHO_AIR,VISC_AIR,PRANDTL_AIR,K_AIR,
     1  CP_AIR,REYNOLDS,NUSSELT,T_1_K,T_2_K,T_SURF
              1
      INTEGER ICOUNT,MODE_IAM,MODE_EFF,LU_DATA,N_TEMPS,N_RADS,NX(2)
      CHARACTER(LEN=MAXMESSAGELENGTH)::MESSAGE1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
      DATA RDCONV/0.017453292/,PI/3.14159265358979/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    FUNCTIONS
      TAUALF(ANGLE_INC)=1.-B0*(1./DMAX1(0.5,DCOS(ANGLE_INC*RDCONV))-1.)
     .          - (1.-B0)*(DMAX1(60.,ANGLE_INC)-60.)/30.
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ERROR MESSAGES
      MESSAGE1='The BIPV model was unable to find a solution for the cel
              1l temperature at the given timestep.  Please report this error to
     1your TRNSYS distributor.'
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
      TIME0=getSimulationStartTime()
      TFINAL=getSimulationStopTime()
      DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
      IF(INFO(7).EQ.-2) THEN
                INFO(12)=16
                RETURN 1
              ENDIF
```

```
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
                 RETURN 1
               ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
      IF(INFO(13).GT.0) THEN
                 RETURN 1
               ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C       RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
         IUNIT=INFO(1)
                 ITYPE=INFO(2)

C       SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
         INFO(6)=NOUT
         INFO(9)=1
                 INFO(10)=0

C       SET THE NUMBER OF PARAMETERS AND INPUTS
         MODE_IAM=JFIX(PAR(12)+0.5)
         MODE_EFF=JFIX(PAR(13)+0.5)
         NPAR=13
                 NIN=15

         IF(MODE_IAM.LT.1) CALL TYPECK(-4,INFO,0,12,0)
         IF(MODE_IAM.GT.2) CALL TYPECK(-4,INFO,0,12,0)
         IF(MODE_EFF.LT.1) CALL TYPECK(-4,INFO,0,13,0)
         IF(MODE_EFF.GT.3) CALL TYPECK(-4,INFO,0,13,0)
                 IF(ERRORFOUND()) RETURN 1

                 IF(MODE_IAM.EQ.1) THEN
                   NPAR=NPAR+2
                 ELSE IF(MODE_IAM.EQ.2) THEN
                   NPAR=NPAR+3
                 ENDIF

                 IF(MODE_EFF.EQ.1) THEN
                   NPAR=NPAR+5
         ELSE IF(MODE_EFF.EQ.2) THEN
                   NPAR=NPAR+3
         ELSE IF(MODE_EFF.EQ.3) THEN
                   NIN=NIN+1
                 ENDIF

C       CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED IN
C       THE TRNSYS INPUT FILE
                 CALL TYPECK(1,INFO,NIN,NPAR,ND)

C       SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
         DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','TE1','IR1','IR1',
              1              'IR1','DM1','DG1','DG1','HT1','HT1','PR2','DM1'/
         DATA OCHECK/'TE1','MF1','PW1','DM1','PW1','DM1','TE1','TE1',
              1              'TE1','TE1','TE1','TE1','DM1','PW1','PW1','PW1',
    1             'PW1','PW1'/

C       CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
         CALL RCHECK(INFO,YCHECK,OCHECK)

C       RETURN TO THE CALLING PROGRAM
         RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C       SET THE UNIT NUMBER FOR FUTURE CALLS
         IUNIT=INFO(1)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
         LENGTH=PAR(1)
         WIDTH=PAR(2)
                 EMISS_COVER=PAR(3)
                 K_COVER=PAR(4)
                 THICK_COVER=PAR(5)
         R_2=PAR(6)
                 EMISS_1=PAR(7)
                 EMISS_2=PAR(8)
         R_3=PAR(9)
                 EMISS_BACK=PAR(10)
                 THICK_CHANNEL=PAR(11)
         MODE_IAM=JFIX(PAR(12)+0.5)
```

```
      MODE_EFF=JFIX(PAR(13)+0.5)
      NPAR=13


C     CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
      IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
      IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
      IF(EMISS_COVER.LT.0.) CALL TYPECK(-4,INFO,0,3,0)
      IF(EMISS_COVER.GT.1.) CALL TYPECK(-4,INFO,0,3,0)
      IF(K_COVER.LE.0.) CALL TYPECK(-4,INFO,0,4,0)
      IF(THICK_COVER.LE.0.) CALL TYPECK(-4,INFO,0,5,0)
      IF(R_2.LE.0.) CALL TYPECK(-4,INFO,0,6,0)
      IF(EMISS_1.LT.0.) CALL TYPECK(-4,INFO,0,7,0)
      IF(EMISS_1.GT.1.) CALL TYPECK(-4,INFO,0,7,0)
      IF(EMISS_2.LT.0.) CALL TYPECK(-4,INFO,0,8,0)
      IF(EMISS_2.GT.1.) CALL TYPECK(-4,INFO,0,8,0)
      IF(R_3.LE.0.) CALL TYPECK(-4,INFO,0,9,0)
      IF(EMISS_BACK.LT.0.) CALL TYPECK(-4,INFO,0,10,0)
      IF(EMISS_BACK.GT.1.) CALL TYPECK(-4,INFO,0,10,0)
      IF(THICK_CHANNEL.LT.0.) CALL TYPECK(-4,INFO,0,11,0)

            IF(ERRORFOUND()) RETURN 1

C     GET THE IAM MODE SPECIFIC PARAMETERS
            IF(MODE_IAM.EQ.1) THEN
              TAUALPHAN=PAR(NPAR+1)
        B0=PAR(NPAR+2)

        IF(TAUALPHAN.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(TAUALPHAN.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(B0.LT.0.) CALL TYPECK(-4,INFO,0,NPAR+2,0)
        IF(B0.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+2,0)

              NPAR=NPAR+2

            ELSE IF(MODE_IAM.EQ.2) THEN
              ABS_PLATE=PAR(NPAR+1)
              RI_COVER=PAR(NPAR+2)
              EXT_COVER=PAR(NPAR+3)

        IF(ABS_PLATE.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(ABS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(RI_COVER.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+2,0)
        IF(EXT_COVER.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+3,0)

              NPAR=NPAR+3

            ENDIF

C     GET THE PV EFFICIENCY SPECIFIC PARAMETERS
            IF(MODE_EFF.EQ.1) THEN
        EFF_PV_REF=PAR(NPAR+1)
        T_REF=PAR(NPAR+2)
        GT_REF=PAR(NPAR+3)
        EFF_CORR_T=PAR(NPAR+4)
        EFF_CORR_I=PAR(NPAR+5)

        IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
        IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+3,0)

      ELSE IF(MODE_EFF.EQ.2) THEN
              LU_DATA=JFIX(PAR(NPAR+1)+0.5)
              N_TEMPS=JFIX(PAR(NPAR+2)+0.5)
              N_RADS=JFIX(PAR(NPAR+3)+0.5)

                    IF(LU_DATA.LT.10) CALL TYPECK(-4,INFO,0,NPAR+1,0)
              IF(N_TEMPS.LT.2) CALL TYPECK(-4,INFO,0,NPAR+2,0)
              IF(N_RADS.LT.2) CALL TYPECK(-4,INFO,0,NPAR+3,0)

            ENDIF

C     PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
      OUT(1)=XIN(1)
      OUT(2:6)=0.
      OUT(7)=T_AMB
      OUT(8)=T_AMB
      OUT(9)=XIN(1)
      OUT(10)=XIN(1)
      OUT(11)=XIN(1)
            OUT(12)=T_ZONE
            OUT(13:18)=0.

C     RETURN TO THE CALLING PROGRAM
      RETURN 1

     ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
     IF(INFO(1).NE.IUNIT) THEN

C     RESET THE UNIT NUMBER
            IUNIT=INFO(1)
            ITYPE=INFO(2)
```

```
C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
      LENGTH=PAR(1)
      WIDTH=PAR(2)
            EMISS_COVER=PAR(3)
            K_COVER=PAR(4)
            THICK_COVER=PAR(5)
      R_2=PAR(6)
            EMISS_1=PAR(7)
            EMISS_2=PAR(8)
      R_3=PAR(9)
            EMISS_BACK=PAR(10)
            THICK_CHANNEL=PAR(11)
      MODE_IAM=JFIX(PAR(12)+0.5)
      MODE_EFF=JFIX(PAR(13)+0.5)
      NPAR=13
            IF(MODE_IAM.EQ.1) THEN
              TAUALPHAN=PAR(NPAR+1)
        B0=PAR(NPAR+2)
              NPAR=NPAR+2
            ELSE IF(MODE_IAM.EQ.2) THEN
              ABS_PLATE=PAR(NPAR+1)
              RI_COVER=PAR(NPAR+2)
              EXT_COVER=PAR(NPAR+3)
              NPAR=NPAR+3
            ENDIF
            IF(MODE_EFF.EQ.1) THEN
        EFF_PV_REF=PAR(NPAR+1)
        T_REF=PAR(NPAR+2)
        GT_REF=PAR(NPAR+3)
        EFF_CORR_T=PAR(NPAR+4)
        EFF_CORR_I=PAR(NPAR+5)
      ELSE IF(MODE_EFF.EQ.2) THEN
              LU_DATA=JFIX(PAR(NPAR+1)+0.5)
              N_TEMPS=JFIX(PAR(NPAR+2)+0.5)
              N_RADS=JFIX(PAR(NPAR+3)+0.5)
            ENDIF

     ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
60    T_FLUID_IN=XIN(1)
     FLOW_IN=XIN(2)
     T_AMB=XIN(3)
     T_SKY=XIN(4)
     T_ZONE=XIN(5)
     T_ZONE_RAD=XIN(6)
     GT=XIN(7)
     GH=XIN(8)
     GDH=XIN(9)
     REFL_GROUND=XIN(10)
     ANGLE_INC=XIN(11)
     SLOPE=XIN(12)
     H_CONV_T=XIN(13)
     H_CONV_B=XIN(14)
            P_ATM=XIN(15)
            IF(MODE_EFF.EQ.3) THEN
              EFF_PV=XIN(16)
            ELSE
              EFF_PV=0.
            ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    CHECK THE INPUTS FOR PROBLEMS
     IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
     IF(GT.LT.0.) CALL TYPECK(-3,INFO,7,0,0)
     IF(GH.LT.0.) CALL TYPECK(-3,INFO,8,0,0)
     IF(GDH.LT.0.) CALL TYPECK(-3,INFO,9,0,0)
     IF(REFL_GROUND.LT.0.) CALL TYPECK(-3,INFO,10,0,0)
     IF(REFL_GROUND.GT.1.) CALL TYPECK(-3,INFO,10,0,0)
     IF(H_CONV_T.LT.0.) CALL TYPECK(-3,INFO,13,0,0)
     IF(H_CONV_B.LT.0.) CALL TYPECK(-3,INFO,14,0,0)
     IF(P_ATM.LE.0.) CALL TYPECK(-3,INFO,15,0,0)
     IF(EFF_PV.LT.0.) CALL TYPECK(-3,INFO,16,0,0)
     IF(EFF_PV.GT.1.) CALL TYPECK(-3,INFO,16,0,0)

            IF(ERRORFOUND()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C    SET SOME GEOMETRIC PARAMETERS
     AREA=LENGTH*WIDTH
            DIAMETER=4.*WIDTH*THICK_CHANNEL/(2.*WIDTH+2.*THICK_CHANNEL)

C    SET THE INCIDENCE ANGLE MODIFIER BASED ON THE MODE
     XKAT=1.
            IF(MODE_IAM.EQ.1) THEN

C      DETERMINE INCIDENCE ANGLE MODIFIER
       IF(GT .GT. 0. .AND. ANGLE_INC .LT. 90.) THEN

C        USE RELATIONS OF BRANDEMUEHL FOR EFFECTIVE INCIDENCE ANGLES FOR DIFFUSE
        EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
        EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
        COSSLP=DCOS(SLOPE*RDCONV)
        FSKY=(1.+COSSLP)/2.
```

```
      FGND=(1.-COSSLP)/2.
      GDSKY=FSKY*GDH
      GDGND=REFL_GROUND*FGND*GH

C      SET THE INCIDENCE ANGLE MODIFIERS
      XKATB=TAUALF(ANGLE_INC)
      XKATDS=TAUALF(EFFSKY)
      XKATDG=TAUALF(EFFGND)
      XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
      IF(XKAT.LE.0.) XKAT=0.
     ELSE
      XKAT=0.
     ENDIF
         ELSE IF(MODE_IAM.EQ.2) THEN

C     GET THE TRANSMITTANCE-ABSORPTANCE PRODUCT AT NORMAL INCIDENCE AND THE
REFLECTANCE OF THE COVER
C     TO DIFFUSE RADIATION
     KL_COVER=THICK_COVER*EXT_COVER
          RHO_DIFFUSE=-1.
     TAUALPHAN=TAU_ALPHA(1,0.D0,KL_COVER,RI_COVER,ABS_PLATE,
         1     RHO_DIFFUSE)

C     USE THE RELATIONS OF BRANDEMUEHL TO GET THE EFFECTIVE INCIDENCE ANGLES FOR
DIFFUSE RADIATION
     EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
     EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
     COSSLOPE=DCOS(SLOPE*RDCONV)
     FSKY=(1.+COSSLOPE)/2.
     FGND=(1.-COSSLOPE)/2.
     GDSKY=FSKY*GDH
     GDGND=REFL_GROUND*FGND*GH

C     USE THE TAU_ALPHA FUNCTION FOR THE COMPONENT IAM VALUES
     XKATDS=TAU_ALPHA(1,EFFSKY,KL_COVER,RI_COVER,ABS_PLATE,
         1     RHO_DIFFUSE)/TAUALPHAN
     XKATDG=TAU_ALPHA(1,EFFGND,KL_COVER,RI_COVER,ABS_PLATE,
         1     RHO_DIFFUSE)/TAUALPHAN
     XKATB=TAU_ALPHA(1,ANGLE_INC,KL_COVER,RI_COVER,ABS_PLATE,
         1     RHO_DIFFUSE)/TAUALPHAN

C     CALCULATE THE OVERALL IAM
     IF(GT.GT.0.) THEN
      XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
          ELSE
           XKAT=0.
          ENDIF

         ENDIF

C   GUESS A COVER TEMPERATURE
    T_COVER=(T_SKY+T_AMB)/2.

C   GUESS A BACK COLLECTOR SURFACE TEMPERATURE
    T_3=(T_ZONE+T_ZONE_RAD)/2.

C   GUESS UPPER AND LOWER FLOW CHANNEL SURFACE TEMPERATURES
    T_1=T_FLUID_IN
          T_2=T_FLUID_IN

C   GUESS A PV CELL TEMPERATURE
    T_PV=(T_COVER+T_2)/2.
          T_PV_OLD=T_PV

C   GUESS A MEAN FLUID TEMPERATURE
    T_FLUID_MEAN=T_FLUID_IN

C   INITIALIZE THE COUNTER
          ICOUNT=1

C   GET THE TOP SURFACE RADIATION COEFFICIENT
10   H_RAD_T=H_RADIATION(T_COVER,T_SKY,EMISS_COVER)

C   GET THE BOTTOM SURFACE RADIATION COEFFICIENT
    H_RAD_B=H_RADIATION(T_3,T_ZONE_RAD,EMISS_BACK)

C   GET THE INNER-CHANNEL RADIATION COEFFICIENT
    H_RAD_12=H_RAD_GRAY(T_1,T_2,EMISS_1,EMISS_2)

C   GET THE PV CELL EFFICIENCY
    IF(MODE_EFF.EQ.1) THEN
     FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
     FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
     PV_EFFICIENCY=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))
          ELSE IF(MODE_EFF.EQ.2) THEN
     X(2)=GT
           X(1)=T_PV
           NX(2)=N_RADS
           NX(1)=N_TEMPS
           CALL DYNAMICDATA(LU_DATA,2,NX,1,X,Y,INFO,*20)
     CALL LINKCK('TYPE 542','DYNAMICDATA',1,99)
20    IF(ERRORFOUND()) RETURN 1
     PV_EFFICIENCY=Y(1)
          ELSE
           PV_EFFICIENCY=EFF_PV
          ENDIF

C   SET THE PROPERTIES OF THE AIR STREAM
          T_PROPS_K=T_FLUID_MEAN+273.15
          P_KPA=P_ATM*101.325
    CALL AIRPROP(T_PROPS_K,P_KPA,AIRPROPS)
```

```
          RHO_AIR=1./AIRPROPS(1)      !KG/M3
          VISC_AIR=AIRPROPS(2)*3600.   !KG/M/HR
          PRANDTL_AIR=AIRPROPS(3)      !DIMENSIONLESS
          K_AIR=AIRPROPS(4)*3.6        !KJ/H/M/K
          CP_AIR=AIRPROPS(5)           !KJ/KG/K

C    CALCULATE THE REYNOLDS NUMBER
      REYNOLDS=4.*FLOW_IN/PI/DIAMETER/VISC_AIR

C    CALCULATE THE FLUID CONVECTION COEFFICIENT
      IF(REYNOLDS.LE.0.) THEN

C      CALL THE SUBROUTINE TO CALCULATE THE NUSSELT NUMBER FOR INCLINED FLAT PLATES
        T_1_K=T_1+273.15
        T_2_K=T_2+273.15
        CALL NUFLATPLATE(SLOPE,T_1_K,T_2_K,THICK_CHANNEL,P_KPA,NUSSELT)

C      SET THE NATURAL CONVECTION HEAT TRANSFER COEFFICIENT
        H_FLUID=NUSSELT*K_AIR/DIAMETER

      ELSE IF(REYNOLDS.LE.2300.) THEN

C      SET THE NUSSELT NUMBER (BASED ON A CONSTANT SURFACE TEMPERATURE)
        NUSSELT=3.66

C      SET THE LAMINAR CONVECTION COEFFICIENT
        H_FLUID=NUSSELT*K_AIR/DIAMETER

          ELSE

C      SET THE NUSSELT NUMBER (BASED ON THE DITTUUS BOELTER EQUATION)
        T_SURF=(T_1+T_2)/2.

            IF(T_SURF.GE.T_FLUID_MEAN) THEN
              N=0.4
            ELSE
              N=0.3
            ENDIF

        NUSSELT=0.023*(REYNOLDS**0.8)*(PRANDTL_AIR**N)

C      SET THE TURBULENT CONVECTION COEFFICIENT
        H_FLUID=NUSSELT*K_AIR/DIAMETER

      ENDIF

C    CALCULATE THE ABSORBED SOLAR RADIATION PER UNIT AREA
      S=TAUALPHAN*XKAT*GT*(1.-PV_EFFICIENCY)

C    CALCULATE THE RESISTANCE DUE TO THE COVER MATERIAL
      R_1=THICK_COVER/K_COVER
```

```
C    SET SOME CONSTANTS WE'LL NEED
      F_PRIME=1.+R_1*H_CONV_T+R_1*H_RAD_T
      H_PRIME=1.+R_3*H_CONV_B+R_3*H_RAD_B
      G_PRIME=1./R_1+1./R_2-1./R_1/F_PRIME
      J=H_RAD_12+H_FLUID+1./R_3-1./H_PRIME/R_3
      M=1.-1./R_2/G_PRIME+R_2*H_FLUID+R_2*H_RAD_12-R_2*H_RAD_12*
     1  H_RAD_12/J

C    REFORMULATE THE 6 ENERGY BALANCES TO FIND qu"=AA*T_fluid+CC
      AA=-2.*H_FLUID+H_FLUID*H_FLUID/J+H_FLUID*R_2*H_FLUID/M+H_FLUID*R_2
     1  *H_RAD_12*H_FLUID*2./M/J+H_FLUID*H_RAD_12*R_2*H_RAD_12*H_FLUID
     1  /J/M/J

      CC=H_FLUID*H_CONV_B*T_ZONE/J/H_PRIME+H_FLUID*H_RAD_B*T_ZONE_RAD/J
     1  /H_PRIME+H_FLUID*S/G_PRIME/M+H_FLUID*H_CONV_T*T_AMB/F_PRIME/
     1  G_PRIME/M+H_FLUID*H_RAD_T*T_SKY/F_PRIME/G_PRIME/M+H_FLUID*R_2*
     1  H_RAD_12*H_CONV_B*T_ZONE/M/J/H_PRIME+H_FLUID*R_2*H_RAD_12*
     1  H_RAD_B*T_ZONE_RAD/M/J/H_PRIME+H_FLUID*H_RAD_12*S/J/M/G_PRIME+
     1  H_FLUID*H_RAD_12*H_CONV_T*T_AMB/J/F_PRIME/G_PRIME/M+H_FLUID*
     1  H_RAD_12*H_RAD_T*T_SKY/J/F_PRIME/G_PRIME/M+H_FLUID*H_RAD_12*
     1  R_2*H_RAD_12*H_CONV_B*T_ZONE/J/M/J/H_PRIME+H_FLUID*H_RAD_12*
     1  R_2*H_RAD_12*H_RAD_B*T_ZONE_RAD/J/M/J/H_PRIME

C    SET THE CASE WITH NO FLOW
      IF(FLOW_IN.LE.0.) THEN

C      SET THE USEFUL ENERGY GAIN
        QU=0.

C      SET THE MEAN FLUID TEMPERATURE FROM QU=AA(T)+CC
        T_FLUID_MEAN=-CC/AA

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
        T_1=S/G_PRIME/M+H_CONV_T*T_AMB/F_PRIME/G_PRIME/M+H_RAD_T*T_SKY
     1    /F_PRIME/G_PRIME/M+R_2*H_FLUID*T_FLUID_MEAN/M+R_2*H_RAD_12*
     1    H_FLUID*T_FLUID_MEAN/M/J+R_2*H_RAD_12*H_CONV_B*T_ZONE/M/J/
     1    H_PRIME+R_2*H_RAD_12*H_RAD_B*T_ZONE_RAD/M/J/H_PRIME

C      SET THE PV TEMPERATURE
        T_PV=S/G_PRIME+H_CONV_T*T_AMB/F_PRIME/G_PRIME+H_RAD_T*T_SKY/
     1     F_PRIME/G_PRIME+T_1/R_2/G_PRIME

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
        T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+H_CONV_B*T_ZONE/J/
     1    H_PRIME+H_RAD_B*T_ZONE_RAD/J/H_PRIME

C      SET THE BACK SURFACE TEMPERATURE
        T_3=T_2/H_PRIME+R_3*H_CONV_B*T_ZONE/H_PRIME+R_3*H_RAD_B*
     1    T_ZONE_RAD/H_PRIME
```

```
C      SET THE COVER TEMPERATURE
       T_COVER=T_PV/F_PRIME+R_1*H_CONV_T*T_AMB/F_PRIME+
     1      R_1*H_RAD_T*T_SKY/F_PRIME

C      SET THE FLUID OUTLET TEMPERATURE
       T_FLUID_OUT=T_FLUID_MEAN

     ELSE

C      FIND THE FLUID OUTLET TEMPERATURE BY INTEGRATING THE FLUID DIFFERENTIAL
EQUATION
       T_FLUID_OUT=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)-
     1      CC/AA

C      FIND THE MEAN FLUID TEMPERATURE BY INTEGRATING THE LOCAL FLUID TEMPERATURE
EQUATION
       T_FLUID_MEAN=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)/
     1      (AA*AREA/FLOW_IN/CP_AIR)-(T_FLUID_IN+CC/AA)/(AA*AREA/
     1      FLOW_IN/CP_AIR)-CC/AA

C      KNOWING THE INLET AND OUTLET TEMPERATURES CALCULATE THE COLLECTOR USEFUL
ENERGY GAIN
              QU=FLOW_IN*CP_AIR*(T_FLUID_OUT-T_FLUID_IN)

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
       T_1=S/G_PRIME/M+H_CONV_T*T_AMB/F_PRIME/G_PRIME/M+H_RAD_T*T_SKY
     1      /F_PRIME/G_PRIME/M+R_2*H_FLUID*T_FLUID_MEAN/M+R_2*H_RAD_12*
     1      H_FLUID*T_FLUID_MEAN/M/J+R_2*H_RAD_12*H_CONV_B*T_ZONE/M/J/
     1      H_PRIME+R_2*H_RAD_12*H_RAD_B*T_ZONE_RAD/M/J/H_PRIME

C      SET THE PV TEMPERATURE
       T_PV=S/G_PRIME+H_CONV_T*T_AMB/F_PRIME/G_PRIME+H_RAD_T*T_SKY/
     1       F_PRIME/G_PRIME+T_1/R_2/G_PRIME

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
       T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+H_CONV_B*T_ZONE/J/
     1       H_PRIME+H_RAD_B*T_ZONE_RAD/J/H_PRIME

C      SET THE BACK SURFACE TEMPERATURE
       T_3=T_2/H_PRIME+R_3*H_CONV_B*T_ZONE/H_PRIME+R_3*H_RAD_B*
     1       T_ZONE_RAD/H_PRIME

C      SET THE COVER TEMPERATURE
       T_COVER=T_PV/F_PRIME+R_1*H_CONV_T*T_AMB/F_PRIME+
     1      R_1*H_RAD_T*T_SKY/F_PRIME

     ENDIF

C    CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
     IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
     1   (ICOUNT.LT.1000)) THEN
            T_PV_OLD=T_PV
       ICOUNT=ICOUNT+1
       GOTO 10
     ENDIF

C    CALCULATE THE THERMAL EFFICIENCY
     IF(GT.GT.0.) THEN
       EFF_THERMAL=QU/AREA/GT
          ELSE
            EFF_THERMAL=0.
     ENDIF

C    CALCULATE THE HEAT TRANSFERS
     Q_TOP_CONV=H_CONV_T*AREA*(T_COVER-T_AMB)
     Q_TOP_RAD=H_RAD_T*AREA*(T_COVER-T_SKY)
     Q_BACK_CONV=H_CONV_B*AREA*(T_3-T_ZONE)
     Q_BACK_RAD=H_RAD_B*AREA*(T_3-T_ZONE_RAD)
          Q_ABS=AREA*GT*TAUALPHAN*XKAT*(1.-PV_EFFICIENCY)

C    CALCULATE THE PV POWER PRODUCTION
     POWER=AREA*TAUALPHAN*XKAT*GT*PV_EFFICIENCY

C    SET THE OUTPUTS
     OUT(1)=T_FLUID_OUT
     OUT(2)=FLOW_IN
     OUT(3)=QU
          OUT(4)=EFF_THERMAL
     OUT(5)=POWER
     OUT(6)=PV_EFFICIENCY
     OUT(7)=T_COVER
     OUT(8)=T_PV
     OUT(9)=T_1
     OUT(10)=T_FLUID_MEAN
     OUT(11)=T_2
          OUT(12)=T_3
          OUT(13)=XKAT
          OUT(14)=Q_TOP_CONV
          OUT(15)=Q_TOP_RAD
          OUT(16)=Q_BACK_CONV
          OUT(17)=Q_BACK_RAD
          OUT(18)=Q_ABS

C    KILL THE SIMULATION IF THE SYSTEM WAS UNABLE TO SOLVE
     IF(ICOUNT.GE.1000) THEN
       CALL MESSAGES(-1,MESSAGE1,'FATAL',IUNIT,ITYPE)
          ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
     RETURN 1
```

```
      END
C-----------------------------------------------------------------------------------------------------------------------
```

TYPE 567-Building integrated photovoltaic system (glazed, air) (No convective and radiative losses at the back of the collector)

```
      SUBROUTINE TYPE567(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C     THIS COMPONENT MODELS A BUILDING-INTEGRATED PHOTOVOLATIC SYSTEM.  IN THIS
MODEL THERE IS ASSUMED TO BE
C     A SINGLE GLASS COVER OVER THE PV MATERIAL.  THE BACK OF THE BIPV SYSTEM
INTERACTS WITH ZONE MODELS WHERE
C     THE SURFACE TEMPERATURE IS CALCULATED BY THE ZONE MODEL. THE FLUID CONVECTION
CALCULATIONS ARE FROM
C     CORRELATIONS PROVIDED BY "INTRODUCTION TO HEAT TRANSFER" BY INCROPERA AND
DEWITT.
C
C     LAST MODIFIED:
C        APRIL 2004 - JEFF THORNTON - ORIGINAL CODING
C-----------------------------------------------------------------------------------------------------------------------
```



```
C-----------------------------------------------------------------------------------------------------------------------
      !Export this subroutine for its use in external DLLs.
      !DEC$ATTRIBUTES DLLEXPORT :: TYPE567
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
                  USE TrnsysConstants
                  USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
      IMPLICIT NONE
                  DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

      INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL,NPAR,NIN
      CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED PARAMETERS FOR THE SIZING OF THE ARRAYS
      PARAMETER (NP=20,NI=14,NOUT=17,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
      DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
      DOUBLE PRECISION RDCONV,ANGLE_INC,TAUALF,B0,Q_BACK,T_1,T_2,
                  1  AREA,FLOW_IN,T_AMB,GT,GH,GDH,REFL_GROUND,SLOPE,EMISS_COVER,
     1  TAUALPHAN,EFFSKY,EFFGND,COSSLP,FSKY,FGND,GDSKY,GDGND,XKATB,
     1  XKATDS,F_PRIME,XKATDG,T_FLUID_MEAN,T_SKY,H_CONV_T,X(2),NUSSELT,
     1  H_FLUID,QU,EFF_THERMAL,S,R_1,R_2,R_3,Q_TOP_CONV,N,H_RAD_12,
     1  Q_TOP_RAD,Q_ABS,H_RADIATION,EMISS_1,EMISS_2,H_RAD_GRAY,G_PRIME
     1  ,M,AA,CC,K_COVER,THICK_COVER,RI_COVER,T_1_K,T_2_K,T_SURF,CP_AIR,
     1  EXT_COVER,ABS_PLATE,EFF_PV,Y(1),EFF_PV_REF,T_REF,GT_REF,K_AIR,
     1  EFF_CORR_T,EFF_CORR_I,T_FLUID_IN,KL_COVER,RHO_DIFFUSE,REYNOLDS,
     1  TAU_ALPHA,COSSLOPE,XKAT,T_COVER,T_PV,T_PV_OLD,FACTOR_T,FACTOR_I
     1  ,PV_EFFICIENCY,POWER,H_RAD_T,J,T_FLUID_OUT,AIRPROPS(5),PI,LENGTH
     1  ,WIDTH,THICK_CHANNEL,P_ATM,P_KPA,DIAMETER,T_PROPS_K,RHO_AIR,
     1  VISC_AIR,PRANDTL_AIR

      INTEGER ICOUNT,MODE_IAM,MODE_EFF,LU_DATA,N_TEMPS,N_RADS,NX(2)
      CHARACTER(LEN=MAXMESSAGELENGTH)::MESSAGE1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
      DATA RDCONV/0.017453292/,PI/3.14159265358979/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    FUNCTIONS
      TAUALF(ANGLE_INC)=1.-B0*(1./DMAX1(0.5,DCOS(ANGLE_INC*RDCONV))-1.)
     .            - (1.-B0)*(DMAX1(60.,ANGLE_INC)-60.)/30.
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ERROR MESSAGES
      MESSAGE1='The BIPV model was unable to find a solution for the cel
                  1l temperature at the given timestep.  Please report this error to
     1your TRNSYS distributor.'
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
      TIME0=getSimulationStartTime()
      TFINAL=getSimulationStopTime()
      DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
      IF(INFO(7).EQ.-2) THEN
                     INFO(12)=16
                     RETURN 1
                  ENDIF
C-----------------------------------------------------------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
            RETURN 1
          ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
      IF(INFO(13).GT.0) THEN
            RETURN 1
          ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C       RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
         IUNIT=INFO(1)
            ITYPE=INFO(2)

C       SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
         INFO(6)=NOUT
         INFO(9)=1
            INFO(10)=0

C       SET THE NUMBER OF PARAMETERS AND INPUTS
         MODE_IAM=JFIX(PAR(11)+0.5)
         MODE_EFF=JFIX(PAR(12)+0.5)
         NPAR=12
            NIN=13

         IF(MODE_IAM.LT.1) CALL TYPECK(-4,INFO,0,11,0)
         IF(MODE_IAM.GT.2) CALL TYPECK(-4,INFO,0,11,0)
         IF(MODE_EFF.LT.1) CALL TYPECK(-4,INFO,0,12,0)
         IF(MODE_EFF.GT.3) CALL TYPECK(-4,INFO,0,12,0)
            IF(ERRORFOUND()) RETURN 1

            IF(MODE_IAM.EQ.1) THEN
              NPAR=NPAR+2
            ELSE IF(MODE_IAM.EQ.2) THEN
              NPAR=NPAR+3
            ENDIF

            IF(MODE_EFF.EQ.1) THEN
              NPAR=NPAR+5
```

```
         ELSE IF(MODE_EFF.EQ.2) THEN
              NPAR=NPAR+3
         ELSE IF(MODE_EFF.EQ.3) THEN
              NIN=NIN+1
            ENDIF

C       CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED IN
C       THE TRNSYS INPUT FILE
            CALL TYPECK(1,INFO,NIN,NPAR,ND)

C       SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
         DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','IR1','IR1','IR1',
     1          'DM1','DG1','DG1','HT1','PR2','DM1'/
         DATA OCHECK/'TE1','MF1','PW1','DM1','PW1','DM1','TE1','TE1',
     1          'TE1','TE1','TE1','TE1','DM1','PW1','PW1','PW1',
     1          'PW1'/

C       CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
         CALL RCHECK(INFO,YCHECK,OCHECK)

C       RETURN TO THE CALLING PROGRAM
         RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C       SET THE UNIT NUMBER FOR FUTURE CALLS
         IUNIT=INFO(1)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
         LENGTH=PAR(1)
         WIDTH=PAR(2)
            EMISS_COVER=PAR(3)
            K_COVER=PAR(4)
            THICK_COVER=PAR(5)
         R_2=PAR(6)
            EMISS_1=PAR(7)
            EMISS_2=PAR(8)
         R_3=PAR(9)
            THICK_CHANNEL=PAR(10)
         MODE_IAM=JFIX(PAR(11)+0.5)
         MODE_EFF=JFIX(PAR(12)+0.5)
         NPAR=12
```

```
C      CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
       IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
       IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
       IF(EMISS_COVER.LT.0.) CALL TYPECK(-4,INFO,0,3,0)
       IF(EMISS_COVER.GT.1.) CALL TYPECK(-4,INFO,0,3,0)
       IF(K_COVER.LE.0.) CALL TYPECK(-4,INFO,0,4,0)
       IF(THICK_COVER.LE.0.) CALL TYPECK(-4,INFO,0,5,0)
       IF(R_2.LE.0.) CALL TYPECK(-4,INFO,0,6,0)
       IF(EMISS_1.LT.0.) CALL TYPECK(-4,INFO,0,7,0)
       IF(EMISS_1.GT.1.) CALL TYPECK(-4,INFO,0,7,0)
       IF(EMISS_2.LT.0.) CALL TYPECK(-4,INFO,0,8,0)
       IF(EMISS_2.GT.1.) CALL TYPECK(-4,INFO,0,8,0)
       IF(R_3.LE.0.) CALL TYPECK(-4,INFO,0,9,0)
       IF(THICK_CHANNEL.LT.0.) CALL TYPECK(-4,INFO,0,10,0)

             IF(ERRORFOUND()) RETURN 1

C      GET THE IAM MODE SPECIFIC PARAMETERS
             IF(MODE_IAM.EQ.1) THEN
               TAUALPHAN=PAR(NPAR+1)
         B0=PAR(NPAR+2)

         IF(TAUALPHAN.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(TAUALPHAN.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(B0.LT.0.) CALL TYPECK(-4,INFO,0,NPAR+2,0)
         IF(B0.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+2,0)

               NPAR=NPAR+2

             ELSE IF(MODE_IAM.EQ.2) THEN
               ABS_PLATE=PAR(NPAR+1)
               RI_COVER=PAR(NPAR+2)
               EXT_COVER=PAR(NPAR+3)

         IF(ABS_PLATE.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(ABS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(RI_COVER.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+2,0)
         IF(EXT_COVER.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+3,0)

               NPAR=NPAR+3

             ENDIF

C      GET THE PV EFFICIENCY SPECIFIC PARAMETERS
             IF(MODE_EFF.EQ.1) THEN
         EFF_PV_REF=PAR(NPAR+1)
         T_REF=PAR(NPAR+2)
         GT_REF=PAR(NPAR+3)
         EFF_CORR_T=PAR(NPAR+4)
         EFF_CORR_I=PAR(NPAR+5)

         IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,NPAR+1,0)
         IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,NPAR+3,0)

       ELSE IF(MODE_EFF.EQ.2) THEN
               LU_DATA=JFIX(PAR(NPAR+1)+0.5)
               N_TEMPS=JFIX(PAR(NPAR+2)+0.5)
               N_RADS=JFIX(PAR(NPAR+3)+0.5)

                     IF(LU_DATA.LT.10) CALL TYPECK(-4,INFO,0,NPAR+1,0)
               IF(N_TEMPS.LT.2) CALL TYPECK(-4,INFO,0,NPAR+2,0)
               IF(N_RADS.LT.2) CALL TYPECK(-4,INFO,0,NPAR+3,0)

             ENDIF

C      PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
       OUT(1)=XIN(1)
       OUT(2:6)=0.
       OUT(7)=T_AMB
       OUT(8)=T_AMB
       OUT(9)=XIN(1)
       OUT(10)=XIN(1)
       OUT(11)=XIN(1)
             OUT(12)=XIN(5)
             OUT(13:17)=0.

C      RETURN TO THE CALLING PROGRAM
       RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
      IF(INFO(1).NE.IUNIT) THEN

C      RESET THE UNIT NUMBER
             IUNIT=INFO(1)
             ITYPE=INFO(2)

C      READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
       LENGTH=PAR(1)
       WIDTH=PAR(2)
```

```
              EMISS_COVER=PAR(3)
              K_COVER=PAR(4)
              THICK_COVER=PAR(5)
      R_2=PAR(6)
              EMISS_1=PAR(7)
              EMISS_2=PAR(8)
      R_3=PAR(9)
              THICK_CHANNEL=PAR(10)
      MODE_IAM=JFIX(PAR(11)+0.5)
      MODE_EFF=JFIX(PAR(12)+0.5)
      NPAR=12
              IF(MODE_IAM.EQ.1) THEN
                TAUALPHAN=PAR(NPAR+1)
        B0=PAR(NPAR+2)
                NPAR=NPAR+2
              ELSE IF(MODE_IAM.EQ.2) THEN
                ABS_PLATE=PAR(NPAR+1)
                RI_COVER=PAR(NPAR+2)
                EXT_COVER=PAR(NPAR+3)
                NPAR=NPAR+3
              ENDIF
              IF(MODE_EFF.EQ.1) THEN
        EFF_PV_REF=PAR(NPAR+1)
        T_REF=PAR(NPAR+2)
        GT_REF=PAR(NPAR+3)
        EFF_CORR_T=PAR(NPAR+4)
        EFF_CORR_I=PAR(NPAR+5)
      ELSE IF(MODE_EFF.EQ.2) THEN
                LU_DATA=JFIX(PAR(NPAR+1)+0.5)
                N_TEMPS=JFIX(PAR(NPAR+2)+0.5)
                N_RADS=JFIX(PAR(NPAR+3)+0.5)
              ENDIF

    ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
60    T_FLUID_IN=XIN(1)
    FLOW_IN=XIN(2)
    T_AMB=XIN(3)
    T_SKY=XIN(4)
    T_SURF=XIN(5)
    GT=XIN(6)
    GH=XIN(7)
    GDH=XIN(8)
    REFL_GROUND=XIN(9)
    ANGLE_INC=XIN(10)
    SLOPE=XIN(11)
    H_CONV_T=XIN(12)
              P_ATM=XIN(13)
              IF(MODE_EFF.EQ.3) THEN
                EFF_PV=XIN(14)
              ELSE
                EFF_PV=0.
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    CHECK THE INPUTS FOR PROBLEMS
    IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
    IF(GT.LT.0.) CALL TYPECK(-3,INFO,7,0,0)
    IF(GH.LT.0.) CALL TYPECK(-3,INFO,8,0,0)
    IF(GDH.LT.0.) CALL TYPECK(-3,INFO,9,0,0)
    IF(REFL_GROUND.LT.0.) CALL TYPECK(-3,INFO,10,0,0)
    IF(REFL_GROUND.GT.1.) CALL TYPECK(-3,INFO,10,0,0)
    IF(H_CONV_T.LT.0.) CALL TYPECK(-3,INFO,12,0,0)
    IF(P_ATM.LE.0.) CALL TYPECK(-3,INFO,13,0,0)
    IF(EFF_PV.LT.0.) CALL TYPECK(-3,INFO,14,0,0)
    IF(EFF_PV.GT.1.) CALL TYPECK(-3,INFO,14,0,0)

              IF(ERRORFOUND()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C    SET SOME GEOMETRIC PARAMETERS
    AREA=LENGTH*WIDTH
              DIAMETER=4.*WIDTH*THICK_CHANNEL/(2.*WIDTH+2.*THICK_CHANNEL)

C    SET THE INCIDENCE ANGLE MODIFIER BASED ON THE MODE
    XKAT=1.
              IF(MODE_IAM.EQ.1) THEN

C      DETERMINE INCIDENCE ANGLE MODIFIER
      IF(GT .GT. 0. .AND. ANGLE_INC .LT. 90.) THEN

C        USE RELATIONS OF BRANDEMUEHL FOR EFFECTIVE INCIDENCE ANGLES FOR DIFFUSE
        EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
        EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
        COSSLP=DCOS(SLOPE*RDCONV)
        FSKY=(1.+COSSLP)/2.
        FGND=(1.-COSSLP)/2.
        GDSKY=FSKY*GDH
        GDGND=REFL_GROUND*FGND*GH

C        SET THE INCIDENCE ANGLE MODIFIERS
        XKATB=TAUALF(ANGLE_INC)
        XKATDS=TAUALF(EFFSKY)
        XKATDG=TAUALF(EFFGND)
```

```
        XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
        IF(XKAT.LE.0.) XKAT=0.
      ELSE
        XKAT=0.
      ENDIF
            ELSE IF(MODE_IAM.EQ.2) THEN

C      GET THE TRANSMITTANCE-ABSORPTANCE PRODUCT AT NORMAL INCIDENCE AND THE
REFLECTANCE OF THE COVER
C      TO DIFFUSE RADIATION
      KL_COVER=THICK_COVER*EXT_COVER
              RHO_DIFFUSE=-1.
      TAUALPHAN=TAU_ALPHA(1,0.D0,KL_COVER,RI_COVER,ABS_PLATE,
     1      RHO_DIFFUSE)

C      USE THE RELATIONS OF BRANDEMUEHL TO GET THE EFFECTIVE INCIDENCE ANGLES FOR
DIFFUSE RADIATION
      EFFSKY=59.68-0.1388*SLOPE+0.001497*SLOPE*SLOPE
      EFFGND=90.-0.5788*SLOPE+0.002693*SLOPE*SLOPE
      COSSLOPE=DCOS(SLOPE*RDCONV)
      FSKY=(1.+COSSLOPE)/2.
      FGND=(1.-COSSLOPE)/2.
      GDSKY=FSKY*GDH
      GDGND=REFL_GROUND*FGND*GH

C      USE THE TAU_ALPHA FUNCTION FOR THE COMPONENT IAM VALUES
      XKATDS=TAU_ALPHA(1,EFFSKY,KL_COVER,RI_COVER,ABS_PLATE,
     1      RHO_DIFFUSE)/TAUALPHAN
      XKATDG=TAU_ALPHA(1,EFFGND,KL_COVER,RI_COVER,ABS_PLATE,
     1      RHO_DIFFUSE)/TAUALPHAN
      XKATB=TAU_ALPHA(1,ANGLE_INC,KL_COVER,RI_COVER,ABS_PLATE,
     1      RHO_DIFFUSE)/TAUALPHAN

C      CALCULATE THE OVERALL IAM
      IF(GT.GT.0.) THEN
        XKAT=(XKATB*(GT-GDSKY-GDGND)+XKATDS*GDSKY+XKATDG*GDGND)/GT
              ELSE
                XKAT=0.
              ENDIF

            ENDIF

C    GUESS A COVER TEMPERATURE
     T_COVER=(T_SKY+T_AMB)/2.

C    GUESS UPPER AND LOWER FLOW CHANNEL SURFACE TEMPERATURES
     T_1=T_FLUID_IN
            T_2=T_FLUID_IN

C    GUESS A PV CELL TEMPERATURE
     T_PV=(T_COVER+T_2)/2.

            T_PV_OLD=T_PV

C    GUESS A MEAN FLUID TEMPERATURE
     T_FLUID_MEAN=T_FLUID_IN

C    INITIALIZE THE COUNTER
            ICOUNT=1

C    GET THE TOP SURFACE RADIATION COEFFICIENT
10    H_RAD_T=H_RADIATION(T_COVER,T_SKY,EMISS_COVER)

C    GET THE INNER-CHANNEL RADIATION COEFFICIENT
     H_RAD_12=H_RAD_GRAY(T_1,T_2,EMISS_1,EMISS_2)

C    GET THE PV CELL EFFICIENCY
     IF(MODE_EFF.EQ.1) THEN
       FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
       FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
       PV_EFFICIENCY=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))
            ELSE IF(MODE_EFF.EQ.2) THEN
       X(2)=GT
              X(1)=T_PV
              NX(2)=N_RADS
              NX(1)=N_TEMPS
              CALL DYNAMICDATA(LU_DATA,2,NX,1,X,Y,INFO,*20)
       CALL LINKCK('TYPE 542','DYNAMICDATA',1,99)
20      IF(ERRORFOUND()) RETURN 1
       PV_EFFICIENCY=Y(1)
            ELSE
              PV_EFFICIENCY=EFF_PV
            ENDIF

C    SET THE PROPERTIES OF THE AIR STREAM
            T_PROPS_K=T_FLUID_MEAN+273.15
            P_KPA=P_ATM*101.325
     CALL AIRPROP(T_PROPS_K,P_KPA,AIRPROPS)

            RHO_AIR=1./AIRPROPS(1)       !KG/M3
            VISC_AIR=AIRPROPS(2)*3600.   !KG/M/HR
            PRANDTL_AIR=AIRPROPS(3)      !DIMENSIONLESS
            K_AIR=AIRPROPS(4)*3.6        !KJ/H/M/K
            CP_AIR=AIRPROPS(5)           !KJ/KG/K

C    CALCULATE THE REYNOLDS NUMBER
     REYNOLDS=4.*FLOW_IN/PI/DIAMETER/VISC_AIR


C    CALCULATE THE FLUID CONVECTION COEFFICIENT
     IF(REYNOLDS.LE.0.) THEN

C      CALL THE SUBROUTINE TO CALCULATE THE NUSSELT NUMBER FOR INCLINED FLAT PLATES
```

```
      T_1_K=T_1+273.15
      T_2_K=T_2+273.15
      CALL NUFLATPLATE(SLOPE,T_1_K,T_2_K,THICK_CHANNEL,P_KPA,NUSSELT)

C     SET THE NATURAL CONVECTION HEAT TRANSFER COEFFICIENT
      H_FLUID=NUSSELT*K_AIR/DIAMETER

     ELSE IF(REYNOLDS.LE.2300.) THEN

C     SET THE NUSSELT NUMBER (BASED ON A CONSTANT SURFACE TEMPERATURE)
      NUSSELT=3.66

C     SET THE LAMINAR CONVECTION COEFFICIENT
      H_FLUID=NUSSELT*K_AIR/DIAMETER

           ELSE

C     SET THE NUSSELT NUMBER (BASED ON THE DITTUUS BOELTER EQUATION)
      T_SURF=(T_1+T_2)/2.

            IF(T_SURF.GE.T_FLUID_MEAN) THEN
              N=0.4
            ELSE
              N=0.3
            ENDIF

      NUSSELT=0.023*(REYNOLDS**0.8)*(PRANDTL_AIR**N)

C     SET THE TURBULENT CONVECTION COEFFICIENT
      H_FLUID=NUSSELT*K_AIR/DIAMETER

     ENDIF

C    CALCULATE THE ABSORBED SOLAR RADIATION PER UNIT AREA
     S=TAUALPHAN*XKAT*GT*(1.-PV_EFFICIENCY)

C    CALCULATE THE RESISTANCE DUE TO THE COVER MATERIAL
     R_1=THICK_COVER/K_COVER

C    SET SOME CONSTANTS WE'LL NEED
     F_PRIME=1.+R_1*H_CONV_T+R_1*H_RAD_T
     G_PRIME=1./R_1+1./R_2-1./R_1/F_PRIME
     J=H_RAD_12+H_FLUID+1./R_3
     M=1.-1./R_2/G_PRIME+R_2*H_FLUID+R_2*H_RAD_12-R_2*H_RAD_12*
           1  H_RAD_12/J

C    REFORMULATE THE 6 ENERGY BALANCES TO FIND qu"=AA*T_fluid+CC
     AA=-2.*H_FLUID+H_FLUID*H_FLUID/J+H_FLUID*R_2*H_FLUID/M+H_FLUID*R_2
           1  *H_RAD_12*H_FLUID*2./M/J+H_FLUID*H_RAD_12*R_2*H_RAD_12*H_FLUID
    1  /J/M/J

     CC=H_FLUID*S/G_PRIME/M
           1  +H_FLUID*H_CONV_T*T_AMB/F_PRIME/G_PRIME/M
           1  +H_FLUID*H_RAD_T*T_SKY/F_PRIME/G_PRIME/M
           1  +H_FLUID*H_RAD_12*S/J/M/G_PRIME
           1  +H_FLUID*H_RAD_12*H_CONV_T*T_AMB/J/F_PRIME/G_PRIME/M
           1  +H_FLUID*H_RAD_12*H_RAD_T*T_SKY/J/F_PRIME/G_PRIME/M
           1  +H_FLUID*H_RAD_12*R_2*H_RAD_12*T_SURF/J/M/J/R_3
           1  +H_FLUID*T_SURF/J/R_3
           1  +H_FLUID*R_2*H_RAD_12*T_SURF/M/J/R_3

C    SET THE CASE WITH NO FLOW
     IF(FLOW_IN.LE.0.) THEN

C     SET THE USEFUL ENERGY GAIN
      QU=0.

C     SET THE MEAN FLUID TEMPERATURE FROM QU=AA(T)+CC
      T_FLUID_MEAN=-CC/AA

C     SET THE UPPER CHANNEL SURFACE TEMPERATURE
      T_1=S/G_PRIME/M+H_CONV_T*T_AMB/F_PRIME/G_PRIME/M+H_RAD_T*T_SKY
           1     /F_PRIME/G_PRIME/M+R_2*H_FLUID*T_FLUID_MEAN/M+R_2*H_RAD_12*
    1     H_FLUID*T_FLUID_MEAN/M/J+R_2*H_RAD_12*T_SURF/M/J/R_3

C     SET THE PV TEMPERATURE
      T_PV=S/G_PRIME+H_CONV_T*T_AMB/F_PRIME/G_PRIME+H_RAD_T*T_SKY/
           1      F_PRIME/G_PRIME+T_1/R_2/G_PRIME

C     SET THE LOWER CHANNEL SURFACE TEMPERATURE
      T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+T_SURF/J/R_3

C     SET THE COVER TEMPERATURE
      T_COVER=T_PV/F_PRIME+R_1*H_CONV_T*T_AMB/F_PRIME+
           1     R_1*H_RAD_T*T_SKY/F_PRIME

C     SET THE FLUID OUTLET TEMPERATURE
      T_FLUID_OUT=T_FLUID_MEAN

     ELSE

C     FIND THE FLUID OUTLET TEMPERATURE BY INTEGRATING THE FLUID DIFFERENTIAL
EQUATION
      T_FLUID_OUT=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)-
           1     CC/AA

C     FIND THE MEAN FLUID TEMPERATURE BY INTEGRATING THE LOCAL FLUID TEMPERATURE
EQUATION
      T_FLUID_MEAN=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)/
           1     (AA*AREA/FLOW_IN/CP_AIR)-(T_FLUID_IN+CC/AA)/(AA*AREA/
    1      FLOW_IN/CP_AIR)-CC/AA
```

```
C      KNOWING THE INLET AND OUTLET TEMPERATURES CALCULATE THE COLLECTOR USEFUL
ENERGY GAIN
                QU=FLOW_IN*CP_AIR*(T_FLUID_OUT-T_FLUID_IN)

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
      T_1=S/G_PRIME/M+H_CONV_T*T_AMB/F_PRIME/G_PRIME/M+H_RAD_T*T_SKY
              1      /F_PRIME/G_PRIME/M+R_2*H_FLUID*T_FLUID_MEAN/M+R_2*H_RAD_12*
   1      H_FLUID*T_FLUID_MEAN/M/J+R_2*H_RAD_12*T_SURF/M/J/R_3

C      SET THE PV TEMPERATURE
      T_PV=S/G_PRIME+H_CONV_T*T_AMB/F_PRIME/G_PRIME+H_RAD_T*T_SKY/
              1       F_PRIME/G_PRIME+T_1/R_2/G_PRIME

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
      T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+T_SURF/J/R_3

C      SET THE COVER TEMPERATURE
      T_COVER=T_PV/F_PRIME+R_1*H_CONV_T*T_AMB/F_PRIME+
              1      R_1*H_RAD_T*T_SKY/F_PRIME

    ENDIF

C    CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
    IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
              1   (ICOUNT.LT.1000)) THEN
                T_PV_OLD=T_PV
      ICOUNT=ICOUNT+1
      GOTO 10
    ENDIF

C    CALCULATE THE THERMAL EFFICIENCY
    IF(GT.GT.0.) THEN
      EFF_THERMAL=QU/AREA/GT
              ELSE
                EFF_THERMAL=0.
    ENDIF

C    CALCULATE THE HEAT TRANSFERS
    Q_TOP_CONV=H_CONV_T*AREA*(T_COVER-T_AMB)
    Q_TOP_RAD=H_RAD_T*AREA*(T_COVER-T_SKY)
    Q_BACK=AREA*(T_2-T_SURF)/R_3
              Q_ABS=AREA*GT*TAUALPHAN*XKAT*(1.-PV_EFFICIENCY)

C    CALCULATE THE PV POWER PRODUCTION
    POWER=TAUALPHAN*XKAT*GT*PV_EFFICIENCY

C    SET THE OUTPUTS
    OUT(1)=T_FLUID_OUT
    OUT(2)=FLOW_IN
    OUT(3)=QU
              OUT(4)=EFF_THERMAL
    OUT(5)=POWER
    OUT(6)=PV_EFFICIENCY
    OUT(7)=T_COVER
    OUT(8)=T_PV
    OUT(9)=T_1
    OUT(10)=T_FLUID_MEAN
    OUT(11)=T_2
              OUT(12)=T_SURF
              OUT(13)=XKAT
              OUT(14)=Q_TOP_CONV
              OUT(15)=Q_TOP_RAD
              OUT(16)=Q_BACK
              OUT(17)=Q_ABS

C    KILL THE SIMULATION IF THE SYSTEM WAS UNABLE TO SOLVE
    IF(ICOUNT.GE.1000) THEN
      CALL MESSAGES(-1,MESSAGE1,'FATAL',IUNIT,ITYPE)
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
    RETURN 1
    END
C-----------------------------------------------------------------------------------------------------------------------
```

TYPE 568-Unglazed building integrated photovoltaic system (air) (No convective and radiative losses at the back of the collector)

```
      SUBROUTINE TYPE568(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C     THIS COMPONENT MODELS A BUILDING-INTEGRATED PHOTOVOLATIC SYSTEM.  IN THIS
MODEL THERE IS NO COVER
C     OVER THE PV MATERIAL.  THE BACK OF THE BIPV SYSTEM INTERACTS WITH ZONE MODELS
WHERE THE SURFACE
C     TEMPERATURE IS CALCULATED BY THE ZONE MODEL. THE FLUID CONVECTION
CALCULATIONS ARE FROM CORRELATIONS
C     PROVIDED BY "INTRODUCTION TO HEAT TRANSFER" BY INCROPERA AND DEWITT.
C
C     LAST MODIFIED:
C        APRIL 2004 - JEFF THORNTON - ORIGINAL CODING
C-----------------------------------------------------------------------------------------------------------------------
```



```
C-----------------------------------------------------------------------------------------------------------------------
      !Export this subroutine for its use in external DLLs.
      !DEC$ATTRIBUTES DLLEXPORT :: TYPE568
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
              USE TrnsysConstants
              USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
      IMPLICIT NONE
              DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

      INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL,NPAR,NIN
      CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED PARAMETERS FOR THE SIZING OF THE ARRAYS
      PARAMETER (NP=15,NI=10,NOUT=15,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
      DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
      DOUBLE PRECISION Q_BACK,T_1,T_2,AREA,FLOW_IN,T_AMB,GT,SLOPE,
     1   TAUALPHAN,F_PRIME,T_FLUID_MEAN,T_SKY,H_CONV_T,X(2),NUSSELT,
     1   H_FLUID,QU,EFF_THERMAL,S,R_2,R_3,Q_TOP_CONV,N,H_RAD_12,
     1   Q_TOP_RAD,Q_ABS,H_RADIATION,EMISS_1,EMISS_2,H_RAD_GRAY,
     1   M,AA,CC,T_1_K,T_2_K,T_SURF,CP_AIR,ABS_PLATE,EFF_PV,Y(1),
     1   EFF_PV_REF,T_REF,GT_REF,K_AIR,EFF_CORR_T,EFF_CORR_I,T_FLUID_IN,
     1   XKAT,T_PV,T_PV_OLD,FACTOR_T,FACTOR_I,REYNOLDS,EMISS_PLATE,
     1   PV_EFFICIENCY,POWER,H_RAD_T,J,T_FLUID_OUT,AIRPROPS(5),PI,
     1   WIDTH,THICK_CHANNEL,P_ATM,P_KPA,DIAMETER,T_PROPS_K,RHO_AIR,
     1   VISC_AIR,PRANDTL_AIR,LENGTH

      INTEGER ICOUNT,MODE_EFF,LU_DATA,N_TEMPS,N_RADS,NX(2)
      CHARACTER(LEN=MAXMESSAGELENGTH)::MESSAGE1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
      DATA PI/3.14159265358979/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ERROR MESSAGES
      MESSAGE1='The BIPV model was unable to find a solution for the cel
              1l temperature at the given timestep.  Please report this error to
     1your TRNSYS distributor.'
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
      TIME0=getSimulationStartTime()
      TFINAL=getSimulationStopTime()
      DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
      IF(INFO(7).EQ.-2) THEN
                INFO(12)=16
                RETURN 1
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
                RETURN 1
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
```

```
      IF(INFO(13).GT.0) THEN
                 RETURN 1
               ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C       RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
        IUNIT=INFO(1)
                 ITYPE=INFO(2)

C       SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
        INFO(6)=NOUT
        INFO(9)=1
                 INFO(10)=0

C       SET THE NUMBER OF PARAMETERS AND INPUTS
        MODE_EFF=JFIX(PAR(10)+0.5)
        NPAR=10
                 NIN=9

        IF(MODE_EFF.LT.1) CALL TYPECK(-4,INFO,0,10,0)
        IF(MODE_EFF.GT.3) CALL TYPECK(-4,INFO,0,10,0)
                 IF(ERRORFOUND()) RETURN 1

                 IF(MODE_EFF.EQ.1) THEN
                   NPAR=NPAR+5
        ELSE IF(MODE_EFF.EQ.2) THEN
                   NPAR=NPAR+3
        ELSE IF(MODE_EFF.EQ.3) THEN
                   NIN=NIN+1
                 ENDIF

C       CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED IN
C       THE TRNSYS INPUT FILE
                 CALL TYPECK(1,INFO,NIN,NPAR,ND)

C       SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
        DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','IR1','DG1','HT1',
             1          'PR2','DM1'/
        DATA OCHECK/'TE1','MF1','PW1','DM1','PW1','DM1','TE1','TE1',
             1          'TE1','TE1','TE1','PW1','PW1','PW1','PW1'/

C       CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
        CALL RCHECK(INFO,YCHECK,OCHECK)

C       RETURN TO THE CALLING PROGRAM
        RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C       SET THE UNIT NUMBER FOR FUTURE CALLS
        IUNIT=INFO(1)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
        LENGTH=PAR(1)
        WIDTH=PAR(2)
                 ABS_PLATE=PAR(3)
                 EMISS_PLATE=PAR(4)
        R_2=PAR(5)
                 EMISS_1=PAR(6)
                 EMISS_2=PAR(7)
        R_3=PAR(8)
                 THICK_CHANNEL=PAR(9)
        MODE_EFF=JFIX(PAR(10)+0.5)

C       CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
        IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
        IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
        IF(ABS_PLATE.LT.0.) CALL TYPECK(-4,INFO,0,3,0)
        IF(ABS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,3,0)
        IF(EMISS_PLATE.LT.0.) CALL TYPECK(-4,INFO,0,4,0)
        IF(EMISS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,4,0)
        IF(R_2.LE.0.) CALL TYPECK(-4,INFO,0,5,0)
        IF(EMISS_1.LT.0.) CALL TYPECK(-4,INFO,0,6,0)
        IF(EMISS_1.GT.1.) CALL TYPECK(-4,INFO,0,6,0)
        IF(EMISS_2.LT.0.) CALL TYPECK(-4,INFO,0,7,0)
        IF(EMISS_2.GT.1.) CALL TYPECK(-4,INFO,0,7,0)
        IF(R_3.LE.0.) CALL TYPECK(-4,INFO,0,8,0)
        IF(THICK_CHANNEL.LT.0.) CALL TYPECK(-4,INFO,0,9,0)

                 IF(ERRORFOUND()) RETURN 1

C       GET THE PV EFFICIENCY SPECIFIC PARAMETERS
                 IF(MODE_EFF.EQ.1) THEN
          EFF_PV_REF=PAR(11)
```

```
          T_REF=PAR(12)
          GT_REF=PAR(13)
          EFF_CORR_T=PAR(14)
          EFF_CORR_I=PAR(15)

          IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,11,0)
          IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,11,0)
          IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,13,0)

        ELSE IF(MODE_EFF.EQ.2) THEN
                LU_DATA=JFIX(PAR(11)+0.5)
                N_TEMPS=JFIX(PAR(12)+0.5)
                N_RADS=JFIX(PAR(13)+0.5)

                        IF(LU_DATA.LT.10) CALL TYPECK(-4,INFO,0,11,0)
                IF(N_TEMPS.LT.2) CALL TYPECK(-4,INFO,0,12,0)
                IF(N_RADS.LT.2) CALL TYPECK(-4,INFO,0,13,0)

               ENDIF

C      PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
        OUT(1)=XIN(1)
        OUT(2:6)=0.
        OUT(7)=T_AMB
        OUT(8)=XIN(1)
        OUT(9)=XIN(1)
        OUT(10)=XIN(1)
        OUT(11)=XIN(5)
               OUT(12:15)=0.

C      RETURN TO THE CALLING PROGRAM
        RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
      IF(INFO(1).NE.IUNIT) THEN

C      RESET THE UNIT NUMBER
               IUNIT=INFO(1)
               ITYPE=INFO(2)

C      READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
        LENGTH=PAR(1)
        WIDTH=PAR(2)
               ABS_PLATE=PAR(3)
               EMISS_PLATE=PAR(4)
        R_2=PAR(5)
               EMISS_1=PAR(6)
               EMISS_2=PAR(7)
        R_3=PAR(8)
               THICK_CHANNEL=PAR(9)
        MODE_EFF=JFIX(PAR(10)+0.5)
               IF(MODE_EFF.EQ.1) THEN
          EFF_PV_REF=PAR(11)
          T_REF=PAR(12)
          GT_REF=PAR(13)
          EFF_CORR_T=PAR(14)
          EFF_CORR_I=PAR(15)
        ELSE IF(MODE_EFF.EQ.2) THEN
                LU_DATA=JFIX(PAR(11)+0.5)
                N_TEMPS=JFIX(PAR(12)+0.5)
                N_RADS=JFIX(PAR(13)+0.5)
               ENDIF

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
60    T_FLUID_IN=XIN(1)
      FLOW_IN=XIN(2)
      T_AMB=XIN(3)
      T_SKY=XIN(4)
      T_SURF=XIN(5)
      GT=XIN(6)
      SLOPE=XIN(7)
      H_CONV_T=XIN(8)
             P_ATM=XIN(9)
             IF(MODE_EFF.EQ.3) THEN
               EFF_PV=XIN(10)
             ELSE
               EFF_PV=0.
             ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    CHECK THE INPUTS FOR PROBLEMS
      IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
      IF(GT.LT.0.) CALL TYPECK(-3,INFO,6,0,0)
      IF(H_CONV_T.LT.0.) CALL TYPECK(-3,INFO,8,0,0)
      IF(P_ATM.LE.0.) CALL TYPECK(-3,INFO,9,0,0)
      IF(EFF_PV.LT.0.) CALL TYPECK(-3,INFO,10,0,0)
      IF(EFF_PV.GT.1.) CALL TYPECK(-3,INFO,10,0,0)
```

```
                IF(ERRORFOUND()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C    SET SOME GEOMETRIC PARAMETERS
     AREA=LENGTH*WIDTH
                DIAMETER=4.*WIDTH*THICK_CHANNEL/(2.*WIDTH+2.*THICK_CHANNEL)

C    SET THE INCIDENCE ANGLE MODIFIER
     XKAT=1.
                TAUALPHAN=ABS_PLATE

C    GUESS UPPER AND LOWER FLOW CHANNEL SURFACE TEMPERATURES
     T_1=T_FLUID_IN
                T_2=T_FLUID_IN

C    GUESS A PV CELL TEMPERATURE
     T_PV=T_AMB
                T_PV_OLD=T_PV

C    GUESS A MEAN FLUID TEMPERATURE
     T_FLUID_MEAN=T_FLUID_IN

C    INITIALIZE THE COUNTER
                ICOUNT=1

C    GET THE TOP SURFACE RADIATION COEFFICIENT
10    H_RAD_T=H_RADIATION(T_PV,T_SKY,EMISS_PLATE)

C    GET THE INNER-CHANNEL RADIATION COEFFICIENT
     H_RAD_12=H_RAD_GRAY(T_1,T_2,EMISS_1,EMISS_2)

C    GET THE PV CELL EFFICIENCY
     IF(MODE_EFF.EQ.1) THEN
       FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
       FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
       PV_EFFICIENCY=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))
                ELSE IF(MODE_EFF.EQ.2) THEN
       X(2)=GT
                  X(1)=T_PV
                  NX(2)=N_RADS
                  NX(1)=N_TEMPS
                  CALL DYNAMICDATA(LU_DATA,2,NX,1,X,Y,INFO,*20)
       CALL LINKCK('TYPE 542','DYNAMICDATA',1,99)
20       IF(ERRORFOUND()) RETURN 1
       PV_EFFICIENCY=Y(1)
                ELSE
                  PV_EFFICIENCY=EFF_PV

                ENDIF

C    SET THE PROPERTIES OF THE AIR STREAM
                T_PROPS_K=T_FLUID_MEAN+273.15
                P_KPA=P_ATM*101.325
     CALL AIRPROP(T_PROPS_K,P_KPA,AIRPROPS)

                RHO_AIR=1./AIRPROPS(1)       !KG/M3
                VISC_AIR=AIRPROPS(2)*3600.   !KG/M/HR
                PRANDTL_AIR=AIRPROPS(3)      !DIMENSIONLESS
                K_AIR=AIRPROPS(4)*3.6        !KJ/H/M/K
                CP_AIR=AIRPROPS(5)           !KJ/KG/K

C    CALCULATE THE REYNOLDS NUMBER
     REYNOLDS=4.*FLOW_IN/PI/DIAMETER/VISC_AIR

C    CALCULATE THE FLUID CONVECTION COEFFICIENT
     IF(REYNOLDS.LE.0.) THEN

C      CALL THE SUBROUTINE TO CALCULATE THE NUSSELT NUMBER FOR INCLINED FLAT PLATES
       T_1_K=T_1+273.15
       T_2_K=T_2+273.15
       CALL NUFLATPLATE(SLOPE,T_1_K,T_2_K,THICK_CHANNEL,P_KPA,NUSSELT)

C      SET THE NATURAL CONVECTION HEAT TRANSFER COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER

     ELSE IF(REYNOLDS.LE.2300.) THEN

C      SET THE NUSSELT NUMBER (BASED ON A CONSTANT SURFACE TEMPERATURE)
       NUSSELT=3.66

C      SET THE LAMINAR CONVECTION COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER

                ELSE

C      SET THE NUSSELT NUMBER (BASED ON THE DITTUUS BOELTER EQUATION)
       T_SURF=(T_1+T_2)/2.

                  IF(T_SURF.GE.T_FLUID_MEAN) THEN
                    N=0.4
                  ELSE
                    N=0.3
                  ENDIF

       NUSSELT=0.023*(REYNOLDS**0.8)*(PRANDTL_AIR**N)

C      SET THE TURBULENT CONVECTION COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER
```

```
      ENDIF

C   CALCULATE THE ABSORBED SOLAR RADIATION PER UNIT AREA
      S=TAUALPHAN*XKAT*GT*(1.-PV_EFFICIENCY)

C   SET SOME CONSTANTS WE'LL NEED
      F_PRIME=R_2*H_CONV_T+R_2*H_RAD_T+1.
      J=H_RAD_12+H_FLUID+1./R_3
      M=1./R_2-1./R_2/F_PRIME+H_FLUID+H_RAD_12-H_RAD_12*H_RAD_12/J

C   REFORMULATE THE 6 ENERGY BALANCES TO FIND qu"=AA*T_fluid+CC
      AA=-2.*H_FLUID
           1   +H_FLUID*H_FLUID/M
           1   +H_FLUID*H_RAD_12*H_FLUID*2./M/J
     1   +H_FLUID*H_RAD_12*H_RAD_12*H_FLUID/J/M/J
     1   +H_FLUID*H_FLUID/J

      CC=H_FLUID*S/F_PRIME/M
           1  +H_FLUID*H_CONV_T*T_AMB/F_PRIME/M
           1  +H_FLUID*H_RAD_T*T_SKY/F_PRIME/M
           1  +H_FLUID*H_RAD_12*T_SURF/M/J/R_3
           1  +H_FLUID*H_RAD_12*S/J/M/F_PRIME
           1  +H_FLUID*H_RAD_12*H_CONV_T*T_AMB/J/F_PRIME/M
           1  +H_FLUID*H_RAD_12*H_RAD_T*T_SKY/J/F_PRIME/M
           1  +H_FLUID*H_RAD_12*H_RAD_12*T_SURF/J/M/J/R_3
           1  +H_FLUID*T_SURF/J/R_3

C   SET THE CASE WITH NO FLOW
      IF(FLOW_IN.LE.0.) THEN

C      SET THE USEFUL ENERGY GAIN
        QU=0.

C      SET THE MEAN FLUID TEMPERATURE FROM QU=AA(T)+CC
        T_FLUID_MEAN=-CC/AA

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
        T_1=S/F_PRIME/M+H_CONV_T*T_AMB/F_PRIME/M+H_RAD_T*T_SKY
           1      /F_PRIME/M+H_FLUID*T_FLUID_MEAN/M+H_RAD_12*H_FLUID*
     1      T_FLUID_MEAN/M/J+H_RAD_12*T_SURF/M/J/R_3

C      SET THE PV TEMPERATURE
        T_PV=R_2*(S/F_PRIME+H_CONV_T*T_AMB/F_PRIME+H_RAD_T*T_SKY/
           1       F_PRIME+T_1/R_2/F_PRIME)

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
        T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+T_SURF/J/R_3

C      SET THE FLUID OUTLET TEMPERATURE
        T_FLUID_OUT=T_FLUID_MEAN

      ELSE

C      FIND THE FLUID OUTLET TEMPERATURE BY INTEGRATING THE FLUID DIFFERENTIAL
EQUATION
        T_FLUID_OUT=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)-
           1      CC/AA

C      FIND THE MEAN FLUID TEMPERATURE BY INTEGRATING THE LOCAL FLUID TEMPERATURE
EQUATION
        T_FLUID_MEAN=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)/
           1      (AA*AREA/FLOW_IN/CP_AIR)-(T_FLUID_IN+CC/AA)/(AA*AREA/
     1      FLOW_IN/CP_AIR)-CC/AA

C      KNOWING THE INLET AND OUTLET TEMPERATURES CALCULATE THE COLLECTOR USEFUL
ENERGY GAIN
             QU=FLOW_IN*CP_AIR*(T_FLUID_OUT-T_FLUID_IN)

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
        T_1=S/F_PRIME/M+H_CONV_T*T_AMB/F_PRIME/M+H_RAD_T*T_SKY
           1      /F_PRIME/M+H_FLUID*T_FLUID_MEAN/M+H_RAD_12*H_FLUID*
     1      T_FLUID_MEAN/M/J+H_RAD_12*T_SURF/M/J/R_3

C      SET THE PV TEMPERATURE
        T_PV=R_2*(S/F_PRIME+H_CONV_T*T_AMB/F_PRIME+H_RAD_T*T_SKY/
           1       F_PRIME+T_1/R_2/F_PRIME)

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
        T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+T_SURF/J/R_3

      ENDIF

C   CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
      IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
           1   (ICOUNT.LT.1000)) THEN
             T_PV_OLD=T_PV
        ICOUNT=ICOUNT+1
        GOTO 10
      ENDIF

C   CALCULATE THE THERMAL EFFICIENCY
      IF(GT.GT.0.) THEN
        EFF_THERMAL=QU/AREA/GT
           ELSE
             EFF_THERMAL=0.
      ENDIF

C   CALCULATE THE HEAT TRANSFERS
      Q_TOP_CONV=H_CONV_T*AREA*(T_PV-T_AMB)
      Q_TOP_RAD=H_RAD_T*AREA*(T_PV-T_SKY)
      Q_BACK=AREA*(T_2-T_SURF)/R_3
           Q_ABS=AREA*GT*TAUALPHAN*XKAT*(1.-PV_EFFICIENCY)
```

```
C    CALCULATE THE PV POWER PRODUCTION
     POWER=TAUALPHAN*XKAT*GT*PV_EFFICIENCY

C    SET THE OUTPUTS
     OUT(1)=T_FLUID_OUT
     OUT(2)=FLOW_IN
     OUT(3)=QU
              OUT(4)=EFF_THERMAL
     OUT(5)=POWER
     OUT(6)=PV_EFFICIENCY
     OUT(7)=T_PV
     OUT(8)=T_1
     OUT(9)=T_FLUID_MEAN
     OUT(10)=T_2
              OUT(11)=T_SURF
              OUT(12)=Q_TOP_CONV
              OUT(13)=Q_TOP_RAD
              OUT(14)=Q_BACK
              OUT(15)=Q_ABS

C    KILL THE SIMULATION IF THE SYSTEM WAS UNABLE TO SOLVE
     IF(ICOUNT.GE.1000) THEN
       CALL MESSAGES(-1,MESSAGE1,'FATAL',IUNIT,ITYPE)
              ENDIF
C---------------------------------------------------------------------------------------------------------------------------

C---------------------------------------------------------------------------------------------------------------------------
C    EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
     RETURN 1
     END
C---------------------------------------------------------------------------------------------------------------------------
```

TYPE 569-Unglazed building integrated photovoltaic system (air) (Convective and radiative losses at the back of the collector)

```
      SUBROUTINE TYPE569(TIME,XIN,OUT,T,DTDT,PAR,INFO,ICNTRL,*)

C-----------------------------------------------------------------------------------------------------------------------
C    DESCRIPTION:
C    THIS COMPONENT MODELS A BUILDING-INTEGRATED PHOTOVOLATIC SYSTEM.  IN THIS
MODEL THERE IS NO COVER
C    OVER THE PV MATERIAL.  THE BACK OF THE BIPV SYSTEM INTERACTS WITH MODELS WHERE
THE BACK-SIDE AIR
C    TEMPERATURE AND BACK-SIDE RADIATIVE SURFACE TEMPERATURE ARE KNOWN. THE
CONVECTION CALCULATIONS
C    ARE FROM CORRELATIONS PROVIDED BY "INTRODUCTION TO HEAT TRANSFER" BY
INCROPERA AND DEWITT.
C
C    LAST MODIFIED:
C       APRIL 2004 - JEFF THORNTON - ORIGINAL CODING
C-----------------------------------------------------------------------------------------------------------------------
```


```
C-----------------------------------------------------------------------------------------------------------------------
      !Export this subroutine for its use in external DLLs.
      !DEC$ATTRIBUTES DLLEXPORT :: TYPE569
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ACCESS TRNSYS FUNCTIONS
              USE TrnsysConstants
              USE TrnsysFunctions
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    TRNSYS DECLARATIONS
      IMPLICIT NONE
              DOUBLE PRECISION XIN,OUT,TIME,PAR,T,DTDT,TIME0,TFINAL,DELT

      INTEGER*4 INFO(15),NP,NI,NOUT,ND,IUNIT,ITYPE,ICNTRL,NPAR,NIN
      CHARACTER*3 YCHECK,OCHECK
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED PARAMETERS FOR THE SIZING OF THE ARRAYS
      PARAMETER (NP=16,NI=12,NOUT=16,ND=0)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    REQUIRED TRNSYS DIMENSIONS
      DIMENSION XIN(NI),OUT(NOUT),PAR(NP),YCHECK(NI),OCHECK(NOUT)
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DECLARATIONS AND DEFINITIONS FOR THE USER-VARIABLES
      DOUBLE PRECISION T_1,T_2,AREA,FLOW_IN,T_AMB,GT,SLOPE,EMISS_BACK,
     1  TAUALPHAN,F_PRIME,T_FLUID_MEAN,T_SKY,H_CONV_T,X(2),NUSSELT,
     1  H_FLUID,QU,EFF_THERMAL,S,R_2,R_3,Q_TOP_CONV,N,H_RAD_12,
     1  Q_TOP_RAD,Q_ABS,H_RADIATION,EMISS_1,EMISS_2,H_RAD_GRAY,
     1  M,AA,CC,T_1_K,T_2_K,T_SURF,CP_AIR,ABS_PLATE,EFF_PV,Y(1),
     1  EFF_PV_REF,T_REF,GT_REF,K_AIR,EFF_CORR_T,EFF_CORR_I,T_FLUID_IN,
     1  XKAT,T_PV,T_PV_OLD,FACTOR_T,FACTOR_I,REYNOLDS,EMISS_PLATE,
     1  PV_EFFICIENCY,POWER,H_RAD_T,J,T_FLUID_OUT,AIRPROPS(5),PI,
     1  WIDTH,THICK_CHANNEL,P_ATM,P_KPA,DIAMETER,T_PROPS_K,RHO_AIR,
     1  VISC_AIR,PRANDTL_AIR,LENGTH,T_ZONE,T_ZONE_RAD,H_CONV_B,T_3,
              1  H_RAD_B,H_PRIME,Q_BACK_CONV,Q_BACK_RAD

      INTEGER ICOUNT,MODE_EFF,LU_DATA,N_TEMPS,N_RADS,NX(2)
      CHARACTER(LEN=MAXMESSAGELENGTH)::MESSAGE1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DATA STATEMENTS
      DATA PI/3.14159265358979/
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    ERROR MESSAGES
      MESSAGE1='The BIPV model was unable to find a solution for the cel
              1l temperature at the given timestep.  Please report this error to
     1your TRNSYS distributor.'
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    GET GLOBAL TRNSYS SIMULATION VARIABLES
      TIME0=getSimulationStartTime()
      TFINAL=getSimulationStopTime()
      DELT=getSimulationTimeStep()
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    SET THE VERSION INFORMATION FOR TRNSYS
      IF(INFO(7).EQ.-2) THEN
                INFO(12)=16
                RETURN 1
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    DO ALL THE VERY LAST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(8).EQ.-1) THEN
                RETURN 1
              ENDIF
C-----------------------------------------------------------------------------------------------------------------------
```

```
C-----------------------------------------------------------------------------------------------------------------------
C   PERFORM ANY "AFTER-ITERATION" MANIPULATIONS THAT ARE REQUIRED
      IF(INFO(13).GT.0) THEN
         RETURN 1
      ENDIF

C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   DO ALL THE VERY FIRST CALL OF THE SIMULATION MANIPULATIONS HERE
      IF (INFO(7).EQ.-1) THEN

C      RETRIEVE THE UNIT NUMBER AND TYPE NUMBER FOR THIS COMPONENT FROM THE INFO
ARRAY
         IUNIT=INFO(1)
         ITYPE=INFO(2)

C      SET SOME INFO ARRAY VARIABLES TO TELL THE TRNSYS ENGINE HOW THIS TYPE IS TO
WORK
         INFO(6)=NOUT
         INFO(9)=1
         INFO(10)=0

C      SET THE NUMBER OF PARAMETERS AND INPUTS
         MODE_EFF=JFIX(PAR(11)+0.5)
         NPAR=11
         NIN=11

         IF(MODE_EFF.LT.1) CALL TYPECK(-4,INFO,0,10,0)
         IF(MODE_EFF.GT.3) CALL TYPECK(-4,INFO,0,10,0)
         IF(ERRORFOUND()) RETURN 1

         IF(MODE_EFF.EQ.1) THEN
           NPAR=NPAR+5
         ELSE IF(MODE_EFF.EQ.2) THEN
           NPAR=NPAR+3
         ELSE IF(MODE_EFF.EQ.3) THEN
           NIN=NIN+1
         ENDIF

C      CALL THE TYPE CHECK SUBROUTINE TO COMPARE WHAT THIS COMPONENT REQUIRES TO
WHAT IS SUPPLIED IN
C      THE TRNSYS INPUT FILE
         CALL TYPECK(1,INFO,NIN,NPAR,ND)

C      SET THE YCHECK AND OCHECK ARRAYS TO CONTAIN THE CORRECT VARIABLE TYPES FOR
THE INPUTS AND OUTPUTS
         DATA YCHECK/'TE1','MF1','TE1','TE1','TE1','TE1','IR1','DG1',
     1        'HT1','HT1','PR2','DM1'/
         DATA OCHECK/'TE1','MF1','PW1','DM1','PW1','DM1','TE1','TE1',
     1        'TE1','TE1','TE1','PW1','PW1','PW1','PW1','PW1'/

C      CALL THE RCHECK SUBROUTINE TO SET THE CORRECT INPUT AND OUTPUT TYPES FOR THIS
COMPONENT
         CALL RCHECK(INFO,YCHECK,OCHECK)

C      RETURN TO THE CALLING PROGRAM
         RETURN 1

      ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C   DO ALL OF THE INITIAL TIMESTEP MANIPULATIONS HERE - THERE ARE NO ITERATIONS AT THE
INTIAL TIME
      IF (TIME.LT.(TIME0+DELT/2.D0)) THEN

C      SET THE UNIT NUMBER FOR FUTURE CALLS
         IUNIT=INFO(1)

C      READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
         LENGTH=PAR(1)
         WIDTH=PAR(2)
         ABS_PLATE=PAR(3)
         EMISS_PLATE=PAR(4)
         R_2=PAR(5)
         EMISS_1=PAR(6)
         EMISS_2=PAR(7)
         R_3=PAR(8)
         EMISS_BACK=PAR(9)
         THICK_CHANNEL=PAR(10)
         MODE_EFF=JFIX(PAR(11)+0.5)

C      CHECK THE PARAMETERS FOR PROBLEMS AND RETURN FROM THE SUBROUTINE IF AN
ERROR IS FOUND
         IF(LENGTH.LE.0.) CALL TYPECK(-4,INFO,0,1,0)
         IF(WIDTH.LE.0.) CALL TYPECK(-4,INFO,0,2,0)
         IF(ABS_PLATE.LT.0.) CALL TYPECK(-4,INFO,0,3,0)
         IF(ABS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,3,0)
         IF(EMISS_PLATE.LT.0.) CALL TYPECK(-4,INFO,0,4,0)
         IF(EMISS_PLATE.GT.1.) CALL TYPECK(-4,INFO,0,4,0)
         IF(R_2.LE.0.) CALL TYPECK(-4,INFO,0,5,0)
         IF(EMISS_1.LT.0.) CALL TYPECK(-4,INFO,0,6,0)
         IF(EMISS_1.GT.1.) CALL TYPECK(-4,INFO,0,6,0)
         IF(EMISS_2.LT.0.) CALL TYPECK(-4,INFO,0,7,0)
         IF(EMISS_2.GT.1.) CALL TYPECK(-4,INFO,0,7,0)
         IF(R_3.LE.0.) CALL TYPECK(-4,INFO,0,8,0)
         IF(EMISS_BACK.LT.0.) CALL TYPECK(-4,INFO,0,9,0)
         IF(EMISS_BACK.GT.1.) CALL TYPECK(-4,INFO,0,9,0)
         IF(THICK_CHANNEL.LT.0.) CALL TYPECK(-4,INFO,0,10,0)

         IF(ERRORFOUND()) RETURN 1
```

```
C       GET THE PV EFFICIENCY SPECIFIC PARAMETERS
              IF(MODE_EFF.EQ.1) THEN
         EFF_PV_REF=PAR(12)
         T_REF=PAR(13)
         GT_REF=PAR(14)
         EFF_CORR_T=PAR(15)
         EFF_CORR_I=PAR(16)

         IF(EFF_PV_REF.LT.0.) CALL TYPECK(-4,INFO,0,12,0)
         IF(EFF_PV_REF.GT.1.) CALL TYPECK(-4,INFO,0,12,0)
         IF(GT_REF.LE.0.) CALL TYPECK(-4,INFO,0,14,0)

      ELSE IF(MODE_EFF.EQ.2) THEN
               LU_DATA=JFIX(PAR(12)+0.5)
               N_TEMPS=JFIX(PAR(13)+0.5)
               N_RADS=JFIX(PAR(14)+0.5)

                    IF(LU_DATA.LT.10) CALL TYPECK(-4,INFO,0,12,0)
               IF(N_TEMPS.LT.2) CALL TYPECK(-4,INFO,0,13,0)
               IF(N_RADS.LT.2) CALL TYPECK(-4,INFO,0,14,0)

              ENDIF

C       PERFORM ANY REQUIRED CALCULATIONS TO SET THE INITIAL VALUES OF THE OUTPUTS
HERE
      OUT(1)=XIN(1)
      OUT(2:6)=0.
      OUT(7)=T_AMB
      OUT(8)=XIN(1)
      OUT(9)=XIN(1)
      OUT(10)=XIN(1)
      OUT(11)=XIN(5)
              OUT(12:16)=0.

C       RETURN TO THE CALLING PROGRAM
      RETURN 1

     ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    *** ITS AN ITERATIVE CALL TO THIS COMPONENT ***
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RE-READ THE PARAMETERS IF ANOTHER UNIT OF THIS TYPE HAS BEEN CALLED
     IF(INFO(1).NE.IUNIT) THEN

C       RESET THE UNIT NUMBER
              IUNIT=INFO(1)
              ITYPE=INFO(2)

C       READ IN THE VALUES OF THE PARAMETERS IN SEQUENTIAL ORDER
      LENGTH=PAR(1)
      WIDTH=PAR(2)
              ABS_PLATE=PAR(3)
              EMISS_PLATE=PAR(4)
      R_2=PAR(5)
              EMISS_1=PAR(6)
              EMISS_2=PAR(7)
      R_3=PAR(8)
              EMISS_BACK=PAR(9)
              THICK_CHANNEL=PAR(10)
      MODE_EFF=JFIX(PAR(11)+0.5)
              IF(MODE_EFF.EQ.1) THEN
        EFF_PV_REF=PAR(12)
        T_REF=PAR(13)
        GT_REF=PAR(14)
        EFF_CORR_T=PAR(15)
        EFF_CORR_I=PAR(16)
      ELSE IF(MODE_EFF.EQ.2) THEN
               LU_DATA=JFIX(PAR(12)+0.5)
               N_TEMPS=JFIX(PAR(13)+0.5)
               N_RADS=JFIX(PAR(14)+0.5)
              ENDIF

     ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    RETRIEVE THE CURRENT VALUES OF THE INPUTS TO THIS MODEL FROM THE XIN ARRAY IN
SEQUENTIAL ORDER
60    T_FLUID_IN=XIN(1)
     FLOW_IN=XIN(2)
     T_AMB=XIN(3)
     T_SKY=XIN(4)
     T_ZONE=XIN(5)
     T_ZONE_RAD=XIN(6)
     GT=XIN(7)
     SLOPE=XIN(8)
     H_CONV_T=XIN(9)
     H_CONV_B=XIN(10)
             P_ATM=XIN(11)
             IF(MODE_EFF.EQ.3) THEN
               EFF_PV=XIN(12)
             ELSE
               EFF_PV=0.
             ENDIF
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
```

```
C    CHECK THE INPUTS FOR PROBLEMS
     IF(FLOW_IN.LT.0.) CALL TYPECK(-3,INFO,2,0,0)
     IF(GT.LT.0.) CALL TYPECK(-3,INFO,7,0,0)
     IF(H_CONV_T.LT.0.) CALL TYPECK(-3,INFO,9,0,0)
     IF(H_CONV_B.LT.0.) CALL TYPECK(-3,INFO,10,0,0)
     IF(P_ATM.LE.0.) CALL TYPECK(-3,INFO,11,0,0)
     IF(EFF_PV.LT.0.) CALL TYPECK(-3,INFO,12,0,0)
     IF(EFF_PV.GT.1.) CALL TYPECK(-3,INFO,12,0,0)

           IF(ERRORFOUND()) RETURN 1
C-----------------------------------------------------------------------------------------------------------------------

C-----------------------------------------------------------------------------------------------------------------------
C    PERFORM ALL THE CALCULATION HERE FOR THIS MODEL.

C    SET SOME GEOMETRIC PARAMETERS
     AREA=LENGTH*WIDTH
           DIAMETER=4.*WIDTH*THICK_CHANNEL/(2.*WIDTH+2.*THICK_CHANNEL)

C    SET THE INCIDENCE ANGLE MODIFIER
     XKAT=1.
           TAUALPHAN=ABS_PLATE

C    GUESS UPPER AND LOWER FLOW CHANNEL SURFACE TEMPERATURES
     T_1=T_FLUID_IN
           T_2=T_FLUID_IN

C    GUESS A BACK COLLECTOR SURFACE TEMPERATURE
     T_3=(T_ZONE+T_ZONE_RAD)/2.

C    GUESS A PV CELL TEMPERATURE
     T_PV=T_AMB
           T_PV_OLD=T_PV

C    GUESS A MEAN FLUID TEMPERATURE
     T_FLUID_MEAN=T_FLUID_IN

C    INITIALIZE THE COUNTER
           ICOUNT=1

C    GET THE TOP SURFACE RADIATION COEFFICIENT
10    H_RAD_T=H_RADIATION(T_PV,T_SKY,EMISS_PLATE)

C    GET THE BOTTOM SURFACE RADIATION COEFFICIENT
     H_RAD_B=H_RADIATION(T_3,T_ZONE_RAD,EMISS_BACK)

C    GET THE INNER-CHANNEL RADIATION COEFFICIENT
     H_RAD_12=H_RAD_GRAY(T_1,T_2,EMISS_1,EMISS_2)

C    GET THE PV CELL EFFICIENCY
     IF(MODE_EFF.EQ.1) THEN
```

```
       FACTOR_T=1+EFF_CORR_T*(T_PV-T_REF)
       FACTOR_I=1+EFF_CORR_I*(GT-GT_REF)
       PV_EFFICIENCY=DMAX1(0.,(FACTOR_T*FACTOR_I*EFF_PV_REF))
           ELSE IF(MODE_EFF.EQ.2) THEN
       X(2)=GT
            X(1)=T_PV
            NX(2)=N_RADS
            NX(1)=N_TEMPS
            CALL DYNAMICDATA(LU_DATA,2,NX,1,X,Y,INFO,*20)
       CALL LINKCK('TYPE 542','DYNAMICDATA',1,99)
20      IF(ERRORFOUND()) RETURN 1
       PV_EFFICIENCY=Y(1)
           ELSE
            PV_EFFICIENCY=EFF_PV
           ENDIF

C    SET THE PROPERTIES OF THE AIR STREAM
           T_PROPS_K=T_FLUID_MEAN+273.15
           P_KPA=P_ATM*101.325
     CALL AIRPROP(T_PROPS_K,P_KPA,AIRPROPS)

           RHO_AIR=1./AIRPROPS(1)       !KG/M3
           VISC_AIR=AIRPROPS(2)*3600.   !KG/M/HR
           PRANDTL_AIR=AIRPROPS(3)      !DIMENSIONLESS
           K_AIR=AIRPROPS(4)*3.6        !KJ/H/M/K
           CP_AIR=AIRPROPS(5)           !KJ/KG/K

C    CALCULATE THE REYNOLDS NUMBER
     REYNOLDS=4.*FLOW_IN/PI/DIAMETER/VISC_AIR

C    CALCULATE THE FLUID CONVECTION COEFFICIENT
     IF(REYNOLDS.LE.0.) THEN

C      CALL THE SUBROUTINE TO CALCULATE THE NUSSELT NUMBER FOR INCLINED FLAT PLATES
       T_1_K=T_1+273.15
       T_2_K=T_2+273.15
       CALL NUFLATPLATE(SLOPE,T_1_K,T_2_K,THICK_CHANNEL,P_KPA,NUSSELT)

C      SET THE NATURAL CONVECTION HEAT TRANSFER COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER

     ELSE IF(REYNOLDS.LE.2300.) THEN

C      SET THE NUSSELT NUMBER (BASED ON A CONSTANT SURFACE TEMPERATURE)
       NUSSELT=3.66

C      SET THE LAMINAR CONVECTION COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER

           ELSE
```

```
C      SET THE NUSSELT NUMBER (BASED ON THE DITTUUS BOELTER EQUATION)
       T_SURF=(T_1+T_2)/2.

          IF(T_SURF.GE.T_FLUID_MEAN) THEN
            N=0.4
          ELSE
            N=0.3
          ENDIF

       NUSSELT=0.023*(REYNOLDS**0.8)*(PRANDTL_AIR**N)

C      SET THE TURBULENT CONVECTION COEFFICIENT
       H_FLUID=NUSSELT*K_AIR/DIAMETER

     ENDIF

C    CALCULATE THE ABSORBED SOLAR RADIATION PER UNIT AREA
     S=TAUALPHAN*XKAT*GT*(1.-PV_EFFICIENCY)

C    SET SOME CONSTANTS WE'LL NEED
     F_PRIME=R_2*H_CONV_T+R_2*H_RAD_T+1.
     H_PRIME=1.+R_3*H_CONV_B+R_3*H_RAD_B
     J=H_RAD_12+H_FLUID+1./R_3-1./R_3/H_PRIME
     M=1./R_2-1./R_2/F_PRIME+H_FLUID+H_RAD_12-H_RAD_12*H_RAD_12/J

C    REFORMULATE THE 6 ENERGY BALANCES TO FIND qu"=AA*T_fluid+CC
     AA=-2.*H_FLUID
          1   +H_FLUID*H_FLUID/M
          1   +H_FLUID*H_RAD_12*H_FLUID*2./M/J
    1   +H_FLUID*H_RAD_12*H_RAD_12*H_FLUID/J/M/J
    1   +H_FLUID*H_FLUID/J

     CC=H_FLUID*S/F_PRIME/M
          1  +H_FLUID*H_CONV_T*T_AMB/F_PRIME/M
          1  +H_FLUID*H_RAD_T*T_SKY/F_PRIME/M
          1  +H_FLUID*H_RAD_12*H_CONV_B*T_ZONE/M/J/H_PRIME
          1  +H_FLUID*H_RAD_12*H_RAD_B*T_ZONE_RAD/M/J/H_PRIME
          1  +H_FLUID*H_CONV_B*T_ZONE/J/H_PRIME
          1  +H_FLUID*H_RAD_B*T_ZONE_RAD/J/H_PRIME
          1  +H_FLUID*H_RAD_12*S/J/M/F_PRIME
          1  +H_FLUID*H_RAD_12*H_CONV_T*T_AMB/J/F_PRIME/M
          1  +H_FLUID*H_RAD_12*H_RAD_T*T_SKY/J/F_PRIME/M
    1  +H_FLUID*H_RAD_12*H_RAD_12*H_CONV_B*T_ZONE/J/M/J/H_PRIME
    1  +H_FLUID*H_RAD_12*H_RAD_12*H_RAD_B*T_ZONE_RAD/J/M/J/H_PRIME

C    SET THE CASE WITH NO FLOW
     IF(FLOW_IN.LE.0.) THEN

C      SET THE USEFUL ENERGY GAIN
       QU=0.
```

```
C      SET THE MEAN FLUID TEMPERATURE FROM QU=AA(T)+CC
       T_FLUID_MEAN=-CC/AA

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
       T_1=S/F_PRIME/M
          1      +H_CONV_T*T_AMB/F_PRIME/M
    1     +H_RAD_T*T_SKY/F_PRIME/M
    1     +H_FLUID*T_FLUID_MEAN/M
    1     +H_RAD_12*H_FLUID*T_FLUID_MEAN/M/J
    1     +H_RAD_12*H_CONV_B*T_ZONE/M/J/H_PRIME
    1     +H_RAD_12*H_RAD_B*T_ZONE_RAD/M/J/H_PRIME

C      SET THE PV TEMPERATURE
       T_PV=R_2*(S/F_PRIME+H_CONV_T*T_AMB/F_PRIME+H_RAD_T*T_SKY/
          1       F_PRIME+T_1/R_2/F_PRIME)

C      SET THE LOWER CHANNEL SURFACE TEMPERATURE
       T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+H_CONV_B*T_ZONE/J/
          1      H_PRIME+H_RAD_B*T_ZONE_RAD/J/H_PRIME

C      SET THE BACK SURFACE TEMPERATURE
       T_3=T_2/H_PRIME+R_3*H_CONV_B*T_ZONE/H_PRIME+R_3*H_RAD_B*
          1      T_ZONE_RAD/H_PRIME

C      SET THE FLUID OUTLET TEMPERATURE
       T_FLUID_OUT=T_FLUID_MEAN

     ELSE

C      FIND THE FLUID OUTLET TEMPERATURE BY INTEGRATING THE FLUID DIFFERENTIAL
EQUATION
       T_FLUID_OUT=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)-
          1      CC/AA

C      FIND THE MEAN FLUID TEMPERATURE BY INTEGRATING THE LOCAL FLUID TEMPERATURE
EQUATION
       T_FLUID_MEAN=(T_FLUID_IN+CC/AA)*DEXP(AA*AREA/FLOW_IN/CP_AIR)/
          1      (AA*AREA/FLOW_IN/CP_AIR)-(T_FLUID_IN+CC/AA)/(AA*AREA/
    1      FLOW_IN/CP_AIR)-CC/AA

C      KNOWING THE INLET AND OUTLET TEMPERATURES CALCULATE THE COLLECTOR USEFUL
ENERGY GAIN
            QU=FLOW_IN*CP_AIR*(T_FLUID_OUT-T_FLUID_IN)

C      SET THE UPPER CHANNEL SURFACE TEMPERATURE
       T_1=S/F_PRIME/M
          1      +H_CONV_T*T_AMB/F_PRIME/M
    1     +H_RAD_T*T_SKY/F_PRIME/M
    1     +H_FLUID*T_FLUID_MEAN/M
    1     +H_RAD_12*H_FLUID*T_FLUID_MEAN/M/J
    1     +H_RAD_12*H_CONV_B*T_ZONE/M/J/H_PRIME
```

```
     1     +H_RAD_12*H_RAD_B*T_ZONE_RAD/M/J/H_PRIME

C       SET THE PV TEMPERATURE
        T_PV=R_2*(S/F_PRIME+H_CONV_T*T_AMB/F_PRIME+H_RAD_T*T_SKY/
     1       F_PRIME+T_1/R_2/F_PRIME)

C       SET THE LOWER CHANNEL SURFACE TEMPERATURE
        T_2=H_RAD_12*T_1/J+H_FLUID*T_FLUID_MEAN/J+H_CONV_B*T_ZONE/J/
     1      H_PRIME+H_RAD_B*T_ZONE_RAD/J/H_PRIME

C       SET THE BACK SURFACE TEMPERATURE
        T_3=T_2/H_PRIME+R_3*H_CONV_B*T_ZONE/H_PRIME+R_3*H_RAD_B*
     1      T_ZONE_RAD/H_PRIME

      ENDIF

C     CHECK TO SEE IF MORE ITERATIONS ARE REQUIRED
      IF((DABS(T_PV_OLD-T_PV).GT.0.001).AND.
     1   (ICOUNT.LT.1000)) THEN
        T_PV_OLD=T_PV
        ICOUNT=ICOUNT+1
        GOTO 10
      ENDIF

C     CALCULATE THE THERMAL EFFICIENCY
      IF(GT.GT.0.) THEN
        EFF_THERMAL=QU/AREA/GT
      ELSE
        EFF_THERMAL=0.
      ENDIF

C     CALCULATE THE HEAT TRANSFERS
      Q_TOP_CONV=H_CONV_T*AREA*(T_PV-T_AMB)
      Q_TOP_RAD=H_RAD_T*AREA*(T_PV-T_SKY)
      Q_BACK_CONV=H_CONV_B*AREA*(T_3-T_ZONE)
      Q_BACK_RAD=H_RAD_B*AREA*(T_3-T_ZONE_RAD)
      Q_ABS=AREA*GT*TAUALPHAN*XKAT*(1.-PV_EFFICIENCY)

C     CALCULATE THE PV POWER PRODUCTION
      POWER=TAUALPHAN*XKAT*GT*PV_EFFICIENCY

C     SET THE OUTPUTS
      OUT(1)=T_FLUID_OUT
      OUT(2)=FLOW_IN
      OUT(3)=QU
      OUT(4)=EFF_THERMAL
      OUT(5)=POWER
      OUT(6)=PV_EFFICIENCY
      OUT(7)=T_PV
      OUT(8)=T_1
      OUT(9)=T_FLUID_MEAN
      OUT(10)=T_2
      OUT(11)=T_3
      OUT(12)=Q_TOP_CONV
      OUT(13)=Q_TOP_RAD
      OUT(14)=Q_BACK_CONV
      OUT(15)=Q_BACK_RAD
      OUT(16)=Q_ABS

C     KILL THE SIMULATION IF THE SYSTEM WAS UNABLE TO SOLVE
      IF(ICOUNT.GE.1000) THEN
        CALL MESSAGES(-1,MESSAGE1,'FATAL',IUNIT,ITYPE)
      ENDIF
C-----------------------------------------------------------------------------------------------------------------------
C-----------------------------------------------------------------------------------------------------------------------
C     EVERYTHING IS DONE - RETURN FROM THIS SUBROUTINE AND MOVE ON
      RETURN 1
      END
C-----------------------------------------------------------------------------------------------------------------------
```

Appendix B

Input and Output Parameters for TRNSYS Solar Thermal, PV, and PV/T Collector Models

TYPE 1: Flat Plate Solar Collector

TYPE 1a-Flat Plate Solar collector (No incidence angle modification)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Tested flow rate		
6	Intercept efficiency		
7	Efficiency slope		
8	Efficiency curvature		
9	Optical Mode 1		

TYPE 1b- Flat Plate Solar collector (2nd order incidence angle modifiers)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Tested flow rate	Total horizontal radiation	
6	Intercept efficiency	Horizontal diffuse radiation	
7	Efficiency slope	Ground reflectance	
8	Efficiency curvature	Incidence angle	
9	Optical mode 2	Collector slope	
10	1st-order IAM		
11	2nd-order IAM		

TYPE 1c- Flat Plate Solar collector (Modifiers=f(incidence angle))

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Tested flow rate	Total horizontal radiation	
6	Intercept efficiency	Horizontal diffuse radiation	
7	Efficiency slope	Ground reflectance	
8	Efficiency curvature	Incidence angle	
9	Optical mode 3	Collector slope	
10	Plate absorptance		
11	No. of IAM's in file		

TYPE 1d- Flat Plate Solar collector (Cover and absorber properties)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Tested flow rate	Total horizontal radiation	
6	Intercept efficiency	Horizontal diffuse radiation	
7	Efficiency slope	Ground reflectance	
8	Efficiency curvature	Incidence angle	
9	Optical mode 4	Collector slope	
10	Plate absorptance		
11	No. of identical covers		

	PARAMETERS	INPUT	OUTPUT
12	Index of refraction		
13	Extinction		

TYPE 1e- Flat Plate Solar collector (Biaxial incidence angle modifiers)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Tested flow rate	Total horizontal radiation	
6	Intercept efficiency	Horizontal diffuse radiation	
7	Efficiency slope	Ground reflectance	
8	Efficiency curvature	Incidence angle	
9	Optical mode 5	Collector slope	
10	Logical unit		
11	Number of values		

TYPE 2: ON/Off Differential

TYPE 2a-On/off Differential Controller for temperatures (New control strategy)

	PARAMETERS	INPUT	OUTPUT
1	New control mode	Upper input temperature Th	Output control function
2	High limit cut-out	Lower input temperature Tl	
3	High limit reset	Monitoring temperature Tin	
4		Input control function	
5		Upper dead band dT	
6		Lower dead band dT	

TYPE 2b- On/off Differential Controller for temperatures (Old control strategy)

	PARAMETERS	INPUT	OUTPUT
1	No. of oscillations	Upper input temperature Th	Output control function
2	High limit cut-out	Lower input temperature Tl	
3		Monitoring temperature Tin	
4		Input control function	
5		Upper dead band dT	
6		Lower dead band dT	

TYPE 2c- On/off Differential Controller-generic (New control strategy)

	PARAMETERS	INPUT	OUTPUT
1	New control mode	Upper input value	Output control function
2	High limit cut-out	Lower input value	
3	High limit reset	Monitoring value	
4		Input control function	
5		Upper dead band	
6		Lower dead band	

TYPE 2d- On/off Differential Controller-generic (Old control strategy)

	PARAMETERS	INPUT	OUTPUT
1	No. of oscillations	Upper input value	Output control function
2	High limit cut-out	Lower input value	
3		Monitoring value	
4		Input control function	

5		Upper dead band	
6		Lower dead band	

TYPE 4: Stratified Fluid Storage Tank

TYPE 4a/b- Stratified Fluid Storage Tank Fixed inlets (a- Uniform losses b-Non-uniform losses)

	PARAMETERS	INPUT	OUTPUT
1	Fixed inlet positions	Hot-side temperature	Temperature to heat source
2	Tank volume	Hot-side flowrate	Flowrate to heat source
3	Fluid specific heat	Cold-side temperature	Temperature to load
4	Fluid density	Cold-side flowrate	Flowrate to load
5	Tank loss coefficient	Environment temperature	Thermal losses
6	Height of node	Control signal for element-1	Energy rate to load
7	Auxiliary heater mode	Control signal for element-2	Internal energy change
8	Node containing heating element 1		Auxiliary heating rate
9	Node containing thermostat 1		Element 1 power
10	Set point temperature for element 1		Element 2 power
11	Deadband for heating element 1		Energy rate from heat source
12	Maximum heating rate of element 1		Average tank temperature
13	Node containing heating element 2		Temperature of node 1+
14	Node containing thermostat 2		
15	Set point temperature for element 2		
16	Deadband for heating element 2		
17	Maximum heating rate of element 2		
18	Not used (Flue UA)		
19	Not used (Tflue)		
20	Boiling point		
21	Incremental loss coefficient for node		

TYPE 4c/d - Stratified Fluid Storage Tank Variable inlets (c- Uniform losses d-Non-uniform losses)

	PARAMETERS	INPUT	OUTPUT
1	Variable inlet positions	Hot-side temperature	Temperature to heat source
2	Tank volume	Hot-side flowrate	Flowrate to heat source
3	Fluid specific heat	Cold-side temperature	Temperature to load
4	Fluid density	Cold-side flowrate	Flowrate to load
5	Tank loss coefficient	Environment temperature	Thermal losses
6	Height of node	Control signal for element-1	Energy rate to load
7	Auxiliary heater mode	Control signal for element-2	Internal energy change
8	Node containing heating element 1		Auxiliary heating rate
9	Node containing thermostat 1		Element 1 power
10	Set point temperature for element 1		Element 2 power

	PARAMETERS	INPUT	OUTPUT
11	Deadband for heating element 1		Energy rate from heat source
12	Maximum heating rate of element 1		Average tank temperature
13	Node containing heating element 2		Temperature of node 1+
14	Node containing thermostat 2		
15	Set point temperature for element 2		
16	Deadband for heating element 2		
17	Maximum heating rate of element 2		
18	Not used (Flue UA)		
19	Not used (Tflue)		
20	Boiling point		
21	Incremental loss coefficient for node		

TYPE 4e/f- Stratified Fluid Storage User-designated inlets (e- Uniform losses f-Non-uniform losses)

	PARAMETERS	INPUT	OUTPUT
1	User-specified inlet positions	Hot-side temperature	Temperature to heat source
2	Tank volume	Hot-side flowrate	Flowrate to heat source
3	Fluid specific heat	Cold-side temperature	Temperature to load
4	Fluid density	Cold-side flowrate	Flowrate to load
5	Tank loss coefficient	Environment temperature	Thermal losses
6	Height of node	Control signal for element-1	Energy rate to load
7	Auxiliary heater mode	Control signal for element-2	Internal energy change
8	Node containing heating element 1		Auxiliary heating rate
9	Node containing thermostat 1		Element 1 power
10	Set point temperature for element 1		Element 2 power
11	Deadband for heating element 1		Energy rate from heat source
12	Maximum heating rate of element 1		Average tank temperature
13	Node containing heating element 2		Temperature of node 1+
14	Node containing thermostat 2		
15	Set point temperature for element 2		
16	Deadband for heating element 2		
17	Maximum heating rate of element 2		
18	Not used (Flue UA)		
19	Not used (Tflue)		
20	Boiling point		
21	Entering node for hot-source flow		
22	Entering node for cold-side		

	PARAMETERS	INPUT	OUTPUT
	fluid		
23	Incremental loss coefficient for node		

TYPE 10: Rock Bed Thermal Storage

	PARAMETERS	INPUT	OUTPUT
1	Specific heat of air	Temperature into top	Bottom node temperature
2	Length of rock bed	Flowrate into top	Flowrate out of bottom
3	Cross-sectional area	Temperature into bottom	Top node temperature
4	Perimeter	Flowrate into bottom	Flowrate out of top
5	Specific heat of rock	Environment temperature	Internal energy change
6	Apparent rock bed density		Rate of energy supply
7	Loss coefficient		Environment losses
8	Effective thermal conductivity		Average temperature

TYPE 38: Algebraic Tank (Plug-Flow)

	PARAMETERS	INPUT	OUTPUT
1	Inlet position mode	Hot-side temperature	Temperature to heat source
2	Tank volume	Hot-side flowrate	Flowrate to heat source
3	Tank height	Cold-side temperature	Temperature to load
4	Height of collector return	Cold-side flowrate	Flowrate to load
5	Fluid specific heat	Environment temperature	Thermal losses
6	Fluid density	Control signal	Energy rate to load
7	Thermal conductivity		Internal energy change
8	Tank configuration		Auxiliary heating rate
9	Overall Loss Coefficient		Energy rate from heat source
10	Insulation ratio		Average tank temperature
11	Initial temperature		
12	Maximum heating rate		
13	Auxiliary height		
14	Thermostat height		
15	Set point temperature		
16	Temperature deadband		
17	Flue loss coefficient		

TYPE 39: Variable Volume Tank

	PARAMETERS	INPUT	OUTPUT
1	Tank operation mode	Inlet temperature	Fluid temperature
2	Overall tank volume	Inlet flow rate	Load flow rate
3	Minimum fluid volume	Flow rate to load	Excess flow temperature
4	Maximum fluid volume	Environment temperature	Excess flow rate
5	Tank circumference		Fluid volume
6	Cross-sectional area		Enthalpy difference
7	Wetted loss coefficient		Environment losses
8	Dry loss coefficient		Internal energy change
9	Fluid specific heat		Level indicator
10	Fluid density		
11	Initial fluid temperature		
12	Initial fluid volume		

TYPE 40-Microprocessor Controller

	PARAMETERS	INPUT	OUTPUT
1	Number of comparators (up to 5)	Comparator high input i	Control function output i
2	Temperature difference on deadband, comp.i	Comparator low input i	
3	Temperature difference off deadband, comp.i		

TYPE 45:Thermosiphon Collector with Integral Storage

	PARAMETERS	INPUT	OUTPUT
1	Collector area	Total incident radiation	Temperature to tank
2	Intercept efficiency	Total horizontal radiation	Useful energy from collector
3	Efficiency slope	Horizontal diffuse radiation	Temperature to collector
4	Tested flow rate	Incidence angle	Flowrate to collector
5	Incidence angle modifier constant	Ground reflectance	Temperature to load
6	Collector slope	Ambient temperature	Flowrate to load
7	Logical unit	Replacement temperature	Thermal losses
8	Number of data points	Load flowrate	Energy rate to load
9	Riser diameter	Environment temperature	Internal energy change
10	Header diameter	Control signal	Auxiliary heating rate
11	Header length		Energy rate from heat source
12	Number of collector nodes		Average tank temperature
13	Collector inlet to outlet distance		
14	Collector inlet to tank outlet distance		
15	Collector inlet diameter		
16	Length of collector inlet		
17	Number of inlet bends		
18	Inlet pipe loss coefficient		
19	Collector outlet diameter		
20	Length of collector outlet		
21	Number of outlet bends		
22	Outlet pipe loss coefficient		
23	Inlet position mode		
24	Tank volume		
25	Tank height		
26	Height of collector return		
27	Fluid specific heat		
28	Fluid density		
29	Thermal conductivity		
30	Tank configuration		
31	Overall Loss Coefficient		
32	Insulation ratio		
33	Initial temperature		
34	Maximum heating rate		
35	Auxiliary height		
36	Thermostat height		
37	Set point temperature		
38	Temperature deadband		
39	Flue loss coefficient		

Type 47: Battery

TYPE 47a- Electrical Storage battery Power as an input (dQ/dt=Peff)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Power to or from battery	State of charge
2	Cell Energy Capacity		Fractional state of charge
3	Cells in parallel		Power
4	Cells in series		Power lost during charge
5	Charging efficiency		

TYPE 47b- Electrical Storage battery Power as an input (Shepherd equation)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Power to or from battery	State of charge
2	Cell Energy Capacity		Fractional state of charge
3	Cells in parallel		Power
4	Cells in series		Power lost during charge
5	Charging efficiency		Total current
6	Max. current per cell charging		Total voltage
7	Max. current per cell discharge		Max. Power for charge
8	Max. charge voltage per cell		Max. Power for discharge
9	Calculate discharge cutoff voltage		Discharge cutoff voltage (DCV)
10			Power corresponding to DCV
11			Charge cutoff voltage (CCV)
12			Power corresponding to CCV

TYPE 47c- Electrical Storage battery Power as an input (Shepherd modified Hyman equation)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Power to or from battery	State of charge
2	Cell Energy Capacity		Fractional state of charge
3	Cells in parallel		Power
4	Cells in series		Power lost during charge
5	Charging efficiency		Total current
6	Max. current per cell charging		Total voltage
7	Max. current per cell discharge		Max. Power for charge
8	Max. charge voltage per cell		Max. Power for discharge
9	Calculate discharge cutoff voltage		Discharge cutoff voltage (DCV)
10	Tolerance for charging current calculation		Power corresponding to DCV
11			Charge cutoff voltage (CCV)
12			Power corresponding to CCV
13			
14			

TYPE 47d- Electrical Storage battery Current as an Input (Shepherd equation)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Current	State of charge

2	Cell Energy Capacity		Fractional state of charge
3	Cells in parallel		Power
4	Cells in series		Power lost during charge
5	Charging efficiency		Total current
6	Max. current per cell charging		Total voltage
7	Max. current per cell discharge		Max. Power for charge
8	Max. charge voltage per cell		Max. Power for discharge
9	Calculate discharge cutoff voltage		Discharge cutoff voltage (DCV)
10			Power corresponding to DCV
11			Charge cutoff voltage (CCV)
12			Power corresponding to CCV

TYPE 47e- Electrical Storage battery Current as an Input (Shepherd modified Hyman equation)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Current	State of charge
2	Cell Energy Capacity		Fractional state of charge
3	Cells in parallel		Power
4	Cells in series		Power lost during charge
5	Charging efficiency		Total current
6	Max. current per cell charging		Total voltage
7	Max. current per cell discharge		Max. Power for charge
8	Max. charge voltage per cell		Max. Power for discharge
9			Discharge cutoff voltage (DCV)
10			Power corresponding to DCV
11			Charge cutoff voltage (CCV)
12			Power corresponding to CCV

TYPE 48: Regulator/Inverter

Type 48a- Regulator/Inverter (System without battery storage)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Input power	Power in
2	Efficiency	Load power	Power out
3			Excess power

Type 48b- Regulator/Inverter (System with battery storage-MPP Tracking-SOC monitoring only)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Input power	Power in from generation
2	Regulator efficiency	Load power	Power to or from battery
3	Inverter efficiency	Battery fractional state of charge	Power to load
4	High limit on fractional state of charge (FSOC)		Dumped generated power
5	Low limit on FSOC		Power from grid
6	charge to discharge limit on FSOC		

7	Inverter output power capacity		

Type 48c- Regulator/Inverter (System with battery storage-MPP Tracking-SOC and SOV monitoring)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Input power	Power in from generation
2	Regulator efficiency	Load power	Power to or from battery
3	Inverter efficiency (DC to AC)	Battery fractional state of charge	Power to load
4	High limit on fractional state of charge (FSOC)	Battery voltage (BV)	Dumped generated power
5	Low limit on FSOC	Max battery input	Power from grid
6	charge to discharge limit on FSOC	Min. battery output	
7	Power output limit	lower limit on battery voltage	
8	Inverter efficiency (AC to DC)	Power corresponding to BV	
9	Current for grid charging of battery	High limit on BV	
10	Upper limit on FSOC for grid charging	Power corresponding to high limit on BV	
11		Start time for grid battery charging	
12		Stop time for grid battery charging	

Type 48d- Regulator/Inverter (System with battery storage-Array V=Battery V)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Input power	Power in from generation
2	Regulator efficiency	Load power	Power to or from battery
3	Inverter efficiency (DC to AC)	Battery fractional state of charge	Power to load
4	High limit on fractional state of charge (FSOC)	Battery voltage (BV)	Dumped generated power
5	Low limit on FSOC	Max battery input	Power from grid
6	charge to discharge limit on FSOC	Min. battery output	Current in from generator
7	Power output limit	lower limit on battery voltage	Current to or from battery
8	Inverter efficiency (AC to DC)	Power corresponding to BV	Current to load
9	Current for grid charging of battery	High limit on BV	Dumped generated current
10	Upper limit on FSOC for grid charging	Power corresponding to high limit on BV	Current from grid
11		Start time for grid battery charging	
12		Stop time for grid battery charging	

Type 50: Flat Plate PV/T Collector

TYPE 50a-Flat Plate Collector (Constant losses)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate

3	Collector Efficiency Factor	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Collector plate absorptance	Cell efficiency	transmittance-absorptance product
6	collector loss coefficient		Electrical power output
7	Cover transmittance		Average cell temperature
8	Temperature coefficient of solar cell efficiency		Apparent thermal loss coefficient
9	Reference temperature for cell efficiency		
10	Packing factor		

TYPE 50b- Flat Plate Collector (Losses=f(temperature, wind, geometry))

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Collector Efficiency Factor	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coeficient
5	Collector plate absorptance	Windspeed	transmitance-absorptance product
6	Number of glass covers	Cell Efficiency at reference conditions	Electrical power output
7	Collector plate emmitance		Average cell temperature
8	Loss coefficient for bottom and edge losses		Apparent thermal loss coefficient
9	Collector slope		
10	Transmittance absorbtance product		
11	Temperature coefficient of PV cell efficiency		
12	Temperature for cell reference efficiency		
13	Packing factor		

TYPE 50c- Flat Plate Collector (Angular dependence of transmittance)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Collector Efficiency Factor	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident beam radiation	Collector loss coefficient
5	Collector plate absorptance	Incident diffuse radiation	transmitance-absorptance product
6	Number of glass covers	Incidence angle of beam radiation	Electrical power output
7	Collector loss coefficient	Cell efficiency	Average cell temperature
8	Extinction coefficient thickness product		Apparent thermal loss coefficient
9	Temperature coefficient of PV cell efficiency		
10	Temperature for cell reference efficiency		
11	Packing factor		

TYPE 50d- Flat Plate Collector (Losses=f(temperature, wind, geometry) and t=f(angle))

	PARAMETERS	INPUT	OUTPUT

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Collector Efficiency Factor	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Collector plate absorptance	Incident diffuse radiation	transmitance-absorptance product
6	Number of glass covers	Incidence angle of beam radiation	Electrical power output
7	Collector plate emmitance	Windspeed	Average cell temperature
8	Loss coefficient for bottom and edge losses	Cell Efficiency at reference conditions	Apparent thermal loss coefficient
9	Collector slope		
10	Extinction coefficient thickness product		
11	Temperature coefficient of PV cell efficiency		
12	Temperature for cell reference efficiency		
13	Packing factor		

TYPE 50e- Concentrating collectors (Constant Losses- No cell operating voltage)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Ratio of aperture to absorber area	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Plate absorptance		transmitance-absorptance product
6	fin efficiency area ratio		Electrical power output
7	Back loss coefficient for no-flow condition		Average cell temperature
8	Thermal conductance between cells and absorber		Apparent thermal loss coefficient
9	Heat transfer coefficient		Array voltage
10	Cover plate transmittance		Array current
11	Front loss coefficient for cells		Cell temperature at collector inlet
12	Logical unit for SOLCEL data file		

TYPE 50f- Concentrating collectors (Top Loss-f(wind, T) No cell operating voltage)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Ratio of aperture to absorber area	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Plate absorptance	Wind speed	transmitance-absorptance product
6	fin efficiency area ratio	Collector slope	Electrical power output
7	Back loss coefficient for no-flow condition		Average cell temperature
8	Thermal conductance between cells and absorber		Apparent thermal loss coefficient

9	Heat transfer coefficient		Array voltage
10	Cover plate transmittance		Array current
11	Number of glass covers		Cell temperature at collector inlet
12	Absorber plate emittance		
13	Logical unit for SOLCEL data		

TYPE 50g- Concentrating collectors (Constant Losses- Cell operating V is input)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Ratio of aperture to absorber area	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Plate absorptance	Voltage applied to array	transmitance-absorptance product
6	fin efficiency area ratio		Electrical power output
7	Back loss coefficient for no-flow condition		Average cell temperature
8	Thermal conductance between cells and absorber		Apparent thermal loss coefficient
9	Heat transfer coefficient		Array voltage
10	Cover plate transmittance		Array current
11	Front loss coefficient for cells		Cell temperature at collector inlet
12	Logical unit for SOLCEL data file		

TYPE 50h- Concentrating collectors (Top Loss-f(wind, T) Cell operating voltage is input)

	PARAMETERS	INPUT	OUTPUT
1	Mode	Inlet fluid temperature	Outlet fluid temperature
2	Collector Area	Fluid mass flow rate	Fluid flowrate
3	Ratio of aperture to absorber area	Ambient temperature	Rate of useful energy gain
4	Fluid thermal capacitance	Incident radiation	Collector loss coefficient
5	Plate absorptance	Wind speed	transmitance-absorptance product
6	fin efficiency area ratio	Collector slope	Electrical power output
7	Back loss coefficient for no-flow condition	Voltage applied to array	Average cell temperature
8	Thermal conductance between cells and absorber		Apparent thermal loss coefficient
9	Heat transfer coefficient		Array voltage
10	Cover plate transmittance		Array current
11	Number of glass covers		Cell temperature at collector inlet
12	Absorber plate emittance		
13	Logical unit for SOLCEL data		

Type 60: Detailed Fluid Storage with Heaters

	PARAMETERS	INPUT	OUTPUT
1	User-specified inlet positions	Flow rate at inlet 1	Flowrate at inlet 1

	PARAMETERS	INPUT	OUTPUT
2	Tank volume	Flow rate at outlet 1	Flowrate at outlet 1
3	Tank height	Used or not (flow inlet 2)	Used or not (inlet 2 flow)
4	Horizontal cylinder or Tank Perimeter	Used or not (flow outlet 2)	Used or not (outlet 2 flow)
5	Height of flow inlet 1	Temperature at inlet 1	Temperature of outlet flow 1
6	Height of flow outlet 1	Used or not (temp inlet 2)	Used or not (temp flow 2)
7	Not used (inlet 2)	Environment temperature	Thermal losses
8	Not used (outlet 2)	Control signal for element 1	Energy supplied by inlet 1
9	Fluid specific heat	Control signal for element 2	Energy removed by outlet 1
10	Fluid density	Flow rate for heat exchanger	Used or not (energy inlet 2)
11	Tank loss coefficient	Inlet temperature for heat exchanger	Used or not (energy outlet 2)
12	Fluid thermal conductivity	Nusselt constant for heat exchanger	Auxiliary heating rate
13	Destratification conductivity	Nusselt exponent for heat exchanger	Element 1 power
14	Boiling temperature		Element 2 power
15	Auxiliary heater mode		Losses to gas flue
16	Height of 1st aux. heater		Internal energy change
17	Height of 1st thermostat		Average tank temperature
18	Set point temperature for element 1		Static pressure difference - inlet 1
19	Deadband for heating element 1		Static pressure difference - outlet 1
20	Maximum heating rate of element 1		Used or not (pressure inlet 2)
21	Height of heating element 2		Used or not (pressure outlet 2)
22	Height of thermostat 2		Energy input from heat exchanger
23	Set point temperature for element 2		Temperature of fluid exiting heat exchanger
24	Deadband for heating element 2		Tank temperature at outlet of heat exchanger
25	Maximum heating rate of element 2		LMTD of heat exchanger
26	Overall loss coefficient for gas flue		UA of heat exchanger
27	Flue temperature		Tank temperature - top
28	Fraction of critical timestep		Tank temperature - bottom
29	Gas heater?		Temperature of node 1+
30	Number of internal heat exchangers		
31	Node heights supplied		
32	Additional loss coeff's supplied		
33	HX Fluid Indicator		
34	Fraction of glycol		
35	Heat exchanger inside diameter		
36	Heat exchanger outside diameter		
37	Heat exchanger fin diameter		
38	Total surface area of heat exchanger		
39	Fins per meter for heat exchanger		
40	Heat exchanger length		

	PARAMETERS	INPUT	OUTPUT
41	Heat exchanger wall conductivity		
42	Heat exchanger material conductivity		
43	Height of heat exchanger inlet		
44	Height of heat exchanger outlet		
45	Height of node		
46	Additional loss coefficient for node		

TYPE 71: Evacuated Tube Solar Collector

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Flow rate at test conditions	Incident diffuse radiation	
6	Intercept efficiency	Solar incidence angle	
7	Negative of first order efficiency coefficient	Solar zenith angle	
8	Negative of second order efficiency coefficient	Solar azimuth angle	
9	Logical unit of file containing biaxial IAM data	Collector slope	
10	Number of longitudinal angles for which IAMs are provided	Collector azimuth	
11	Number of transverse angles for which IAMs are provided		

TYPE 71-Unglazed transpired collector system (Flat plate, no convective losses at the top of the collector, no PV)[1]

	PARAMETERS	INPUT	OUTPUT
1	Collector area	Month of the year	Plenum air temperature
2	Collector height	Hour of month	Collector outlet air temperature
3	Collector hole diameter	Radiation incident on the collector	Mixed air temperature
4	Collector hole pitch, distance between centers of holes	Ambient temperature	Supply air temperature
5	Collector emissivity	Sky temperature	Collector surface temperature
6	Collector absorptivity	Atmospheric pressure	Energy savings rate
7	Plenum depth	Internal gains due to people, equipment, etc.	Convection from collector
8	Emissivity of the wall behind the collector	Supply air flow rate from collector air-handling units	Convection from wall
9	R-value of the wall behind the collector	Minimum outdoor air flow rate through collector/summer bypass damper	Radiation from collector
10	Total UA-value of the	Supply air flow rate from no-	Radiation from wall

	PARAMETERS	INPUT	OUTPUT
	building walls and roof	collector air-handling units	
11	Room air temperature	Outdoor air flow rate through no collector	Conduction through wall
12	Ambient air temperature above which the summer bypass damper is opened		Reduced conduction through wall because of collector
13	Maximum auxiliary heat rate available		Absorbed energy rate
14	Night bypass		Auxiliary heating rate
15			Outdoor air flow rate through collector/summer bypass damper
16			Heat exchanger effectiveness of collector
17			Solar efficiency of the collector
18			Pressure drop across collector plate
19			Bypass damper position = 0 if open =1 if closed
20			Heat rate supplied by a traditional heating system
21			Additional fan power required

Type 72: Performance Map Solar Collector

TYPE 72a-Performance Map Solar Collector (No incidence angle modification)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Logical Unit	Windspeed	
6	Number of DT/IT points		
7	Number of radiation curves		
8	Number of wind speed curves		
9	Optical Mode 1		

TYPE 72b- Performance Map Solar Collector (2nd order incidence angle modifiers)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Logical Unit	Windspeed	
6	Number of DT/IT points	Total horizontal radiation	
7	Number of radiation curves	Horizontal diffuse radiation	
8	Number of wind speed curves	Ground reflectance	
9	Optical mode 2	Incidence angle	
10	1st-order IAM	Collector slope	
11	2nd-order IAM		

TYPE 72c- Performance Map Solar Collector (Modifiers=f(incidence angle))

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Logical Unit	Windspeed	
6	Number of DT/IT points	Total horizontal radiation	
7	Number of radiation curves	Horizontal diffuse radiation	
8	Number of wind speed curves	Ground reflectance	
9	Optical mode 3	Incidence angle	
10	Logical unit	Collector slope	
11	No. of IAM's in file		

TYPE 72d- Performance Map Solar Collector (Cover and absorber properties)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Logical Unit	Windspeed	
6	Number of DT/IT points	Total horizontal radiation	
7	Number of radiation curves	Horizontal diffuse radiation	
8	Number of wind speed curves	Ground reflectance	
9	Optical mode 4	Incidence angle	
10	Plate absorptance	Collector slope	
11	No. of identical covers		
12	Index of refraction		
13	Extinction		

TYPE 72e- Performance Map Solar Collector (Biaxial incidence angle modifiers)

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Efficiency mode	Incident radiation	
5	Logical Unit	Windspeed	
6	Number of DT/IT points	Total horizontal radiation	
7	Number of radiation curves	Horizontal diffuse radiation	
8	Number of wind speed curves	Ground reflectance	
9	Optical mode 5	Incidence angle	
10	Logical unit	Collector slope	
11	Number of values		

TYPE 73:Theoretical Flat Plate Solar Collector

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Collector fin efficiency factor	Incident radiation	
5	Bottom, edge loss coefficient	Windspeed	

	PARAMETERS	INPUT	OUTPUT
6	Absorber plate emittance	Horizontal radiation	
7	Absorptance of absorber plate	Horizontal diffuse	
8	Number of covers	Ground reflectance	
9	Index of refraction of cover	Incidence angle	
10	Extinction coeff. thickness product	Collector slope	

TYPE 74:Compound Parabolic Concentrating Collector

	PARAMETERS	INPUT	OUTPUT
1	Number in series	Inlet temperature	Outlet temperature
2	Collector area	Inlet flowrate	Outlet flowrate
3	Fluid specific heat	Ambient temperature	Useful energy gain
4	Collector fin efficiency factor	Incident radiation	
5	Overall Loss Coefficient	Horizontal radiation	
6	Wall reflectivity	Horizontal diffuse	
7	Half-acceptance angle	Ground reflectance	
8	Truncation ratio	Incidence angle	
9	Axis orientation	Zenith angle	
10	Absorptance of absorber plate	Solar azimuth angle	
11	Number of covers	Collector slope	
12	Index of refraction of cover	Collector azimuth angle	
13	Extinction coeff. thickness product		

TYPE 94: Photovoltaic Panel

TYPE 94a- Photovoltaic Panel (Crystalline module)

	PARAMETERS	INPUT	OUTPUT
1	Module short-circuit current at reference conditions	Total incident radiation	Array voltage
2	Module open-circuit voltage at reference conditions	Ambient temperature	Array current
3	Reference temperature	Load voltage	Array power
4	Reference insolation	Flag for convergence promotion	Power at maximum power point
5	Module voltage at max power point and reference conditions	Array slope	Fraction of maximum power used
6	Module current at max power point and reference conditions	Beam radiation	Voltage at MPP
7	Temperature coefficient of Isc at (ref. cond)	Diffuse radiation	Current at MPP
8	Temperature coefficient of Voc (ref. cond.)	Incidence angle of beam radiation	Open circuit voltage
9	Number of cells wired in series		Short circuit current
10	Number of modules in series		Array fill factor
11	Number of modules in parallel		Array temperature
12	Module temperature at		

	PARAMETERS	INPUT	OUTPUT
	NOCT		
13	Ambient temperature at NOCT		
14	Insolation at NOCT		
15	Module area		
16	tau-alpha product for normal incidence		
17	Semiconductor bandgap		
18	Slope of IV curve at Isc		
19	Module series resistance		
20			

TYPE 94b- Photovoltaic Panel (Thin film module)

	PARAMETERS	INPUT	OUTPUT
1	Module short-circuit current at reference conditions	Total incident radiation	Array voltage
2	Module open-circuit voltage at reference conditions	Ambient temperature	Array current
3	Reference temperature	Load voltage	Array power
4	Reference insolation	Flag for convergence promotion	Power at maximum power point
5	Module voltage at max power point and reference conditions	Array slope	Fraction of maximum power used
6	Module current at max power point and reference conditions	Beam radiation	Voltage at MPP
7	Temperature coefficient of Isc at (ref. cond))	Diffuse radiation	Current at MPP
8	Temperature coefficient of Voc (ref. cond.)	Incidence angle of beam radiation	Open circuit voltage
9	Number of cells wired in series		Short circuit current
10	Number of modules in series		Array fill factor
11	Number of modules in parallel		Array temperature
12	Module temperature at NOCT		
13	Ambient temperature at NOCT		
14	Insolation at NOCT		
15	Module area		
16	tau-alpha product for normal incidence		
17	Semiconductor bandgap		
18	Slope of IV curve at Isc		
19	Module series resistance		

TYPE 186: Serpentine Collector

	PARAMETERS	INPUT	OUTPUT
1	number of turns	temperature of fluid entering collector	temperature of fluid exiting collector
2	inner tube diameter	mass flowrate of fluid entering collector	mass flowrate of fluid exiting collector

	PARAMETERS	INPUT	OUTPUT
3	outer tube diameter	ambient temperature of collector surroundings	rate of energy gain from collector
4	plate thickness	Radiation incident on collector surface	mean absorber plate temperature
5	length of each turn	wind speed	collector heat removal factor
6	tube spacing	total radiation on horizontal surface	transmittance-absorbtance product
7	plate conductivity	diffuse radiation on horizontal surface	P_loss pressure loss
8	loss coefficient from back and edge of collector per unit aperture area	ground reflectance	overall heat loss coefficient
9	absorber plate emmitance	incidence angle of beam radiation (	
10	absorber plate absorptance	collector slope	
11	number of glass covers	dynamic viscosity	
12	refractive index of glass covers	specific heat of collector	
13	product of extinction coefficient and thickness of cover plates	fluid thermal conductivity	
14		fluid density	

TYPE 555- Unglazed air PV/T flat plate collector

	PARAMETERS	INPUT	OUTPUT
1	Total collector area	Inlet temperature	Outlet fluid temperature
2	Fluid specific heat	Inlet flow rate	Flow rate at outlet
3	Reflectance	Ambient temperature	Useful energy gain
4	Emissivity	Sky temperature	PV power
5	1^{st} order IAM	Back-surface environment temperature	PV efficiency
6	PV cell reference temperature	Incident solar radiation	Thermal efficiency
7	PV cell reference radiation	Total horizontal radiation	Collector F_R
8	PV efficiency at reference condition	Horizontal diffuse radiation	Plate (PV) temperature
9	Efficiency modifier-temperature	Ground reflectance	Mean fluid temperature
10	Efficiency modifier-radiation	Incidence angle	Overall IAM
11	Resistance of substrate material	Collector slope	Collector top losses-convective
12	Resistance of back material	Top loss convection coefficient	Collector top losses-radiative
13		Back loss coefficient	Back losses
14		Fluid heat transfer coefficient	Fluid heat transfer
15			Absorbed solar radiation
16			Collector F'
17			Collector U_L
18			$F_R\tau\alpha_N$
19			F_RU_L

TYPE 56(Mode) – PVT Systems

Type 560 - Unglazed PV/T collector with fluid stream passing through tubes bonded to an absorber plate located beneath the PV cells

	PARAMETERS	INPUT	OUTPUT
1	Length	Inlet temperature	Outlet fluid temperature
2	Width	Inlet flow rate	Flow rate at outlet
3	Absorber thickness	Ambient temperature	Useful energy gain
4	Thermal conductivity of the absorber	Sky temperature	PV power
5	Number of tubes	Back-surface environment temperature	PV efficiency
6	Tube diameter	Incident solar radiation	Thermal efficiency
7	Bond width	Total horizontal radiation	Collector F_R
8	Bond thickness	Horizontal diffuse radiation	Plate (PV) temperature
9	Bond thermal conductivity	Ground reflectance	Mean fluid temperature
10	Resistance of substrate material	Incidence angle	Overall IAM
11	Resistance of back material	Collector slope	Collector top losses-convective
12	Fluid specific heat	Top loss convection coefficient	Collector top losses-radiative
13	Reflectance	Back loss coefficient	Back losses
14	Emissivity	Fluid heat transfer coefficient	Absorbed solar radiation
15	1st order IAM		Collector U_L
16	PV cell reference		$F_R\tau\alpha_N$
17	PV cell reference radiation		F_RU_L
18	PV efficiency at reference condition		
19	Efficiency modifier-temperature		
20	Efficiency modifier-radiation		

TYPE 563- Unglazed PV/T collector with fluid stream passing through tubes bonded to an absorber plate located beneath the PV cells (considering conduction between the back of the collector and the roof)

	PARAMETERS	INPUT	OUTPUT
1	Length	Inlet temperature	Outlet fluid temperature
2	Width	Inlet flow rate	Flow rate at outlet
3	Absorber thickness	Ambient temperature	Useful energy gain
4	Thermal conductivity of the absorber	Sky temperature	PV power
5	Number of tubes	Back-surface temperature	PV efficiency
6	Tube diameter	Incident solar radiation	Thermal efficiency
7	Bond width	Total horizontal radiation	Collector F_R
8	Bond thickness	Horizontal diffuse radiation	Mean PV temperature
9	Bond thermal conductivity	Ground reflectance	Mean fluid temperature
10	Resistance of substrate material	Incidence angle	Overall IAM
11	Resistance of back material	Collector slope	Collector top losses-convective
12	U-value of roof material	Top loss convection coefficient	Collector top losses-radiative
13	Fluid specific heat	Fluid heat transfer coefficient	Back losses
14	Reflectance		Absorbed solar radiation
15	Emissivity		Collector U_L
16	1st order IAM		$F_R\tau\alpha_N$
17	PV cell reference		F_RU_L

	PARAMETERS	INPUT	OUTPUT
18	PV cell reference radiation		Q_{BASE}
19	PV efficiency at reference condition		Q_{FIN}
20	Efficiency modifier-temperature		T_{back}
21	Efficiency modifier-radiation		

TYPE 566-Building integrated photovoltaic system (glazed, air)

	PARAMETERS	INPUT	OUTPUT
1	Collector length	Inlet air temperature	Outlet air temperature
2	Collector width	Inlet air flow rate	Outlet air flow rate
3	Collector emissivity	Ambient temperature	Useful energy gain
4	Thermal conductivity of cover material	Sky temperature	Thermal efficiency
5	Thickness of cover	Back-surface environment temperature	Electrical power
6	Resistance of substrate material	Back-surface radiant temperature	Electrical efficiency
7	Emissivity – top surface of flow channel	Incident solar radiation	Cover temperature
8	Emissivity –bottom surface of flow channel	Total horizontal radiation	PV cell temperature
9	Resistance of back material	Horizontal diffuse radiation	Upper air channel surface temperature
10	Emissivity – back surface	Ground reflectance	Mean fluid temperature
11	Channel height	Incidence angle	Lower air channel surface temperature
12	IAM mode	Collector slope	Back surface temperature
13	PV mode	Top loss convection coefficient	Overall IAM
14		Back loss convection coefficient	Top losses – convective
15		Atmospheric pressure	Top losses – radiative
16			Back losses - convective
17			Back losses - radiative
18			Absorbed solar radiation
If IAM Mode=1 (Nb of parameters=15)			
14	Transmittance-absorptance product at normal incidence		
15	1^{st} order IAM		
If IAM Mode=2 (Nb of parameters=16)			
14	Absorptance of PV surface		
15	Cover index of refraction		
16	Extinction coefficient		
If PV Mode =1			
NPAR+1	PV efficiency at reference condition		
NPAR+2	PV cell reference temperature		
NPAR+3	PV cell reference radiation		
NPAR+4	Efficiency modifier-temperature		
NPAR+5	Efficiency modifier-radiation		
If PV Mode=2			
NPAR+1	Logical unit for data file		
NPAR+2	Number of temperature		

	PARAMETERS	INPUT	OUTPUT
	points		
NPAR+3	Number of radiation points		
If PV Mode=3			
16		PV efficiency	

TYPE 567-Building integrated photovoltaic system (glazed, air) (No convective and radiative losses at the back of the collector)

	PARAMETERS	INPUT	OUTPUT
1	Collector length	Inlet air temperature	Outlet air temperature
2	Collector width	Inlet air flow rate	Outlet air flow rate
3	Collector emissivity	Ambient temperature	Useful energy gain
4	Thermal conductivity of cover material	Sky temperature	Thermal efficiency
5	Thickness of cover	Back-surface temperature	Electrical power
6	Resistance of substrate material	Incident solar radiation	Electrical efficiency
7	Emissivity – top surface of flow channel	Total horizontal radiation	Cover temperature
8	Emissivity –bottom surface of flow channel	Horizontal diffuse radiation	PV cell temperature
9	Resistance of back material	Ground reflectance	Upper air channel surface temperature
10	Channel height	Incidence angle	Mean fluid temperature
11	IAM mode	Collector slope	Lower air channel surface temperature
12	PV mode	Top loss convection coefficient	Back surface temperature
13		Atmospheric pressure	Overall IAM
14			Top losses – convective
15			Top losses – radiative
16			Back losses
17			Absorbed solar radiation
If IAM Mode=1 (Nb of parameters=14)			
13	Transmittance-absorptance product at normal incidence		
14	1st order IAM		
If IAM Mode=2 (Nb of parameters=15)			
13	Absorptance of PV surface		
14	Cover index of refraction		
15	Extinction coefficient		
If PV Mode =1			
NPAR+1	PV efficiency at reference condition		
NPAR+2	PV cell reference temperature		
NPAR+3	PV cell reference radiation		
NPAR+4	Efficiency modifier-temperature		
NPAR+5	Efficiency modifier-radiation		
If PV Mode=2			
NPAR+1	Logical unit for data file		
NPAR+2	Number of temperature points		
NPAR+3	Number of radiation points		
If PV Mode=3			
16		PV efficiency	

TYPE 568-Unglazed building integrated photovoltaic system (air) (No convective and radiative losses at the back of the collector)

	PARAMETERS	INPUT	OUTPUT
1	Collector length	Inlet air temperature	Outlet air temperature
2	Collector width	Inlet air flow rate	Outlet air flow rate
3	Absorptance of PV surface	Ambient temperature	Useful energy gain
4	Emissivity of PV surface	Sky temperature	Thermal efficiency
5	Resistance of substrate material	Back-surface temperature	Electrical power
6	Emissivity – top surface of flow channel	Incident solar radiation	Electrical efficiency
7	Emissivity –bottom surface of flow channel	Collector slope	PV cell temperature
8	Resistance of back material	Top loss convection coefficient	Upper air channel surface temperature
9	Channel height	Atmospheric pressure	Mean fluid temperature
10	PV mode		Lower air channel surface temperature
11			Back surface temperature
12			Top losses – convective
13			Top losses – radiative
14			Back losses
15			Absorbed solar radiation
If PV Mode=1			
11	PV efficiency at reference condition		
12	PV cell reference temperature		
13	PV cell reference radiation		
14	Efficiency modifier-temperature		
15	Efficiency modifier-radiation		
If PV Mode=2			
11	Logical unit for data file		
12	Number of temperature points		
13	Number of radiation points		
If PV Mode=3			
10		PV efficiency	

TYPE 569-Unglazed building integrated photovoltaic system (air) (Convective and radiative losses at the back of the collector)

	PARAMETERS	INPUT	OUTPUT
1	Collector length	Inlet air temperature	Outlet air temperature
2	Collector width	Inlet air flow rate	Outlet air flow rate
3	Absorptance of PV surface	Ambient temperature	Useful energy gain
4	Emissivity of PV surface	Sky temperature	Thermal efficiency
5	Resistance of substrate material	Back-surface environment temperature	Electrical power
6	Emissivity – top surface of flow channel	Back-surface radiant temperature	Electrical efficiency
7	Emissivity –bottom surface of flow channel	Incident solar radiation	PV cell temperature
8	Resistance of back material	Collector slope	Upper air channel surface temperature
9	Emissivity – back surface	Top loss convection	Mean fluid temperature

	PARAMETERS	INPUT	OUTPUT
		coefficient	
10	Channel height	Back loss convection coefficient	Lower air channel surface temperature
11	PV mode	Atmospheric pressure	Back surface temperature
12			Top losses – convective
13			Top losses – radiative
14			Back losses - convective
15			Back losses - radiative
16			Absorbed solar radiation
If PV Mode=1			
12	PV efficiency at reference condition		
13	PV cell reference temperature		
14	PV cell reference radiation		
15	Efficiency modifier-temperature		
16	Efficiency modifier-radiation		
If PV Mode=2			
12	Logical unit for data file		
13	Number of temperature points		
14	Number of radiation points		
If PV Mode=3			
10		PV efficiency	