Modelling a flat plate solar collector

Smit, WA

CSC 411

2017-05-02

Modelling a flat plate solar collector

Smit, WA

14052131

Department of Chemical Engineering

University of Pretoria

CSC 411

2017-05-02

Modelling a flat plate solar collector

# Synopsis

# Contents

[Synopsis 3](#_Toc481502395)

[1 Literature Survey 5](#_Toc481502396)

[1.1 Average Wind Speed and Direction 7](#_Toc481502397)

[1.2 Effective Radiation 7](#_Toc481502398)

[1.3 Materials and Construction of a Collector 8](#_Toc481502399)

[1.3.1 Glass Cover 8](#_Toc481502400)

[1.3.2 Absorber Plate 8](#_Toc481502401)

[1.3.3 Insolation 9](#_Toc481502402)

[1.3.4 Riser and Header Pipes 9](#_Toc481502403)

[1.4 Thermal Network of the Collector 9](#_Toc481502404)

[1.5 Radiation heat flow 11](#_Toc481502405)

[1.6 Convective heat transfer coefficients 11](#_Toc481502406)

[1.7 Efficiency of a Collector 13](#_Toc481502407)

[1.8 A few previous models 13](#_Toc481502408)

[2 Experimental and Model 15](#_Toc481502409)

[2.1 Assumptions in Model 15](#_Toc481502410)

[2.2 Setting up the model equations 15](#_Toc481502411)

[2.2.1 The Glass cover 15](#_Toc481502412)

[2.2.2 The Air duct 15](#_Toc481502413)

[2.2.3 The absorber plate 15](#_Toc481502414)

[2.2.4 The insolation 16](#_Toc481502415)

[2.2.5 The working fluid 16](#_Toc481502416)

[2.3 The simulation equations 16](#_Toc481502417)

[2.3.1 The differential equations 16](#_Toc481502418)

[2.3.2 The simulation constants 17](#_Toc481502419)

[2.4 Experiments 17](#_Toc481502420)

[3 Results 20](#_Toc481502421)

[3.1 Model performance 20](#_Toc481502422)

[3.1.1 The effect of number of nodes on the performance 20](#_Toc481502423)

[3.1.2 The effect of iteration steps on the performance 21](#_Toc481502424)

[3.2 Theoretical validation 22](#_Toc481502425)

[3.3 Experimental validation 22](#_Toc481502426)

[4 Discussion 24](#_Toc481502427)

[5 Conclusion and Recommendations 25](#_Toc481502428)

[References 26](#_Toc481502429)

[Appendix 27](#_Toc481502430)

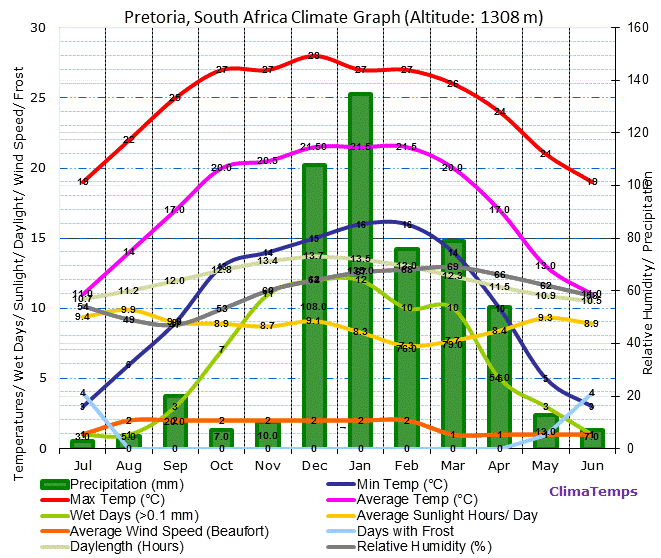
# Nomenclature

# Introduction

# Literature Survey

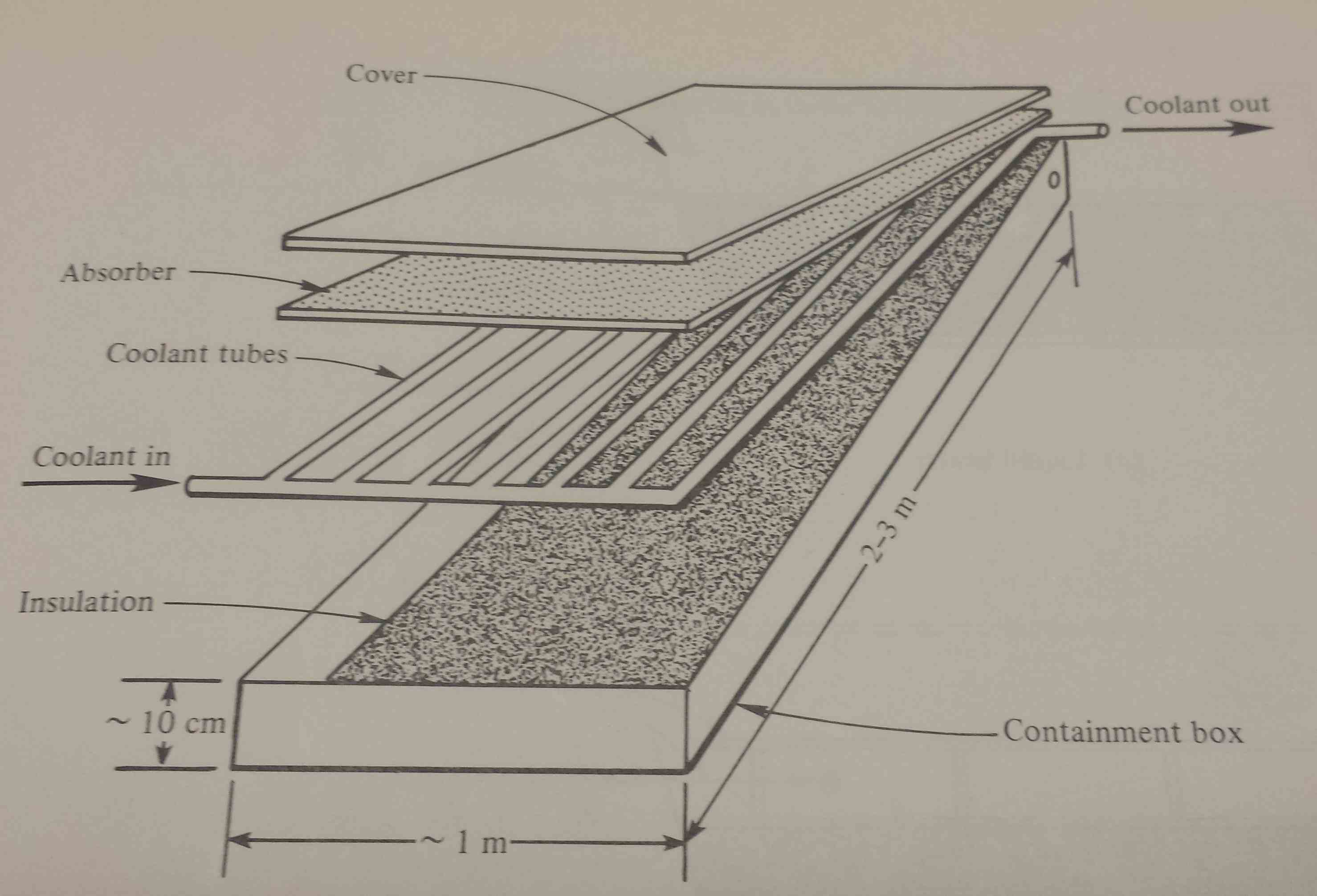
The demand of solar collectors has increased drastically in the past decade, as the world is striving to harvest cleaner and more sustainable forms of energy. The drive toward solar energy is approaching a climax at a very high rate, especially in regions of the world where there are a lot of sunny days. Public awareness campaigns are launched everywhere, and it seems that the world consumers are slowly starting to move away from electrical geysers. According to H. Aktamis (2011) the level of awareness of renewable energy sources is very among the general public, especially among second grade students in the developed countries of the world.

Solar power is readily useable in South Africa, with applications including: photovoltaic (PV) cells; domestic and swimming pool heating for the middle and upper class; industry applications; agriculture; and pumping of water in rural areas (Dept. of Energy, 2016). South Africa has an ideal climate for solar power applications with a lot of sunshine days, as well as a lot of daylight hours. Data from ClimaTemps (2016) can be seen in Figure 1.



**Figure 1:** An overview of the weather in Pretoria, South Africa.

Solar water heating systems can be active or passive. Active systems contain a pump or fluid moving device, that enables circulation of the heat storage fluid, between the storage tank and the collector. Passive systems rely on gravity or the natural convective tendency of a big bulk of water to circulate and spread the heat. Most systems that heat water with solar radiation are active systems. The two most common active type solar collectors are the evacuated tube collectors and the flat plate collectors. There are also a lot of different designs for flat plate collectors. The design used in this project has copper tubing that runs through the absorber plate, and therefore heats up the fluid running through the tube. This is depicted in Figure 1.

: 

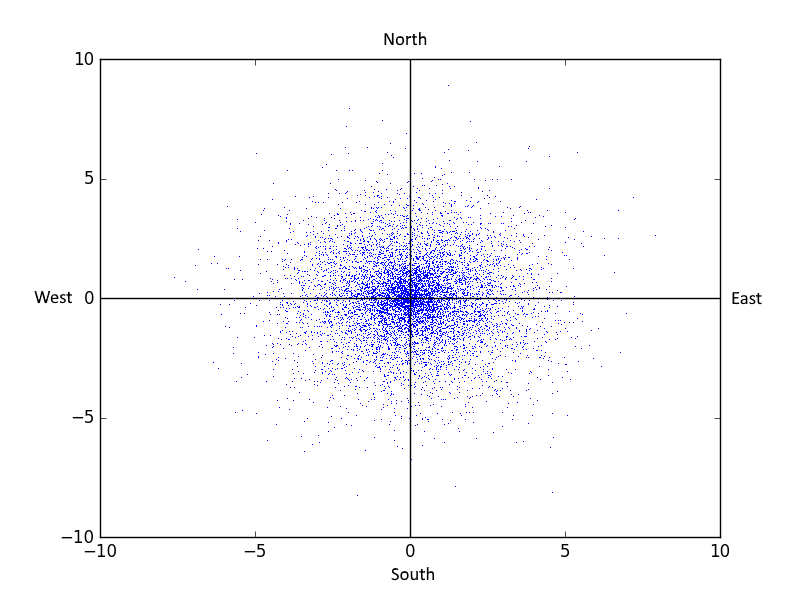
**Figure 2:** The layout and design of the flat plate solar collector used.

Flat plate solar collectors are mostly designed for energy delivery at moderate temperatures (up to 100 ⁰C above ambient temperature). These collectors convert both beam and diffuse radiation from the sun and do not need a lot of maintenance. Another advantage of these collectors is that they do not require solar tracking (tracking can be implemented at a loss of overall panel efficiency, or a very small gain in efficiency) (Duffie & Beckman, 2006). These units are mechanically and thermally a lot simpler the evacuated tube type collectors.

Despite the physical design of the collector, there are a multitude of other conditions to consider when modelling or setting up a solar collector. A few of these conditions include: the average wind speed and direction; the effective radiation available for absorption by the collector; the materials of the collector plate, cover, and insolation; and the mean difference in angle between the normal plane of the collector and the Direct Normal Irradiance (DNI).

## Average Wind Speed and Direction

In order to minimise the heat loss through, the solar panel should preferably lie in a direction where the glass/polymer cover gets the least amount of wind exposure. In Figure 2, the hourly wind speed and direction over the course of one year (2016) is displayed (data from SAURAN, 2016). The wind speed and direction appears to be very evenly distributed.



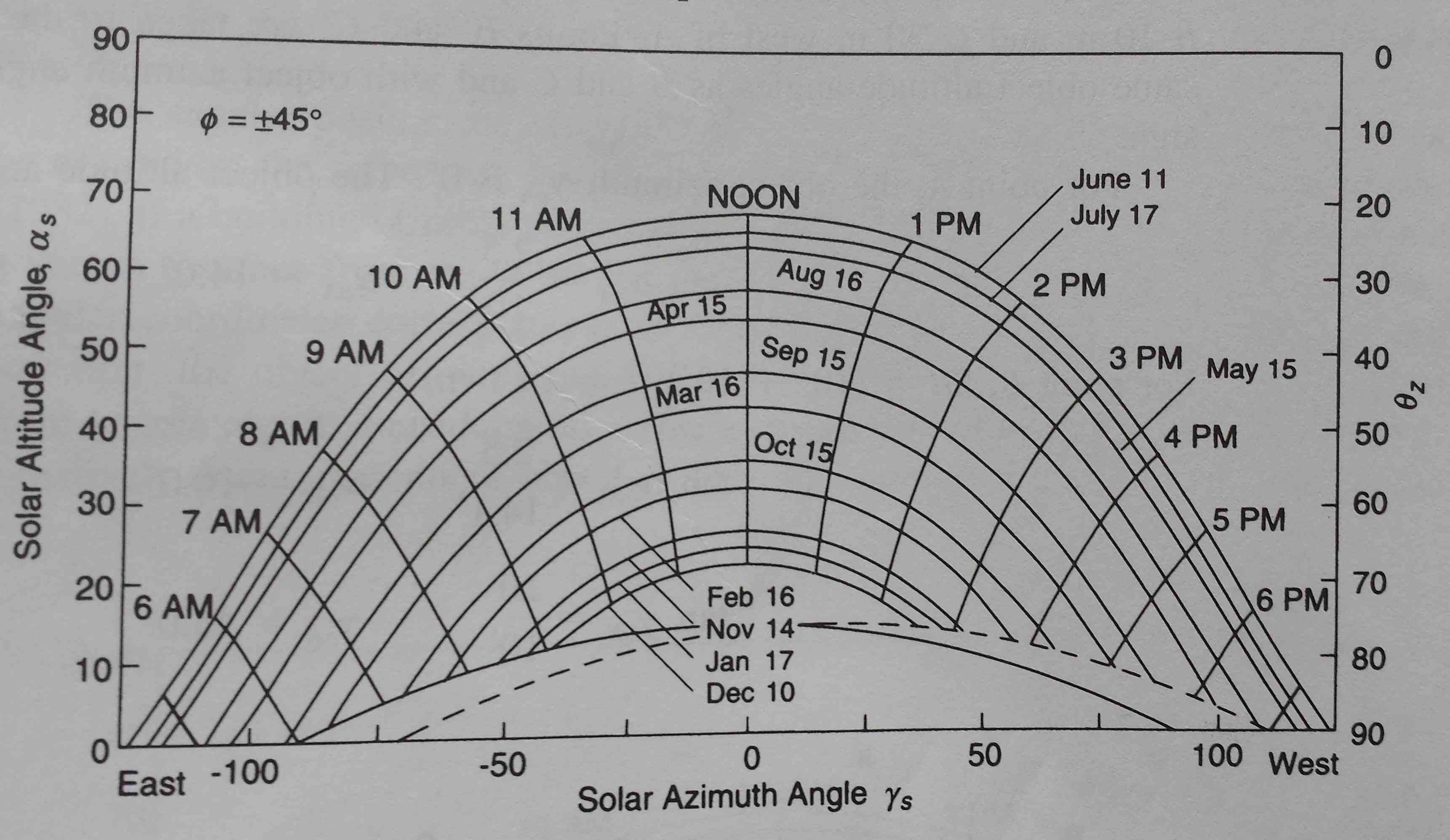
**Figure 3:** A wind speed and direction distribution for Pretoria, South Africa.

The most notable observation in Figure 2, is the low average wind velocity (a mere 2 m/s). This is very advantageous for setting up a solar farm in the Pretoria region.

## Effective Radiation

Radiation from the sun is the most resource of energy for the earth. There are however a multitude of factors to consider when calculating, or describing the heat from the sun. Because the sun’s light contains a lot of wavelengths or spectrums, the heat absorbed by different materials may vary according to ability of the material to absorb as large a spectrum of light as possible. Mostly these effects are lumped into the absorption and emission coefficients of materials.

Due to the rotation of the earth and its movement around the sun, the sun does not radiate toward earth at a constant angle. In the summer months of a region the sun will shine at an almost 90 ⁰ angle with the horizontal at noon. This angle will change in the winter season of the same region. This is better illustrated in Figure 5 (Duffie & Beckman, 2006).



**Figure 4:** The angle the sun makes to the horizontal throughout the year in California, United States of America.

There are several different ways to measure the incoming radiation on a surface. The three most common measurements are the Direct Normal Irradiance (DNI), Diffuse Horizontal Irradiance (DHI), and the Global Horizontal Irradiance (GHI).

DNI is the radiation perpendicular to the surface of the irradiated surface. DHI is irradiance from the scattered light in the atmosphere. A relationship between the three irradiance quantities is

|  |  |  |
| --- | --- | --- |
|  |  | () |

where is the angle between the sun and the normal line of the irradiate surface, the graph in Figure 4 is of some relevance in the calculation of this angle. Mostly this angle is measured and lumped into the GHI. The GHI is therefore the irradiance value that is of use when modelling a collector.

## Materials and Construction of a Collector

### Glass Cover

The glass cover’s (usually made of tempered or toughened glass) purpose is to protect the rest of the collector from environmental effects, while at the same time allowing radiation to strike the absorber plate. Usually the glass used is very transparent, allowing more than 90% of the incident radiation to pass through.

### Absorber Plate

Made from a thin sheet of aluminium, with a coating of a highly selective material that is very efficient at absorbing solar radiation, the absorber plate’s main purpose is to convert the incident radiation to effective heat that can be transferred to the water (or another working fluid). The welding between the copper pipes and the aluminium sheet is usually done ultrasonically, in order to weld joint with very low thermal resistance.

### Insolation

The insolation’s, usually made from ultra-light weight melamine foam, function is to minimize the heat loss through the back of the absorber plate. Newer flat plate models use a vacuum as insolation (AEE INTEC).

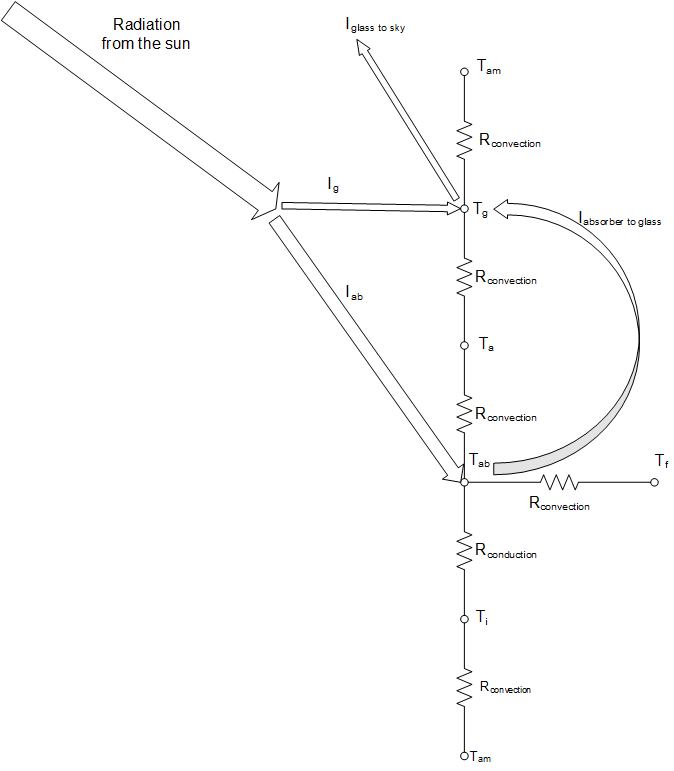
### Riser and Header Pipes

Made from copper, these tubes are welded ultrasonically in order to minimize heat loss over the weld joints. Further functions of these tubes are to transfer the stored/converted heat to the working fluid. Small riser pipe diameters contribute to turbulent flow through the pipe, maximizing the convective heat transfer within the pipes.

Further detail regarding the physical and heat transfer parameters of all the relevant materials can be found in the Appendix.

## Thermal Network of the Collector

The easiest way to conceptualize the heat transfer within a flat plate solar collector is to draw up a thermal resistance network. This can be seen in Figure 4.



**Figure 5:** A thermal network analysis of a flat plate solar collector.

While resistances were not actually used to set up the model, the resistance network still holds as a very convenient way to illustrate the system. The convective and conductive ‘resistances’, were only used to in order to calculate the heat flux, by using the common transfer function for heat transfer

|  |  |  |
| --- | --- | --- |
|  |  | () |

The equation describing the thermal resistance to convection is

|  |  |  |
| --- | --- | --- |
|  |  | () |

whereas the equation describing thermal resistance to conduction is

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Ḉengel & Ghajar, 2011). For more information regarding the heat transfer coefficients refer to Section 1.6 and Appendix.

## Radiation heat flow

In this model, it was assumed that the emitted radiation from the glass panel and the absorber plate was substantial. The other components’ radiative effects were assumed to be negligible.

The radiative heat flux from the glass panel to the sky, can be described by

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Ḉengel & Ghajar, 2011), where the sky temperature is given by Swinbank’s formula

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Saleh, 2012).

Radiation from the absorber plate is toward the glass plate, and can be calculated using the formula

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Goswami *et al*, 2000).

All of the radiation constants can be seen in Appendix.

## Convective heat transfer coefficients

Multiple convective heat transfer coefficients are needed for the model. All of these equations are from Ḉengel & Ghajar (2011).

Firstly, the forced external convection coefficient of all the surfaces (the glass panel as well as the back of the collector) can be calculated, assuming there is a certain wind speed, by

|  |  |  |
| --- | --- | --- |
|  |  | () |

where

|  |  |  |
| --- | --- | --- |
|  |  | () |

for laminar flow and

|  |  |  |
| --- | --- | --- |
|  |  | () |

for turbulent flow. The Nusselt number described above are average Nusselt number over the whole plate region. A fluid with a Reynolds number of less than is considered to be laminar, while the moment the Reynolds number is larger than this number, the flow is considered to be turbulent, as the general assumption is that there is no transitional region between laminar and turbulent flow. The Reynolds number for external flow over a flat plate can be expressed as

|  |  |  |
| --- | --- | --- |
|  |  | () |

When considering forced internal convection in the insides of the riser pipes, the following expressions can be used to calculate the convective heat transfer coefficient. The Reynolds number can be calculated using

|  |  |  |
| --- | --- | --- |
|  |  | () |

It is again assumed that there is no transitional region between laminar and turbulent flow. Flow is considered to be laminar when the Reynolds number is less than , while a Reynolds number higher than this number will indicate turbulent flow. The convective heat transfer coefficient can then be calculated by

|  |  |  |
| --- | --- | --- |
|  |  | () |

for laminar flow (with the assumption of a constant surface temperature) and

|  |  |  |
| --- | --- | --- |
|  |  | () |

for turbulent flow, where for heating and for cooling applications.

The air pocket inside the collector is susceptible to natural convection from the glass plate as well as the absorber plate. The plates were assumed to be perfectly horizontal. The Rayleigh number should be calculated as

|  |  |  |
| --- | --- | --- |
|  |  | () |

where

|  |  |  |
| --- | --- | --- |
|  |  | () |

The Rayleigh number can then be used to calculate the convective heat transfer coefficient using

|  |  |  |
| --- | --- | --- |
|  |  | () |

for heat loss from the upper surface of a hot plate (the absorber plate) and

|  |  |  |
| --- | --- | --- |
|  |  | () |

for heat loss from the bottom surface of a hot plate (the glass cover). In Equation (16) and Equation (17) can be calculated with

|  |  |  |
| --- | --- | --- |
|  |  | () |

## Efficiency of a Collector

Due to time-related factors, two efficiencies are defined for a flat plate solar collector. These efficiencies are the instantaneous and average efficiencies. The instantaneous efficiency is simply

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Goswami *et al*, 2000). The average efficiency can be defined as

|  |  |  |
| --- | --- | --- |
|  |  | () |

(Duffie & Beckman, 2006). In both cases

|  |  |  |
| --- | --- | --- |
|  |  | () |

## Previous models

A dynamic model of a flat plate solar collector was set up by Saleh (2012), as part of a Master of Science in Engineering thesis. In the model, a partial volume approach is used, solving the time differential in several nodes, serving as the length dependant differential. Nearly all of the heat transfer calculations were done using Duffie & Beckman (2006) as a guideline. The maximum error over two two-hour testing periods was 4.5 %. His model performed a lot worse in windy conditions, than in still atmosphere conditions. The efficiency of the collector was not calculated.

Gunjo & Mahanta (2016) set up a Computational Fluid Dynamics (CFD) model for a single riser tube for steady state conditions. This study was very thorough, and different temperature profiles for the length across a riser pipe, the cross-sectional area of the riser pipe, and the insolation adjacent to a riser pipe was generated using Ansys. The absorption and heat transfer efficiencies for absorber plates of different materials (copper, aluminium, and steel) were determined and compared. The model could predict the steady state outlet temperature with a maximum relative error of 5.4%. Aluminium was recommended as the best absorber material, since the performance of copper, was only marginally bigger than that of aluminium.

Hung *et al* (2016) set up a CFD model, using Solar Ray Tracing as a method for thermal radiation flux boundary layer analysis. The purpose was to harness thermal energy more effectively using the principles of solar ventilation technology and a solar air collector. They concluded that an inverse proportionality exists between the outlet temperature and efficiency, and the emissivity of the material used in the construction of a solar collector.

# Experimental and Model

## Assumptions in Model

* Uniform mass flow in collector tubes
* One-dimensional heat transfer
* No heat transfer in the direction of the flow
* No heat transfer or heat loss from the edges of the collector
* All thermal properties are independent of temperature
* Heat loss through the front and the back are to the same ambient temperature and atmospheric conditions
* The sky can be considered as a black body for radiation at an equivalent sky temperature
* No dust and dirt on the collector panel
* No temperature drop over the glass panel and the absorber plate to the insides of the tubes.

## Setting up the model equations

All of the equations were set up from basic principles, therefore using the basic transfer function equation

|  |  |  |
| --- | --- | --- |
|  |  | () |

where in this case will represent energy, as the energy balance is of importance.

### The Glass cover

The transfer function across the cover is

|  |  |  |
| --- | --- | --- |
|  |  | () |

### The Air duct

The transfer over the air duct is

|  |  |  |
| --- | --- | --- |
|  |  | () |

### The absorber plate

The transfer function across the absorber plate is

|  |  |  |
| --- | --- | --- |
|  |  | () |

### The insolation

The transfer function across the insolation is

|  |  |  |
| --- | --- | --- |
|  |  | () |

### The working fluid

The transfer function across the working fluid pipes is

|  |  |  |
| --- | --- | --- |
|  |  | () |

## The simulation equations

### The differential equations

Euler’s integration method was used during the simulation of these equations. In order to do this Equation (28) had to be discretised with respect to as the integration was done in terms of . The partial derivative term in Equation (28) can be discretized in the following manner,

|  |  |  |
| --- | --- | --- |
|  |  | () |

Constants for the equations in Section 2.2 were lumped together in order to produce a more logical simulation. The differential equations that were solved using Euler’s integration method were

|  |  |  |
| --- | --- | --- |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |

As is seen in the above equations, only the working fluid temperature is written in terms of the node numbers. All the other temperatures are calculated in the node of the iteration. The Euler integration method used is described by

|  |  |  |
| --- | --- | --- |
|  |  | () |

Where can be replaced by any subscript , , , , or .

### The simulation constants

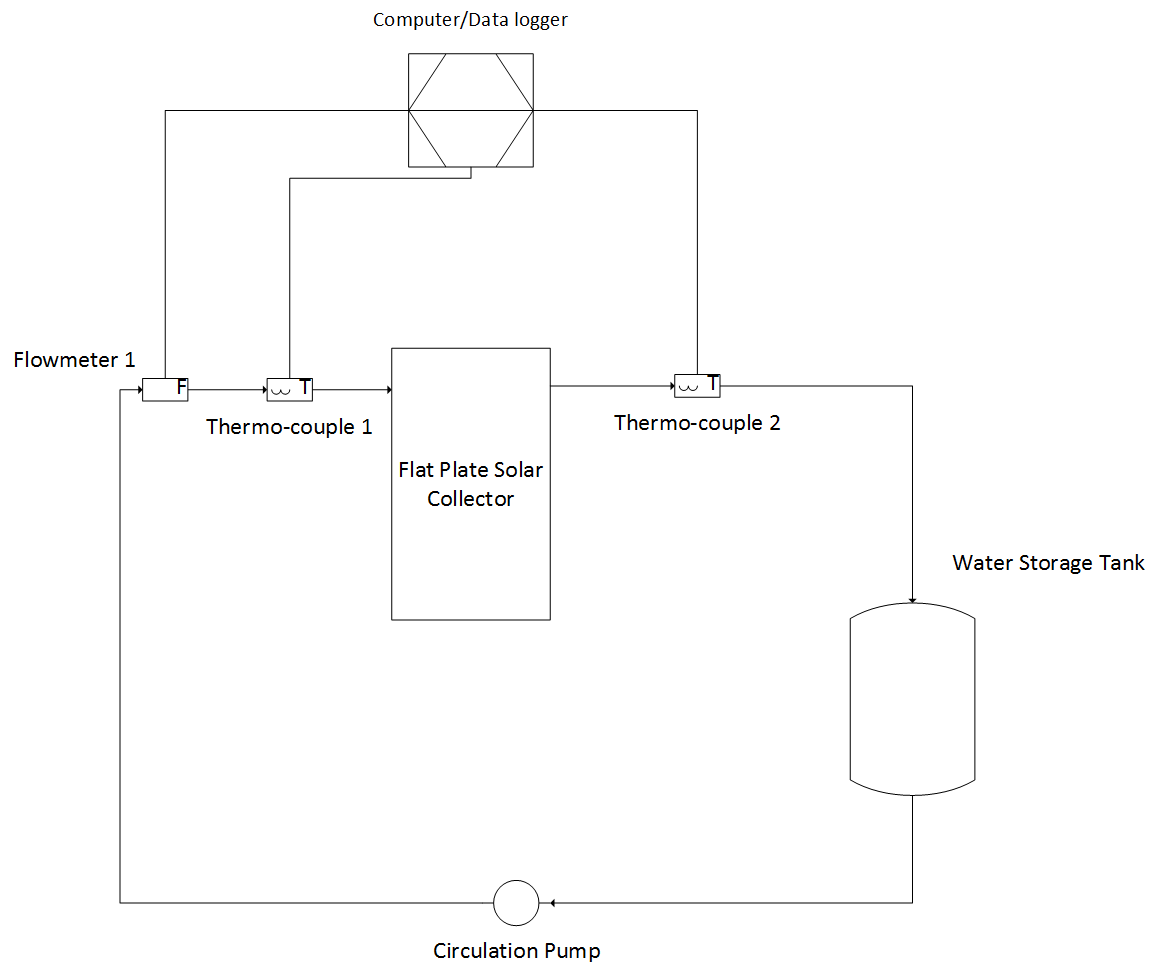
The various constants defined for the differential equations are

|  |  |  |
| --- | --- | --- |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |
|  |  | () |

## Experiments

During the first experiment, the aim was to generate data that could be used for validating the model. In order to generate data that can validate your model up to an excellent degree, it is needed to step an input to the model. The only parameter that could be stepped accurately, due to limitations in the equipment used, was the flowrate through the collector.

A Process Flow Diagram (PFD) of the setup is displayed in Figure 6. A ball valve was installed just before the collector to ensure a permanent head for the pump when the pump is stationary, as well as to control the flowrate.

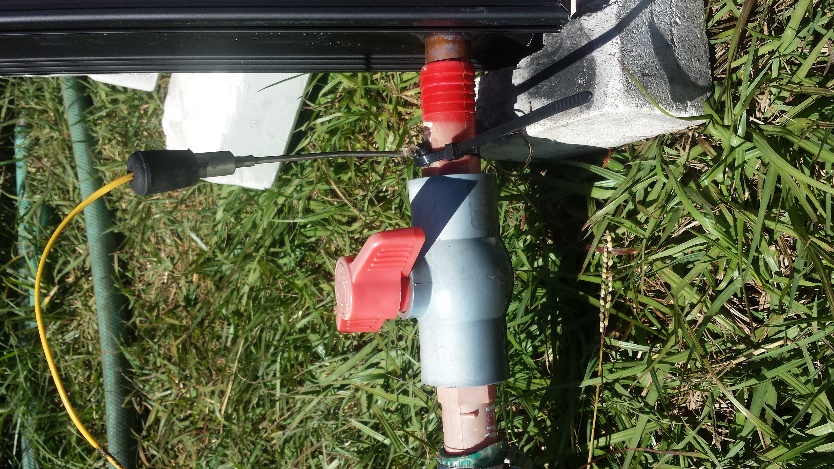


**Figure 6:** A PFD of the experimental setup.

Figure 7 displays the actual setup. Figure 8 displays the valve used to control flowrate and the physical thermocouple setup.



**Figure 7:** The experimental setup.



**Figure 8:** The valve and thermocouple connections.

Since the collector was not set up so that the DNI is its maximum, a correction for the angle to direct irradiance had to be made. This correction was

|  |  |  |
| --- | --- | --- |
|  |  | () |

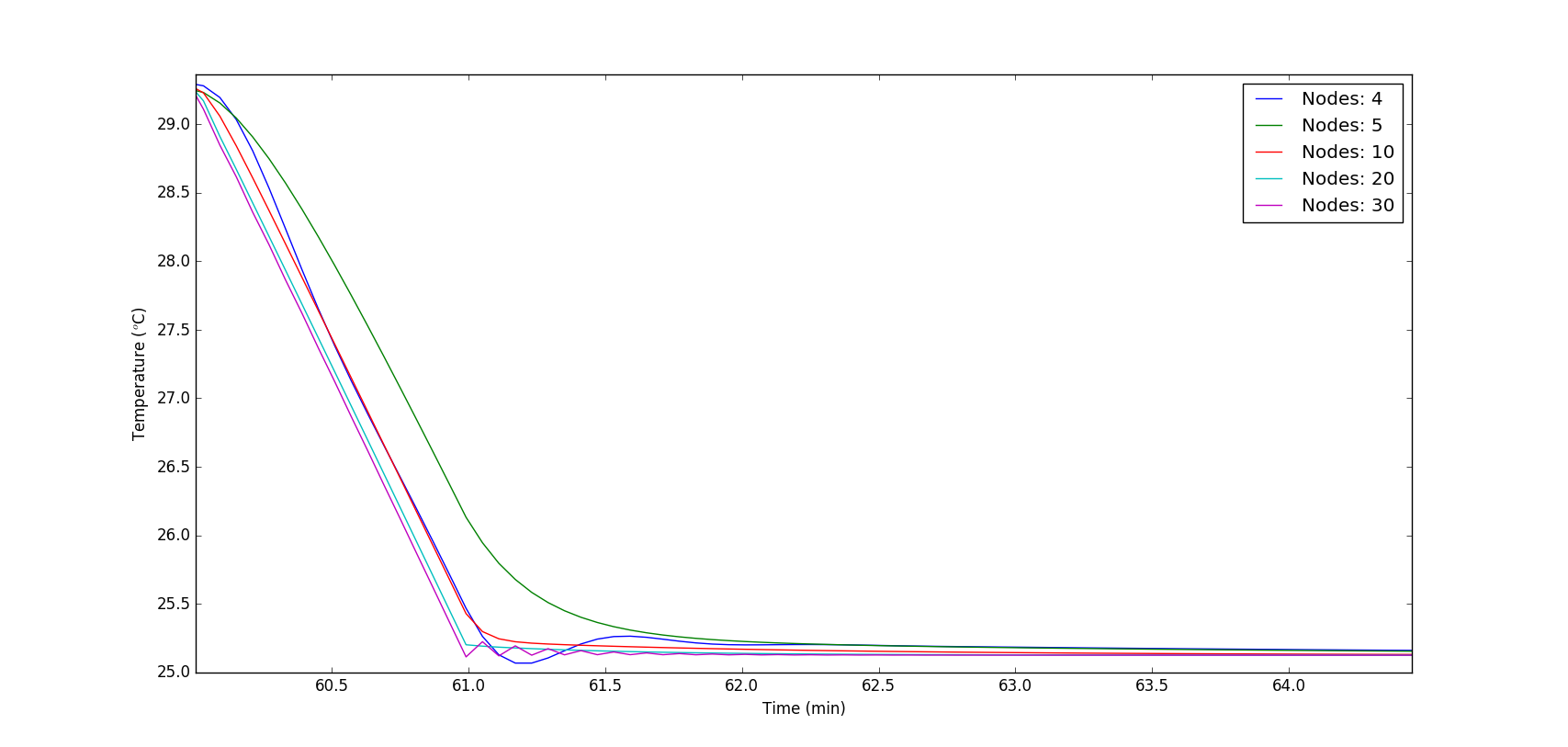
# Results and Discussion

## Model performance

For the simulations in this section the same inputs were used unless specified otherwise. The was ramped from 150 to 800 W/m2K during the first hour, and then made equal to zero during the second hour of simulation. The default flowrate was 0.09 kg/s, while windspeed was kept constant at 3 m/s. The inlet temperature was also kept constant at 25 ⁰C.

### The effect of number of nodes on the performance

The step change in at the one hour mark produces the best dynamics for comparing the effect of the number of nodes in the simulation. The number of iterations were kept constant at 2000 iterations for this simulation. Figure 9 displays the dynamic responses obtained.



**Figure 9:** The dynamic response curves of the simulation for a step change in .

It is clear that the effect of the number of nodes starts to get diminished when the number of nodes are larger than 20. Iteration problems were encountered during the simulation of 30 nodes, due to a too big time interval, while the small number of nodes (4 nodes) showed an underdamped second order response curve, while the real output is second order overdamped. The runtimes for the simulation of different number of nodes are displayed in Table 1.

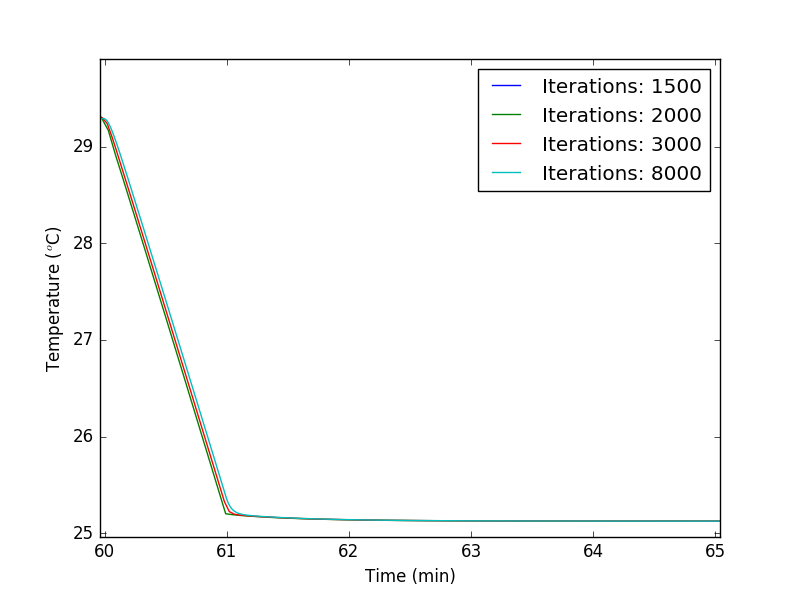
**Table 1:** The simulation runtimes for a different number of nodes.

|  |  |
| --- | --- |
| Number of nodes | Runtime (s) |
| 4 | 9.94 |
| 5 | 13.1 |
| 10 | 23.5 |
| 20 | 46.3 |
| 30 | 72.8 |

It is clear that the number of nodes has a great effect on the accuracy of the simulation as well as its runtime. Using Figure 9 to obtain where the simulation results will start to converge with an increase in the number of nodes, it was established that 20 nodes will provide an optimum accuracy when runtime is also considered.

### The effect of iteration steps on the performance

For this simulation, the number of nodes were kept constant at 20, while a varying amount of iteration step were simulated. Figure 10 displays the dynamic results of the simulations.



**Figure 10:** The dynamic response of the simulations for a step change in .

The graph of 1500 iterations is not displayed, since this iteration step size was much too small, so the iteration did not converge at all. It is clear that the response becomes a lot smoother as the number of iterations are increased, this only effects the very fast dynamic the behaviour of the simulation, as the steady state temperature is virtually identical for all iteration step sizes. The runtimes for the different number of iterations are displayed in Table 2.

**Table 2:** The runtimes for the different simulations.

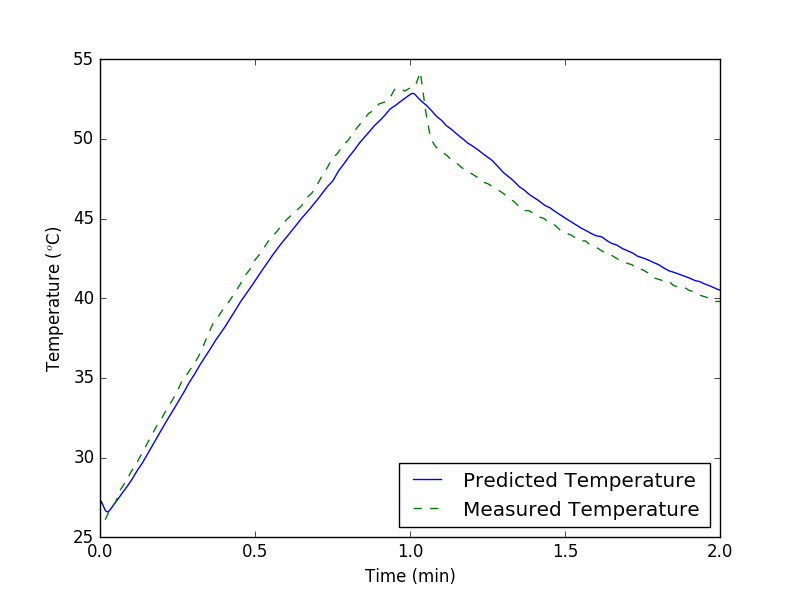
|  |  |  |
| --- | --- | --- |
| Number of iterations | Time step (s) | Runtime (s) |
| 1500 | 4.8 | 38.5 |
| 2000 | 3.6 | 53.7 |
| 3000 | 2.4 | 83.0 |
| 8000 | 0.9 | 211.1 |

As expected the runtime increases as the number of iterations increase. It is therefore necessary to use Figure 10 to establish what the number of intervals should be for the required accuracy used in the simulation. Considering both the runtimes as well as the accuracy, it was decided that 2000 iterations will adequate in both runtime and in accuracy of the simulation.

The optimum simulation parameters were established to be 20 nodes and 2000 iteration. The overall model performance is good, although some effort can be spent in order to optimize the runtime of the simulation even further.

## Theoretical validation

Since the available equipment did not allow me to control the irradiation from the sun, data from another report (Saleh, 2012) was used to validate the performance of the model based on the varying irradiance. During this simulation 20 nodes and 2000 iterations were selected as simulation constants. Figure 11 displays the results obtained from the simulation.



**Figure 11:** Simulated prediction compared to data from external source.

The predicted temperature is clearly lower than the measured temperature, right up until there is a great drop in the at about 1 hour. The predicted response is a lot slower than expected. This may be due to faulty input data.

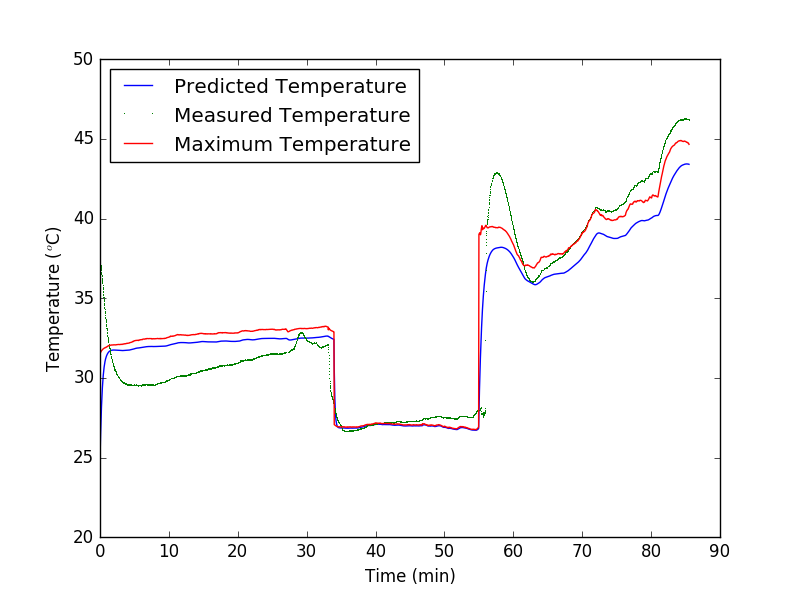
## Experimental validation

During the experimental validation, the flowrate through the collector was stepped twice during the experimental period. Table 3 shows the different flowrates at the different times during the experiment.

**Table 3:** The variation of flowrate during the experimental validation of the model.

|  |  |
| --- | --- |
| Time period (min) | Flowrate (kg/s) |
| 0 – 34 | 0.0130 |
| 34 – 55 | 0.0327 |
| 55 – 90 | 0.0078 |

Figure 11 displays the results obtained, from selecting 20 nodes and 2000 iterations as simulation parameters.



**Figure 12:** Experimental results compared to simulated prediction.

The average error of the predicted outlet temperature was 1.49 with a standard deviation of 1.29. These simulated values are therefore trustworthy up to a precision of about 2.8 ⁰C. The maximum temperature depicted in Figure 12, is calculated as the outlet temperature when the instantaneous efficiency is set equal to 1.

It is clear that some of the measured temperatures are wrong, due to the fact that they are above the maximum possible temperature. The position of the thermocouple on the outlet side, as well as the water level in the outlet pipe, may be the root cause of these incorrect readings. The thermocouple got in contact with too much air above the outlet water stream and measured an elevated temperature. This is further motivated by the fact that the wrong measurements were made at the lowest flowrate.

Another thing to notice is that there seems to be a relationship between the efficiency of the collector, and the flowrate through the collector. Flowrate is the only parameter that can affect the efficiency significantly. Efficiency will only be affected temperature, when the operation temperature of the collector becomes moderate high (more than 90 ⁰C), and radiation heat losses will start to play a major role in the energy balance.

# Conclusion and Recommendations

## Conclusions

On the basis of the results obtained in this study, it can be concluded that:

* A detailed mathematical transient model of a flat plate solar collector was set up from first principles, using simplifying assumptions.
* The system of equations was solved numerically, and a finite difference approach was suggested and implemented.
* The simulation allows for time dependant flowrate, variable ambient temperature, variable wind speed, and variable solar irradiance. The simulation was also set up in order to read solar data from SAURAN.
* The model was then simulated using Python as the software to perform this task. This was carried out successfully. The simulation was analysed according to its runtime as well as its accuracy. The runtime was longer than other simulations of the flat plate collector.
* An experiment was done in order to validate the accuracy of the model. There were some measurement errors in the outlet temperature of the collector, since the measured temperature was higher than the maximum achievable temperature.
* The simulation outputs were compared to the experimental data. The simulated values are accurate up to about 2.8 ⁰C.
* The simulation can be used to predict the flat plate collector performance of a collector consisting of a variety of other materials than discussed in this study, as well as various other atmospheric and irradiation conditions, without the use of doing experimental work.

## Recommendations

* More experimental work is needed to confirm the accuracy of the simulation for numerous other operating conditions.
* The simulation needs to be optimized even further, by using the inner product function (Renze *et* al, 2017) instead of running the Euler integration technique using loops.

# References

Aktamis, H (2011) “Determining energy saving behaviour and energy awareness of secondary school students according to socio-demographic characteristics”, *Educational Research and Reviews*, 6(3), 243 – 250.

Ḉengel, YA and Ghajar, AJ (2011) *Heat and Mass Transfer: Fundamentals and Applications*, McGraw-Hill, New York.

Duffie, JA and Beckman WA (2006) *Solar Engineering of Thermal Processes*, John Wiley & Sons, Hoboken.

Goswami, DY, Kreith, F and Kreider, JF (2000) *Principles of Solar Engineering*, Taylor & Francis, New York.

Gunjo, DG, Mahanta, P and Robi, PS (2016) “CFD and experimental investigation of flat plate solar water heating system under steady state condition” *Renewable Energy*, 106(2017), 24 – 36.

Howell, JR, Bannerot, RB and Vliet, GC (1982) *Solar-Thermal Energy Systems*, McGraw-Hill, New York.

Hung, T, Huang, T, Lee, D, Lin, C, Pei, B and Li, Z (2016) “Numerical analysis and experimental validation of heat transfer characteristic for flat-plate solar air collector” *Applied Thermal Engineering*, 111(2017), 1025 – 1038.

Renze, J, Stover, C and Weisstein, EW (2017) “Inner Product”, <http://mathworld.wolfram.com/InnerProduct.html> [2017, May 3].

Saleh, AM (2012) *Modelling of Flat-Plate Solar Collector Operation in Transient States*, Masters of Science Thesis, Purdue University, Fort Wayne, United States of America.

# Appendix

## Thermal properties of all materials used

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Air | Water | Aluminium | Glass | Melamine foam |
|  | 1008.2 | 4180 | 910 | 840 | 840 |
|  | 2.62E-02 | 6.70E-01 | - | - | 0.033 |
|  | - | - | 0.98 | 0.2 | - |
|  | - | - | 0.9 | 0.2 | - |
|  | 0.0172 | - | 0.0346 | 0.00172 | 0.0172 |
|  | 2 | - | 2 | 2 | 2 |
|  | 1.177 | 1000 | 2700 | 2400 | 1040 |

## The simulation

The simulation was run out of several different python text files. Below are all the lines of code used to solve the model in their separate files.

Listing 1: The Simulation\_Experimental.py file.

|  |
| --- |
| from Integration\_main import Integration\_step  from Data\_Experimental import Import\_data  from Constants\_main import Constants  import matplotlib.pyplot as plt  import numpy as np  A, alpha\_sab, alpha\_sg, Ca, Cab, Cf, Cg, Ci, din, eg, eg, ep, g, ka, kf, ki, \  Lc, Li, Ltube, rhoa, rhoab, rhof, rhog, rhoi, sigma, tau, Va, Vab, Vg, Vi \  = Constants()  Nodes = 1  Times, Tin, Tout, G, Air\_Temp = Import\_data()  Tin = np.array(Tin) + 273.15  Tout = np.array(Tout) + 273.15  def mass\_flow(t):  if t < 34\*60:  return 0.01296  elif t < 55\*60:  return 0.0327  else:  return 0.007788  for Nodes in [20]:  del\_z = Ltube/Nodes  Temps = np.array([35, 35, 25, 37, 35]) + 273.15  iterations = 2000  tspan = np.linspace(0, Times[-1], iterations)  del\_t = (tspan[1] - tspan[0])  Tf\_ult = []  Tg\_ult = []  Ta\_ult = []  Tab\_ult = []  Ti\_ult = []  Tmax = []  for t in tspan:  Tam = np.interp(t, Times, Air\_Temp) + 273.15  Gstep = np.interp(t, list(Times), G)  mf = mass\_flow(t)  Tf\_prev = float(np.interp(t, np.array(Times), Tin))  Tmax.append(Gstep/2.9\*A/mf/Cf + Tf\_prev)  for z in range(Nodes):  Temps = list(Temps) + [Tam]  Temps = Integration\_step(Gstep, 3, mf, Temps, Tf\_prev, del\_t,  del\_z)  Tg, Ta, Tab, Tf, Ti = Temps  Tf\_prev = Tf  Tf\_ult.append(Tf)  Tg\_ult.append(Tg)  Ta\_ult.append(Ta)  Tab\_ult.append(Tab)  Ti\_ult.append(Ti)  print(str(t/tspan[-1]\*100)[:5] + ' % complete')  plt.plot(tspan/60, np.array(Tf\_ult) - 273.15,  label='Predicted Temperature')  plt.plot(np.array(Times)/60, Tout-273.15, ',', label='Measured Temperature')  plt.plot(tspan/60, np.array(Tmax)-273.15, label='Maximum Temperature')  plt.xlabel('Time (min)')  plt.ylabel('Temperature ($^o$C)')  plt.legend(loc=0)  error = [abs(np.interp(t, tspan, Tf\_ult) - np.interp(t, np.array(Times), Tout))  for t in tspan]  print('Standard deviation of error = ', np.std(error))  print('Average error = ', np.average(error)) |

Listing 2: The Integraion\_main.py file.

|  |
| --- |
| from Parameters\_main import Parameters  from Constants\_main import Constants  def Integration\_step(G, vwind, mf, Tlist, Tf\_prev, del\_t, del\_z):  A, alpha\_sab, alpha\_sg, Ca, Cab, Cf, Cg, Ci, din, eg, eg, ep, g, ka, kf, \  ki, Lc, Li, Ltube, rhoa, rhoab, rhof, rhog, rhoi, sigma, tau, Va, Vab,\  Vg, Vi = Constants()  Tg, Ta, Tab, Tf, Ti, Tam = Tlist  B, C, D, E, F, H, I, J, K, L, M, N, O, P, Q = Parameters(vwind, Tlist, mf,  del\_z)  Tsky = 0.0552\*Tam\*\*1.5  G = G/2.9  dTgdt = C/B\*(Tam-Tg) + D/B\*(Tab\*\*4-Tg\*\*4) + E/B\*(Ta-Tg) + F/B\*G \  + Q/B\*(Tsky\*\*4 - Tg\*\*4)  dTadt = E/H\*(Tg-Ta) + I/H\*(Tab-Ta)  dTabdt = J/K\*G + D/K\*(Tg\*\*4-Tab\*\*4) + I/K\*(Ta-Tab) + L/K\*(Ti-Tab) \  + M/K\*(Tf-Tab)  dTfdt = M/O\*(Tab-Tf) - P/O\*(Tf-Tf\_prev)  dTidt = L/N\*(Tab-Ti) + C/N\*(Tam - Ti)  Tg += del\_t\*dTgdt  Ta += del\_t\*dTadt  Tab += del\_t\*dTabdt  Tf += del\_t\*dTfdt  Ti += del\_t\*dTidt  return [Tg, Ta, Tab, Tf, Ti] |

Listing 3: The Parameters\_main.py file.

|  |
| --- |
| import numpy as np  from Constants\_main import Constants  A, alpha\_sab, alpha\_sg, Ca, Cab, Cf, Cg, Ci, din, eg, eg, ep, g, ka, kf, ki, \  Lc, Li, Ltube, rhoa, rhoab, rhof, rhog, rhoi, sigma, tau, Va, Vab, Vg, Vi\  = Constants()  def hgam\_fun(v\_wind, Tlist, del\_z):  Tg, Ta, Tab, Tf, Ti, Tam = Tlist  mu = 1.849E-5  Re = rhoa\*v\_wind\*Lc/mu  Pr = 0.7202  if Re > 5E5:  Nu = 0.037\*Re\*\*0.8\*Pr\*\*(1/3)  else:  Nu = 0.664\*Re\*\*0.5\*Pr\*\*(1/3)  return Nu\*ka/Lc  def hf\_fun(mf, Tlist, del\_z):  Tg, Ta, Tab, Tf, Ti, Tam = Tlist  muf = 8.9E-4  u = mf/(np.pi/4\*din\*\*2)/rhof/8  Re = rhof\*u\*din/muf  Pr = 1.5  if Re > 2000:  Nu = 0.023\*Re\*\*0.8\*Pr\*\*0.4  else:  Nu = 3.66 + (0.065\*din/Ltube\*Re\*Pr)/(1+0.04\*((din/Ltube)\*Re\*Pr)\*\*(2/3))  return Nu\*kf/din  def hga\_fun(Tlist, del\_z):  Tg, Ta, Tab, Tf, Ti, Tam = Tlist  Pr = 0.7202  v = 1.568E-5  p = 6  L = A/p  Beta = 1/((Tg+Ta)/2)  Ra = abs(g\*Beta\*(Tg-Ta)\*L\*\*3/(v\*\*2)\*Pr)  if Ra < 10E7:  Nu = 0.54\*Ra\*\*(1/4)  else:  Nu = 0.15\*Ra\*\*(1/3)  return Nu\*ka/Lc  def haab\_fun(Tlist, del\_z):  Tg, Ta, Tab, Tf, Ti, Tam = Tlist  Pr = 0.7202  v = 1.568E-5  p = 6  L = A/p  Beta = 1/((Ta+Tab)/2)  Ra = abs(g\*Beta\*(Tab-Ta)\*L\*\*3/(v\*\*2)\*Pr)  if Ra < 10E7:  Nu = 0.54\*Ra\*\*(1/4)  else:  Nu = 0.15\*Ra\*\*(1/3)  return Nu\*ka/Lc  def Parameters(vwind, Tlist, mf, del\_z):  A = Constants()[0]  hgam = hgam\_fun(vwind, Tlist, del\_z)  hga = hga\_fun(Tlist, del\_z)  haab = haab\_fun(Tlist, del\_z)  hf = hf\_fun(mf, Tlist, del\_z)  hr1 = sigma/(1/ep + 1/eg - 1)  Vol\_correct = del\_z/Lc  b = A/Lc  B = Cg\*rhog\*Vg\*Vol\_correct  C = hgam\*b\*del\_z  D = hr1\*b\*del\_z  E = hga\*b\*del\_z  F = tau\*alpha\_sg\*b\*del\_z  H = Ca\*rhoa\*Va\*Vol\_correct  I = haab\*b\*del\_z  J = alpha\_sab\*b\*del\_z  K = Cab\*rhoab\*Vab\*Vol\_correct  L = ki\*b\*del\_z/Li  M = np.pi\*din\*del\_z\*hf\*8  N = Ci\*rhoi\*Vi\*Vol\_correct  O = Cf\*rhof\*(np.pi/4\*din\*\*2)\*del\_z\*8  P = mf\*Cf  Q = sigma\*b\*del\_z  return [B, C, D, E, F, H, I, J, K, L, M, N, O, P, Q] |

Listing 4: The Constant\_main.py file.

|  |
| --- |
| import csv  def Constants():  ss\_values = []  with open('Proses\_Constants.csv', 'r') as csvfile:  ss = csv.reader(csvfile, delimiter=',')  for row in ss:  ss\_values.append(float(row[1]))  return ss\_values |

Listing 5: The Data\_Experimental.py file.

|  |
| --- |
| import pandas as pd  import numpy as np  from datetime import datetime  def \_\_datetime(str\_datetime):  str\_datetime = str(str\_datetime)  return datetime.strptime(str\_datetime[2:], '%y/%m/%d %H:%M')  def Import\_data():  columns\_to\_keep1 = ['Times', 'Tin', 'Tout', 'Tam']  dataframe1 = pd.read\_csv("./Test\_2\_Data.csv", usecols=columns\_to\_keep1)  columns\_to\_keep2 = ['GHI\_CMP11', 'Time']  dataframe2 = pd.read\_csv("./Test\_2\_Sun\_Data.csv", usecols=columns\_to\_keep2)  Times\_prelim = np.array(dataframe1['Times'])  Tin = np.array(dataframe1['Tin'])  Tout = np.array(dataframe1['Tout'])  G\_prelim = np.array(dataframe2['GHI\_CMP11'])  Time\_prelim = np.array(dataframe2['Time'])  Air\_Temp = np.array(dataframe1['Tam'])  G = [np.interp(t, Time\_prelim, G\_prelim) for t in Times\_prelim]  Times = []  Out = []  In = []  AT = []  for i in range(len(Tin)):  In.append(float(Tin[i].replace(',', '.')))  Out.append(float(Tout[i].replace(',', '.')))  AT.append(float(Air\_Temp[i].replace(',', '.')))  Times.append(float(Times\_prelim[i]))  return (Times, In, Out, G, AT) |