One- and Two-Dimensional Modeling of Ventilated Façades with Integrated Photovoltaics

Rémi Charron

A Thesis

in

The Department

of

Civil, Building and Environmental Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Building Engineering) at
Concordia University
Montreal, Quebec, Canada

April 2004

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-94679-7
Our file Notre référence
ISBN: 0-612-94679-7

ABSTRACT

One- and Two-Dimensional Modeling of Ventilated Façades with Integrated Photovoltaics

Rémi Charron

Ventilated façades with integrated photovoltaic panels can be used to generate electricity, thermal energy, and for daylighting. Developing working models to study their performance for this thesis is the start of a long-term research project at Concordia University that aims to develop a better understanding of these systems.

The research involved developing both a one-dimensional finite-difference model and a two-dimensional control-volume model. An algorithm was developed that determines iteratively the most appropriate convective heat transfer coefficient relationship to use for surfaces inside the cavity, based on system characteristics and temperature distributions. In the case of the 1D model, average, as opposed to local, coefficients are calculated. The 2D model provides a more detailed representation of the radiation heat transfer between surfaces inside the cavity, and includes vertical heat conduction within the system components. The 1D model on the other hand is more robust and faster to use.

Validation was carried out which compared results obtained from the models to results found in literature, and from experiments. In addition, an inter-model comparison was done between the 1D and 2D models. Results show that the models come within the 10 to 21 percent uncertainty levels predicted by other researchers.

After validation, the models were used to optimise the system performance, which showed that combined thermal-electric efficiencies of over 70% could be attained.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Dr. A.K. Athienitis, for his expert guidance, encouragement, and support during my graduate studies.

Special thanks and appreciation also go to my wife, Carrie, for her patience, understanding, encouragement, and moral support.

I would also like to thank Lisa Dignard-Bailey and Yves Poissant for their assistance in setting up the Concordia test facility, and Joe Hrib and Jian Zhang for their assistance with the data collection.

Finally, I would like to thank le Ministère des ressources naturelles du Québec, ATS Automation Tooling Systems, the Natural Sciences and Engineering Research Council of Canada, and CANMET for their support with this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ν	Kinematic viscosity
$\overline{T}_{ma}$	Mean air temperature
f	Friction coefficient
σ	Stefen-Boltzman const. ($5.67 \cdot 10^{-8} W/m^2K^4$)
ρ_{air}, ρ	Air density (kg/m^3)
ε_i	Emissivity of surface i
ρ_i	Reflectivity of i
τ_i	Transmissivity of surface i
η_{pv}	PV cell efficiency
Δx	Height of each element
AFW	Airflow window
AFW-IP	Airflow window with integrated photovoltaics
AST	Apparent solar time
BIPV	Building integrated photovoltaics
C	Specific heat of air (J/kg·K)
CFD	Computational fluid dynamics
CHTC	Convective heat transfer coefficient
D, D_h	Hydraulic diameter
DMS	Direct numerical simulation
E_{fan}	Electricity consumed by fan
E_{pv}	Electricity generated by PV

ES	Energy simulation
esr	Earth spin rate
ET	Equation of time
E_{trans}	Solar irradiation transmitted through AFW
FEM	Finite element method
g	Gravitational acceleration (9.81 m/s^2)
G	Incident solar irradiation (W/m^2)
Gr	Grashof number
ha	Hour angle
h_i	Convective heat transfer coefficient on surface i
h_o	Heat transfer coefficient on outside surface (conv. + radiative)
H_{pv}	Height of PV section
h_{room}	Heat transfer coefficient on room-side surface (conv. + radiative)
H_w	Height of Vision section
I_{bm}	Beam irradiation
I_{df}	Diffuse irradiation
I_{gnd}	Ground-reflected irradiation
In_i	Radiation directed outside from i
I_{on}	Extraterrestrial solar radiation
Ip_i	Radiation directed to room from i
J	Jacobian matrix
k_i	Thermal conductivity of i (W/K)
L	Gap width

L_c	Characteristic length
L_{fin}	Fin length
lw	Longwave
M	Air flow rate of air (m^3/s)
Nu	Nusselt number
P	Perimeter
Pe	Peclet number
Pr	Prandtl number
PV	Photovoltaic
Q_{air}	Heat absorbed in the air
q_f	Heat transferred from fins
Q_i	Heat transferred to i
q_{max}	Maximum theoretical q_f
Q_o	Heat transfer lost to the outdoor environment
Q_{room}	Heat transferred to room by AFW
r	Component reflectivity
Ra	Rayleigh number
Re	Reynolds number
sha	Sunset hour angle
S_i	Irradiation absorbed in component i
$S_{pv.heat}$	Irradiation absorbed as heat in PV
STM	Local standard time meridian (degrees)
T_i	Temperature of component i

t_i	Thickness of i (m)
T_m	Mean temperature
T_o	Outdoor air temperature
tod	Time of day (hours)
T_{room}	Room temperature
U_i	Thermal conductance of i
V, Vel, u	Air velocity
V_w	Wind velocity
w	Width of AFW
X	Height of entrance region
z	Zenith angle
α	Thermal diffusivity
α_i	Absorbance of surface i
β	Coefficient of volumetric thermal expansion (1/K)
γ	Surface solar azimuth
δ	Declination angle
η_{fin}	Fin efficiency
θ	Angle of incidence
μ	Air viscosity (kg/m·s)
φ	Solar azimuth
ψ	Surface azimuth

CHAPTER 1

INTRODUCTION

1.1 Introduction

In order to make many solar energy technologies cost effective, they need to be integrated into a building's envelope (Krauter, et al., 1999). Photovoltaic (PV) technologies can be integrated to the roof or façade of a building. When PV is integrated into the façade of a building, less solar radiation is incident on the panels in comparison with rooftop installations. When examining costs, it was found that much better payback periods could be achieved when systems with PV integrated on a façade also captured heat (Costa, et al., 2000). The heat can be collected in a similar fashion to that of a regular solar hot-water heater or hot-air collectors. Despite the low heat capacity in an air system, it is the simplest and least expensive way to allow modular installation during construction (Bazilian, Prasad, 2002).

Even with prefabricated modules, the array and labour costs need to be reduced substantially before their payback period is reduced to an acceptable level (Jaros, et al., 2002). Labour costs are expected to fall over time as more technicians gain experience installing PV modules and other solar collectors. As for array costs, they are continuously declining with the advancement of PV technology and manufacturing. Cost effectiveness can be improved by reducing the overall cost of the system, or by improving its overall efficiency. Efficiency can be raised by increasing the PV cell efficiencies, or by increasing the efficiency of the system as a whole, including heat extraction, daylighting, electricity generation, air quality, thermal comfort, etc.

The first record of a mechanically ventilated façade was described by Jean-Baptiste Jobard, director of the Industrial Museum in Brussels, in 1849 (Saelens, 2002). After this time, little is heard of the idea until the early to mid 20th century. Airflow windows as studied in this paper, started being studied in Scandinavia in the 1950's, with the first patent being filed in 1957 in Sweden. It took another ten years before its first use in office buildings, in Helsinki, Finland.

Not until the energy crises of 1973 and 1979, did the concept start to appear in European buildings. However, as the energy crisis abated, so did the interest in ventilated façades. By the end of the 1990's, interest started to increase again as more people started to focus on energy efficiency, and the air quality of buildings. Double-façades have since been used to combat the "sick-building syndrome" as it gives the building occupants the ability to open windows and control the fresh air intake, even in high office towers (Saelens, 2002), (Zollner, et al., 2002).

1.2 Definition of Ventilated Façades

There are four different paths for air to flow in a ventilated facade, as depicted in Figure 1.1. All four achieve a different intended result. The first diagram is that of a supply window, where the outside air flows through the cavity and is then either ventilated directly into the building, or used as a fresh air supply in an HVAC system. The supply window is used in colder climates in an attempt to preheat fresh air going into a building to save on heating costs. The second type, known as an exhaust window, blows the exhausted air from an HVAC system through the facade before being released to the outside environment. This type of ventilated window utilizes the energy already

used to heat or cool the building to bring the window temperature closer to that of the indoor environment.

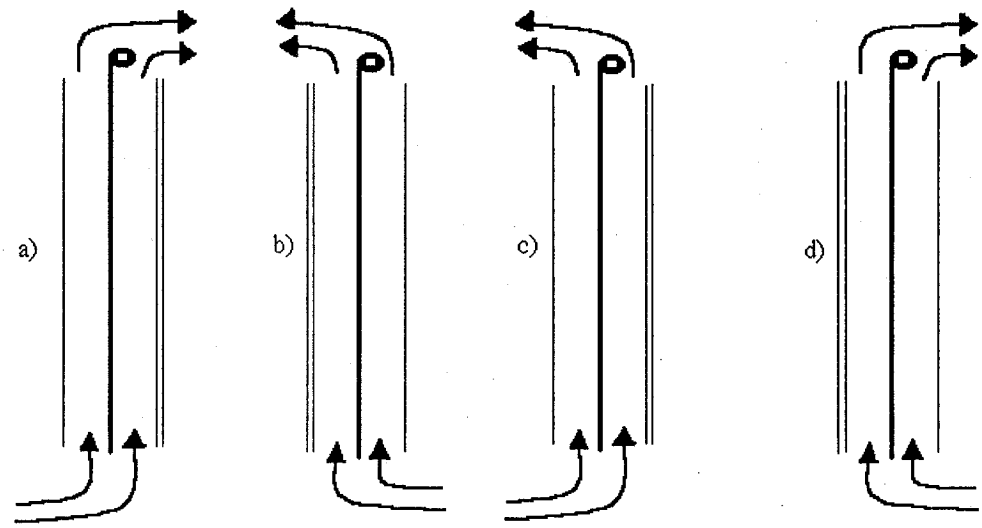

Figure 1.1: Airflow Paths Used in Ventilated Façades

The last two paths have the air leaving the cavity on the same side that it comes in, a configuration which is sometimes referred to as an exterior air curtain (Saelens, 2002). When outside air flows through the system, it is referred to as an exterior air curtain. This is often used with natural ventilation to help cool the window lowering the cooling load of the building. Interior air curtains use air from the mechanical ventilation system to flow in the cavity to create a buffer between the window and the outdoor environment. As depicted in the figure, when outdoor air is used in the cavity, the double-glazed window is closest to the room and faces the outdoor environment. When used as an exhaust window or interior air curtain, the double-glazed window needs to be the barrier between the outdoor and indoor air to reduce the building's energy load. For added insulation, two double-glazed windows may be used instead of one.

PV technology can be integrated into a double-façade in a variety of different ways. Figure 1.2 shows an application done at the Mataro library in Barcelona where PV cells are sandwiched between two glazings to form a semi-transparent PV integrated façade. The PV generates electricity, and produces shading at the same time. There is also a second double-glazed window that forms a cavity with the PV glazing where air is mechanically drawn from. This particular façade has been studied by a group of researchers, with some of the results and a more detailed discussion presented in later chapters (Costa, et al., 2002), (Infield, et al., 1999), (Mei, et al., 2003).

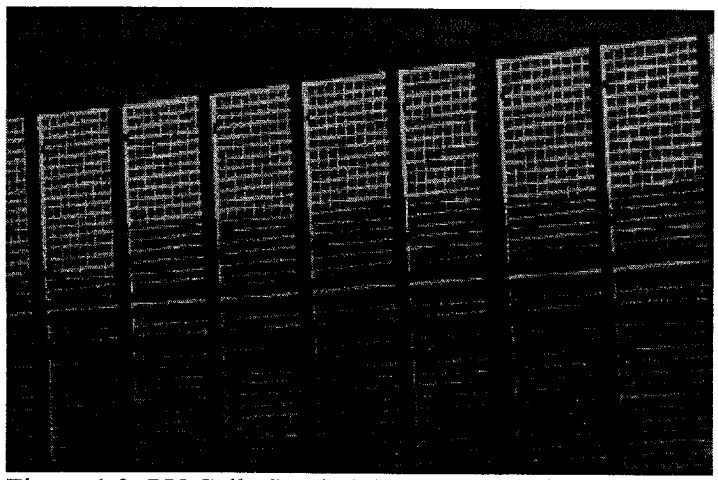

Figure 1.2: PV Cells Sandwiched Between Two Glazings (Infield D., et al., 1999)

This thesis will focus on two different types of double-façades with integrated PV. The first, depicted as Configuration 1 in Figure 1.3, has the PV on the lower half of the façade, facing the outdoor environment directly. The second, depicted as Configuration 2 in Figure 1.3, has the PV on the lower half of the façade, but placed in the middle of the cavity, allowing air to flow on either side, or on both sides at the same time. Both configurations have a motorized blind in the middle of the top section, and the option of bringing outdoor air into the building, or exhausting it outside. These

arrangements will be referred to as air flow windows with integrated photovoltaics (AFW-IP). The top section, referred to as the Vision section, allows a portion of the solar radiation incident upon it to go into the building where it may be utilized for daylighting purposes. The bottom section is called the PV section, and is opaque unless otherwise stated.

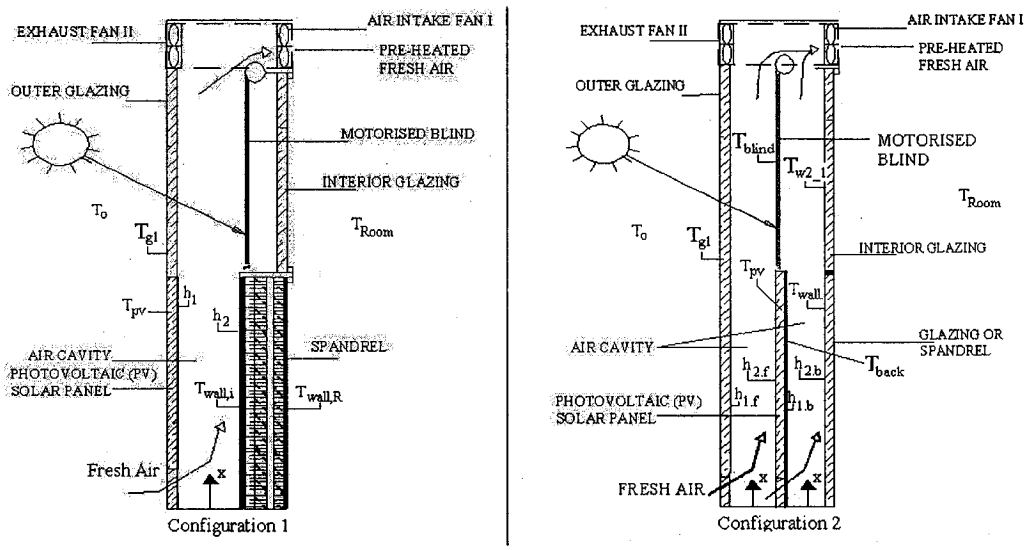

Figure 1.3: Configuration 1 and 2 of the Airflow-Window

1.3 Motivation

Many researchers have examined various configurations of double-façades, and have developed thermo-fluid models to investigate their performance. However, most of the research examines specific topics such as finding proper heat transfer coefficients, the effects of transmittance, the effects of inlet and outlet conditions, etc., with little research being done to find general design guidelines that may be used by designers to optimize performance (Saelens, 2002). Some argue that predicting the performance of double-facades is not trivial, as the performance depends on the geometric, thermophysical, and

optical properties of its various components (Hensen, et al., 2002). Due to the complexity of the problem, it is suggested that general design guidelines are practically impossible to obtain, and that modeling and simulation should be used to support each design directly.

Indeed the system is complex; however, with more research being directed at studying the system as a whole, key parameters may be identified to help us learn how they interact to affect the overall performance of the system. To date, most experimental work has been done studying naturally-ventilated double-façades (Saelens, 2002). Even though daylighting can play an important role in reducing a building's electricity consumption, the effects that multiple-skin façades have on daylighting has not been studied explicitly (Saelens, 2002). Some work has been done to determine the effects of certain parameters and found that predictions are not heavily dependent on surface absorbance, emissivity, or transmittance (Ong, 1995).

1.4 Objectives

The objectives of this thesis are the following:

1 To study the current techniques available to model double-façades and similar technologies,

2 To develop a one-dimensional thermofluid model to help predict the behaviour of an airflow-window with integrated photovoltaics (AFW-IP),

3 To develop a two-dimensional thermofluid model to both validate the use of the one-dimensional model, and to help further study the systematic interactions of the AFW-IP,

4 To use the developed models to identify key parameters and their interactions.

1.5 Thesis layout

Chapter 2 examines the various aspects of modeling double-façades, solar collectors, and other relevant technologies that are found in literature. Chapter 3 presents a description of the one- and two-dimensional models developed for this investigation. The models are then validated in Chapter 4 by comparing their results to each other, to results found in literature, and to preliminary results from the Concordia Building Integrated Photovoltaic (BIPV) test facility. After the models are validated, various results from modeling the AFW-IP using the two models developed in this study are presented in Chapter 5. As the system under study is very complex, the scope of this work is limited to the objectives mentioned above; therefore, the thesis will end with Chapter 6, which suggests future work that should be done in this field of research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Numerous models have been developed to investigate double-façades, solar chimneys, solar collectors, and other similar technologies. With the exception of solar-collectors, which have standardized methods of modeling, no modeling technique is hailed as the preferred method. Models range in complexity from being single zone models with the heat transfer described by a single U-factor, to numerical models based on different computational fluid dynamics methodologies.

Some researchers insist that the complex interactions that occur within the system necessitate the use of a model with a sufficient level of complexity (Saelens, 2002). Others state that the simplicity of the numerical design and its inputs is necessary for users, wary of complex and time-sensitive building simulation tools, to encourage their use (Bazilian, Prasad, 2002).

2.2 One- and Two-Dimensional Models

One of the simplest ways to model these systems is to use a single zone model, in which each cavity is represented by a single node. In this type of model, radiation and convection in the cavity are combined and the heat transfer through the cavity surfaces is described by a single U-factor. Infield, et al. (2002) provided a preliminary report that used this approach as part of an attempt to model the overall building energy simulation

at the Mataro library. They found that the theoretical and experimental results correlated well. Other studies have shown that modeling an AFW with a simple U-factor lacks the complexity to accurately model this type of façade (Saelens, 2002). However, as Infield's group used this approach as part of a model to simulate the building's overall energy use, the inaccuracies of the model were minimal when looking at the simulation as a whole. Saelens (2002) found that the influence of other phenomena on the building heating and cooling load were very high, directly limiting the influence of multiple skin façade typologies on overall energy use. Therefore, if the AFW model is to be used as part of a building simulation, less complexity is required. On the other hand, if the façade itself is being investigated, then the U-value approach is not suitable.

The more sophisticated one-dimensional models assign a node to each layer of the AFW. Another improvement is that radiation and convection are treated separately; this approach is often taken to model solar air heaters. This approach can be represented by a simplified thermal network, and is the basis for the one-dimensional model that will be developed for this study. More information concerning this approach can be found in various literature sources (Ong, 1995), (Saelens, 2002), (Bazilian, Prasad, 2002).

Several approaches exist for modeling the AFW-IP in two dimensions. One way is to do a series of one-dimensional models. Ong (1995) suggested that a one-dimensional model be used for every meter of collector. The output air temperature in one model is used as the input in the next. This approach is similar to the model developed for this study; however, it lacks some of the complexity. The model developed in this study uses a radiosity analysis to calculate radiation exchange between all nodes. The basic approach is to divide the AFW into a series of slices along the

vertical plane. Then an energy balance is performed on each slice, using an iterative process, until the law of conservation of energy holds true for each slice. This method was used for Saelen's PhD thesis (Saelens, 2002), and is explained in more detail in the next chapter.

2.3 Computational Fluid Dynamics Models

In a computational fluid dynamics (CFD) model, the conservation equations for mass, thermal energy, and momentum are solved for all nodes in a two- or three-dimensional grid of the system under investigation. This provides more details on the nature of the flow field compared to network methods. One of the interesting aspects of CFD is that the same governing equations are used for all indoor applications of airflow and heat transfer, differing simply by changing the boundary conditions (Chen, Srebric, 2002).

In theory, the CFD approach is applicable to any thermo-fluid problem; however, when trying to model a building, problems may arise. Computing power, the complex nature of the flow fields, occupant-dependant boundary conditions, are some of the main reasons why CFD applications have been restricted to steady-state cases or very short simulation periods (Hensen, et al., 2002).

One promising use of CFD programs is to couple them with energy simulation (ES) programs. ES programs provide energy analyses for an entire building, providing space averaged indoor environmental conditions, cooling/heating loads, coil loads, and energy consumption. A drawback to these programs is that they cannot make detailed predictions of thermal comfort or predict the distributions of air velocity, and temperature

like a CFD program could (Zhai, et al., 2002). Convective heat transfer coefficients used in ES programs generally come from empirical formulas that may or may not be accurate. It would be beneficial to use a CFD program to calculate these coefficients in order to improve their accuracy. Boundary conditions are a requisite for CFD programs and may be obtained from an ES program, demonstrating one of the benefits of coupling ES and CFD programs together. Coupling the two programs would allow both to acquire the necessary information in an iterative process.

In addition to calculating proper convective heat transfer coefficients, a CFD program could be used to calculate the friction coefficients needed in ES modeling. Empirical formulas for friction factors are often derived for forced convection, and are different in the case of natural convection (Hensen, et al., 2002).

There are different methods of doing CFD modeling, such as using a finite-volume, finite-element, or control-volume approach. The method found most often in the literature search was control-volume modeling; however this does not mean that control-volume modeling is the best method as the literature search was not done with the intention of learning about CFD modeling. The type of CFD modeling performed is highly dependant on the available computational power. As is well known, computational power increases from year to year, which ensures a continual improvement of CFD modeling. For example, a paper published in 1996 presented an FEM model using 65 x 181 nodes, which took 800 to 1,100 iterations, running for approximately 14 hours (Moshfegh, Sandberg, 1996). At the time, they found that particular amount of nodes sufficient to model the system. If they were to run the same simulation today, they

could afford to increase the complexity and still have significantly lower computational costs.

CFD models can be divided into micro (direct numerical simulation, DMS) and macro analysis. Micro analyses have very small elements that take into account all of the flow's behavior, including turbulence. These models require an enormous amount of computational power because to define a whole system, millions of nodes are required. Small aspects of the system, such as the inlet could be modeled using micro-analyses to account for a specific phenomenon, such as inlet effects. When a macro-analysis is used, a turbulence model is needed to represent the overall effect of turbulence. The used most often in the models was the low Reynolds number k-e model (Zollner, et al., 2002), (Jaros, et al., 2002), (Chen, Srebric, 2002). To provide a more in-depth study of CFD models is beyond the scope of this thesis. Modeling the AFW-IP with a combination of an ES program and CFD model is something that should be looked into in the future, especially with the ever increasing availability computational power.

2.4 Convective Heat Transfer Coefficients

The most critical factor in generating an ES model is determining the proper convective heat transfer coefficients (CHTC) to use in the cavity. The coefficients used in energy simulation programs are usually determined empirically for a given experimental setup and may not be accurate for use in the ventilated façade (Zhai, et al., 2002). In evaluating different empirical relations for determining the Nusselt number, which is directly proportional to the CHTC, Ong (1995) found that the results could vary by as much as ±50%. When comparing his predicted results with experimental results,

Ong (1995) found that he had to multiply his model's calculated CHTCs by a factor of 0.75 to have a better correlation between the two. Evaluation of the CHTC will lead to uncertainties (Costa, et al., 2000).

One aspect in calculating the CHTC that varies from one researcher to another is whether or not to use the flat plate approximation for the flow. If the AFW has a wide cavity, then it may be possible that the fluid-flow boundaries do not coalesce into one flow as is depicted in Figure 2.1b. On the other hand, if the cavity is narrow or sufficiently high, then the flow cannot be treated entirely as two separate flows. Instead, coefficients can be calculated for the entrance region using plate flow, and then using channel flow for the remaining length of flow travel. This method provides the program with local CHTC until the flow coalesces and only one value for the remaining length. An alternative is to use a single CHTC for the whole length of flow that takes into account the entrance region of the channel.

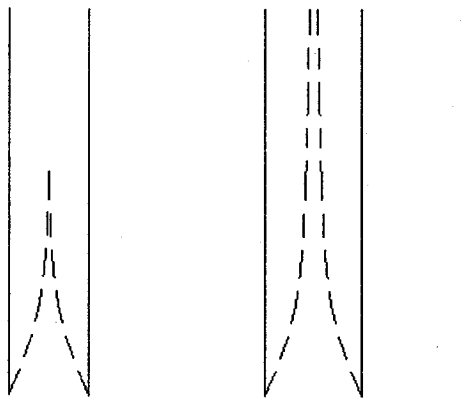

Figure 2.1: Depiction of Boundary Layer Formations in a Cavity

There is no single accepted combination of channel height and width that is agreed upon to validate the use of the separate plate approximation. On the contrary,

literature sources seem to give conflicting results. Mei L. et al. (2003) studied the flow at the Mataro public library in Barcelona and found that using separate flow approximation on a 0.14 m wide cavity, which was over 2.5 m high, was justified, since their model and experimental results correlated well. On the other hand, Zollner et al. (2002) found that channel flow should be used for gap widths of up to 0.60 m, and only when the cavity was wider than 0.60 m should plate flow be considered. Rodrigues et al. (2000) found that for a channel height of 2.5 m, that a channel width of 0.30 m could be approximated using plate flow. Saelens (2002) indicated that for the case of a one-storey high façade that the cavity could be modeled as separate plates. Other models found in literature use separate plate models, or not, without justification.

Another grey area in evaluating the CHTC is the determination of the driving mechanism generating the flow. As discussed in the next chapter, it is important to determine if the flow falls within forced, free, or mixed convection regimes. The following ratio of the Grashofs number and Reynolds number is used to determine just that (Rolle, 2000), (Incropera, De Witt, 1990), (Saelens, 2002):

$Gr << Re^2$ forced convection (2.1a)

$Gr >> Re^2$ free convection (2.1b)

$Gr \sim Re^2$ mixed convection (2.1c)

Discrepancies occur when researchers set values that separate the different flow regimes. Rolle (2000) is the only source which was found to give an actual value, and he stated that it should only be used as an initial approximation. His approximation to go from mixed convection to natural convection is given to be when the ratio of the Grashof number over the square of the Reynolds number is more than 4, whereas Saelens (2002)

still uses mixed convection with a ratio of 100. The lack of data in this area will generate uncertainty in the model developed for this thesis. However, Brinkworth et al. (2002) found that textbook parameters derived for forced convection could be used in buoyancy driven flows, implying that it was not too critical to determine the exact driving mechanism.

Outdoor heat transfer coefficients are also a source of uncertainty. Saelens (2002) found various studies examining the outdoor heat transfer coefficients due to wind, and found that their results varied significantly. He suggests that different building heights and geometries, combined with the different techniques of measuring wind speed account for the differences. He notes that the heat transfer coefficients will vary considerably for the different sides of the building, being higher for windward, and lower for leeward sides. At a wind speed of 10 m/s, the results varied by as much as 15 W/m^2K between windward and leeward orientations. Using ambient conditions to measure the outdoor heat transfer coefficient will generate some error as micro-climates exist near buildings (Hensen, et al., 2002). Values used for the outdoor heat transfer coefficient in literature varied from 12.9 W/m^2K to 28 W/m^2K.

2.5 Inlet/Outlet Effects

One aspect of modeling AFW that needs to be researched is the influence of the inlet and outlet regions. There are two factors that make these regions a challenge to model correctly. The flow conditions at the inlet and outlet are hard to specify (Brinkworth, et al., 2000), and the disturbance in the flow in these localized zones needs to be examined as they affect the global performance of the system (Rodrigues, et al.,

2000). The shape, roughness, and design of the inlet/outlet will all have an effect on the flow. The influence of the height of the inlet/outlet relative to the gap width is important. Zollner et al. (2002), found that the overall effect of this parameter was negligible for small gap widths, but could have a significant impact when the cavity was wider then 0.6 m.

Another factor that generates errors in the inlet region is the assumption that the air temperature going into the system is the same as the outdoor air temperature. Some investigations have shown that the air is heated while going through the inlet, and that it is difficult to accurately account for this in the model (Saelens, 2002), (Costa, et al., 2000). Preliminary results at a Concordia test room have found that the roof could increase the incoming air temperature by 1 to 3°C, during the Fall, with the effect diminishing to less than 1°C during cold weather.

2.6 Air Leakage

An assumption that is often made when modeling ventilated facades is the assumption that the cavity is completely sealed. In actuality, there will usually be some leakage that occurs in the façade. Only Saelens (2002), which did an extensive review of both experimental and theoretical work, mentions the importance of considering air leakage in a model. Leakage can alter the flow velocity, and the total energy balance. Unfortunately, there is no simple way to account for air leakage in the system. More experimental work is needed in this area to find a way to accurately predict its impact on the system. Air leakage will therefore not be included in the models for this thesis.

Air leakage can also come into play within the system. The vision section of the model assumes that the blind separates the top cavity into two separate cavities; however, the blind needs to be dense enough to prevent inter-cavity flow in order to make this assumption valid (Saelens, 2002). Not only does it have to be dense enough, but it also has to take up the whole space, not forming any gaps between the two cavities. If the cavity is wide enough, and the gaps at the edges narrow enough, then the limited amount of air flow between cavities in this area can be lumped in with other edge effects that are often ignored. The models developed in this thesis will assume that the blind is of sufficient density and wide enough so that no significant leakage occurs.

2.7 Control Strategies and Flow Velocity

As with most parameters that vary from one system to the next, air flow velocities, and mass flow rate by default, vary widely from one system to the next. For a set cross-sectional area, increasing the velocity will increase the mass-flow rate and vice-versa. Higher flow velocities lead to higher convective heat transfer coefficients, which lead to higher recovered heat, and hence, higher efficiencies. When keeping velocity constant, but increasing mass-flow rate (by changing area), more air is available to transport energy, and therefore higher efficiencies are reached but the temperature rise of the air is lower. This phenomenon of achieving higher efficiencies with higher flow rates is a concept that has been around for a long time when dealing with solar air collectors (Ong, 1995). However, as outlined below and later in later chapters, running an AFW-IP with too high velocities can actually increase a building's energy consumption.

In systems relying on thermal buoyancy effects to drive the flow, there are limits to the control of the flow rate. After a system is constructed, the only method of controlling the flow rate is by adjusting inlet and outlet dampers. At this stage, not enough experimental work has been done to accurately predict the flow rate in thermal buoyancy driven systems by means of controlling dampers. The control may be achieved using a feedback mechanism by means of a flow meter. A system can be designed in order to maximize flow rates achieved by thermal buoyancy effects; this can be done by selecting an appropriate gap width. For systems that rely on thermal buoyancy, gap widths of 0.20 to 0.30 m were found to maximize the buoyancy effects in solar chimneys of 6 and 14 m high, over a range of outdoor temperatures (Balocco, 2002).

Solar collectors that use a liquid, as opposed to air, to capture heat, control the rate of flow by means of temperature measurements. If the liquid heats up to a certain setpoint, then the pump starts operating until another setpoint (lower than the first) is reached, at which point the pump stops until more heat is stored. One paper found in literature followed the same principle with the AFW (Zollner, et al., 2002). The flow rate of the façade was controlled to maintain a temperature differential between the inlet and the outlet of 2°C since this was determined by some set of parameters to be the optimal operating conditions. This particular paper presented results from a test room that had variable gap width, height, window width, and inlet/outlet dimensions. The façade also had a range of 0.06 to 0.6 kg/s mass flow rate controlled by a fan at the exit. It would be interesting to try to evaluate if the optimum operating condition is in fact dependent on the temperature differential. As the scope of this thesis is limited, this will not be investigated further at this stage.

One method of determining a setpoint for the flow rate in an AFW-IP that is connected to a building's HVAC system to maximize performance is to set the flow rate going through the entire façade equal to the building's fresh air requirements (Saelens, 2002), (Durisch, et al., 2001). If the flow rate exceeds this amount, it is quite possible that extra heating would be required as the temperature of the air exiting the cavity might not be hot enough to meet the heating requirements of the building. In that case, the balance of the building's air requirements would be best served by using recycled air. Since the fresh air requirements need to be met, there would be no purpose in drawing air from anywhere but the AFW-IP as this would maximize the pre-heating effect of the system. The only reason to draw air elsewhere would be to lower flow velocities due to noise constraints. Note that this control strategy would only be useful during the heating season, since in the summer it would result in elevated cooling loads. In the cooling season, the air should be exhausted outside as this would help to reduce the cooling load of the building. More work is needed to find an optimum strategy for the period between the heating and cooling seasons.

There are other control areas of the AFW-IP that are often not examined. If there is a motorized blind in the cavity, a control strategy is required. If the blind was always down, it would lower the transmitted daylight during cloudy periods. If the blind was left down overnight, then the system would have a lower U-value. Saelens (2002) suggested that the blind be lowered if a certain setpoint of solar radiation is reached; however, he suggested two different setpoints (75 and 150 W/m^2) in his doctoral thesis without giving an explanation.

Another point that should be mentioned is concerning the flow velocity in dual cavity systems. Without proper design, the flow velocity in the two cavities will differ (Saelens, 2002). There may be advantages and disadvantages of having different velocities in the front and back cavities. The important thing to be aware when modeling the AFW and during experimental investigations is that a difference exists. As with other areas, determining optimal balance in flow rate in both ducts of a dual-duct system is beyond the scope of this thesis.

2.8 Modeling Photovoltaic Cells

The efficiency at which a PV cell converts solar radiation to electricity is not constant. The efficiency of crystalline cells depends on the amount of solar radiation incident on the cell, the PV cell temperature, and the light spectrum. The efficiency of a crystalline cell is essentially zero at an irradiation level of 50 W/m^2 (Poissant, et al., 2003). The efficiency of amorphous-silicone cells is not directly dependant on irradiation levels; however, its temperature coefficient does vary with irradiation.

Reliable knowledge of the different climactic influences on PV efficiency and power output is indispensable for the accurate prediction of daily, weekly, monthly and yearly electricity yields (Durisch, et al., 200). The effect of operating temperature can either be reported in percent reduction in generated power per degree Celsius, or a percent reduction of the overall cell efficiency per degree. For crystalline technologies, the effect of temperature on power generation is 0.2 to 0.5%/°C (Sandberg, 1999). Unless otherwise stated, the only effect on PV efficiency that will be considered in this

study will be the temperature coefficient suggested by Sandberg (1999) with a coefficient of 0.038 as follows:

$$\eta_{PV} = 13\% - 0.038\% \cdot (Tpv - 25°C)/\,°C$$

2.9 Modeling Solar Radiation

When reviewing literature on AFW, very few sources discussed how they would model solar radiation. There are two important aspects of modeling solar radiation. The first is to be able to accurately predict where the sun is situated in the sky relative to the façade. The angle of incidence of the solar radiation affects the optical properties of the glazings. It is important to include this dependence on angle of incidence on the transmittance and reflectivity for each glazing in the system (Saelens, 2002). For example the transmittance of 3 mm clear glass is approximately 89% from 0° to 35° incidence, but drops considerably at higher angles; the value is below 50% at 75°. The models developed in this study take angle of incidence into account when they are used to simulate a specific time and place. However, when general simulations are done, average transmittance and reflectivity is used.

The second important aspect of modeling solar radiation is the ability to predict the quantity of irradiation expected on a surface based on time and location. This can be done by taking data from weather databases, or by using a model. Unless otherwise stated, this study will use the Kt clearness index model to estimate diffuse and beam solar radiation (Orgill, Hollands, 1977). One of the limitations mentioned in using this model was computational costs; however, as this paper was written in 1977, this limitation no longer applies today. Note that the model is developed for latitudes of 43 to 54°N. This

model is suitable for predictions in Montreal. It would be of interest, in the future, to find other models that set no limitations on latitudes.

A lot of data and models (including the Kt clearness index model) give insolation values for a horizontal surface. In this thesis, Klucher's method (1978) is used to predict the insolation on a tilted surface based on data from a horizontal surface. This model is based on an anisotropic all sky model. This model was selected based on research indicating that old isotropic sky models are deficient in predicting insolation on titled surfaces for non-uniform (non-cloudy) sky conditions, and the other widely used Temps & Coulson's model was not as accurate in non-clear days (Klucher, 1979).

2.10 Example of Application

As mentioned, building integrated photovoltaics (BIPV) and double-façades are starting to gain popularity. However, there are still very few projects around the world that have PV directly integrated into a double-façade to create a BIPV. One project that is a good example of this technology is found at the Mataro public library in Barcelona, Spain. The building is rectangular with its main 225 m^2 façade facing south comprising of a 6.5 m high PV section. Figures 2.2a and 2.2b are pictures of the library's double-façade viewed from outside and inside the building.

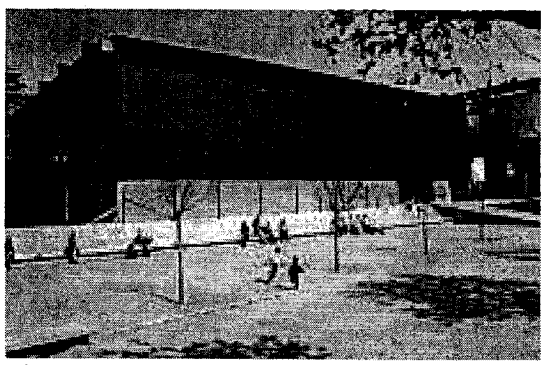

Figure 2.2a: Mataro Public Library Viewed from Outside (Infield D., et al., 1999)

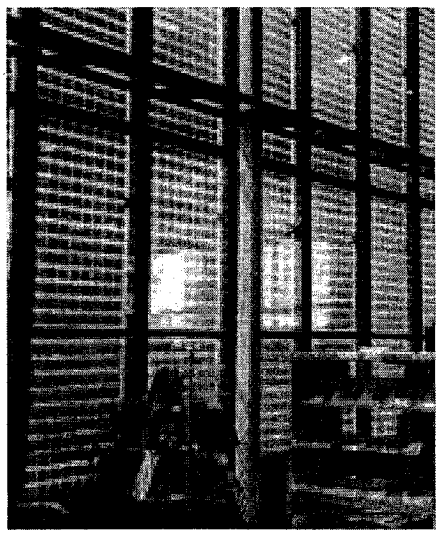

Figure 2.2b: Mataro Public Library Double-Façade Viewed from Inside (Infield D., et al., 1999)

The façade is composed of a PV panel and a double-glazed window separated by a 0.14 m gap. The PV panel, which faces the outside of the building, has blue polycrystalline silicon solar cells sandwiched between the glazings. The square PV cells are placed such that they leave a 0.014 m horizontal gap, allowing for daylight to enter the building as seen in Figure 2.2b. This arrangement allows 15% of the incident short-wave irradiation to pass directly through the PV panel and the double-glazed window (ignoring absorption in the glass) for daylighting (Mei, et al., 2003).

Various authors have modeled the ventilated PV façade by evaluation of energy inputs and outputs through radiation, convection, conduction and power generation (Mei, et al., 2003). Mei, et al., (1999, 2003) have studied the Mataro library using both a simplified steady state analysis (Infield, et al., 1999) and a fully dynamic thermal model for the ventilated PV façade (Mei, et al., 2003). As this design has been examined in detail, results taken from the Mataro Library will be used in Chapter 4 as part of the validation of the two models developed for this study.

CHAPTER 3

MODEL DESCRIPTION

3.1 One-Dimensional Model

As the AFW is divided into two sections, the mathematical formulation of the problem considers the two sections separately. The two are linked such that the output air temperature calculated for the PV section is used as an input value to the Vision section. A nodal model is used to determine an expression for the temperature of each component. This is done by doing an energy balance at each non-air element. For the simple case of the PV section depicted in Configuration 1 of Figure 1.3, equations 3.1 to 3.3 are used. The following major assumptions were used for the 1D model:

- One dimensional heat transfer
- Steady state system, i.e. thermal capacitance effects neglected
- Isothermal surfaces
- Use of linearised radiative heat transfer coefficients to calculate the long-wave radiation heat transfer
- Neglect end and edge effects of surfaces and air cavity
- PV efficiency a function of temperature only.

PV element

$$U_o(T_{pv}-T_o) + S_{pv.heat} = U_{h1}(T_{air} - T_{pv}) + U_{rad}(T_{wall,i} - T_{pv}) \quad (3.1a)$$

$$T_{pv} = \frac{U_o T_o + U_{h1} T_{air} + U_{rad} T_{wall,i} + S_{pv.heat}}{U_o + U_{h1} + U_{rad}} \quad (3.1b)$$

Interior wall element

$$U_{h2}(T_{wall,i} - T_{air}) + U_{rad}(T_{wall,i} - T_{pv}) = U_{wall}(T_{wall,o} - T_{wall,i}) \quad (3.2a)$$

$$T_{wall,i} = \frac{U_{rad}T_{pv} + U_{h2}T_{air} + U_{wall}T_{wall,o}}{U_{rad} + U_{h2} + U_{wall}} \quad (3.2b)$$

Exterior wall element

$$U_{wall}(T_{wall,o} - T_{wall,i}) = U_{room}(T_{room} - T_{wall,o}) \quad (3.3a)$$

$$T_{wall,o} = \frac{U_{wall}T_{wall,i} + U_{room}T_R}{U_{wall} + U_{room}} \quad (3.3b)$$

Theoretical studies have demonstrated that the temperature profile in the ventilated cavity is exponential (Saelens, 2002). In the model, the change in enthalpy of air in the control volume is equal to the energy transferred to the air by convection, see Equation (3.4a). Solving the resulting differential equation (3.4b) for air temperature gives an exponential profile, as expected. This approach eliminates the need to assume an air temperature profile, as other models have done in the past.

The energy balance on the air:

$$M \cdot \rho \cdot c \cdot dT_{air} = [h_1(T_{pv} - T_{air}) + h_2(T_{wall,i} - T_{air})]w \cdot dx\,, \quad (3.4a)$$

$$\frac{M \cdot c \cdot \rho}{w} \cdot \frac{dT_{air}}{dx} + (h_1 + h_2) \cdot T_{air} = h_1 \cdot T_{pv} + h_2 \cdot T_{wall,i}\,. \quad (3.4b)$$

Equations (3.1b-3.3b) are re-arranged to express the component temperatures in terms of air temperatures. The result for the k^{th} component is shown in equation (3.5), where c_1 and c_2 are constants.

$$T_k(x) = c_1 T_{air}(x) + c_2 \quad (3.5)$$

These equations are then substituted back into equation (3.4) to obtain a first order differential equation of $T_{air}(x)$, which is differentiated to obtain an expression for the air temperature, as shown in equation (3.6), where c_3, and c_4 are constants, and $a = M \cdot c \cdot \rho / w$.

$$T_{air}(x) = \left(T_o - \frac{c_3}{c_4} \right) \cdot e^{\frac{-x \cdot c_4}{a}} + \frac{c_3}{c_4} \tag{3.6}$$

The mean air temperature is then calculated using equation (3.7).

$$\overline{T}_{ma} = \frac{1}{H_{pv}} \cdot \int_0^{H_{pv}} T_{air}(x) \cdot dx \tag{3.7}$$

The mean air temperature is then substituted into the equations of the component temperatures expressed as in equation (3.5) to obtain the mean temperature of the components, equation (3.8). The mean component and air temperatures are then used to find the heat transfer coefficients and PV cell efficiency for the next iteration. The procedure is repeated until convergence is reached, which usually only takes a couple of iterations.

$$\overline{T}_k = c_1 \overline{T}_{ma} + c_2 \tag{3.8}$$

For configurations that have two air cavities, a set of coupled differential equations is found for the air temperatures in the front and back cavities. The Runge-Kutta method for systems of differential equations is used to solve for the mean air temperature in these cavities (Burden, Faires, 1997). FORTRAN program Yearly.F90, and its main subroutine 1D_2PV_2W.F90, which are part of the 1D model of Configuration 2, are listed in Appendices D and E, respectively.

3.2 Two-Dimensional Model

The two-dimensional model separates each component into elements of equal thickness. The model then calculates the temperature of each element, allowing for the temperature profile of each component to be predicted. Figure 3.1 shows how the components in the Vision section are subdivided in the x-direction. FORTRAN program Model.F90, and its main subroutine funcv.F90, which are part of the 2D model of Configuration 1, are listed in Appendices F and G, respectively.

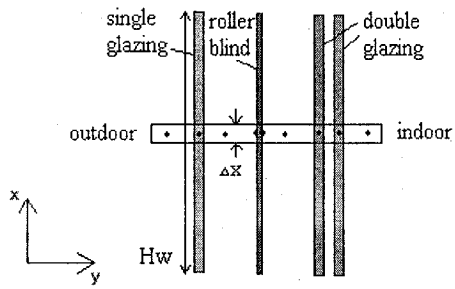

Figure 3.1: Vision Section Control Volume in Two-Dimensional Model

3.2.1 Indoor and Outdoor Heat Transfer

As in the case for the one-dimensional model, the heat transfer to the exterior and interior surroundings is calculated using a single heat transfer coefficient, which includes both long-wave radiation and convection.

$$Q_{o,k} = h_o \cdot (T_k - T_o) \cdot \Delta x \quad (W/m) \tag{3.9}$$

$$Q_{room,k} = h_{room} \cdot (T_{k2} - T_{room}) \cdot \Delta x \quad (W/m) \tag{3.10}$$

3.2.2 Conductive Heat Transfer within the Layers

In the one-dimensional model, conduction in the x-direction is not considered, since the temperature is assumed to be constant along that direction, except for in the air cavities, which are not affected by conduction. Conduction in the y-direction is considered only when nodes are separated by a solid material. In the two-dimensional model, conduction in the x- and y-direction is taken into account. Every component has conduction in the x-direction. However, in the spandrel portion of the PV section, the material is considered to be a good insulator, so heat conduction in the vertical is ignored.

Conduction into each element from the x-direction is calculated using equation (3.11).

$$Q_{x-cond,k} = k \cdot A \frac{\partial T}{\partial x} \approx \frac{k \cdot t \cdot 1m}{\Delta x} \cdot ((T_k - T_{k-1}) + (T_k - T_{k+1})) = \frac{k \cdot t}{\Delta x} \cdot (2 \cdot T_k - T_{k-1} - T_{k+1}) \qquad (3.11)$$

where k is the thermal conductivity of the element, and t is the element thickness. No conductive heat transfer is assumed to occur through the component boundaries.

For the conduction in the y-direction, the calculation is similar to what was given for the one-dimensional case. Equation (3.12) is an example of conduction heat transfer between two adjacent elements, k and j, with nodes centered in the element.

$$Q_{y-cond,k} = \left[\frac{0.5 \cdot t_k}{k_k \cdot \Delta x} + \frac{0.5 \cdot t_j}{k_j \cdot \Delta x} \right]^{-1} \cdot (T_k - T_j) \quad (W/m) \qquad (3.12)$$

3.2.3 Long Wave Radiation between Elements

In the two dimensional model, no linearised coefficients are used to calculate the radiation exchange between surfaces. Instead, a radiosity analysis is performed to directly

compute the radiation exchange. The details of this analysis are covered later in this chapter. The important thing to point out at this stage is the equation to calculate the amount of absorbed longwave solar radiation in each element. This is simply the difference between the incoming and the outgoing radiation as seen in equation (3.13).

$$Q_{lw-rad,k} = Q_{lw-rad,out,k} - Q_{lw-rad,in,k} \tag{3.13}$$

3.2.4 Heat Transfer to the Air

The heat transferred to the air in the cavity elements occurs through convection, and advection. The airflow rate is assumed to be large enough that conduction can be neglected (Saelens, 2002). The enthalpy gained in the cavity elements is obtained using equation (3.14).

$$Q_{enth,k} = M_{air} c_{air} \rho_{air} \cdot (T_k - T_{k-1}) \tag{3.14}$$

where ρ_{air} represents the air density (kg/m^3), c_{air} is the specific heat capacity of air (J/kg·K), and M_{air} is the airflow rate of air (m^3/s).

No air is assumed to flow from the back cavity to the front cavity through the blind. This assumption was validated in (Saelens, 2002), provided that the shading device is dense enough. The use of Venetian blinds would not meet this requirement.

3.2.5 Convective Heat Transfer within the Cavities

In the two-dimensional model, local as opposed to average convective heat transfer coefficients are used to calculate the convective heat transfer. The expressions used to calculate these coefficients are outlined in detail in Section 3.3.2. Once the proper coefficient is calculated, the convective heat transfer between an element k, and

the adjoining air cavity element *j*, is calculated using equation (3.15). FORTRAN subroutine HTcoef.F90, which calculates the convective heat transfer coefficient used in the 2D model, is listed in Appendix A.

$$Q_{conv,k} = h_{conv,k} \cdot (T_k - T_j) \cdot \Delta x \quad (W/m) \tag{3.15}$$

3.2.6 Heat Transfer between Panes of Double Glazing

Heat transfer between the two panes in the double glazed window occurs through convection and radiation. The convective heat transfer coefficient between the glazings is assumed constant at 2 $W/m^2 \cdot K$, as suggested by Athienitis (1998). The radiosity analysis is not used for this part of the model; instead, a linearised radiative heat transfer coefficient is used, see equation (3.48).

3.2.7 Defining the System of Equations

In order to determine the temperature of each element, a coupled, non-linear system of equations is defined. Each element within the system generates one equation. As there is one equation per element, the system is solved to find each elemental temperature. Each equation is simple to define; it is the sum of all input heat transfer minus the sum of all output heat transfer acting on the element, which is equal to zero by the conservation of energy, see equation (3.16).

$$F_k = \sum Q_{input,k} - \sum Q_{output,k} = 0 \tag{3.16}$$

The algorithm used to solve the nonlinear system of equations is based on a technique known as Broyden's method. It is a quasi-Newton method that replaces the Jacobian matrix of Newton's method with an approximation that is updated at each

iteration, reducing the number of arithmetic calculations from $O(n^3)$ to $O(n^2)$ (Burden, Faires, 1997); however, the convergence is reduced from quadratic to superlinear.

An initial approximation of $T^{(0)}$ for the system $F(T)=0$ is required to start the solver. The next approximation is found using Newton's method by finding the Jacobian $J(T^{(0)})$ using a finite-difference approximation for the derivatives as:

$$\frac{\partial f_j}{\partial f_k}(T^{(i)}) \approx \frac{f_j(T^{(i)} + e_k\Delta) - f_j(T^{(i)})}{\Delta}, \tag{3.17}$$

where Δ is small in absolute value and e_k is the vector whose only nonzero entry is a one in the k^{th} coordinate.

Broyden's method differs from Newton's method in the computation of all of the terms following. The matrix $J(T^{(1)})$ is replaced by matrix A_1 with the property set in equation (3.18).

$$A_1(T^{(1)} - T^{(0)}) = F(T^{(1)}) - F(T^{(0)}) \tag{3.18}$$

To uniquely define the matrix A_1, the manner in which it acts on the orthogonal complement needs to be specified using the following constraint:

$$A_1 z = J(T^{(0)})z, \textit{ whenever } (T^{(1)} - T^{(0)})^t z = 0. \tag{3.19}$$

Following these two conditions, A_1 can be found as:

$$A_1 = J(T^{(0)}) + \frac{[F(T^{(1)}) - F(T^{(0)}) - J(T^{(0)})(T^{(1)} - T^{(0)})](T^{(1)} - T^{(0)})^t}{\left\|T^{(1)} - T^{(0)}\right\|_2^2}. \tag{3.20}$$

The term $T^{(2)}$ is then computed as:

$$T^{(2)} = T^{(1)} - A_1^{-1}F(T^{(1)}). \tag{3.21}$$

The general equation for obtaining the A matrix for subsequent iterations is as follows:

$$A_i = A_{i-1} + \frac{y_i - A_{i-1}s_i}{\|s_i\|_2^2} s_i^t, \tag{3.22}$$

where the notation $y_i=F(T^{(i)})-F(T^{(i-1)})$ and $s_i=T^{(i)}-T^{(i-1)}$.

A key improvement to this method is obtained by using the matrix inversion formula of Sherman and Morrison (Burden, Faires, 1997). Their result stipulates that if A is a nonsingular matrix and if $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors, then $A + \boldsymbol{xy}^t$ is nonsingular, provided that equations (3.23) and (3.24) hold.

$$\boldsymbol{y}^t A^{-1}\boldsymbol{x} \neq -1 \tag{3.23}$$

$$(A + xy^t)^{-1} = A^{-1} - \frac{A^{-1}xy^t A^{-1}}{1 + y^t A^{-1}x} \tag{3.24}$$

This formula can be applied to determine the inverse of A:

$$A_i^{-1} = A_{i-1}^{-1} + \frac{(s_i - A_{i-1}^{-1}y_i)s_i^t A_{i-1}^{-1}}{s_i^t A_{i-1}^{-1} y_i}. \tag{3.25}$$

In the code prepared for modeling of the airflow window, a numerical recipe provided by Microsoft Fortran PowerStation was used to implement Broyden's method.

3.3 Convective Heat Transfer Coefficients

The AFW generally deals with low temperature ranges; therefore, the physical properties of air were assumed to vary linearly with temperature (Ong, 1995), unless otherwise stated:

Specific heat:

$$c = 1000 \cdot [1.0057 + 0.000066 \cdot (T - 300.15)] \cdot \frac{J}{kg \cdot K} \tag{3.26}$$

Density:

$$\rho = [1.1774 - 0.00359 \cdot (T - 300.15)] \cdot \frac{kg}{m^3} \tag{3.27}$$

Thermal Conductivity:

$$k = [0.02624 + 0.0000758 \cdot (T - 300.15)] \cdot \frac{W}{m \cdot K} \tag{3.28}$$

Viscosity:

$$\mu = [0.1983 + 0.00184 \cdot (T - 300.15)] \cdot 10^{-5} \cdot \frac{kg}{m \cdot s} \tag{3.29}$$

The coefficient of volumetric thermal expansion (assuming perfect gas):

$$\beta = \frac{1}{T} \tag{3.30}$$

Kinematic viscosity:

$$\nu = \frac{\mu}{\rho} \tag{3.31}$$

Thermal diffusivity:

$$\alpha = \frac{k}{\rho \cdot c} \tag{3.32}$$

Various dimensionless numbers are used in the course of this investigation. The Nusselt number (*Nu*) defines the ratio of heat transfer that occurs between convection and pure conduction. Empirical formulas to calculate convective heat transfer are often given in terms of *Nu*:

$$Nu_l = \frac{h \cdot l_c}{k_{air}}, \tag{3.33}$$

where h is the convective heat transfer coefficient, l_c is the characteristic length, and k is the fluid (air) thermal conductivity. The characteristic length for flow along a plate would be the plate length, whereas for internal flows, the hydraulic diameter (D_h) is used:

$$D_h = \frac{4A}{P} = \frac{4 \cdot (L \cdot w)}{2 \cdot (w + L)}, \tag{3.34}$$

where w is the length of the AFW, and L is the cavity width. Note that for parallel plates with $L<<w$, equation (3.34) can be approximated as $D_h=2L$.

The Reynolds number (Re) defines the ratio between the momentum due to eddy diffusion and molecular transport:

$$\mathrm{Re} = \frac{V \cdot l_c}{\nu}, \tag{3.35}$$

where V is the fluid velocity.

The Prandtl number (Pr) compares the momentum diffusivity with the thermal diffusivity and links the temperature to the velocity boundary layer.

$$\mathrm{Pr} = \frac{\nu}{\alpha} \tag{3.36}$$

When the flow is dominated by natural convection, the Grashof number (Gr) replaces Re:

$$Gr = \frac{\beta \cdot g \cdot l_c^3 \cdot \Delta T}{\nu^2}, \tag{3.37}$$

where g is the gravitational acceleration (9.81 m/s^2) and ΔT is the difference between the surface and fluid temperature. Another dimensionless number that will be used, which is directly related to Gr, is the Rayleigh number (Ra):

$$Ra = \mathrm{Pr} \cdot Gr. \tag{3.38}$$

3.3.1 Indoor and Outdoor Heat Transfer Coefficients

The convective and radiative heat transfer from the external surfaces of the airflow window, facing the inside of the room and the outdoor environment, has been

studied in more detail than the heat transfer inside the cavity. However, more study generates more empirical relations, and rarely are the same expressions used between researchers. In most cases, a constant value of the heat transfer coefficient is used for the indoor surface, and less often for the outdoor coefficients (Costa, et al., 2002), (Hensen, et al., 2002). Often, papers fail to mention what values were used for these heat transfer coefficients.

In some instances, the heat transfer coefficient is a combination of radiative and convective heat transfer coefficients, and in other cases they are kept separate. Athienitis (1998) suggested the following correlations:

$$h_o = 8.6 \frac{V_w^{0.6}}{L_c^{0.4}}, \tag{3.39}$$

$$h_i = 1.31 \cdot (T_s - T_{air})^{1/3}, \tag{3.40}$$

$$h_r = \varepsilon \cdot \sigma \cdot 4 \cdot T_m^3. \tag{3.41}$$

Equation (3.39) determines the heat transfer coefficient due to wind, stipulating a minimum value of 5 W/m^{2}K, where L_c is the characteristic length and V_w is the wind speed (m/s). Equation (3.40) determines the indoor heat transfer coefficient, where T_s is the average wall surface temperature. Equation (3.41) determines the radiative heat transfer coefficient, where ε is the surface emissivity, σ is the Stefan-Boltzman constant (σ= 5.67 x 10^{-8} W/m^{2}K^4), and T_m is the mean temperature between the surface and its surroundings in Kelvin. In order to have a combined heat transfer coefficient, the expressions for convective and radiative heat transfer are simply added together.

Mei, et al., (2003) used McAdam's formula to calculate the external combined (radiative-convective) heat transfer coefficient due to wind:

$$h_o = (5.7 + 3.8 \cdot V_w) \frac{W}{m^2 K}. \tag{3.42}$$

Balocco (2002) suggested that McAdam's formula be used for wind speeds greater than 4 m/s and that an expression for combined laminar and turbulent flows over a vertical wall at uniform temperature to be used for lower velocities:

$$h_o = Nu \cdot k_{air} / H, \tag{3.43}$$

where k_{air} is the air's thermal conductivity, H is the wall height, and the Nusselt number, Nu, is found as:

$$Nu = (Nu_1^3 + Nu_t^3)^{1/3}, \tag{3.44a}$$

where

$$Nu_1 = \frac{2}{\ln\left(1 + \frac{2}{C_1} Ra^{0.25}\right)}, \tag{3.44b}$$

$$Nu_t = \frac{C_t Ra^{1/3}}{1 + 1.4 \cdot 10^9 \, \mathrm{Pr} / Ra}, \tag{3.44c}$$

$$C_1 = \frac{0.671}{\left[1 + \left(\frac{0.492}{\mathrm{Pr}^{9/16}}\right)\right]^{4/9}}, \tag{3.44d}$$

$$C_t = 0.16 \cdot \left(\frac{\mathrm{Pr}^{0.22}}{1 + 0.61 \mathrm{Pr}^{0.81}}\right)^{0.42}. \tag{3.44e}$$

As for indoor heat transfer coefficients, Saelens (2002) suggested the following expression for a vertical wall without heating or cooling devices:

$$h_i = \left\{ \left[1.50 \cdot \left(\frac{T_s - T_{air}}{H} \right)^{0.25} \right]^6 + [1.23 \cdot (T_s - T_{air})^{0.33}]^6 \right\}^{1/6} . \quad (3.45)$$

If heating devices are near the wall, the flow can enter a mixed convection regime and the following expressions are suggested (Saelens, 2002):

Room heated with a radiator

$$h_i = 2.04(T_s - T_{air})^{0.23}, \quad (3.46)$$

Surfaces opposite a fan heater

$$h_i = 2.92(T_s - T_{air})^{0.25} . \quad (3.47)$$

It is clear that indoor and outdoor heat transfer coefficients are influenced by many factors. Expressions suggested by various authors have been shown to vary significantly (Saelens, 2002). The goal of this thesis is not to study in detail the effects that wind has on the outdoor heat transfer coefficient, or the effects that heating/cooling devices may have on the indoor heat transfer coefficient. Therefore, the same approach as used by Costa, et al. (2000), of using constant indoor and outdoor heat transfer coefficients that account for both radiation and convection will be used.

Equations 3.39 to 3.41 will be used to calculate the overall heat transfer coefficients. Unless otherwise stated, an average wind speed of 2 m/s, and characteristic length of 3 m will be used in (3.39). For equation 3.40, a temperature difference of 2°C between surface and room air temperature will be assumed. Finally, an emissivity of 0.9, with a mean surface temperature of 281K for the outside surface and 294K for the inside surface will be used. Using these values results in having h_o=12.9 W/m^{2}K and h_i=6.85 W/m^{2}K. Both these values fall within the range of values that have been found in

literature, which vary between 5 and 9.1 W/m^2K for indoor, and 12 and 29 W/m^2K for outdoor surfaces (Costa, et al., 2002), (Saelens, 2002), (Barletta, 2002).

3.3.2 Heat Transfer Coefficients within the Cavity

In the two-dimensional formulation of the system, heat transfer coefficients are not used to find the radiation heat transfer within components, instead, a radiosity analysis is used to directly compute it, which will be covered later in section 3.5. On the other hand, the one-dimensional formulation uses the following linearised coefficients to determine heat transferred through radiation:

$$h_{rad} = \frac{4\sigma \cdot T_m^3}{\frac{1}{\varepsilon_1} + \frac{1}{\varepsilon_2} - 1} \quad , \tag{3.48}$$

where $T_m = 0.5(T_1 + T_2)$.

Countless expressions are available for determining the heat transfer coefficient in a channel. The flow can be approximated as flowing in a circular duct, using an appropriate hydraulic diameter, in a rectangular channel, between parallel plates, and with sufficient spacing, as two separate flows going along vertical plates. The flow can be assumed to be driven by forced, natural, or mixed convection, and be either laminar, turbulent, or in transition, and be either developing or fully-developed, all of which have their own associated convective heat transfer coefficients.

Different researchers have suggested using the separate vertical plate approximation, provided there is sufficient spacing between the plates; Mei, et al., (2003) found that 0.14 m spacing was sufficient, whereas, Zollner, et al., (2002) stated that only

gaps wider than 0.60 m should use a plate model. Such large discrepancies suggest that further work in determining this criterion is needed.

The width at which a wide plate approximation is valid would not be a set width as the above mentioned authors have suggested. It would depend on factors such as channel height, air velocity, surface characteristics, etc., all of which would affect the boundary layer growth. To be accurate, using the separate plate analysis would depend on whether or not the boundary layers of the two walls coalesce (Saelens, 2002). There exist several expressions that can be used to determine if a flat plate approximation is appropriate in the case of natural convection, which take into account the following factors:

For vertical cylinders (Incropera, De Witt, 1990):

$$\frac{D}{H} \geq \frac{35}{Gr^{0.25}}, \tag{3.49}$$

For parallel plates (Incropera, De Witt, 1990):

$$Ra_L \cdot \frac{L}{H} \geq 100, \tag{3.50}$$

Channel flow (Saelens, 2002):

$$\frac{L}{H} \geq Ra_L^{-1} \quad or \quad \frac{L}{H} \geq Ra_H^{-0.25}. \tag{3.51}$$

The vertical plate approximation can be used in the entrance region for forced convection. This applies for $x<X$ (Saelens, 2002):

$$X/D_h \cong 0.05 \cdot \mathrm{Re}_{D_h}, \qquad (\mathrm{Re_D} < 2300) \tag{3.52}$$

$$X/D_h \cong 10. \qquad (\mathrm{Re_D} > 2300) \tag{3.53}$$

In the model's algorithm, equations (3.50, 3.52-3.53) are used to determine whether the vertical plate approximation can be used.

It is important to determine if the flow falls within forced, free or mixed convection regimes. Incropera, De Witt (1990), Rolle (2000), and Saelens (2002) give the criteria for determining this as follows:

$Gr << \mathrm{Re}^2$ forced convection (3.54a)

$Gr >> \mathrm{Re}^2$ free convection (3.54b)

$Gr \sim \mathrm{Re}^2$ mixed convection. (3.54c)

The criteria given in equations (3.54a,b,c) are somewhat vague. To implement these requirements in the model algorithm, concrete values are needed. Only in Rolle (2000) are values given as a first approximation to what the boundaries should be:

$Gr/\mathrm{Re}^2 > 4$ free convection (3.55a)

$Gr/\mathrm{Re}^2 < 0.25$ forced convection (3.55b)

$0.25 \leq Gr/\mathrm{Re}^2 \leq 4$ mixed convection. (3.55c)

In the algorithm, equations (3.55a,b,c) are used to determine which convective regime is appropriate. Note that these values are used as first approximations and should be verified through experimental testing. Although no specific values are given, Saelens (2002) does state at one point in his doctoral thesis that a ratio of Gr/Re^2 in the range of 100 was still considered in the mixed regime.

3.3.2.1 Equations for Forced Convection

The following expressions are for laminar ($Re < 5 \cdot 10^5$) plate flow, with low solar irradiance ($G < 200$ W/m^2) values valid for all Prandtl numbers associated with using an AFW (Incropera, De Witt, 1990), (Saelens, 2002):

$$h = \frac{k_{air}}{H} \cdot 0.664 \cdot \mathrm{Re}_H^{0.5}\, \mathrm{Pr}^{1/3}, \tag{3.56a}$$

$$h(x) = \frac{k_{air}}{x} \cdot 0.332 \cdot \mathrm{Re}_x^{0.5}\, \mathrm{Pr}^{1/3}. \tag{3.56b}$$

Laminar plate flow with high irradiance values (Incropera, De Witt, 1990):

$$h = \frac{k_{air}}{H} \cdot 0.906 \cdot \mathrm{Re}_H^{0.5}\, \mathrm{Pr}^{1/3}, \tag{3.57a}$$

$$h = \frac{k_{air}}{H} \cdot 0.453 \cdot \mathrm{Re}_x^{0.5}\, \mathrm{Pr}^{1/3}. \tag{3.57b}$$

Combined laminar and turbulent plate flow (Saelens, 2002):

$$h = \frac{k_{air}}{H} 0.664 \mathrm{Pr}^{\frac{1}{3}}\, \mathrm{Re}_{x,tr}^{0.5} + 0.0296 \mathrm{Pr}^{\frac{1}{3}} \cdot \left(\mathrm{Re}_H^{\frac{4}{5}} - \mathrm{Re}_{x,tr}^{\frac{4}{5}} \right). \tag{3.58}$$

For turbulent flows (Incropera, De Witt, 1990):

$$h_{turb}(x) = \frac{k_{air}}{x} \cdot 0.0296 \cdot \mathrm{Re}^{0.8} \cdot \mathrm{Pr}^{1/3}. \tag{3.59}$$

Note that for the one-dimensional model, there is no expression for purely turbulent flow; at the beginning of the channel, the flow will generally be laminar and then may progress to turbulent. If the entrance region is not smooth, the flow may start as turbulent. Equation (3.58) is used when turbulent flow is present since it averages both the laminar and turbulent sections. The two-dimensional model relies on local convective heat

transfer coefficients, so there is no expression for combined laminar and turbulent plate flow. Equations (3.56b or 3.57b) are used when $Re < 5 \cdot 10^5$ and equation (3.59) is used for $Re > 5 \cdot 10^5$.

For forced convection in a tube, the same expression for both the one- and two-dimensional model is used, due to the fact that the convective heat transfer coefficient is constant past the entrance region (Gnielinski, 1983):

$$h = \frac{k_{air}}{D_h} \cdot \left(\frac{\left(\frac{f}{8} \left(\mathrm{Re}_{D_h} - 1000 \right) \mathrm{Pr} \right)}{1 + 12.7 \sqrt{\frac{f}{8}} \left(\mathrm{Pr}^{\frac{2}{3}} - 1 \right)} \right) \left(1 + \left(\frac{D_h}{H} \right)^{\frac{2}{3}} \right). \tag{3.60}$$

Equation (3.60), where the friction factor, f, is calculated as follows, is valid when $2300 < Re < 10^6$, $0 < D/H < 1$, and $0.6 < Pr < 2000$ (Gnielinski, 1983):

$$f = (1.82 \cdot \log_{10} \mathrm{Re} - 1.64)^{-2}. \tag{3.61}$$

3.3.2.2 Equations for Natural Convection

When the Rayleigh number for natural convection is lower than 10^9, the following laminar plate flow expressions are used (Incropera, De Witt, 1990):

$$h = \frac{k_{air}}{H} \frac{4}{3} \left(\frac{Gr_H}{4} \right)^{0.25} \cdot g(\mathrm{Pr}), \tag{3.62a}$$

$$h(x) = \frac{k_{air}}{x} \cdot \left(\frac{Gr_x}{4} \right)^{1/4} \cdot g(\mathrm{Pr}), \tag{3.62b}$$

where

$$g(\mathrm{Pr}) = \frac{0.75 \, \mathrm{Pr}^{0.5}}{\left(0.609 + 1.221 \, \mathrm{Pr}^{0.5} + 1.238 \, \mathrm{Pr} \right)^{0.25}}. \tag{3.62c}$$

If the Rayleigh number exceeds 10^9, the following turbulent plate flow expression is used (Burmeister, 1993):

$$h(x) = 0.02979 k_{air} \, \mathrm{Pr}^{\frac{1}{15}} \, x^{\frac{1}{5}} \frac{\left(\frac{\mathrm{Pr}}{\nu^2} \Delta T \cdot g \cdot \beta \right)^{\frac{2}{5}}}{(1 + 0.494 \, \mathrm{Pr}^{\frac{2}{3}})^{\frac{2}{5}}} \, . \tag{3.63}$$

If the flow becomes turbulent, as in the case of the one-dimensional model, an average value is taken between equations (3.62a and 3.63):

$$h = \frac{1}{x_{crit}} \cdot \int_0^{x_{crit}} h_{lam}(x) \, dx + \frac{1}{H - x_{crit.}} \int_{x_{crit}}^{H} h_{turb}(x) \, dx \, . \tag{3.64}$$

For flow in a narrow, open cavity, the same expression for convective heat transfer is used for both the one- and two-dimensional models (Saelens, 2002):

$$h = \frac{k_{air}}{L} \cdot 0.68 \left(Ra_L \frac{L}{H} \right)^{0.25} \, . \tag{3.65}$$

3.3.2.3 Equations for Mixed Convection

When dealing with flow over a flat plate in a mixed convection regime, the following expression is used to take the average between natural and forced convection (Rolle, 2000), (Incropera, De Witt, 1990), (Burmeister, 1993):

$$h_{mix} = \sqrt[3]{h_{natural}^3 + h_{forced}^3} \, . \tag{3.66}$$

Research on the characteristics of mixed convection in narrow channels is relatively new and mostly analytical (Saelens, 2002). In both models, the following expression for turbulent, fully-developed flow in a vertical tube is used (Burmeister, 1993):

$$Nu^3 = |Nu_f^3 - Nu_n^3|, \tag{3.67a}$$

where

$$Nu_f = \frac{0.0357\,\mathrm{Re}\,\mathrm{Pr}^{1/3}}{\ln(\mathrm{Re}/7)[1+\mathrm{Pr}^{-4/5}]^{5/6}}, \tag{3.67b}$$

$$Nu_n = \frac{0.15Ra^{1/3}}{[1+(0.492/\mathrm{Pr})^{9/16}]^{16/27}}, \tag{3.67c}$$

$$h = Nu \cdot (k_{air}/D_h). \tag{3.67d}$$

Note that the minus sign in equation (3.67a) is used for the parallel case, since the increased velocity near the wall decreases turbulent core mixing due to a lower velocity in the core.

3.3.2.4 Convection in Closed Cavities

In certain configurations, closed rectangular cavities may be formed. For those instances, the convective heat transfer coefficients are calculated using the larger of the following two Nusselt numbers (Athienitis, 1998):

$$Nu = 0.0605 \cdot Ra^{1/3}, \tag{3.68a}$$

$$Nu = \left[1 + \frac{0.104 \cdot Ra^{0.293}}{\left[1+\left(\frac{6310}{Ra}\right)^{1.36}\right]^3}\right]^{1/3}. \tag{3.68b}$$

3.4 Solar Radiation Absorbed at Each Layer

Solar radiation transmitted and reflected within the system is calculated using a net heat radiation algorithm for n layers (Siegel, Howell, 1992). As shown in Figure 3.2, every layer has a positive and negative element, each composed of two components as follows:

$$Ip_i = Ip_{i-1}\tau_i + In_{i+1}\rho_i\,, \tag{3.69a}$$

$$In_i = Ip_{i-1}\rho_i + In_{i+1}\rho_i\,. \tag{3.69b}$$

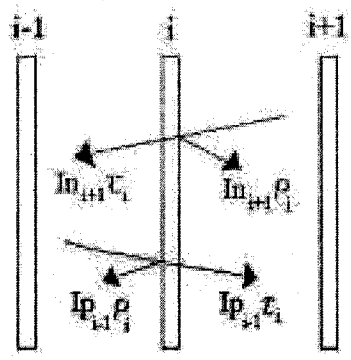

Figure 3.2: Positive and Negative Radiation Components

Each layer provides two equations and two unknowns. The positive component of the first layer is equal to the incident solar radiation, and the negative component coming from the room is assumed to be negligible. After solving the set of equations, a simple operation is done to find the net heat absorbed in each layer:

$$S_i = A_i \cdot (Ip_i - In_i)\,. \tag{3.70}$$

3.5 Long-Wave Radiation Heat Transfer in 2D Model

As discussed in section 3.3.2, the two-dimensional model uses a radiosity analysis to directly compute the radiation exchange between elements:

Net radiation in element *k*:

$$q_{r,k} = q_{o,k} - q_{i,k}, \tag{3.71}$$

Outgoing radiation from element *k*:

$$q_{o,k} = \varepsilon_{lw,k}\sigma T_k^4 + \rho_k \cdot q_{i,k}, \tag{3.72}$$

Incident radiation on element *k*:

$$q_{i,k} = \sum_{j=1}^{N} F_{k,j} \cdot q_{o,j}. \tag{3.73}$$

Equations (3.71-3.73) can be rearranged and put into matrix form as follows:

$$\mathbf{[A]\,[Q_o] = [S]}, \tag{3.74}$$

where

$$A_{i,j} = \delta_{i,j} - \rho_i \cdot F_{i,j}, \tag{3.75}$$

$$S_k = \varepsilon_{lw,k}\sigma T_k^4. \tag{3.76}$$

The matrix **A** is known as the radiosity matrix, the vector $\mathbf{Q_o}$ represents the amount of radiation leaving the surfaces, and the vector **S** represents the radiation emitted by the surfaces. After solving for $\mathbf{Q_o}$, the vector $\mathbf{Q_i}$ can be found as the difference between the radiation leaving the surface, and what the surface emits, which is equal to the reflected component of the incident radiation:

$$q_{i,k} = \frac{q_{o,k} - S_k}{\rho_k}. \tag{3.77}$$

Numerical recipes for **LU** decomposition and back substitution that come with Microsoft's Fortran PowerStation 4.0 were used to solve equation (3.74). The basic principle behind **LU** decomposition is to find the lower (**L**) and upper (**U**) triangular matrices of **A** such that

$$[A] = [L][U]. \tag{3.78}$$

Finding **L** and **U** requires less computational power than finding the inverse of **A**. Equation (3.74) can be rewritten as

$$[A][Q_o] = [L][U][Q_o] = [S]. \tag{3.79a}$$

By introducing

$$[Y]=[U][\,Q_o], \tag{3.79b}$$

equation (3.79a) becomes

$$[A][\,Q_o] = [L][Y] = [S]. \tag{3.79c}$$

As **L** and **U** are triangular matrices, **Y** is solved using back substitution with equation (3.79c), and then $\mathbf{Q_o}$ is solved using back-substitution with equation (3.79b).

3.6 Finding View Factors

The main equation that is used to find view factors between elements is the standard equation for determining view factors between parallel rectangles (Figure 3.3) (Incropera, De Witt, 1990):

$$F_{i,j} = \frac{2}{\pi \overline{X}\overline{Y}}\left\{\ln\left[\frac{(1+\overline{X}^2)(1+\overline{Y}^2)}{1+\overline{X}^2+\overline{Y}^2}\right]^{1/2} + \overline{X}(1+\overline{Y}^2)^{1/2}\tan^{-1}\frac{\overline{X}}{(1+\overline{Y}^2)^{1/2}} + \overline{Y}(1+\overline{X}^2)\tan^{-1}\frac{\overline{Y}}{(1+\overline{X}^2)^{1/2}} - \overline{X}\tan^{-1}\overline{X} - \overline{Y}\tan^{-1}\overline{Y}\right\}. \tag{3.80}$$

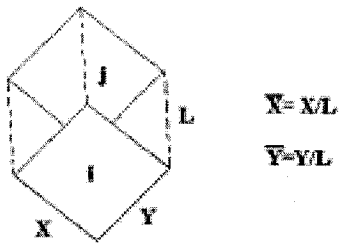

Figure 3.3: Parallel Rectangles

This equation is coupled with energy conservation to obtain all of the view factors between the parallel rectangular elements. Energy conservation is required due to the fact that not all of the elements are aligned as depicted in Figure 3.4.

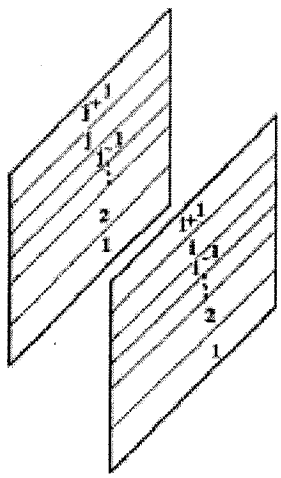

Figure 3.4: Parallel Plates in the Cavity

It is evident that $f_{1,1}$ can be computed directly from equation 3.80. The following energy balance is used to compute $f_{1,2}$:

$$f_{(1,2),(1,2)} \cdot (A_1 + A_2) = A_1 \cdot (f_{1,1} + f_{1,2}) + A_2 \cdot (f_{2,1} + f_{2,2}). \tag{3.81}$$

In this case, all of the elements have the same areas; therefore, we can solve for $f_{1,2}$:

$$f_{1,2} = f_{(1,2),(1,2)} - f_{1,1}. \tag{3.82}$$

By doing an energy balance on the other possible combinations, the following general relation can be found:

$$f_{1,i} = \frac{i}{2} f_{(1\ldots j),(1\ldots i)} - \frac{1}{2} f_{1,1} - \frac{i-1}{2} f_{(1\ldots j-1),(1\ldots i-1)} - \sum_{n=2}^{i-1} f_{1,n}. \tag{3.83}$$

Now that the view factors have computed between the first element of the first plate, and all the elements of the second plate, the remaining terms can be found using symmetry and reciprocity as follows:

$$f_{2,1} = \frac{A_1}{A_2} f_{1,2}. \tag{3.84}$$

To find the view factor between the elements and the inlet and outlet, the equation for perpendicular rectangles with a common edge is needed, as depicted in Figure 3.5 (Incropera, De Witt, 1990):

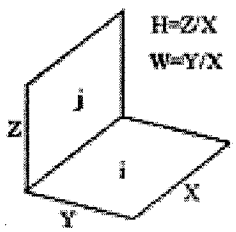

Figure 3.5: Perpendicular Rectangles with a Common Edge

$$f_{ij} = \frac{1}{\pi W}\left(\begin{array}{l} W \tan^{-1}\frac{1}{W} + H \tan^{-1}\frac{1}{H} - (H^2+W^2)^{1/2} \tan^{-1}\frac{1}{(H^2+W^2)^{1/2}} \\ + \frac{1}{4}\ln\left\{ \frac{(1+W^2)(1+H^2)}{1+W^2+H^2}\left[\frac{W^2(1+W^2+H^2)}{(1+W^2)(1+H^2)}\right]^{W^2} \right. \\ \left. \times\left[\frac{H^2(1+W^2+H^2)}{(1+H^2)(1+W^2)}\right]^{H^2} \right\} \end{array} \right) \tag{3.85}$$

To determine the view factor between individual elements and the entrance or exit is simpler than in the parallel orientation, and has the following general expression:

$$f_{i,bottom\,(or\,top)} = i \cdot f_{(1\ldots i),bottom} + (i-1) \cdot f_{(1\ldots i-1),bottom} \cdot \cdot \tag{3.86}$$

View factors between the elements and the side walls of the channel are still required; however, the expressions for calculating this directly would be complicated. If the two walls are grouped together as one surface, then the simple enclosure rule can be used to compute the remaining terms. The rule states that the sum of the view factors from one element to all other elements in an enclosure must equal to one. FORTRAN subroutine ViewFact.F90, which calculates the view factors used in the 2D model, is listed in Appendix B.

3.7 Solar Energy Availability

The total solar radiation incident on the AFW consists of the direct (beam) part, the sky diffuse solar radiation, and the radiation reflected off the ground, which is also assumed to be diffuse. In order to determine the approximate level of irradiation at a specific time of day, for a given location, the apparent solar time is required:

$$AST = tod + ET + (STM - LNG)/(15°/hr), \tag{3.87}$$

where

AST = apparent solar time,

tod = time of day (hrs),

ET = equation of time ,

$$ET = 9.87\sin\left[\frac{4\pi(n-81)}{364}\right] - 7.53\cos\left[\frac{2\pi(n-81)}{364}\right] - 1.5\sin\left[\frac{2\pi(n-81)}{364}\right], \tag{3.88}$$

n = day of year (1-365),

STM = local standard time meridian (degrees),

LNG = local longitude (degrees).

The position of the sun and the geometric relationships between a plane and the solar radiation incident on it may be described as in Figure 3.6 in terms of the following angles:

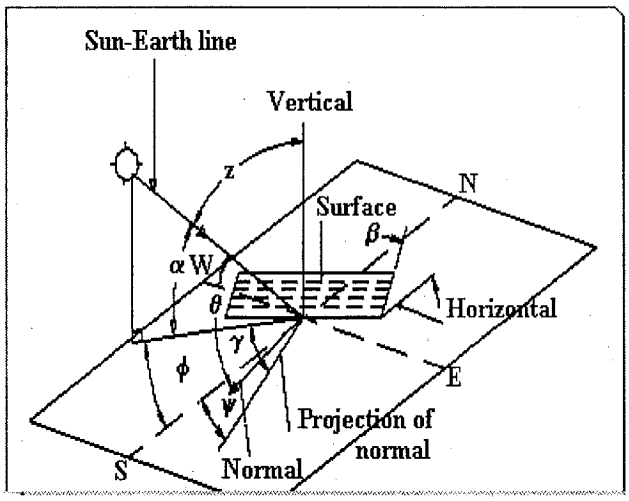

Figure 3.6: Solar Radiation Geometry for an Inclined Plane (Athienitis, 1998)

- *Lat*: Latitude, equal to the angle of the location relative to the equator; north is positive.
- δ: Declination, equal to the angular position of the sun at solar noon, relative to equatorial plane.
- α: Solar altitude, equal to the angle between the sun's rays and the horizontal.
- *z*: Zenith angle, equal to the angel between the sun's rays and the vertical.
- φ: Solar azimuth, equal to the angle between the horizontal projection of the sun's rays from due south.

- γ: Surface solar azimuth, equal to the angle between the projection of the sun's rays and the normal to the surface on the horizontal plane.
- ψ: Surface azimuth, equal to the angle between the projection of the normal to the surface on a horizontal plane and due south.
- β: Tilt, angle between the surface and the horizontal.
- θ: Angle of incidence, angle between the solar rays and a line normal to the surface.
- *ha*: Hour angle, an expression of AST centred at local solar noon.

$$\alpha = \sin^{-1}[\cos(Lat)\cdot\cos(\delta)\cdot\cos(ha)+\sin(L)\cdot\sin(\delta)] \tag{3.89}$$

$$\delta = 23.45^{\circ}\sin\left[\frac{360(284+n)}{365}\right] \tag{3.90}$$

$$ha = (AST-12hr)\cdot\frac{15^{\circ}}{hr} \tag{3.91}$$

$$\varphi = \cos^{-1}\left[\frac{\sin(\alpha)\cdot\sin(Lat)-\sin(\delta)}{\cos(\alpha)\cdot\cos(Lat)}\right]\cdot\frac{ha}{|ha|} \tag{3.92}$$

$$\gamma = \varphi - \psi \tag{3.93}$$

$$\theta = \cos^{-1}\left[\frac{\cos(\alpha)\cdot\cos(|\varphi|)+|\cos(\alpha)\cdot\cos(|\varphi|)|}{2}\right] \tag{3.94}$$

The K_T clearness index model was used to determine the solar radiation incident on the surface (Orgill, Hollands, 1977). Average monthly K_T values that were used for Montreal are given in Table 3.1.

Table 3.1: Mean monthly K_T values for Montreal

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sept	Oct	Nov	Dec
K_T	0.45	0.51	0.50	0.48	0.49	0.49	0.52	0.49	0.49	0.41	0.35	0.38

The hourly clearness index is obtained from monthly values based on the following correlation:

$$k_t=(A+ B\cdot\cos(esr\cdot(tod-11.99hrs)))\cdot K_T, \quad (3.95)$$

where

$$A = 0.409 + 0.5016\cdot\sin(sha - 1.047), \quad (3.96)$$

$$B = 0.6609 - 0.4767\cdot\sin(sha- 1.047), \quad (3.97)$$

$$esr = \text{earth's spin rate} = 2\pi / 24 \ \text{(rad/hr)}, \quad (3.98)$$

$$sha = \text{sunset ha} = \cos^{-1}[-\tan(Lat)\cdot\tan(\delta)]. \quad (3.99)$$

The hourly diffuse clearness index was determined using the following correlations:

For $(0 < k_t < 0.35)$:

$$k_d = (1 - 0.249\cdot k_t)\cdot k_t, \quad (3.100)$$

For $(0.35 < k_t < 0.75)$:

$$k_d = (1.557 - 1.84\cdot k_t)\cdot k_t, \quad (3.101)$$

For $(k_t > 0.75)$:

$$k_d = 0.177\cdot k_t. \quad (3.102)$$

The hourly beam clearness index (k_b) is obtained from the difference of the hourly clearness index, and the hourly diffuse clearness index:

$$k_b = k_t - k_d. \quad (3.103)$$

The total beam and diffuse irradiation can be determined based on the following indices.

Extraterrestrial solar radiation:

$$I_{on} = 1353\cdot\left[1 + 0.033\cdot\cos\left(\frac{360\cdot n}{365}\right)\right]\cdot\frac{W}{m^2}, \quad (3.104)$$

Beam irradiation:

$$I_{bm} = I_{on} \cdot \cos(\theta) \cdot k_b, \tag{3.105}$$

Diffuse irradiation:

$$I_{df} = I_{on} \cdot \sin(\theta) \cdot \left[\frac{1+\cos(\beta)}{2}\right] \cdot k_d, \tag{3.106}$$

Ground-reflected irradiation:

$$I_{gnd} = I_{on} \cdot \sin(\theta) \cdot \left[\frac{1-\cos(\beta)}{2}\right] \cdot \rho_{gnd} \cdot (k_d + k_b). \tag{3.107}$$

3.8 Angle Dependency of Optical Properties

The solar transmittance, reflectance and absorbance of the glazings are a function of the angle of incidence, θ, of the incident beam, the glass thickness, L, refractive index, n, and the extinction coefficient, k. The procedure for calculating these properties is given in detail in Athienitis and Santamouris (2002), which is reviewed in the following section. The component reflectivity, r, which is the fraction of each ray component reflected, is determined by the Fresnel relations of electromagnetic theory:

$$r = \frac{1}{2} \cdot \left[\left(\frac{\sin(\theta-\theta')}{\sin(\theta+\theta')}\right)^2 + \left(\frac{\tan(\theta-\theta')}{\tan(\theta+\theta')}\right)^2\right], \tag{3.108}$$

where θ' is calculated using Snell's law:

$$\theta' = \sin^{-1}\left(\frac{\sin(\theta)}{n}\right). \tag{3.109}$$

The refractive index is assumed to be constant at 1.52 in all calculations. The distance the rays travel within the glazing, L', is calculated as:

$$L' = \frac{L}{\sqrt{1-\left(\frac{\sin(\theta)}{n}\right)^2}}. \tag{3.110}$$

Now the transmittance, reflectance, and absorbance can be calculated as follows:

$$\tau = \frac{(1-r)^2 \cdot a}{(1-r^2a^2)}, \tag{3.111}$$

$$\rho = r + \frac{r \cdot (1-r)^2 \cdot a^2}{1-r^2a^2}, \tag{3.112}$$

$$\alpha = 1 - \tau - \rho, \tag{3.113}$$

where $a = e^{-k \cdot L'}$, which represents the exponential decay function of the intensity. For diffuse radiation, the above expressions are used to find optical properties using an angle of incidence of 60° (Athienitis, Santamouris, 2002). FORTRAN subroutine EnvCond.F90, which calculates the optical properties, hourly irradiation levels, and hourly temperatures used in the 1D model, is listed in Appendix C.

CHAPTER 4

MODEL VALIDATION

4.1 Introduction

This chapter presents validation for the models in three sections. The first compares results between the one- and two-dimensional models. It shows that the assumptions made in the one-dimensional model are reasonable, as they generate similar results to the two-dimensional model. The next section compares results from the two-dimensional model to experimental and theoretical results obtained from the Mataro Public Library. The last section looks at initial results obtained from a test room to see if the models are capable of making accurate reasonable predictions.

4.2 Comparing 1D and 2D Model Results

In order to compare the two models, thirteen different scenarios were tested. As the model generates more differences using Configuration 2, all scenarios used this configuration. Table 4.1 summarises the parameters used for each scenario. Each scenario differs from the next by one parameter, making it possible to identify the impact of each parameter separately. Four different blinds are used with the optical properties outlined in Table 4.2.

Table 4.1: Parameters used in different scenarios

Scenario	h_o (W/m^2K)	G (W/m^2)	T_o (°C)	V_{front} (m/s)	V_{back} (m/s)	L_{front} (m)	L_{back} (m)	Blind #	L_{fin} (m)
1	12.91	600	20	1	1	0.05	0.05	1	n/a
2	12.91	600	20	1	1	0.05	0.05	2	n/a

3	12.91	600	20	1	1	0.05	0.05	3	n/a
4	12.91	600	20	1	1	0.05	0.05	4	n/a
5	25	600	20	1	1	0.05	0.05	1	n/a
6	12.91	600	20	1	1	0.05	0.05	1	0.015
7	12.91	100	20	1	1	0.05	0.05	1	n/a
8	12.91	600	20	0.5	0.5	0.05	0.05	1	n/a
9	12.91	600	20	2	2	0.05	0.05	1	n/a
10	12.91	600	20	1	1	0.025	0.025	1	n/a
11	12.91	600	20	1	1	0.15	0.15	1	n/a
12	12.91	600	-20	1	1	0.05	0.05	1	n/a
13	12.91	1000	20	1	1	0.05	0.05	1	n/a

Table 4.2: Blind properties

Blind #	τ_{blind}	ρ_{blind}	α_{blind}
1	0.20	0.30	0.50
2	0.20	0.50	0.30
3	0.10	0.20	0.70
4	0.05	0.40	0.55

Figures 4.1 to 4.4 show the absolute difference in results between the one- and two-dimensional models. Results from scenarios 1 to 6, 12, and 13 are all within 0 to 3% difference.

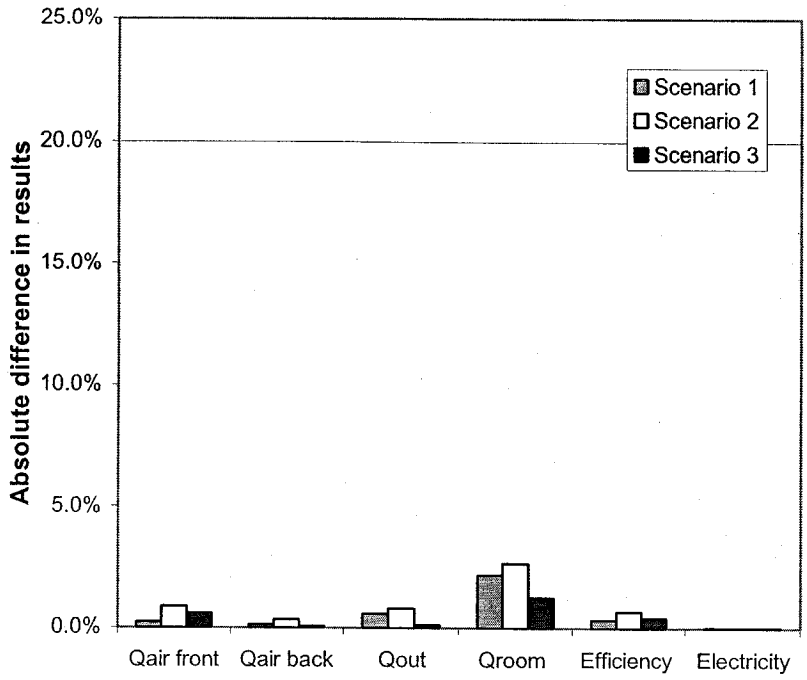

Figure 4.1: Absolute Difference in Results between 1D and 2D Models, Scenarios 1-3

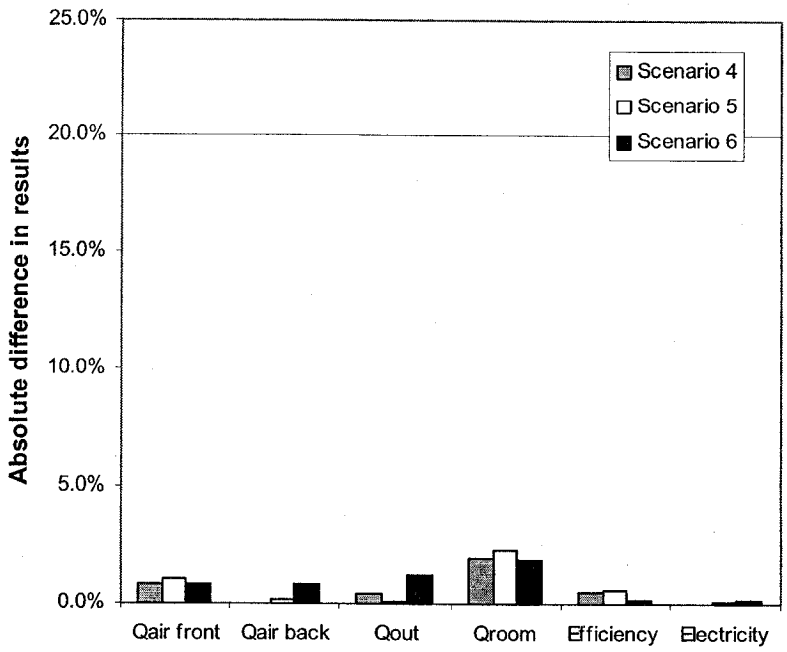

Figure 4.2: Absolute Difference in Results between 1D and 2D Models, Scenarios 4-6

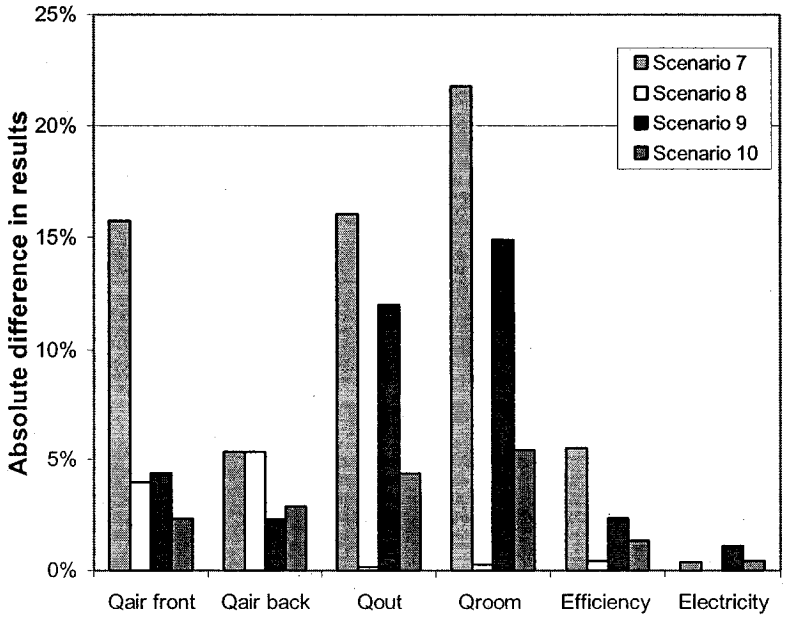

Figure 4.3: Absolute Difference in Results between 1D and 2D Models, Scenarios 7-10

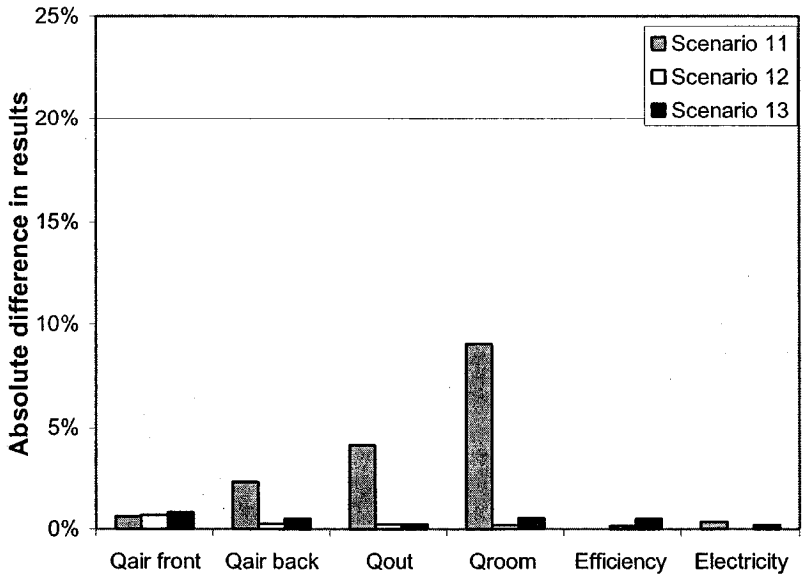

Figure 4.4: Absolute Difference in Results between 1D and 2D Models, Scenarios 11-13

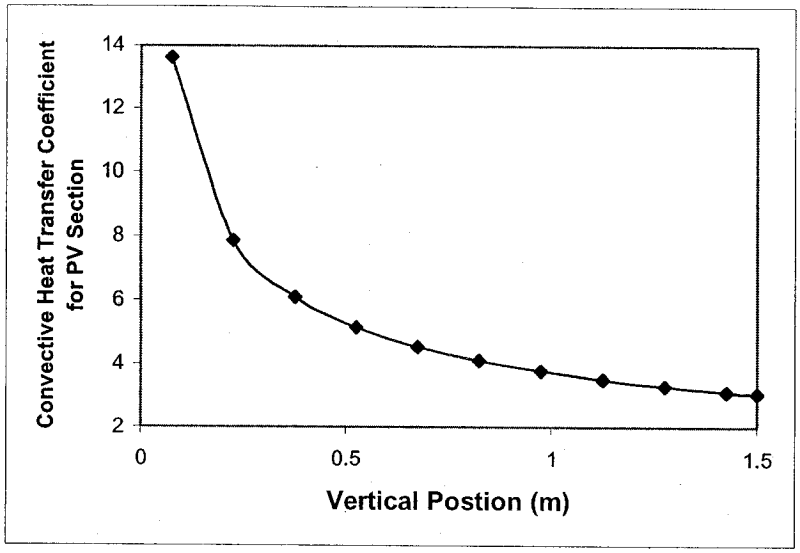

Figure 4.5: Convective Heat Transfer Coefficient in PV Section, Calculated for Scenario 9 with 2D Model

As previously mentioned, one of the main differences between the two models is that the two-dimensional model uses local convective heat transfer coefficients as opposed to average values. This is the cause of the majority of the differences in results between the two models. As seen in Figure 4.3, the calculated heat transfer to the outdoor environment, Qo, and to the room, Qroom, was 12 and 15%, respectively, for Scenario 9. In the one-dimensional model, the convective heat transfer coefficients used inside the cavity were equal to 5.73 W/m^2K, whereas the profile shown in Figure 4.5 is used for the two-dimensional model. Even though the average values of the coefficients are similar, the distribution causes major differences in results between the two models. To show this, the two-dimensional model was used with the same constant convective heat transfer coefficients as calculated in the one-dimensional model. Results from scenario 9 are shown in Figure 4.6; differences which were as high as 15%, are less than 1%.

The profile obtained for the local convective heat transfer coefficient in the PV section is obtained directly from known empirical formulas. Determining the profile in the Vision section is not as straight forward; it depends on the physical interface between the two sections. If a protrusion exists in the channel at the interface, the flow might separate, increasing the convective heat transfer coefficient in that region. Correlations need to be obtained from experimental data and/or from CFD models. Some authors have suggested that even in the first PV section, that it would be best to couple the type of model presented in this paper, with a CFD model, to determine accurate coefficients (Zhai, et al., 2002).

The air temperature profiles in the front and back cavities of Scenario 6 are shown in Figure 4.7 and 4.8. Two different curves for the two-dimensional model are shown, one using 10 elements per section, the other 50. The resulting air temperature profiles are quite similar between the two models. The air temperature calculated for the first node in the two-dimensional model is too high, as depicted in Figure 4.9, which shows an enlarged view of the first portion of the curve of Figure 4.8. As more elements are used per section, this over estimation is reduced, since when Δx approaches zero, the temperature of the first node converges to inlet air temperature. This is not a problem with the profile calculated using the one-dimensional model.

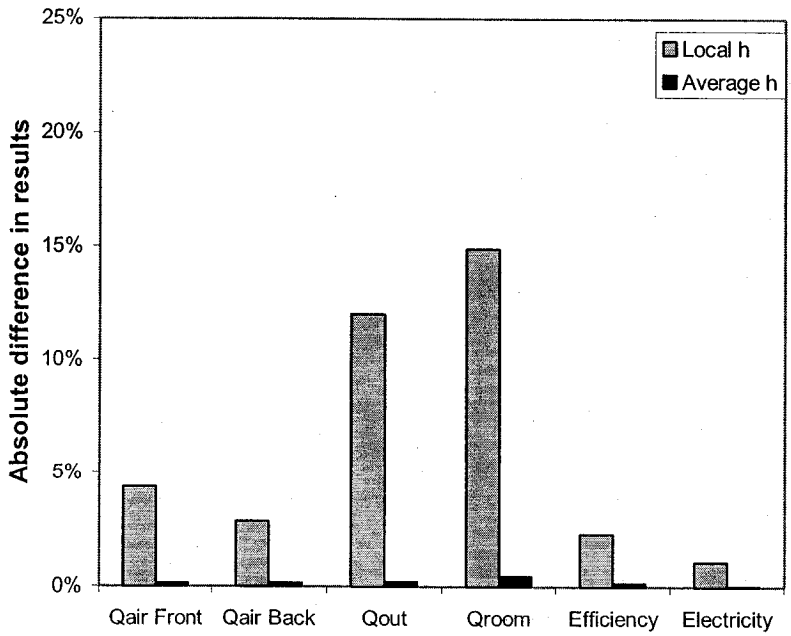

Figure 4.6: Absolute Difference in Results between 1D and 2D Models, using Average and Local Heat Transfer Coefficients, Scenario 9

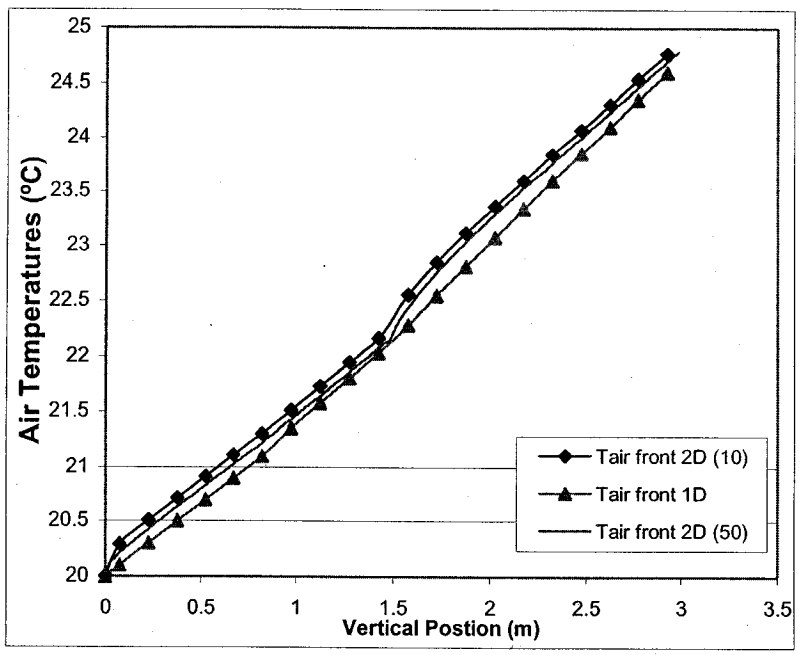

Figure 4.7: Air Temperature Profile in the Front Cavity, Scenario 6

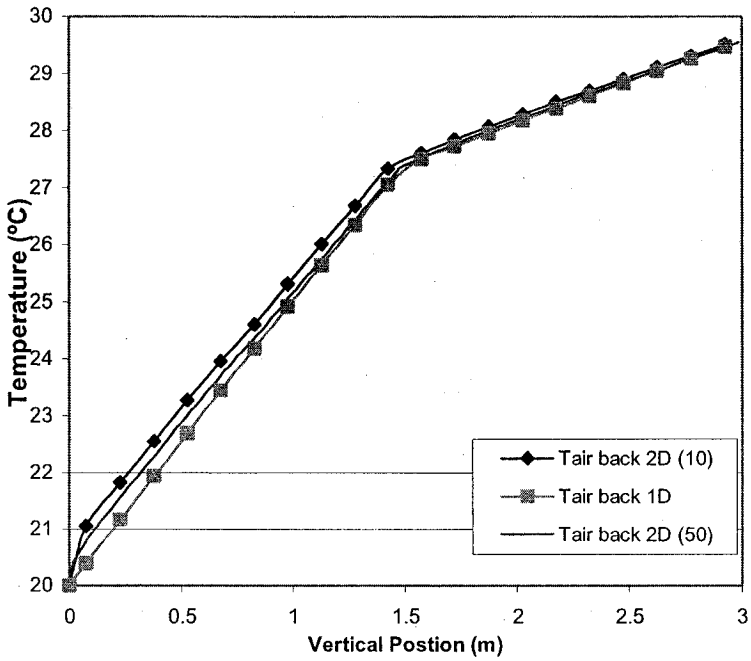

Figure 4.8: Air Temperature Profile in the Back Cavity, Scenario 6

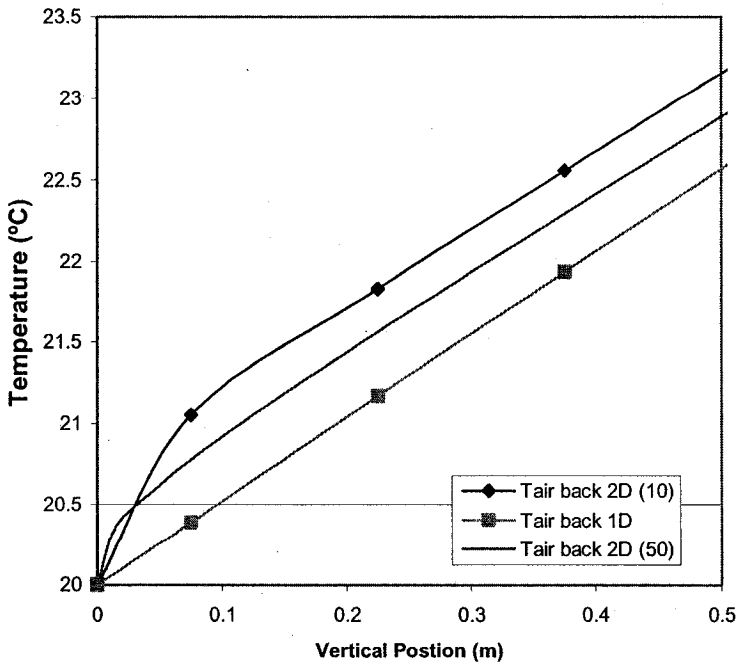

Figure 4.9: Enlarged View of the Start of the Curve for the Air Temperature Profile of Back Cavity, Scenario 6

The one-dimensional model produces the temperature profile of the air, and average temperatures for all solid components. The profile obtained for the air temperature is in the form of equation (4.1). An approximate temperature profile for the solid components is obtained using results from equation (4.1) in equation (4.2). Figures 4.10 and 4.11 show results of the temperature profiles of the various system components of the PV and Vision sections using Scenario 9. In the two-dimensional model, the profile of the component temperatures is similar to that of the profile seen in Figure 4.5 of the convective heat transfer coefficients, whereas it is more or less linear for the one-dimensional model. This variance contributed to the difference in results that was seen between the two models for this scenario. Using the same constant convective heat transfer coefficient, results in predicting almost identical component temperature profiles as seen in Figures 4.12 and 4.13. The temperature of the ends is slightly different in the two-dimensional model due to the end effects in the radiosity analysis.

$$T_{air}(x) = \left(T_o - \frac{c_3}{c_4}\right) \cdot e^{\frac{-x \cdot c_4}{a}} + \frac{c_3}{c_4} \tag{4.1}$$

$$T_k(x) = c_1 T_{air}(x) + c_2 \tag{4.2}$$

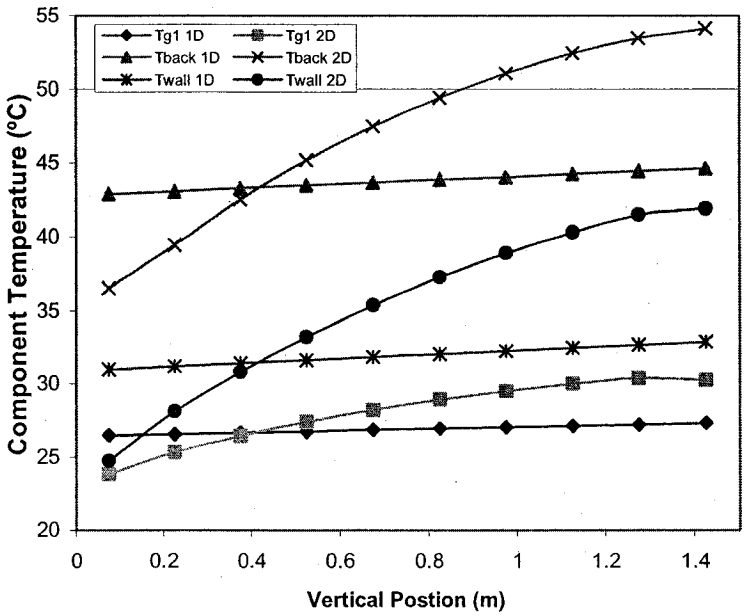

Figure 4.10: Temperature Profiles in PV Section from 1- and 2-Dimensional Models, Scenario 9

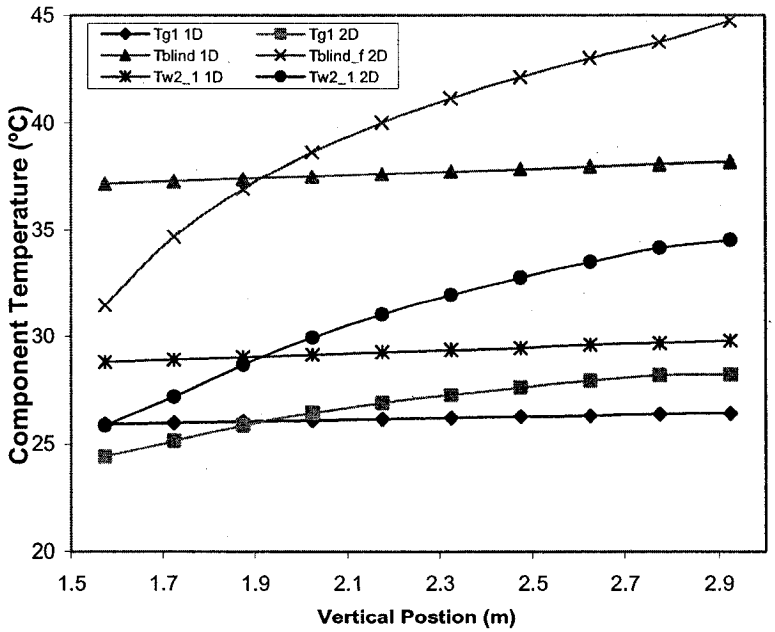

Figure 4.11: Temperature Profiles in Vision Section from 1- and 2-Dimensional Models, Scenario 9

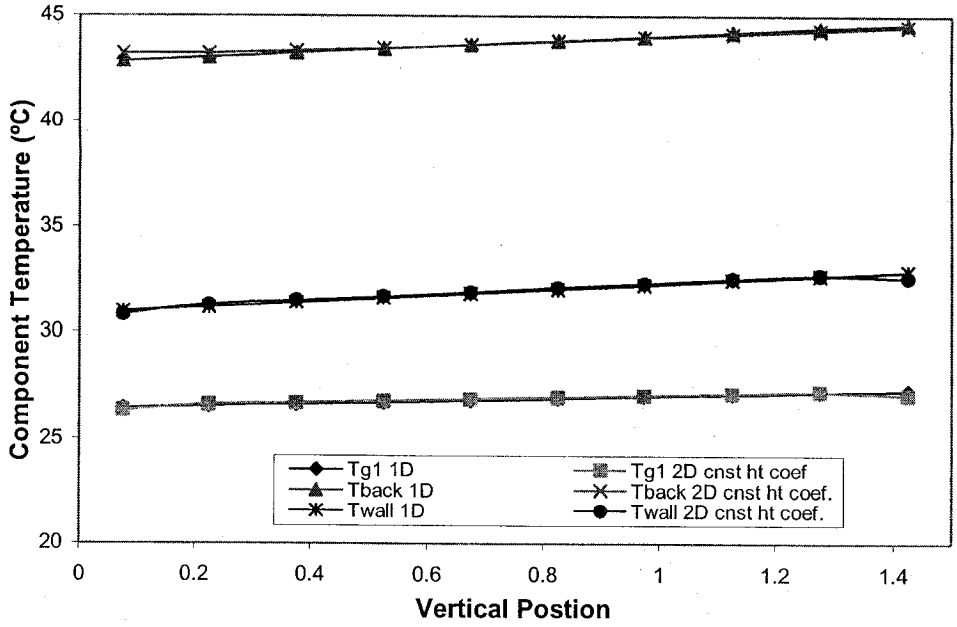

Figure 4.12: Temperature Profiles in PV Section from 1- and 2-Dimensional Models Using the Same Average Heat Transfer Coefficients in Both Models, Scenario 9

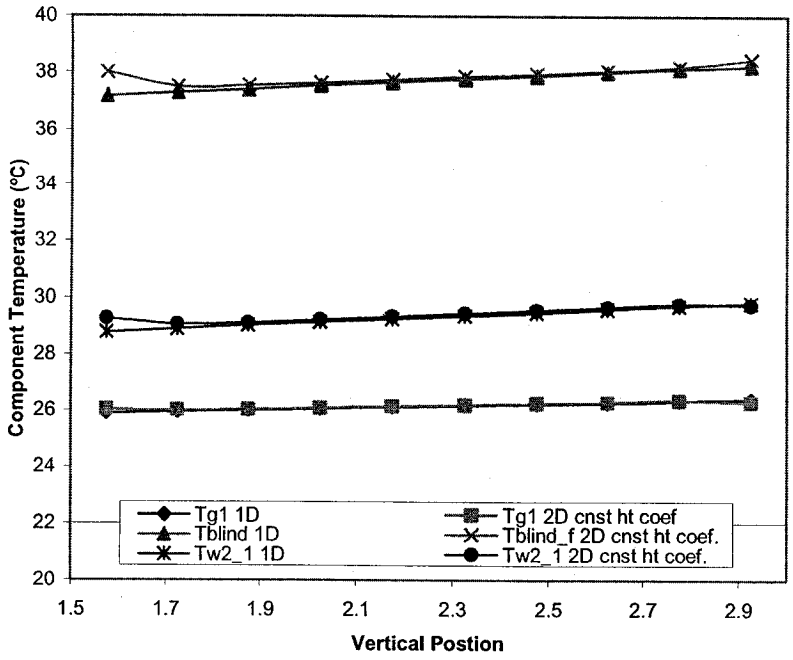

Figure 4.13: Temperature Profiles in Vision Section from 1- and 2-Dimensional Models Using the Same Average Heat Transfer Coefficients in Both Models, Scenario 9

4.3 Comparing Results with those from Literature

Validating a model by simply comparing results from another model is not an ideal approach as the other models may not have been validated themselves. However, a 1D model is useful in identifying programming errors and boundary condition errors of a 2D or 3D model. Due to a lack of resources, models are often simply assumed to have been verified when in fact they have not (Hensen, et al., 2002). Models often undergo verification, but this does not imply that computational results represent physical reality. Verification simply implies that the model is capable of accounting for the relevant physical phenomena faced by the system. Only in validation are the theoretical results verified against experimental data (Chen, Srebric, 2002).

Very few published papers present results that can be used for the purpose of validation. The results are usually not accompanied with all the required parameters needed to accurately define a system. This is true for both results presented from theoretical work and from experimental investigations. Important parameters such as fluid velocity, heat transfer coefficient values, physical properties of system components, outdoor conditions such as wind speed, temperature, and solar radiation levels, all of which can have a significant impact on final results are often not reported. It is also difficult to compare results between models when the models are designed for a specific façade (Krauter, et al., 1999).

Infield, et al. (1999), presented theoretical and experimental results for the Mataro Library presented earlier. Although that specific paper did not give all the required parameters to define the system, enough information was obtained in subsequent publications, and by contacting the authors to adequately represent the experimental

conditions in simulations with our model. For parameters that were not given, approximate values were used in the simulation. For example, the air velocities were said to be below 0.5 m/s, with no specific value given; a value of 0.3 m/s was used in this investigation. This assumption generates a source of uncertainty in using their results as air velocities can have a significant impact on performance. Figures 4.14a to 4.14e show the results that were published at the Jerusalem Solar World Congress by Infield et al. (1999) (Infield, et al., 1999).

The setup found at the Mataro Library is not identical to the one for which these models were developed. In order to properly compare the results, modifications were made to the two-dimensional model so that it more accurately represents that system. Only the PV section was considered, as the Mataro Library is only composed of one section. The interior wall needed to be changed from an opaque insulating wall, to a double-glazed window. The PV section also needed to be modified, so that PV cells did not cover the whole surface. Instead the cells needed to only cover 85 % of the surface to allow some of the irradiation to be transmitted into the room.

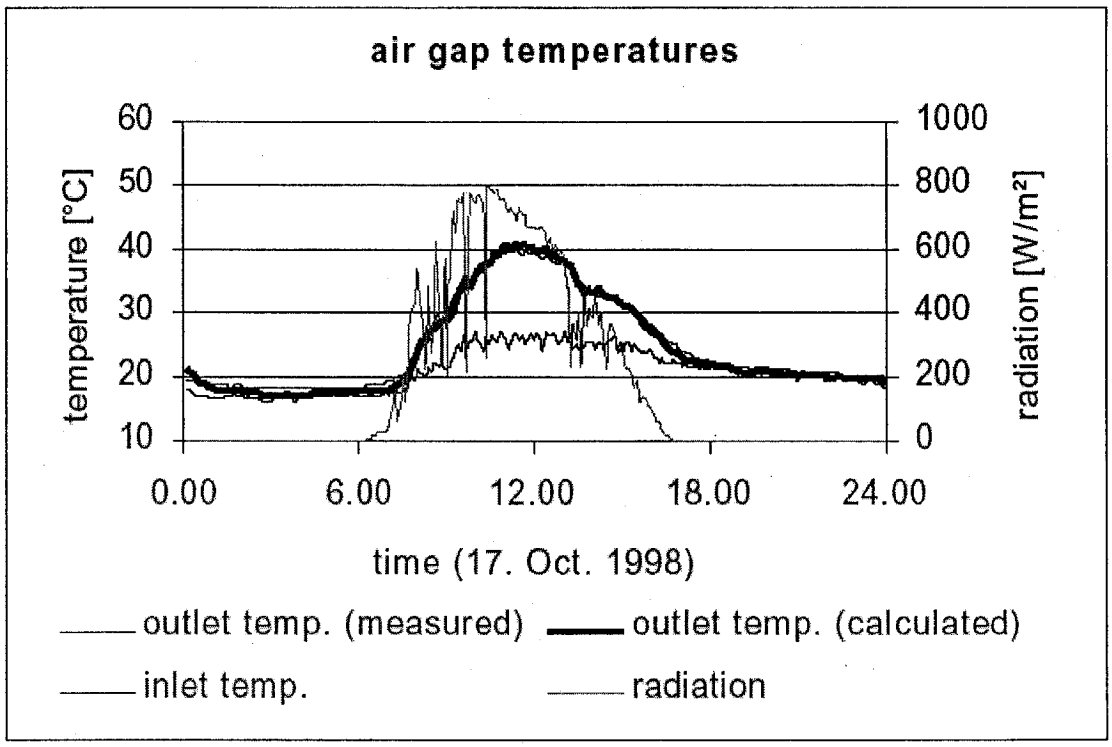

Figure 4.14a: Theoretical and Experimental Outlet Temperatures for October 17, 1998 in the Mataro Library (Infield, et al., 1999)

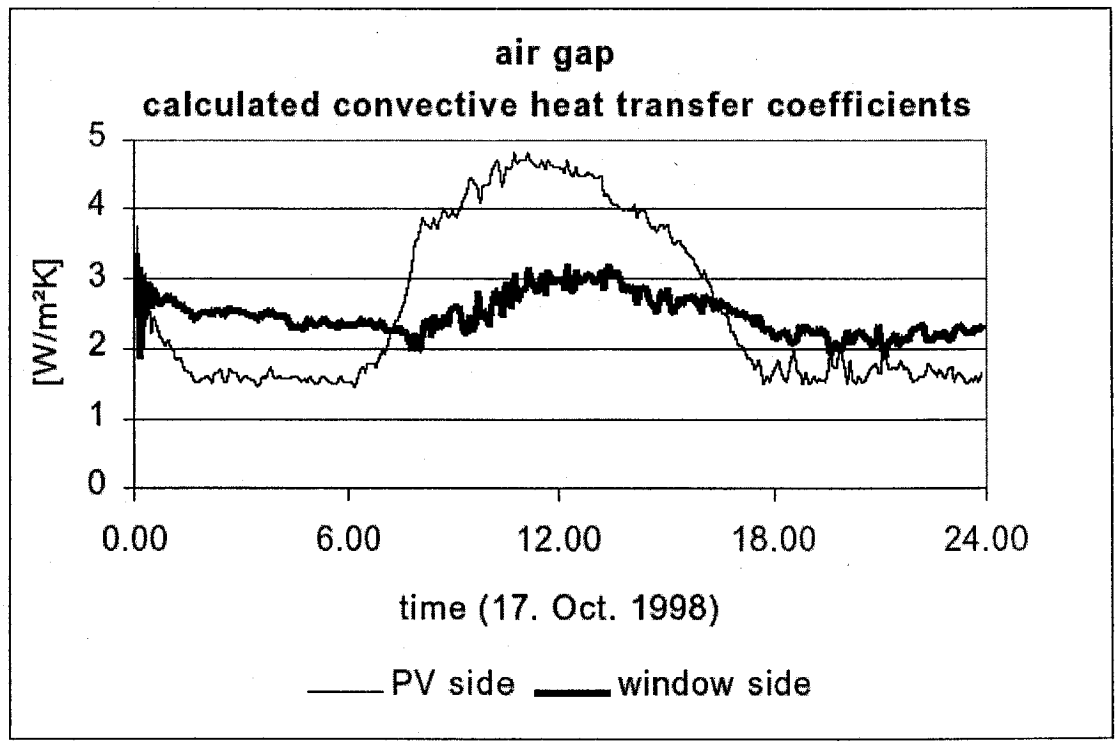

Figure 4.14b: Calculated Convective Heat Transfer Coefficients for Results Presented in Figure 4.14a (Infield, et al., 1999)

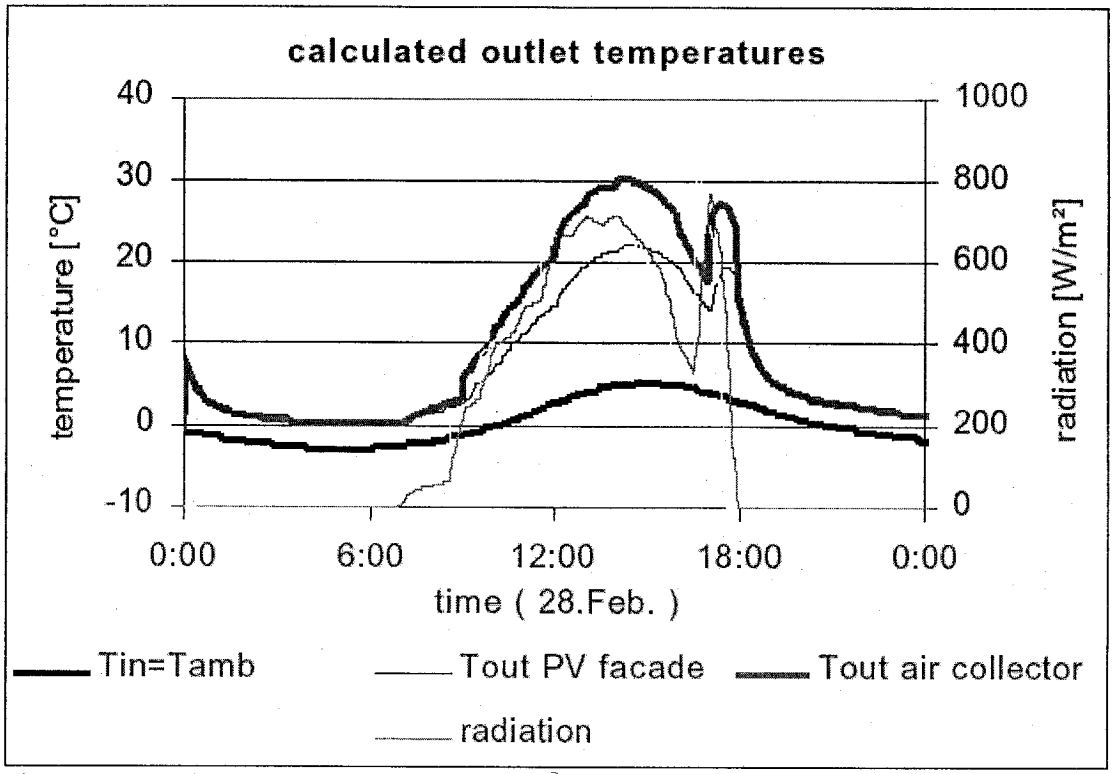

Figure 4.14c: Theoretical Outlet Temperatures for Mataro Library Based on Feb 28 Weather Data (Infield, et al., 1999)

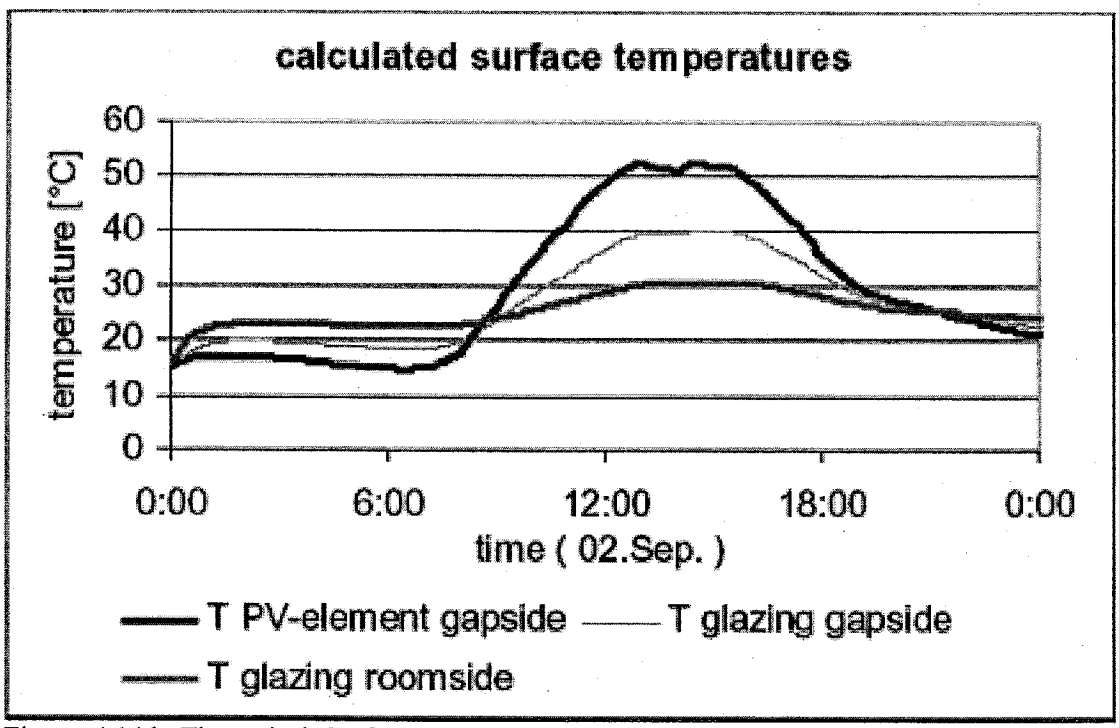

Figure 4.14d: Theoretical Surface Temperatures for Mataro Library Based on September 2 Weather Data (Infield, et al., 1999)

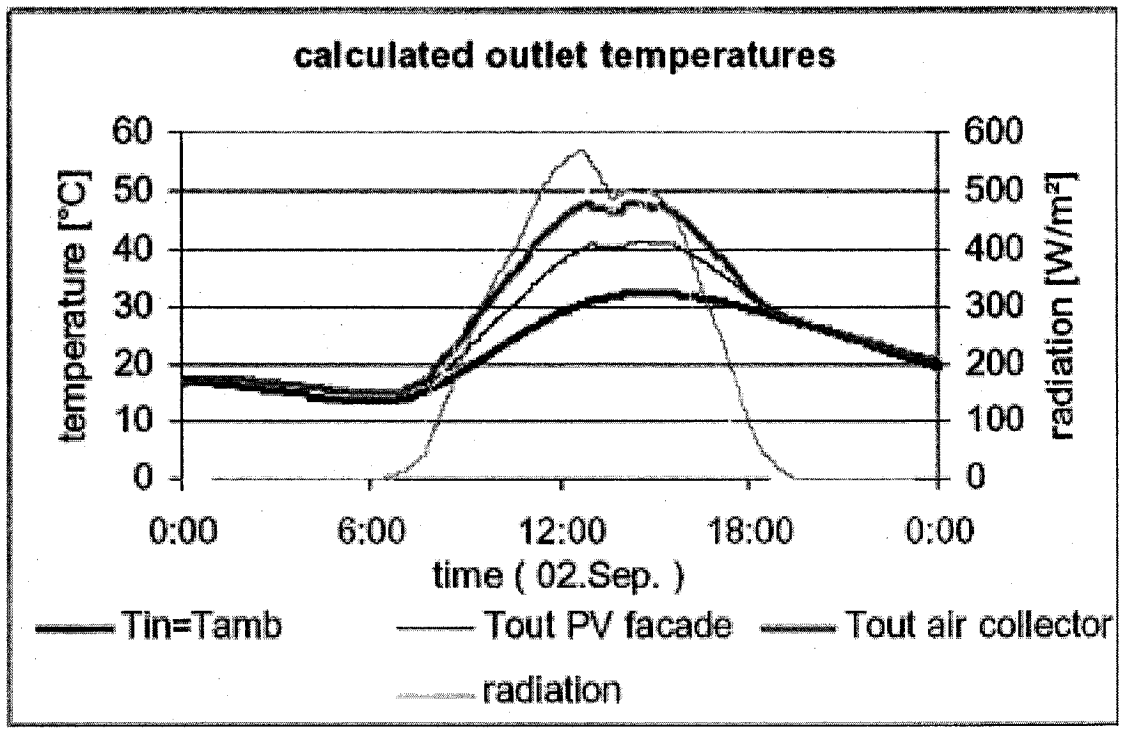

Figure 4.14e: Theoretical Outlet Temperature for Mataro Library Based on September 2 Weather Data (Infield, et al., 1999)

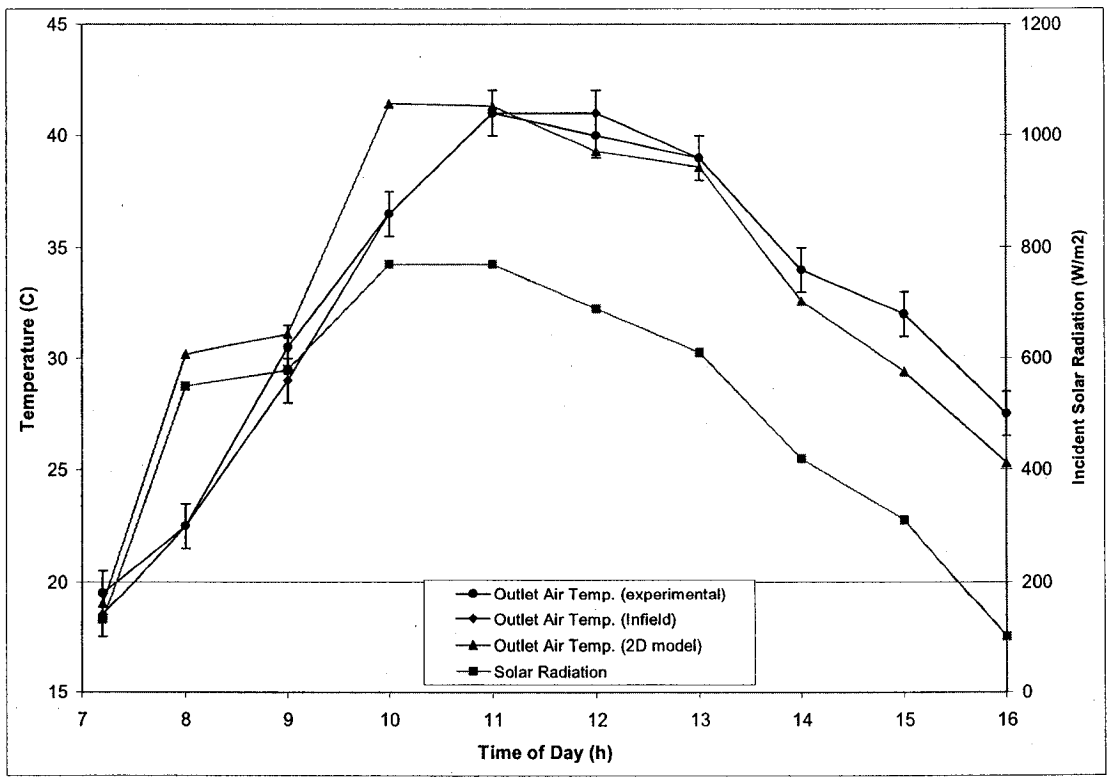

Figure 4.15: Verification of Outlet Air Temperature Results Based on October 17th Data (Infield, et al., 1999)

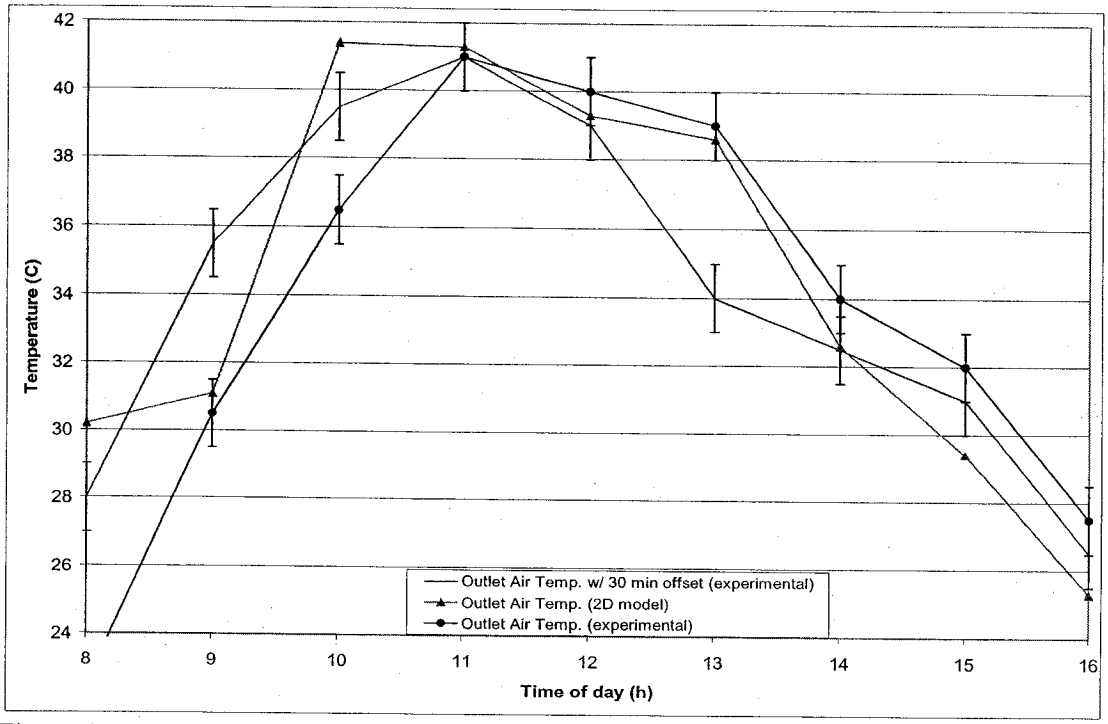

Figure 4.16: Verification of Outlet Air Temperature Results Based on October 17 Data Using a 30 Minute Offset (Infield, et al., 1999)

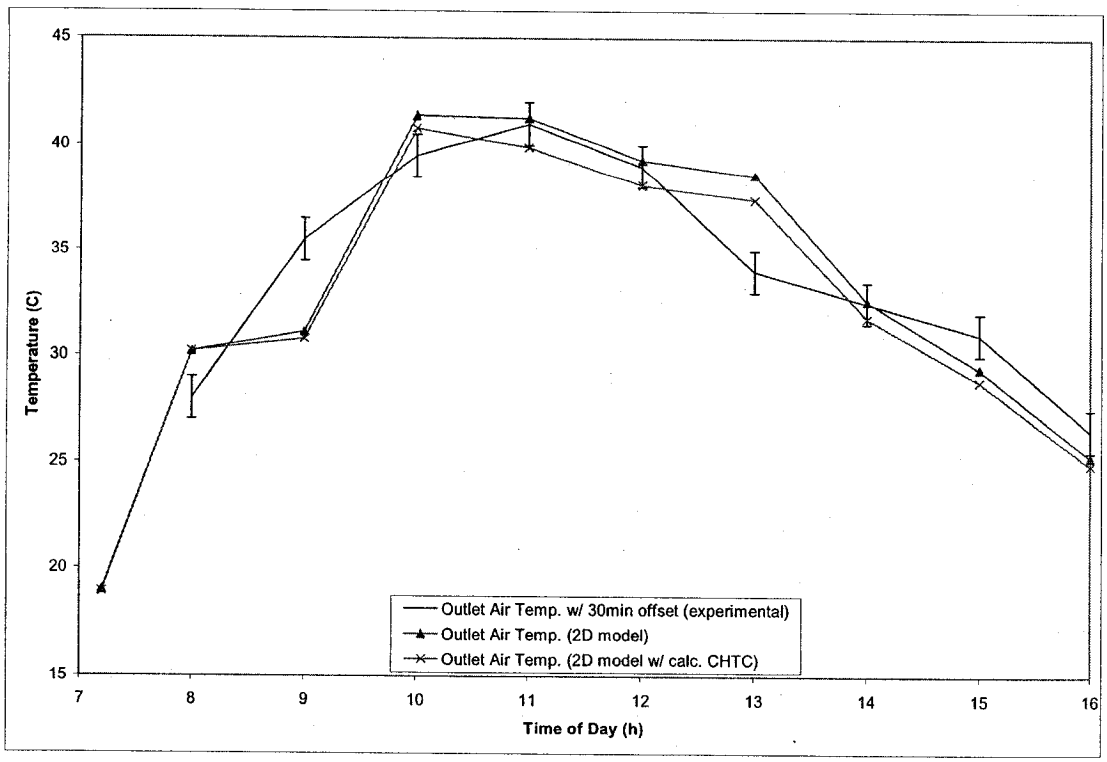

Figure 4.17: Verification of Outlet Air Temperature Results Based on October 17 Data Using given and Calculated Convective Heat Transfer Coefficients (Infield, et al., 1999)

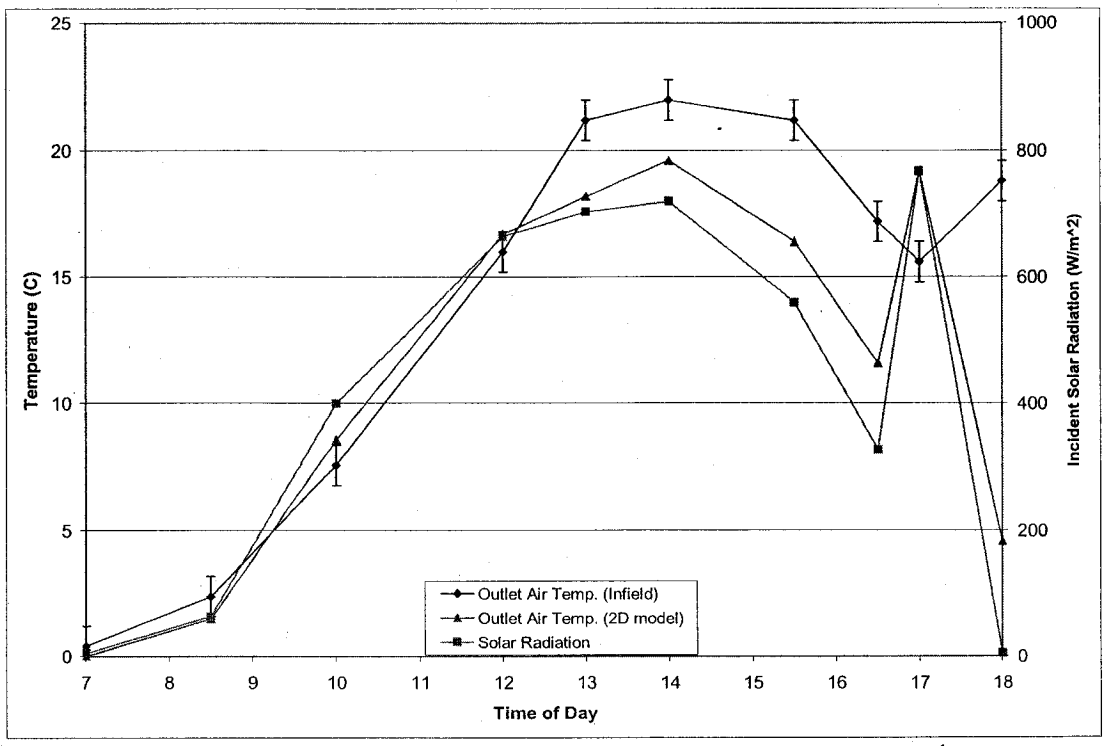

Figure 4.18: Verification of Outlet Air Temperature Results Based on February 28th Data (Infield, et al., 1999)

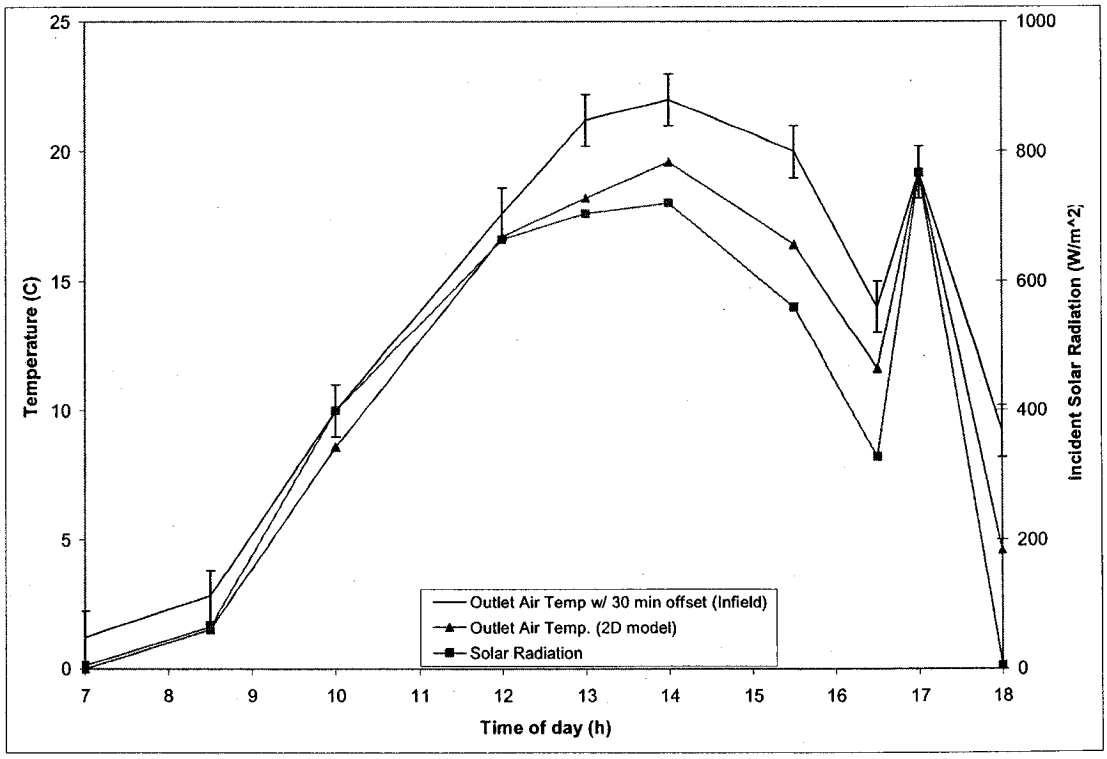

Figure 4.19: Verification of Outlet Air Temperature Results Based on February 28th Data Using a 30 Minute Offset (Infield, et al., 1999)

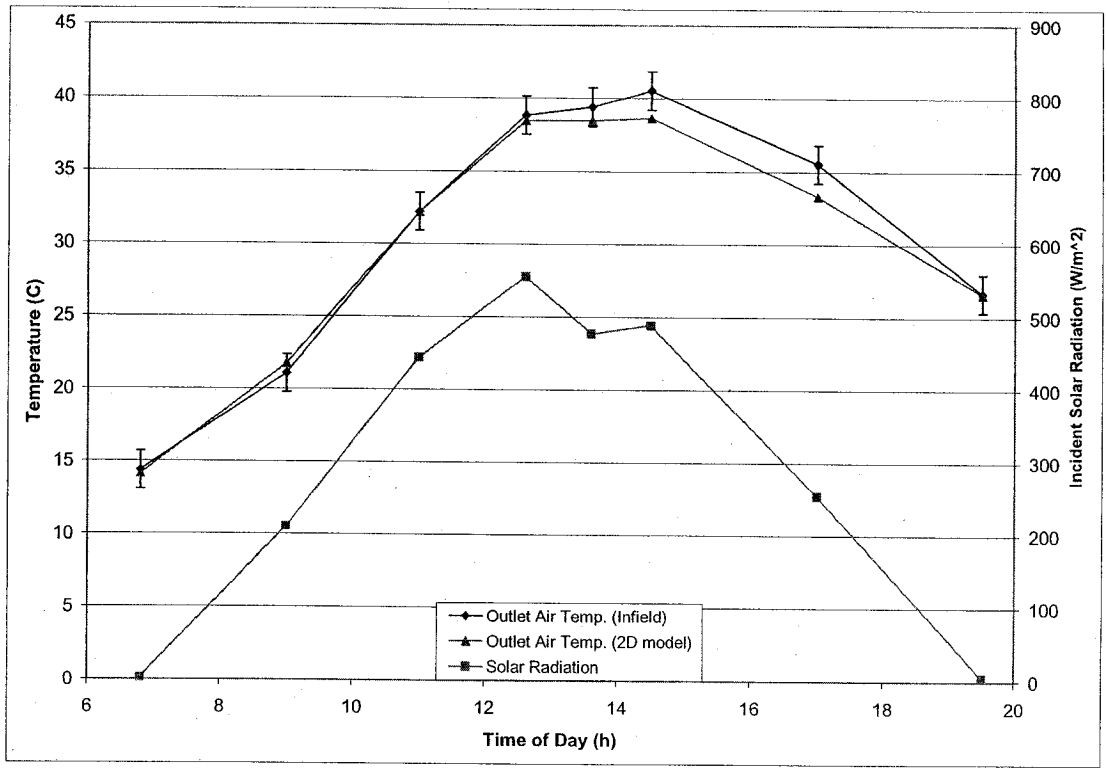

Figure 4.20: Verification of Outlet Air Temperature Results Based on September 2nd Data (Infield, et al., 1999)

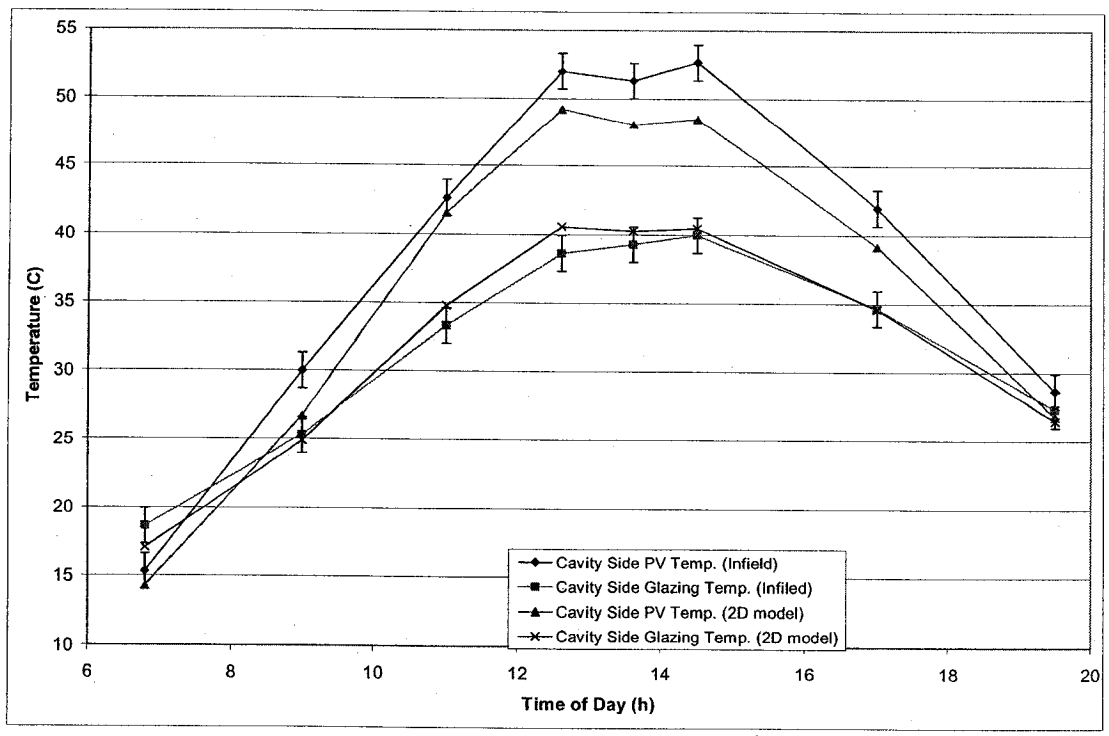

Figure 4.21: Verification of PV Temperatures Using September 2nd Data (Infield, et al., 1999)

Figures 4.15 through 4.21 show results taken from Figures 4.14a through 4.14e, comparing them to results obtained using the two-dimensional model. The error bars represent errors that may have occurred by reading the data from the original figures. The figures were enlarged such that the time scale had one hour equal to one centimetre. Measurements were taken from the vertical axis using a ruler; an error of $\pm$ 2 mm was assumed from the readings. In general, the result followed the same trend, with some differences in magnitude occurring.

Figure 4.15 represents data for October 17, 1999. Data presented from that day is especially relevant as it was the only day for which experimental results are presented, and also the only day for which the convective heat transfer coefficients used in the cavity were given. As seen on the figure, the results from Infield's model correlate well with the experimental results, and to some extent, the results from the 2D model correlate well. Half of the data points are within the error bars, with three others within 5%. The two points that are most erroneous can easily be attributed to the fact that the two-dimensional model assumes steady-state conditions. The results are also very dependent on the level of incident solar radiation. As seen in Figure 4.14a, the solar radiation recorded on October 17 was sporadic, with some short peaks present throughout the morning. Some of these peaks were used in the 2D model, which caused the results to be more elevated than what actually occurred. If on the other hand an average irradiation value had been used for 8 am and 10 am, the results would have been much closer to the experimental results.

The fact that the two-dimensional model assumes steady-state conditions introduces another source of error; it lacks the ability to account for the thermal inertia of

the system. The thermal inertia of the system generates a phase-difference between the temperature peaks and the peak irradiation levels. Infield (1999) measured the phase-difference to be 30 minutes for the Mataro Library. This value is dependent on the system. In Salens' (2002) experiments, solar radiation led the flow rate and the temperature difference by as much as one and a half hours. Figure 4.16 shows results taking into account the 30 minute phase difference. As can be seen, the peaks of the curves are closer together, as would be expected.

The data collected on October 17th is the only such data which gives the convective heat transfer coefficients used in Infield's model; these are given in Figure 4.14b. Figure 4.17 shows the effect of using the local convective heat transfer coefficients calculated with the 2D model, or using the average values given in Figure 4.14b. The results are very similar to one another. The effect of using the calculated local coefficients seems to predict lower temperature increases with higher irradiation levels. The effect seems to be minimal, with the greatest temperature difference being only 1°C.

Figure 4.18 shows results calculated for February 28th (Infield, et al., 1999). The results predicted from the 2D model are lower than those predicted by Infield. Taking into account the possible measurement errors (±0.8 °C) the biggest difference is only 7.3%. If the error that is introduced due to the different convective heat transfer coefficients is taken into account, this difference is reduced to only 2.7%. As observed from the figure, the results predicted with the 2D model follow the same trend as the incident solar radiation, whereas results predicted by Infield follow the same trend, but

with a thermal lag of 30 minutes. This can be seen clearly in Figure 4.19, when the thermal lag is taken into account. In this figure, all curves follow the same trend.

Figure 4.20 compares results from September 2. Results from this day are good for comparison as the solar radiation follows a smooth curve, unlike data collected October 17th, which experienced erratic irradiation levels. The benefits of this smooth curve are shown in the figure since the results are within 2.5% taking into account possible translation errors. The remaining difference can be attributed to the use of differing convective heat transfer coefficients. Figure 4.21 compares the results of the predicted glazing temperatures on both sides of the cavity for September 2. The cavity side glazing temperatures calculated in both models are almost identical. The cavity side PV temperatures, calculated with the 2D model, are lower than those from Infield's model, with the largest difference being only 3.5%. This difference is acceptable since both models represent the sandwiched PV configuration differently.

4.4 Experimental Results From Concordia Test Room

A test room was built at Concordia University to investigate the application of PV-integrated double-façades. The room was made from a prefabricated housing unit that was delivered and placed on the roof of the Building Engineering building at Concordia. It has interior dimensions of 2.972 m x 2.896 m x 3.045 m, with a total interior floor area of 8.61 m^2. There are two door openings, one is a single door and the other is a large double door. The double door opening is the location for the application of the double façade panel. Figure 4.22 shows a drawing of the general layout and of the test room.

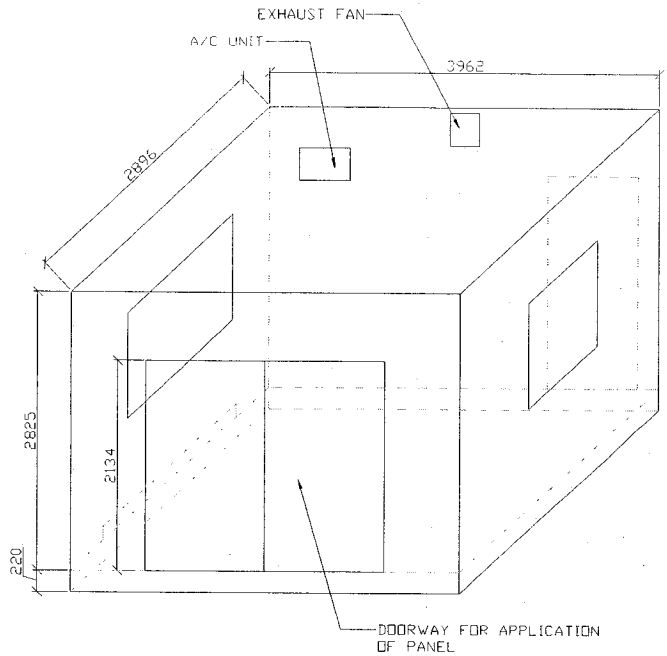

Figure 4.22: Test Room 3D Schematic (Interior Dimensions in mm; Drawn By: Julie Szabo)

Two different types of PV panels, installed in two different configurations, are located in the double door opening of the test room. Two Photowatt panels are installed on the exterior of the façade forming a 0.091 m channel. The other side has two Spheral Solar panels placed in the middle of the cavity in order to allow air to flow on both sides. The room side of the cavity is plywood with insulation making the energy transfer to and from the room negligible. Figure 4.23 is a photo of the test room showing the south wall with the double-façade configuration.

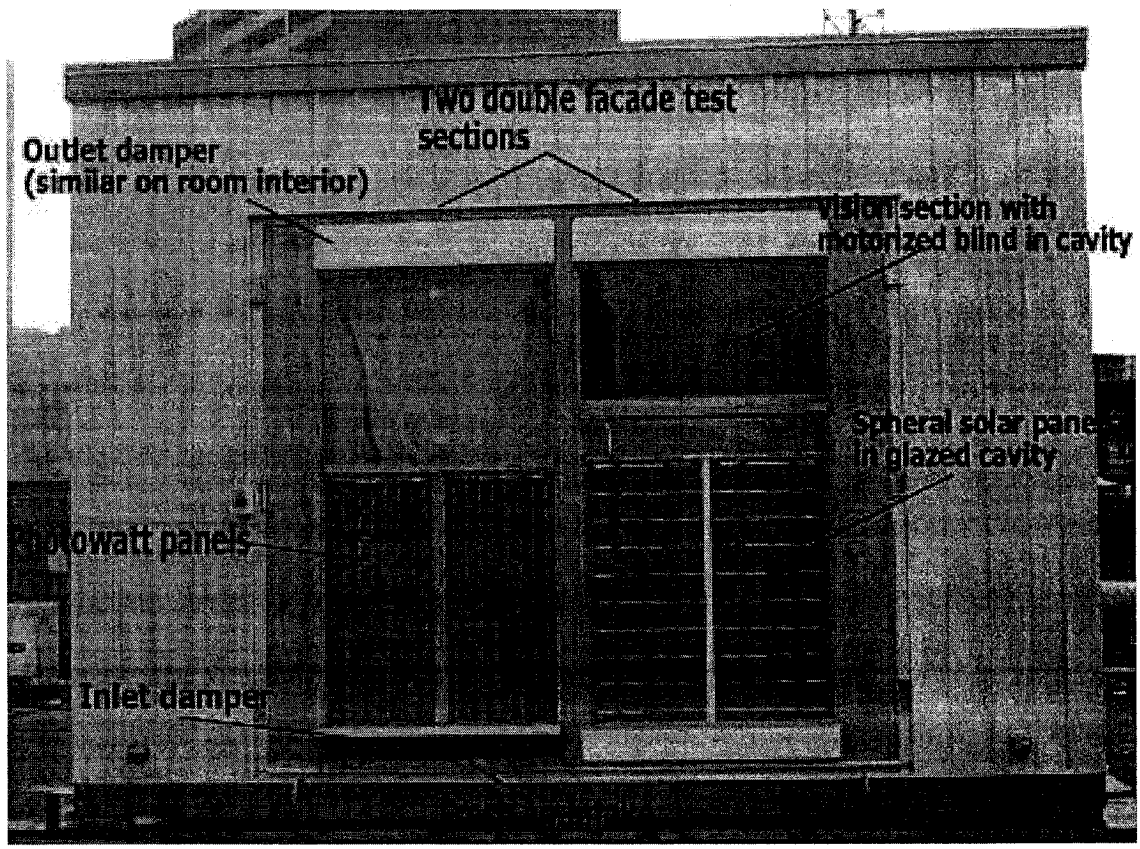

Figure 4.23: Photo of the Double-Façade Configuration of the Concordia Test Room

Experimental data were collected from the PV section of the Spheral Solar side of the façade on December 19th, 2003, and from both sides on January 26th, 2004. Since the model assumes steady state conditions, experimental data needs to be collected in periods of steady solar radiation and outdoor temperatures. The days in question had clear skies with high levels of solar radiation and no wind, and measurements were taken at peak conditions between 11am and noon.

Some issues still need to be resolved with the experimental setup. The thermocouples measuring the exterior glazing temperature in the Photowatt side may not be shielded adequately, which results in temperature readings that are possibly higher

than actual conditions. Therefore, for validation purposes these readings have been ignored. The apparatus used to measure the exterior heat transfer coefficient was not calibrated and the readings obtained in its current state seem to be higher than expected. The days considered had low wind speeds, so an exterior heat transfer coefficient of 12 W/m^2K is used, which corresponds to values obtained from empirical relations.

The convective heat transfer coefficients calculated in the model do not take into account wind effects or the 90 degree bend that the flow must go through at the intake. There are also some wires in the channel for measurement purposes. All of these factors can cause turbulence in the flow, which could increase the convective heat transfer coefficient (CHTC) beyond what was calculated, which also affects the profile of the local CHTC. The experimental results will therefore be verified using calculated and assumed CHTC.

Table 4.3: December 19th Spheral Solar results

Dec 19	Experimental Data	Calculated using 1D model CHTC	Calculated using 1.9 x 1D model CHTC	Calculated using 2D model. CHTC	Calculated using 1.9 x 2D model CHTC
Ave h_{1f_pv}	n/a	2.0	3.8	2.2	4.2
Ave h_{2f_pv}	n/a	4.4	8.3	4.2	8.0
Ave h_{1b_pv}	n/a	5.1	9.6	5.4	10.3
Ave h_{2b_pv}	n/a	3.2	6	3.9	7.4
Tpv	39.6	50.8	36.4	54.8	39.8
Twall	4.4	7.7	4.8	11.5	7.1
Tair Exit Front	19.9	17.8	18.2	16.1	16.7
Tair Exit Back	6	5.7	7	6	7.4
ΔTair Ave	11.5		11.8		11.7

Note: n/a represents not applicable

Table 4.4: January 26th Spheral Solar results

Jan 26	Experi-mental Data	Calculated using 1D model CHTC	Calculated using 1.9 x 1D model CHTC	Calculated using 2D model. CHTC	Calculated using 1.9 x 2D model CHTC	2D input CHTC as in Fig 4.25
Ave h_{1f_pv}	n/a	1.9	3.6	2	3.8	3.8
Ave h_{2f_pv}	n/a	5.0	9.5	4.2	8.0	7.4
Ave h_{1b_pv}	n/a	5.4	10.3	4.8	9.1	7.9
Ave h_{2b_pv}	n/a	3.1	5.9	2.7	5.1	5.1
Tpv	32.4	40.2	27.1	47.7	32.6	31.7
Twall	-7.4	-2.1	-4.9	3.2	-1.5	0.5
Tair Exit Front	21.6	15.1	15.2	14.5	14.1	15.7
Tair Exit Back	2	-3.8	-1.2	-3.5	-1.3	-1.2
ΔTair Ave	24.6		19.8		19.9	

Note: n/a represents not applicable

Table 4.5: January 26th Photowatt results

Dec 19	Experi-mental Data	Calculated using 1D model Calc.CHTC	Calculated using 1.9 x 1D model Calc. CHTC	Calculated using 2D model. Calc. CHTC	Calculated using 1.9 x 2D model Calc. CHTC
Ave h_{1f_pv}	n/a	5.4	10.8	5.0	10.0
Ave h_{2f_pv}	n/a	4.8	9.6	4.7	9.4
Tpv	13.3	21.3	13.6	20.1	13.2
Twall	-4.0	7.1	-1.5	10.8	2
Tair Exit	-10.6	-12.6	-11.3	-13.1	-11.8
ΔTair	6.4	4.4	5.7	3.9	5.2

Note: n/a represent not applicable

As is shown in Tables 4.3 to 4.5, the theoretical results do not correlate well with what was measured when calculated values of the CHTC were used. There is a much better agreement between the results when the CHTC is multiplied by a factor of 1.9 for the Spheral solar side and 2 for the Photowatt results. The factor of 1.9 was appropriate for both dates, even though the experimental conditions varied quite a bit between the two days. Table 4.6 gives a summary of the key parameters used in the investigation. Note that the two models predicted slightly different results from one another since the one-dimensional model used average CHTC and the 2D model used local values. In all cases, the inlet air temperature was assumed to be equal to the outdoor temperature.

Table 4.6: Key parameters from experimental measurements

Parameter	Dec. Spheral	Jan. Spheral	Jan. Photowatt
Outdoor Temp. (°C)	-2.1	-17	-17
Vel. Exterior Cavity (m/s)	0.3	0.15	0.6
Vel. Interior Cavity (m/s)	0.6	0.35	n/a
Solar Radiation (W/m^2)	1000	989	989

As mentioned before, an important aspect of the 2D model is its ability to predict the temperature distribution of the PV. Figure 4.24 shows the temperature distribution of both the experimental results, and the theoretical results using CHTC values that are 1.9 times larger than calculated values. The experimental results show a smooth increase in PV temperature with increasing height. This is due to the high correlation between the CHTC profile and the PV temperature profile. The model algorithm determines the best CHTC correlation to use at each point. In this case, the algorithm changes empirical relation half way up the PV panel. It seems to be a good feature to have since the initial correlation was starting to generate larger errors that the new correlation corrected. A

possible improvement in the model could be to average different correlations, to ensure a smooth transition from one to the other. Empirical relations to calculate the CHTC will rarely be exact; hence, there will always be a slight variation between experimental and theoretical results. The goal is to minimize this error. The CHTC can be manually input to obtain closer a correlation between experimental and theoretical results. Figure 4.25 shows results having modified the CHTC used for Figure 4.24 in order to have a better correlation. This indicates that the entrance affects only the first point, and that the remaining points have fairly constant CHTC values.

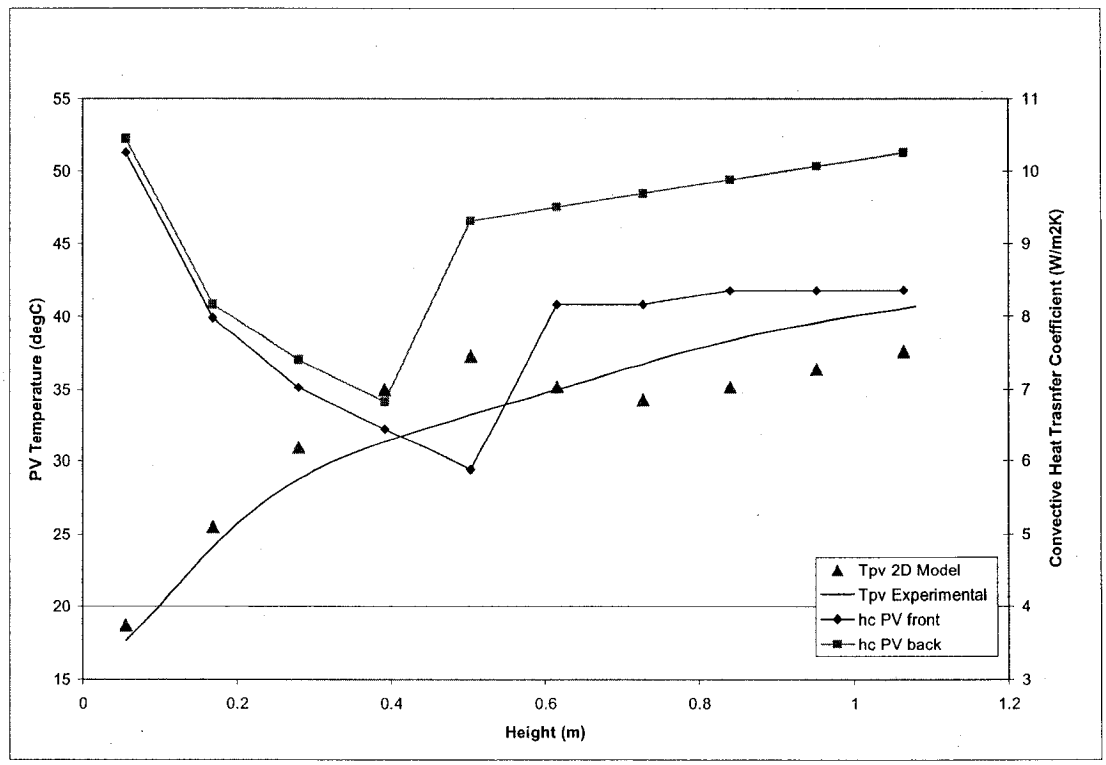

Figure 4.24: Calculated Spheral Solar PV Temperatures Using 1.9 x Calculated CHTC for Jan. 26th Data

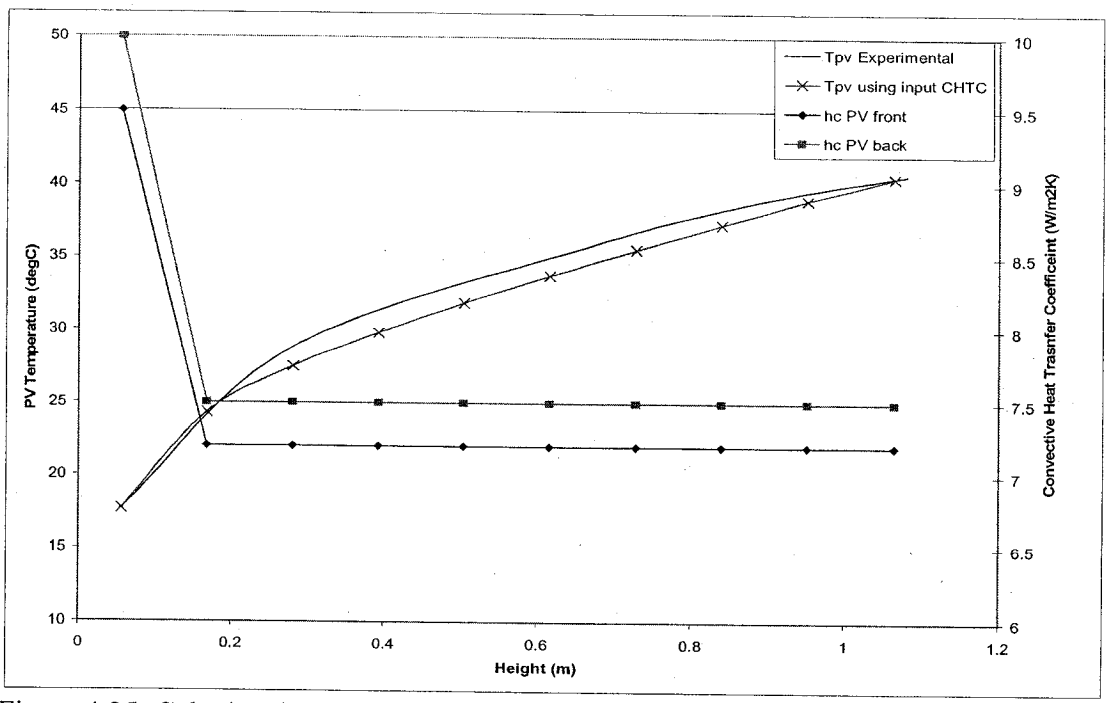

Figure 4.25: Calculated PV Temperature Using Input CHTC

Other than being able to accurately predict the PV temperature, which affects the PV cell efficiency, another important parameter to be able to accurately predict is the exit air temperature. As shown in Tables 4.3 to 4.5, the models tend to under predict the temperature rise. It is also apparent that the CHTC is not the only factor affecting the temperature rise. Multiplying the CHTC by 1.9 in the exterior cavity of the Spheral solar section using December 19th conditions only increased the exit air temperature from 16.1 to 16.7°C in the model, whereas the value was measured as 19.9°C. Many factors can account for this difference. The exterior convective heat transfer coefficient might be lower than 12 W/m^{2}K. The average velocity in the cavity might be lower than the measured 0.3 m/s, where a small difference can have a noticeable impact on the exit air temperature. Another key factor is that it was assumed that the air entering the cavity was at the same temperature as the outside air. However, the air might have been

warmed by the rooftop before entering the cavity, and/or the air temperature might have increased while going through the inlet. This is an effect that researchers have noted should not be ignored. Keeping these factors in mind, the calculated average air temperature increase was within 20% of the measured values, when multiplying the CHTC by 1.9 and 2 as mentioned.

4.5 Conclusion

This analysis has shown that the results obtained from the use of a one-dimensional finite-difference thermal model developed by Charron and Athienitis (2003), correlate well with results obtained from the use of a two-dimensional control-volume model. For certain scenarios, the difference in results between the two models is as high as 20%; however, this difference is accounted for by the fact that the two-dimensional model uses local convective heat transfer coefficients, whereas average values are used in the one-dimensional model. This demonstrates the importance of using an appropriate correlation for calculating convective heat transfer coefficients. Ong (1995) examined various empirical formulas for calculating forced convection heat transfer coefficients between parallel plates and found that the variation between calculated values may range by as much as ±50%. As so many empirical relations exist to calculate convective heat transfer coefficients, more research is required to identify which expressions are most suitable for representing the flow in a double-façade.

A significant finding is that for the gap widths considered in this investigation, there seems to be little impact on the results from using linearised radiative heat transfer coefficients to calculate the long-wave radiation heat transfer. This is important since it

eliminates the need to have the non-linear terms of the radiosity analysis, which reduces the computational requirements of solving the system of equations. To verify that this was the case, radiative heat transfer coefficients were calculated using results of the radiosity analysis, and compared to the linearised values used in the 1D model; the two values were in very good agreement. If the gap width becomes significantly larger, then more errors would be introduced by using linearised radiative heat transfer coefficients. The thermal conduction in the vertical direction within the components seems to be negligible. This may not always be the case; it depends on the width and type of components used.

Even with the difficulties that arose in comparing results with other published results, the 2D model was able to predict the temperature rise in the air cavity occurring in the Mataro Library within 8%. This is very good correlation since some of the system characteristics had to be estimated. Two days of experimental results obtained at the Concordia BIPV test facility were used to validate the model. Results indicate that the convective heat transfer coefficient in the cavities is 1.9 to 2 times higher than the calculated values. This can be attributed to wind effects, entrance effects, and measurement wires in the flow path, all of which could have led to more turbulent flows with higher CHTC values. Using the higher CHTC resulted in a fair correlation between the measured and calculated results; within the 10 to 21% uncertainty levels predicted by other authors (Zollner, et al., 2002), (Balocco, 2002). A perfect correlation between theoretical and experimental results would be very difficult to achieve as the air temperature calculations are sensitive. There is difficulty in adequately modeling the flow resistance of the various components (Hensen, et al., 2002).

CHAPTER 5

RESULTS

5.1 Evaluating System Efficiency

One important, but difficult task is to determine a general expression for calculating the system efficiency of an AFW-IP that can be used by researchers to compare different technologies. The most general expression used to calculate efficiency is to take the ratio of the useful energy delivered over the total incoming solar energy as follows:

$$\eta = \frac{E_{total}}{A \cdot G}. \tag{5.1}$$

Equation 5.1 does not show how any of the system parameters interact to influence the efficiency.

Two expressions are common in rating solar air collectors; these are the collector efficiency factor, F', and the heat removal factor F_R. F' is the ratio of actual useful energy gain, to the gain that would occur if the plate temperature equalled the flow temperature (ideal case $F'=1$). F_R compares the performance of a real collector with the thermodynamic optimum, i.e. it is a ratio of the actual heat transfer to the air, to the maximum possible heat transfer. Take for example the solar collector depicted in Figure 5.1:

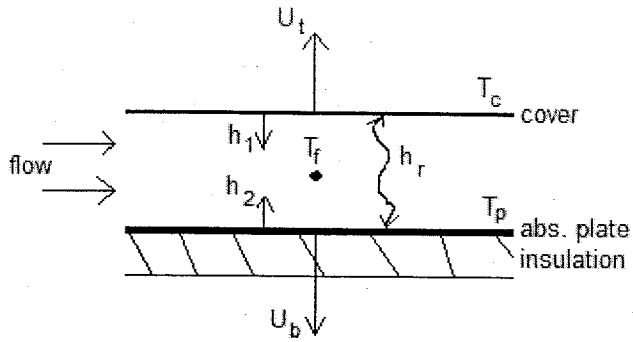

Figure 5.1: Simple Solar Air Collector

The values for F' and F_R can be calculated or found from literature:

$$F' = \frac{h_r h_1 + U_t h_2 + h_r h_2 + h_1 h_2}{(U_t + h_r + h_1)(U_b + h_2 + h_r) - h_r^2}, \tag{5.2}$$

$$F_R = \frac{\dot{m} c_p}{A_p U_L}\left(1 - e^{-A_p U_L F' / \dot{m} c_p}\right), \tag{5.3}$$

where

$$U_L = \frac{(U_b + U_t)(h_r h_1 + h_2 h_r + h_1 h_2) + U_b U_t (h_1 + h_2)}{h_r h_1 + h_r h_2 + U_t h_2 + h_1 h_2}. \tag{5.4}$$

Unlike equation 5.1, which was totally general, the expressions for F' and F_R are given in terms of known physical parameters. Although it may not be apparent from the equations exactly how any given term will affect the overall result, it can be easily calculated using a variety of computer programs. Even though expressions to calculate the collector efficiency factor and the heat-removal factor of different solar collector configurations will differ, the result can be used to compare the performance from one configuration to another.

Can similar expressions be derived for evaluating an AFW-IP? The answer is not so simple. Solar collectors have one main purpose: to convert incident solar radiation

into useful thermal energy. AFW-IP on the other hand have a variety of different uses. The most important, and most costly, is its photovoltaic component that converts incident solar radiation to electricity. Electricity is more valuable than thermal energy, and any expression comparing the performance of different types of AFW-IP should take this into account. In addition, if the window is used for daylighting purposes, then a portion of the transmitted daylight will offset the building's electricity consumption. Not all configurations of the AFW-IP will use the gained thermal energy in the building. In certain applications, the thermal energy will simply be exhausted to the outdoor environment.

In order to develop an appropriate expression, the intended use of the AFW-IP would have to be known. Questions such as the following would have to be answered for each particular design:

- Is the thermal energy being utilised to offset the building's thermal energy load?
- Is daylighting being used in conjunction with the design?
- How much is electrical generation worth compared to thermal energy?
- How much of the exhausted heat is offsetting the cooling load?

Different expressions need to be developed for each different application of the AFW-IP; unfortunately, this goes beyond the scope of this thesis.

In this thesis, the system efficiency used to compare results from one configuration to the next is defined as follows:

$$\eta = \frac{E_{pv} + Q_{air} + E_{trans} \pm Q_{room} - E_{fan}}{G \cdot A}. \tag{5.5}$$

This expression is simply the general expression for calculating system efficiency that was presented earlier. The heat transfer to the room is added in the heating season, and

subtracted when cooling is required. This expression can be used as a total system efficiency or for each section separately. The energy used by the fan to circulate the air should be subtracted from the gained energy. However, in this case, the power required to drive the fan was found to be small relative to the other terms and was omitted.

5.2 Heat Conduction in the Air

As with several models found in literature, the diffusion term is ignored in the air cavity. This assumption is valid if the heat transfer due to convection is much greater than the heat transfer due to diffusion. A good way to verify if this assumption is valid is to calculate the Peclet number, *Pe*, which is defined as the ratio of the strengths of convection to diffusion:

$$Pe = \frac{\rho \cdot u \cdot dx}{k_{air}}. \tag{5.6}$$

When the Peclet number is large, then convection is more dominant; for *Pe* larger than 2, convection dominates and diffusion can be ignored (Charron, Athienitis, 2003). In the case of the airflow window, the air density and thermal conductivity remain relatively constant. Hence, the velocity and height of each element are the important criteria to examine. In order to have insignificant diffusion, the following criteria must be met:

$$\frac{2 \cdot k_{air}}{\rho} \approx \frac{2 \cdot 0.026}{1.16} \approx 0.045 < u \cdot dx. \tag{5.7}$$

In general, the criterion set in equation 5.7 is met when modeling the AFW; however, for fine grids, combined with low velocity, this may not be the case. For example, a velocity of 0.3 m/s is quite possible, and if the nodes are smaller than 0.15 m, then *Pe* will be smaller than 2. It is relatively simple to verify the effect of the thermal

conductivity in the air in the 2D model by adding an additional term to the energy balance equation. These are represented as equation 5.8a, without including the thermal conductivity in the air, and 5.8b with the added term. The one-dimensional model produces results that differ from the two-dimensional model mainly due to the differences in the calculated convective heat transfer coefficients. Verifying the effect of diffusion in the two-dimensional model, should in essence verify the effect of diffusion in the one-dimensional model.

$$0 = Vel\cdot\rho\cdot C\cdot\left(T_i - T_{i-1}\right)\cdot w\cdot L_{gap} - h_1\cdot\left(T_{i-N} - T_i\right)\cdot w\cdot dx - h_2\cdot\left(T_{i+N} - T_i\right)\cdot w\cdot dx \tag{5.8a}$$

$$0 = Vel\cdot\rho\cdot C\cdot\left(T_i - T_{i-1}\right)\cdot w\cdot L_{gap} - h_1\cdot\left(T_{i-N} - T_i\right)\cdot w\cdot dx - h_2\cdot\left(T_{i+N} - T_i\right)\cdot w\cdot dx - \frac{k_{air}\cdot L_{gap}\cdot w}{dx}\cdot\left(2\cdot T_i - T_{i+1} - T_{i-1}\right) \tag{5.8b}$$

The added term in equation 5.8b should be insignificant in most cases, since the thermal conductivity of air is very low, 0.026 W/m·K, at room temperature (Incropera, De Witt, 1990).

A simple way to verify the importance of the added term is to take the sum of the absolute value of all the heat transfer that occurs due to the added term over the total available energy:

$$\frac{\sum_{i=0}^{N}\left|\left(\frac{k_{air}\cdot L_{gap}\cdot w}{dx}\right)\cdot(2T_i - T_{i+1} - T_{i-1})\right|}{Total\ Available\ Energy}. \tag{5.9}$$

If the ratio in equation 5.9 is small, then the overall effect of the added term in equation 5.8b would be insignificant. After running many different test cases with the two-dimensional model, it was found that the added term for diffusion had no apparent affect on the model. This can be demonstrated by taking an extreme case, i.e. gap width of 1 m, elemental height of 0.02 m, velocity of 0.2 m/s. Even at these limiting

conditions, the ratio of equation 5.9 is under 1.7 x 10^{-4}, explaining the lack of diffusion effects in the model.

5.3 One-Dimensional Model Results

In order to compare the overall performance of different configurations of the AFW-IP, a solar radiation model based on a monthly average clearness index is used to obtain average yearly efficiencies. The environmental conditions used in the model are representative of what would be experienced in Montreal, Canada, which is at a latitude of 45° and a longitude of 74°. The transmittance, reflectance, and absorbance of the glazings, which depend on angle of incidence, are calculated on an hourly basis.

5.3.1 PV Section

In order to increase the heat removal rate of the PV, the back-plate where the PV cells are mounted could have fins incorporated into it. The effect of integrating fins to the back-plate is modeled using fin efficiency for an adiabatic fin with corrected length (L_c) (Incropera, De Witt, 1990):

$$\eta_{fin} = \frac{q_{fin}}{q_{\max}} \cong \frac{\tanh(mL_c)}{mL_c}, \tag{5.10}$$

where

$$mL_c = \left(\frac{2h}{k \cdot A_p}\right)^{0.5} L_c^{1.5}, \tag{5.11}$$

and $L_c = L_{fin} + t/2$, $A_p = L_c \cdot t$.

Various fin configurations are modeled to study how efficiency is affected. Figure 5.2 shows efficiency as a function of fin length, for a fin thickness of 2 mm, gap

width 0.10 m, air velocity of 1 m/s, and PV section height of 1.5 m. In all cases, the fin spacing is set to three times the fin thickness to ensure adequate convection/radiation heat transfer (Rolle, 2000).

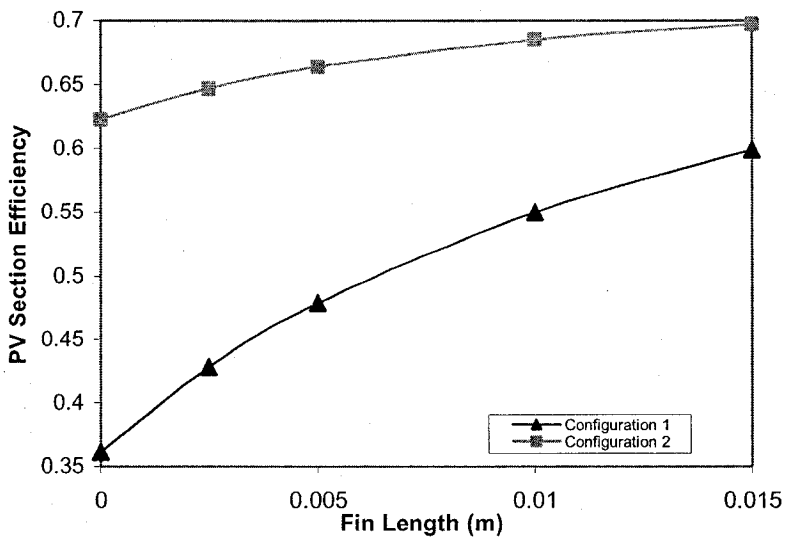

Figure 5.2: PV Section Efficiency vs Fin Length

Figure 5.2 shows the effect of the two most influential factors for the PV section efficiency. The most important is the difference in configuration as depicted in Figure 1.3. Having airflow on both sides of the PV, as opposed to on only one side, increases the average yearly combined thermo-electric efficiency from 36% to 62%. This increase occurs since less heat is rejected outside and is instead captured in the air flowing through the cavity. This increased efficiency may or may not be beneficial, as less electricity is generated and not all of the heat captured may be fully utilised.

Adding 0.015 m long fins increases the average yearly efficiency of Configuration 1 by 24%, whereas the increase is less than 8% for Configuration 2. Gains in efficiency obtained by having air flow on both sides of the PV can be replicated by

incorporating fins on the back-plate of the PV. It is important to note that these calculations are done assuming that the fins are an integral part of the back-plate. If instead fins are glued or mechanically attached to the back-plate, thermal contact resistance between the back-plate and the fins could seriously impact performance.

5.3.2 Vision Section

The optical properties and location of the blind have a significant impact on the overall efficiency of the window section. Two different placements are considered. The first is as depicted in Figure 1.3, where air flows on only one side, having a closed cavity on the other. The second blind placement is in the middle of the channel in the Vision section. The blind properties listed in Table 4.2 are considered.

Average annual vision section efficiency at flow velocities of 1 m/s, gap width of 0.10 m, and height of 1.5 m, for the four blinds, both centred and off to the side, are shown in Figure 5.3. On average, the extra heat extracted by having flow on both sides accounts for a 5% increase in efficiency. Blinds with a higher reflectivity have a lowered efficiency as more solar energy is reflected out of the system. Approximately 46% of the solar radiation incident on the Vision section is reflected out with Blind 2, compared to only 29% with Blind 3.

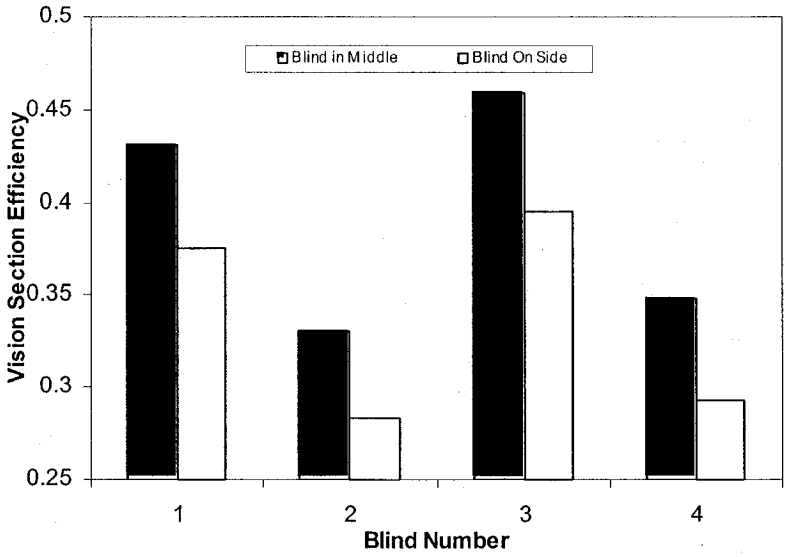

Figure 5.3: Effects of Blind on Vision Section Efficiency

5.3.3 Combined Efficiencies

Certain parameters affect the efficiency of both sections of the AFW. Three of these parameters, flow velocity, mass-flow rate, and gap width, are inter-related, i.e one cannot be changed without affecting the others.

Higher flow velocities lead to higher heat transfer coefficients, which in turn increase the efficiency of the system. Higher velocities are achieved by increasing the mass-flow rate, decreasing the gap width, or a combination of the two. Making the cavity too small, increases the pressure drop of the airflow to a level that fan power becomes an important factor. In addition, gap widths that are too small could be difficult to clean and maintain. One the other hand, the width of the exterior façade must not be too wide as this would take up valuable real-estate space.

Each system has a maximum mass-flow rate that can be handled efficiently. Up to this maximum, as the mass-flow rate increases, the amount of heat that can be removed

by the air increases, leading to an increase in system efficiency. If the pre-heated air is used to meet the outdoor air requirements of a building, the mass-flow rate should not exceed this requirement. If higher rates are used, higher heating costs could result, as the air may require more heating after exiting the AFW. Also, if the velocity increases too much noise can become a problem.

Figure 5.4 demonstrates how the overall efficiency is affected on an average yearly basis, when changing the velocity. The velocity is changed by altering the gap width, while keeping a constant mass-flow rate. The results are for both configurations with the blind centred in the cavity. From Figure 5.4, it can be seen that changing the gap width while keeping the mass-flow rate constant can significantly affect efficiency.

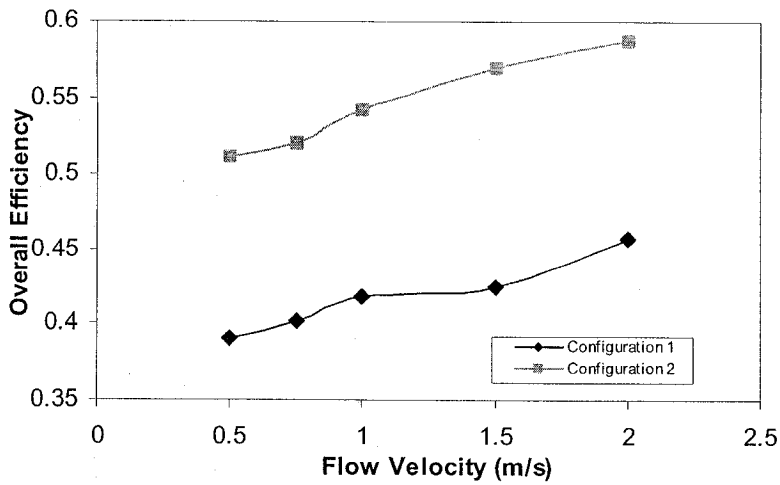

Figure 5.4: Overall Efficiency of AFW as a Function of Velocity for a Constant Flow Rate (Gap width varied from 0.05 to 0.20 m)

Figure 5.5 shows how the efficiency is affected on an average yearly basis, as a function of velocity due to a changing mass-flow rate. The gap width is assumed constant at 0.10 m. As expected, the results are very similar to those for changing gap width. The differences between the two cases are due to the differing heat transfer

coefficient expressions that are used depending on the parameters. Provided noise and maximum fresh air ventilation requirements are not exceeded, increasing the flow rate can increase efficiency considerably.

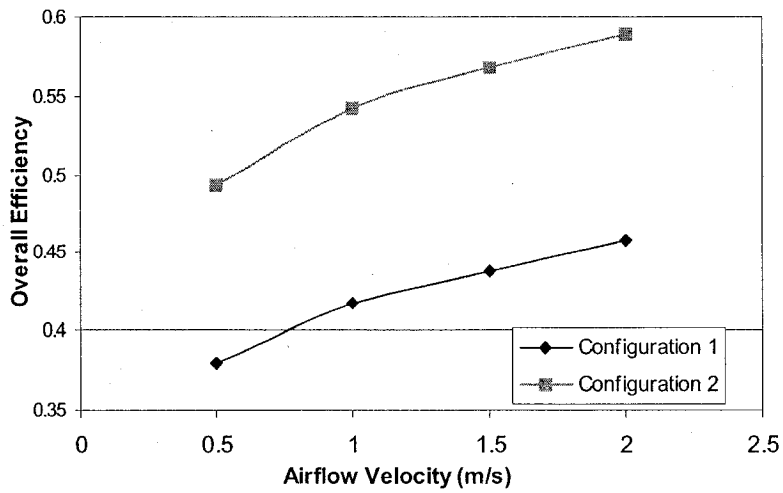

Figure 5.5: Overall Efficiency as a Function of Velocity at Constant Gap Width of 0.10 m

5.3.4 Expected Yearly Output

Up to this point, the analysis has focused on system efficiencies; however, other factors are also important in determining an optimal configuration. For example, the large increase in system efficiency realised in Configuration 2, comes at a significant price in terms of electricity production. This is due to the fact that the added glazing reflects significant amounts of irradiation out of the system at low angles of incidence.

At a flow velocity of 1 m/s, total gap width of 0.10 m, and a PV height of 1.5 m, the PV section of Configuration 1 produced 155 kWh of electricity per meter width and 331 kWh/m of heat. The electricity produced in Configuration 2 is 21% lower at 123 kWh/m, and the heat is more than 109% higher at 692 kWh/m.

The heat gained in both configurations may or may not be useful. During periods when cooling is required, this heat could either be rejected outdoors or accessed via a heat exchanger to preheat domestic hot water. Furthermore, in the heating season, only the portion of the pre-heated air needed to meet a building's fresh air requirements would be useful. In most instances, it would be more economical to use recycled air for the balance of air requirements. In addition, the PV daily average temperature in Configuration 2 is 14.0°C, whereas it is only 12.9°C in Configuration 1.

As mentioned, a compromise between the two configurations would be to use Configuration 1 with fins. This option produces 156 kWh/m of electricity and 642 kWh/m of heat in the PV section, while keeping the average PV temperature at 11.1°C, using 0.015 m long, 0.002 m wide fins.

As discussed, an important application of the AFW-IP would be to preheat air going into an HVAC system. Figure 5.7 shows the predicted temperature rise of the air using Configuration 2, with an outside air temperature of 0°C, solar irradiation of 600 W/m^2, air velocity of 1 m/s, and a total gap width of 0.10 m. Figure 5.6 shows the predicted of air temperature rise using Configuration 1, with and without fins of 0.015 m length, and 0.002 m width, for the same conditions. As shown, the AFW increases the air temperature of 100 L/s of air per meter width of window by 4 to 7°C, depending on the configuration.

Again, by comparing Figures 5.6 and 5.7, it can be seen that incorporating fins to the PV back-plate in Configuration 1 gives similar results to changing from Configuration 1 and 2. Another important point is that the average PV temperature is 25°C without fins, and only 14°C with fins. If the outside temperature is increased to

30°C, and irradiance to 1000 W/m^2, then the average PV temperatures increase to 70°C and 53°C, respectively for Configuration 1. Fins would help reduce average PV temperature significantly at peak conditions, which would help to prolong the life expectancy of the PV cells.

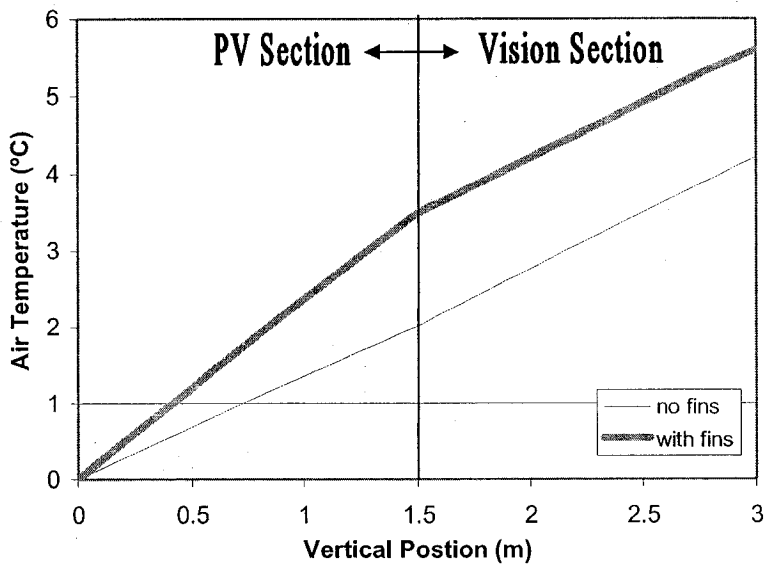

Figure 5.6: Air Temperature Profile, Configuration 1

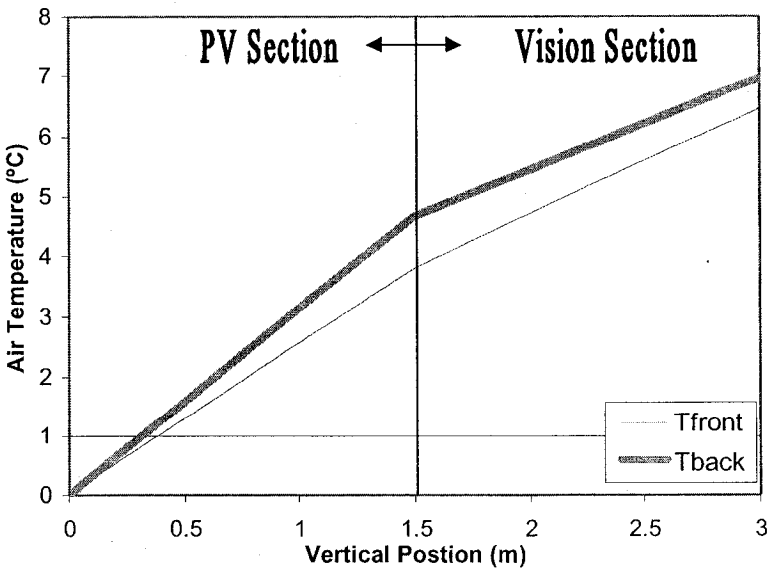

Figure 5.7: Air Temperature Profile, Configuration 2

5.4 Two-Dimensional Model Results

One of the major advantages of the two-dimensional model is that it can provide a better representation of the temperature profile in the PV panels. The average temperature obtained with the one-dimensional model is not the maximum temperature that the PV cells will reach. Generally, the top of the panel will be hotter than the bottom. The maximum temperature will lead to thermal stresses and reduction of PV module efficiency and lifetime; it is therefore important to predict this temperature accurately for system design. Table 5.1 gives results of the minimum and maximum PV cell temperatures obtained with different conditions. All the results are calculated using an outdoor temperature of 30°C, 800 W/m^2 incident solar radiation, and a total gap width of 0.10 m.

Table 5.1: PV minimum and maximum temperatures with varying air velocity

Configuration	Air Velocity (m/s)	Fins ?	Minimum Local PV Temperature (°C)	Maximum Local PV Temperature (°C)
1	0.30	No	65.2	67.2
1	0.30	Yes	52.5	58.9
1	0.65	No	60.2	65.0
1	0.65	Yes	45.2	54.7
1	1.00	No	58.3	64.2
1	1.00	Yes	43.0	52.9
2	0.30	No	71.4	79.4
2	0.65	No	61.2	72.8
2	1.00	No	57.2	70.2
2	1.00	Yes	42.7	57.8

As is shown in Table 5.1, certain system parameters have a large influence on the maximum expected PV temperature. The greatest difference, 26.5°C, occurs between Configuration 2 with low flow velocities and no fins, and Configuration 1 with high flow velocities and integrated fins. This shows the importance of properly sizing the AFW-IP for a building. The gap width should be selected in order to optimise the flow velocity based on fresh-air requirements. If pre-fabricated modules are selected that do not offer the flexibility of gap width selection, then the collector width needs to be selected to provide optimum conditions.

5.4.1 Results from Modeling Concordia Test Room

The models developed in this thesis will be used to verify experimental results obtained from the new test room constructed on the roof of Concordia's engineering building; therefore, results calculated from differing test conditions would be useful. The experimental results presented in Chapter 4 were only preliminary, and were from only one day. This section presents results of the single cavity configuration of the test room for a wide range of parameters.

The PV parameters were set to those given by the manufacturer. These are listed in Table 5.2. The table gives values for one panel; there are two panels in the test room. Note that the results presented in this section are for constant glazing transmittance and reflectance.

Table 5.2: Properties of Photowatt panels

Property*	**Photowatt PWX 500**
Number of solar cells	36
Dimensions of PV module with frame (mm)	1042 mm (length)

	462 mm (width)
Typical power (W)	47.5 W
Voltage at typical power (V)	17 V
Current at typical power (A)	2.8 A
Nominal Operating Cell Temperature (°C) (0.8 kW/m^2, 20 °C, 1 m/s)	45
Open circuit voltage (V)	21.5 V
Short circuit current (A)	3.1 A
Solar cell characteristics	
Technology	Multi - crystalline
Size of solar cell (mm^2)	101.25 X 101.25
Thickness (μm)	300 ± 50
Temperature coefficients Power Current Voltage	 Gamma = (dP/P)/dT # - 0.43 %/°C Alpha = (dI/I)/dT # + 0.034 %/°C Beta = dV/dT # - 2.17 mV/°C

*At standard test conditions: 1 kW/m^2, AM 1.5 @ 25 °C, Source: Photowatt Inc.

Figure 5.8a shows the effect of solar radiation on PV efficiency and electricity generation for different outdoor temperatures. For the case with an outside temperature of 30°C the efficiency drops linearly with increasing solar radiation. At -20°C solar radiation has no impact on cell efficiency. In reality this would not be the case, as cell efficiency is not only a function of temperature, as was modeled, but is also dependent on incident solar radiation (Poissant, et al., 2003). At 5°C, it takes about 550 W/m^2 to bring the PV cell temperature above the nominal 25°C to start affecting the efficiency. Therefore it takes the right combination of temperature and incident solar radiation to reach nominal temperatures when PV efficiency starts to decline.

Figure 5.8b shows the temperature increase in the flowing air as a function of solar radiation for different outdoor conditions. At low irradiation levels, a greater temperature increase is achieved with lower outdoor temperatures. This is expected as lower outdoor temperatures offer a larger temperature gradient between the flowing air and the components of the AFW, and thus a larger heat flow. It is interesting to note that as the level of incident solar radiation increases, this trend reverses and higher outdoor temperatures result in larger temperature gradients. This can be explained by the effect of PV cell efficiency on operating temperature. The higher the temperature, the lower the efficiency, which leads to a greater portion of incident solar radiation being converted to heat as opposed to electricity. The more heat that is available, the more the air temperature can increase.

Figure 5.8c shows how PV cell temperatures and the temperature of the back glazing of the PV panel are affected with varying incident solar radiation for different outdoor temperatures. The trend is similar for all outdoor temperatures: there is a linear increase in PV temperature with increasing solar radiation. The slope is slightly more elevated for outdoor temperatures of 30°C as opposed to -20°C, but the difference is minimal; at 30°C the slope is 0.0363°C/(W/m^2), and at -20°C it is 0.0349°C/(W/m^2), which is a difference of only 3.9%. Not only does the slope change but the y-intercept is different; this value changes by 2.2°C relative to the outdoor temperature between the two extreme cases.

Figure 5.8d shows results with a constant incident solar radiation value of 700 W/m^2 with varying outdoor temperature. The air temperature increase in the cavity is more or less constant at 2.4°C. The PV cell temperature increases linearly with

outdoor temperature at a slope of 0.986 PV °C/OA °C. The PV efficiency, and by association generated electricity, remain constant until the outdoor air temperature reaches approximately 0°C, at which point the efficiency starts a linear decline with a slope of 0.05%/OA °C. If the rate at which PV efficiency drops with PV temperature is required, the efficiency slope is divided by the temperature slope, which gives a value of 0.051% / PV °C. This value is slightly higher than the range of coefficients suggested by Sandberg (1999), which had an upper limit of 0.046% / PV °C.

Figure 5.8e shows results with a constant irradiation and outdoor temperature with varying air velocity. The air temperature increase drops with increasing velocity. The value goes from 3.9 to 1.6°C when the velocity increases from 0.3 to 1 m/s, respectively. Even though the temperature increase with higher velocities is lower, the total amount of thermal energy transported is greater since more air is available to transport the energy. As mentioned previously, the increase in thermal efficiency is generally only useful when the air is used as pre-heated air for the ventilation system up to the amount need to meet outdoor air requirements for the building. Beyond this limit, extra energy will generally be required to heat the fresh air to an acceptable level. For the same increase in air velocity, the PV efficiency increases by 0.19%, which results in an increase in power generation of only 0.8 W. Therefore if the only concern is electricity generated, small gains are achieved by increasing the flow velocity, especially if fan power is considered. The maximum PV temperature was lowered by 2.6 °C over the range of velocities considered.

Figure 5.8f shows results with varying outdoor heat transfer coefficient. The tests done in the test room did not take measurements of the outside glazing temperature, or

wind speeds in order to calculate the outdoor heat transfer coefficient. As mentioned previously, coefficients in literature are often constant and cover a wide range of values. The results show that this factor should not be ignored when doing simulations as it has a significant impact on results; a difference in coefficient of only 5 W/m^2K changes cell efficiency by as much as 0.6%, PV temperature by 12.5°C, and air temperature by 1.3°C. Changes in wind speeds from 0.5 to 2.2 m/s changes the outdoor heat transfer coefficient from 10 to 15 W/m^2K. Considering that the RETScreen database lists the average wind speed in Montreal as 4 m/s, this variation is likely to occur. A difficult aspect to model is that the changes that occur due to outdoor heat transfer coefficients are not linear as with a lot of the other variables.

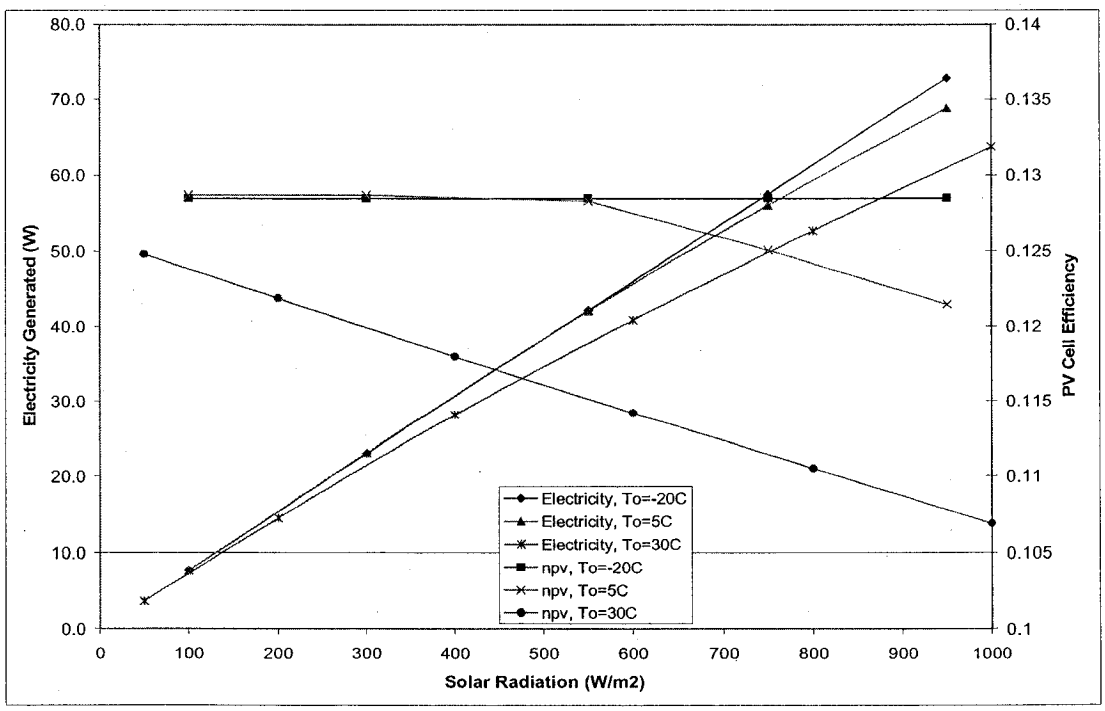

Figure 5.8a: Generated Electricity vs Solar Radiation with Vel=0.6 m/s, ho=14 W/m^2K for Photowatt Side of Test Room

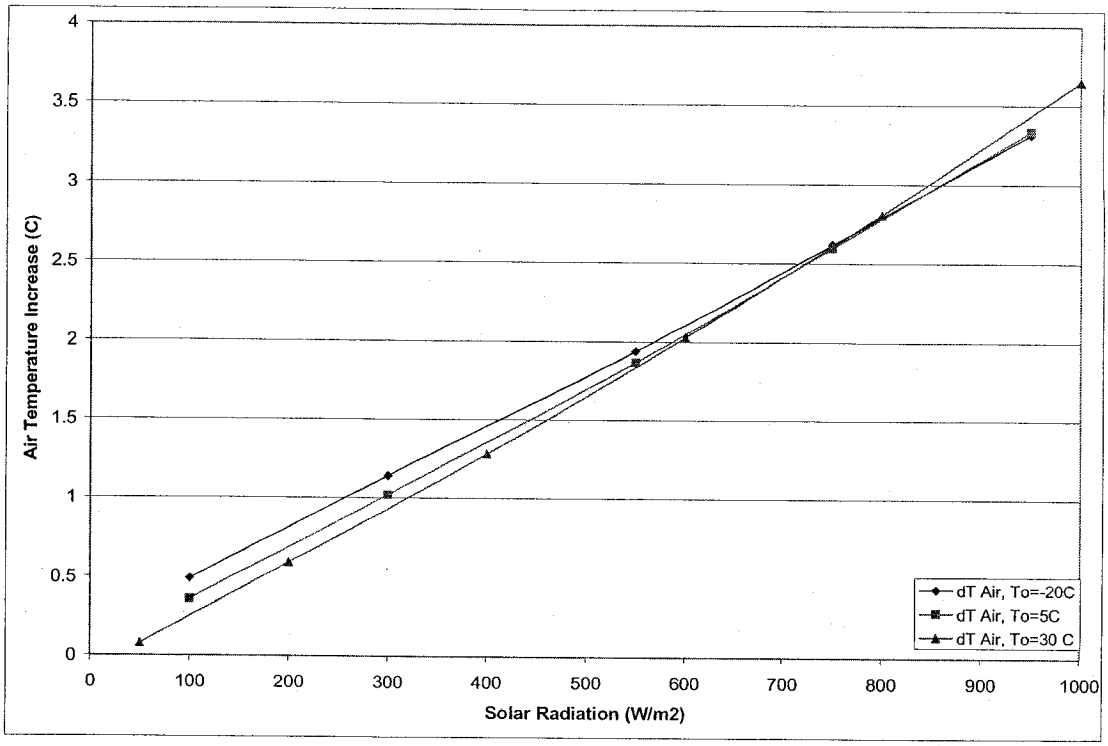

Figure 5.8b: Air Temperature Increase vs Solar Radiation, with Vel=0.6 m/s, ho=14 W/m^{2}K for Photowatt Side of Test Room

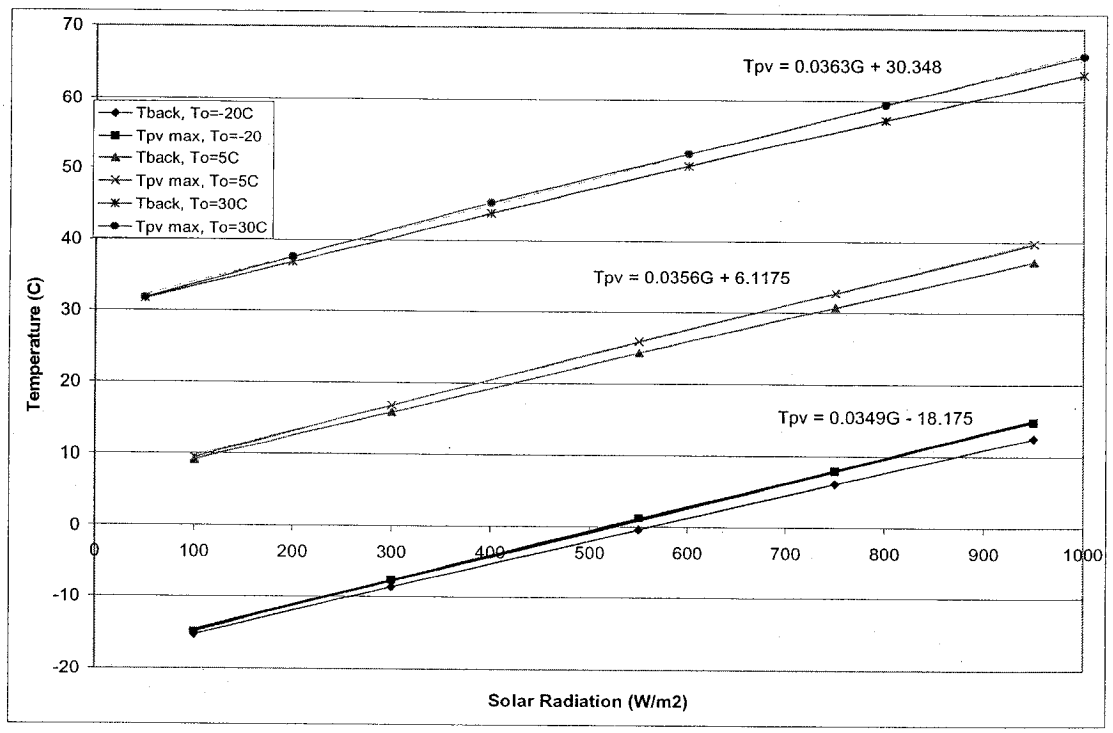

Figure 5.8c: PV Temperature vs Solar Radiation, with Vel=0.6 m/s, ho=14 W/m^{2}K for Photowatt Side of Test Room

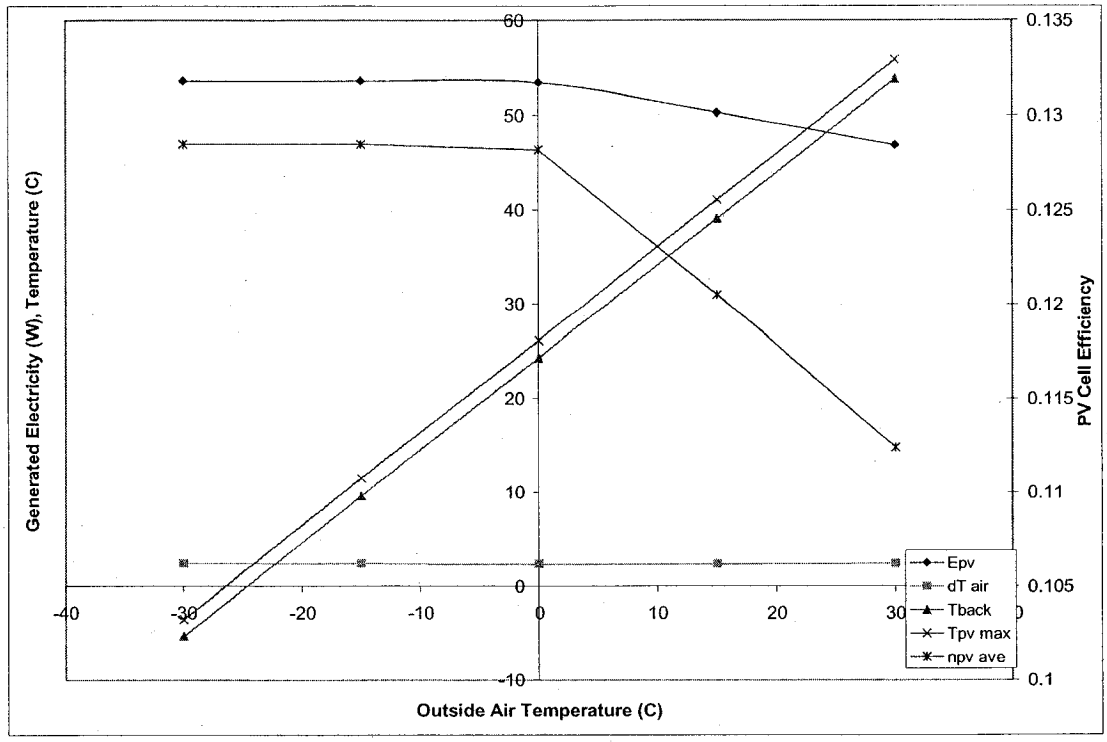

Figure 5.8d: Results with Varying Outside Air Temperature, with Vel=0.6 m/s, G=700W/m^2, ho=14 W/m^{2}K for Photowatt Side of Test Room

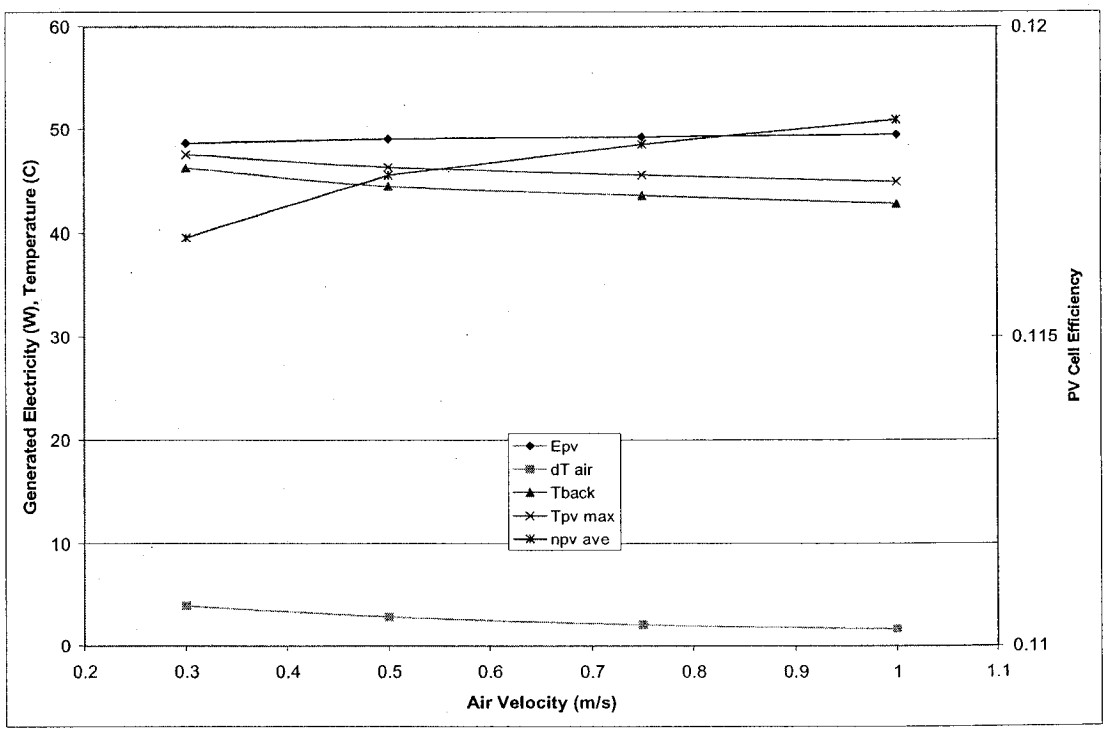

Figure 5.8e: Results with Varying Air Velocity, with G=700 W/m^2, To=20°C, ho=14 W/m^{2}K for Photowatt Side of Test Room

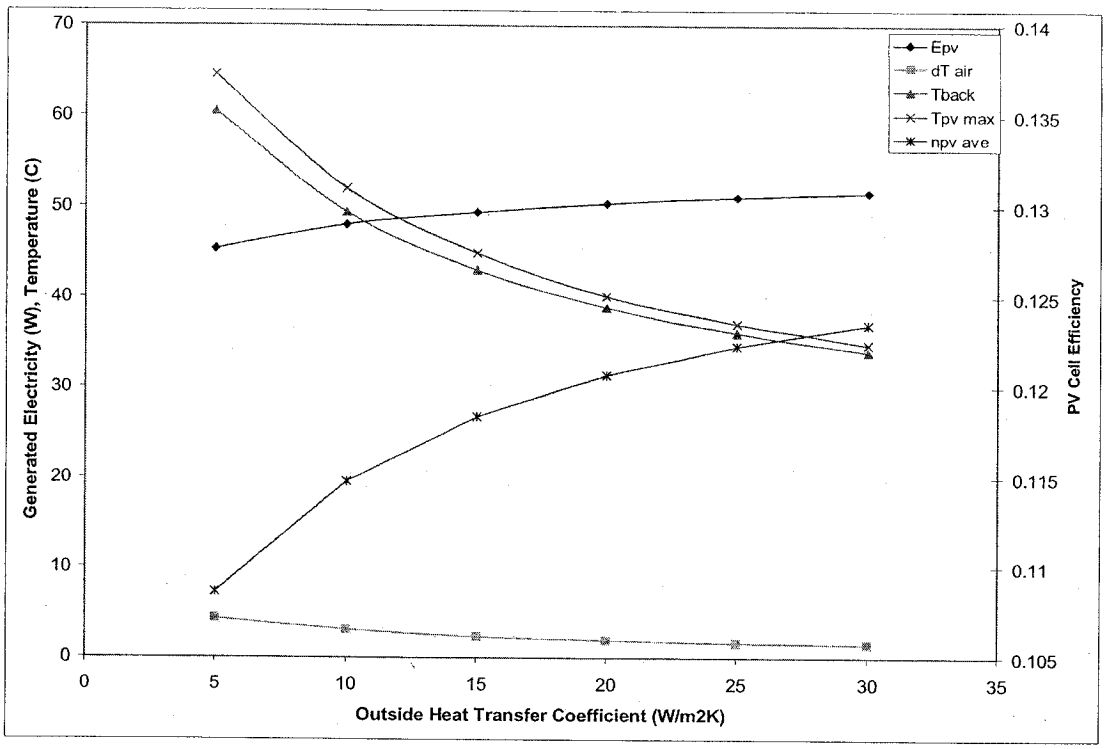

Figure 5.8f: Results with Varying Outside Heat Transfer Coefficient, To=20 °C, G= 700 W/m^2, Vel=0.6 m/s for Photowatt Side of Test Room

5.5 Conclusion

Investigation of the key parameters that affect the performance of PV-integrated façades is important. From the results presented in this chapter, it was seen that the different intended uses of the facades result in different optimal configurations. For example, if the particular façade does not require a lot of daylight, more than 50% can be covered with PV. Whether the heat absorbed in the air is utilized, or if it is simply rejected outdoors, plays an important role in determining certain parameters. In both cases, maximizing the heat removal rate is beneficial as it either reduces heating or cooling loads.

To review, PV modules placed in the middle of the cavity allow more heat to be captured as less is rejected; unfortunately, the added glazing reduces the amount of solar

radiation incident on the PV cells, thus reducing electricity generation. Integrating fins with the PV back plate has a similar effect to placing the PV inside the cavity; operating temperatures are reduced without affecting the electricity generating capacity. A low reflectivity blind may be installed in the middle of the Vision section. The transmittance of the blind should be large enough to provide adequate daylighting without causing occupant discomfort. If the preheated air is used to meet fresh air requirements, the total mass flow rate of the air should not exceed the building occupants' requirements. However, if air is being rejected outside, higher flow rates in the cooling season, and lower flow rates in the heating season are preferable. The gap width should be minimised to reduce space requirements and increase the air velocity; however, noise considerations must be taken into account. Gap widths should be large enough to allow for maintenance.

To demonstrate the importance of the above recommendations, two different configurations are tested. The first has Blind 3 in the middle of the cavity, using Configuration 1 with 0.015 m long fins, a gap width of 0.10 m and a velocity of 1 m/s. The second configuration has blind 2 on the side of the cavity, using Configuration 1 with no fins, a gap width of 0.20 m and velocity of 0.5 m/s. The average annual efficiency of the more optimal configuration, listed first, is 55%, whereas it is only 30% in the other case. This demonstrates a gain in efficiency of over 80% by optimising the configuration.

Another aspect that should not be ignored is the maximum PV cell temperature. The two-dimensional model was used to calculate this parameter by varying several parameters. While maintaining environmental conditions, the maximum cell temperature

varied from 52.9°C to 79.4°C. This difference of 26.5°C could seriously impact projected cell efficiencies and life expectancies. PV panels are the most expensive element of an AFW-IP, so it is important to optimise its performance, while at the same time extending its lifetime.

The last section of this chapter examines the expected performance of the Photowatt side of the test room built at Concordia University. The effects that solar radiation, outdoor temperature, flow velocity, and the outdoor heat transfer coefficient have on PV temperature, PV efficiency, and the increase in air temperature are investigated. The majority of the parameters are affected in a linear fashion when varying the operating parameters. The main exception is the effect of varying the outdoor heat transfer coefficient; this produces results in the form of second order polynomials. The problem that occurs in real test conditions is that all parameters vary simultaneously, producing non-linear results, which are hard to predict.

CHAPTER 6

CONCLUSION

6.1 Conclusion

One-dimensional and two-dimensional models were developed to study a double façade with integrated photovoltaic panels that provides simultaneous electricity generation and fresh air preheating. Chapter 4 shows that the two models give very similar results. The only factor that seemed to differentiate the results between the two models was the use of local convective heat transfer coefficients in the two-dimensional model, as opposed to average values in the one-dimensional model. Due to the inherent uncertainties that calculating the coefficients introduce, it was not clear if there was any advantage in using local values. Chapter 4 went on to show that the models could produce results to within 10% of a model that was developed for, and validated with the AFW-IP at the Mataro Library in Spain (Infield, et al., 1999). Promising results were also shown when the model was used to verify results that were obtained from the Concordia University BIPV test facility. A major problem associated with the attempt at validation was that the convective heat transfer coefficients in the cavities were 1.9 to 2 times higher than calculated values. This could be attributed to wind effects, entrance effects, and measurement wires in the flow path, all of which could have lead to more turbulent flows and hence higher CHTC values. Using the higher CHTC resulted in fair correlation between the measured and calculated results; values were within the 10 to 21

percent uncertainty levels predicted by other authors (Zollner, et al., 2002), (Balocco, 2002).

Chapter 5 examined how various parameters of the AFW-IP affect the performance of the system. Placing the PV in the middle of the cavity, as was done for Configuration 2, increased the overall efficiency of the system significantly. However, the gains in efficiency were all thermal, and the extra glazing actually reduced the amount of electricity generated by approximately 20%. Integrating 0.015 m long fins to the back-plate of the PV in Configuration 1 also increased the efficiency by a significant amount. In that case, the rise in efficiency was only thermal, but there was a slight increase in electricity production due to the lowered PV operating temperatures. In the Vision section, it was shown that it is beneficial to place the blind in the middle of the cavity as the system efficiency generally increased by approximately 5%.

The last section of Chapter 5 examined the effects that solar radiation, outside air temperature, air velocity in the cavity, and outdoor heat transfer coefficients will have on the Concordia University test room. With the exception of the outdoor heat transfer coefficient, the parameters generally caused a change in the predicted exit air temperature and the power generation in a linear fashion. The results showed the importance of using an appropriate value for the outdoor heat transfer coefficient as this parameter had a significant impact on results.

6.2 Future Work

As mentioned, AFW-IP are very complex systems with many interrelated parameters. Therefore, modeling such a system accurately is a challenging process.

Concordia University has embarked on a long-term research project to thoroughly investigate AFW-IP. This investigation was the first of many to come. The following is a list of areas that would benefit from further research as they were only briefly touched upon in this work:

- An investigation into AFW that rely solely on thermal buoyancy as the driving force, with the creation of a new model that deals specifically with this case.
- The addition of a time component in the models is required to better simulate the transient effects of the system.
- A CFD model should be created to assist in the simulation. The CFD model could be used in conjunction with current models to help determine certain terms that are currently derived from empirical formulas, which introduce uncertainty.
- The influence that the inlet/outlet configuration has on parameters such as the convective heat transfer coefficient, mass-flow rate, etc., should be looked into. This could include the use of dampers to control the flow rate, especially in thermal buoyancy driven flows.
- The effect that air leakage has on the system, both between the AFW and its surrounding environment, and between the two internal channels formed by the blind and/or PV module should be studied.
- The effect that wind has on the system should be investigated. This should include a look into which empirical relation for outdoor heat transfer coefficient is most accurate in accounting for wind effects and how the airflow within the cavity is affected by wind.

- An investigation of the use of the temperature gradient of the air flowing through the cavity as a means to control the flow rate (Zollner, et al., 2002).
- Development of control strategies for the heating and cooling seasons, as well as the intermediate period when both cooling and heating may be required.
- Initial measurements at the Concordia test room have shown that when the PV is in the middle of the cavity, the flow rate differs in the front and back cavities. A closer look at the factors involved with this phenomenon is needed to be able to make more accurate predictions for such systems.
- A more thorough investigation in the calculation of the convective heat transfer coefficient within the cavity is needed. This includes looking at the ratio of the Grashoff number over the square of the Reynolds number to determine what values should be used to determine if the system is being driven by forced, natural or mixed convection. A more detailed analysis of mixed convection in narrow cavities is necessary. In addition a more in depth look into the profile of the CHTC in the Vision section of the system is required.
- The development of similar expressions to the F' and FR values used for solar air collectors to help rate the AFW-IP.
- Determination of a way to properly weigh the value of electricity, daylighting, and thermal energy when determining energy efficiency.
- Determination of the maximum velocity that can be used in an AFW-IP without encountering noise problems.

From the extensive list of work that still needs to be done to more thoroughly understand the AFW-IP, it is clear that the long-term research project at Concordia University will be of great value. Some of the points listed will take little time to investigate, whereas others may take years to fully explore. Velocity profiles will be measured more accurately using a particle-image velocimetry system in the near future. One project that will be a direct continuation of this thesis will be the integration of the one-dimensional model into CANMET Energy Technology Centre's (CETC) HOT3000 repository. This work will be done as part of the research network formed by the Buildings Group at the CETC. HOT3000 is a residential energy analysis tool that is being developed to replace HOT2000. HOT2000 used a bin-based model, whereas HOT3000 will be based on ESP-r (http://www.esru.strath.ac.uk), which employs a transient simulation approach to model the relevant physical processes in the building and HVAC systems.

REFERENCES

Anderson K.T., (1995) Theoretical considerations on natural ventilation by thermal buoyancy, *ASHRAE Transactions* **101,** 2, 1103-1117.

Athienitis A., (1998) *Building Thermal Analysis,* 3rd edition, Concordia University, Montreal.

Athienitis A., Santamouris M., (2002) *Thermal Analysis and Design of Passive Solar Building,* James and James, Brooklyn, NY.

Athienitis A.K., (1997) Investigation of Thermal Performance of a Passive Solar Building with Floor Radiant Heating, *Solar Energy* **61**, 337-345.

Balocco C., (2002) A simple model to study ventilated facades energy performance. *Energy and Buildings* ***34,*** 469-475.

Barletta A., (2002) Fully developed mixed convection and flow reversal in a vertical duct with uniform wall heat flux, *Int. J. of Heat and Mass Transfer* **45,** 641-654.

Bartak M., et al., (2002) Integrating CFD and building simulation, *Building and Environment* **37**, 865-871.

Bazilian M., Prasad D., (2002) Modeling of a photovoltaic heat recovery system and its role in a design decision support tool for building professionals, *Renewable Energy* **27**, 57-68.

Brinkworth B., (2002) Coupling of convective and radiative heat transfer in PV cooling ducts, *Journal of Solar Energy Engineering* **124**, 250-255.

Brinkworth B.J., et al., (2000) A validated model of naturally ventilated PV cladding, *Solar Energy* **69**, 67-81.

Burden R., Faires J.D. (1997) *Numerical Analysis.* 6th edn., Brooks/Cole Publishing Company, Pacific Grove, CA.

Burmeister L., (1993) *Convective Heat Transfer,* 2nd edition, John Wiley and Sons, New York.

Charron R., Athienitis A.K., (2003) A two-dimensional model of a PV-integrated façade, *SESCI 2003 Annual Conference,* Kingston, ON.

Charron R., Athienitis A.K., (2003) Optimization of the performance of PV-integrated double-façades, *ISES Solar World Congress 2003,* Sweden.

Chen Q., Srebric J., (2002) A procedure for verification, validation, and reporting of indoor environment CFD Analyses, *HVAC&R Research* **8,** 201-216.

Costa M., et al., (2000) Optimal Design of Multi-Functional Ventilated Facades, *Final Report: JOULE III- European Commission.*

Costa M., et al., (2002) *Numerical Analysis of thermal behaviour of glazed ventilated facades in Mediterranean climates,* Preprint submitted to Elsevier Science.

Durisch W, Lam K-h, Close J. (2001) Behaviour of a Copper Indium Diselenide Module under Real Operating Conditions, *PV Conference*, Munich.

Federov A., Viskanta R., (1997) Turbulent natural convection heat transfer in an asymmetrically heated, vertical parallel-plate channel, *Int. J. of Heat and Mass Transfer* **40**, 3849-3860.

Fox R., McDonald A., (1998) *Introduction to Fluid Mechanics,* 5th edition, John Wiley and Sons, New York.

Gnielinski V., (1983) *Forced convection in ducts,* Chapter 2.5.1, Ernst Schlunder et al (Eds)., Hemisphere Publishing Corporation. In Heat Exchanger Design Handbook.

Hensen J., et al, (2002) Modeling and Simulation of a Double-Skin Facade System, *ASHRAE Transactions* **108**, 2.

Hollands K.G.T., Shewen E.C., (1981) Optimization of flow passage geometry for air-heating, plate-type solar collectors, *Journal of Solar Energy Engineering* **103,** 323-330.

Incropera F., De Witt D., (1990) *Fundamentals of Heat and Mass Transfer,* 3rd edition, John Wiley and Sons, New York.

Infield D., et al., (1999) Thermal performance of building integrated ventilated PV facades, *Jerusalem Solar World Congress.*

Infield D., et al., (2002) Thermal Modelling a Building with an Integrated Ventilated PV Facade, *submitted for publishing to Elsevier Science.*

Jaros M., et al. (2002) Numerical and Experimental Investigation of the Conditions in the Double Solar Energy Façade, *Proceedings Indoor Air,* 1062-1067.

Klucher T. M., (1979) Evaluation of models to predict insolation on tilted surfaces, *Solar Energy* **23**, 111-114.

Krauter S., et al., (1999) Combined Photovoltaic and Solar Thermal Systems for Façade Integration and Building Insulation, *Solar Energy* **67,** 239-248.

Mathsoft, (2001) *Mathcad 2001 User's Guide*, Mathsoft Engineering and Education, Cambridge, MA.

Mei L., et al., (2003) Thermal modeling of a building with an integrated ventilated PV façade, *Energy and Buildings* **35**, 605-617.

Moshfegh B., Sandberg M., (1996) Investigating of fluid flow and heat transfer in a vertical channel heated from one side by PV elements Part I – Numerical Study, *WREC 1996*, 248-253.

Moshfegh B., Sandberg M., (1996) Investigating of fluid flow and heat transfer in a vertical channel heated from one side by PV elements Part II – Experimental Study, *WREC 1996*, 254-258.

Murdoch J., (1994) *Illumination Engineering: From Edison's Lamp to the Laser,* Visions Communications.

Omer S.A, Wilson R., Riffat S.B., (2003) Monitoring results of two examples of building integrated PV (BIPV) systems in the UK, *Renewable Energy* **28,** 1387-1399.

Ong K., (1995) Thermal Performance of Solar Air Heaters: Mathematical Model and Solution Procedure, *Solar Energy* **55**, 2, 93-109.

Ong K.S., (1995) Thermal performance of solar air heaters – experimental correlation, *Solar Energy* **55,** 209-220.

Orgill J. F., Hollands K.G.T., (1977) Correlation equation for hourly diffuse radiation on a horizontal surface, *Solar Energy* **19,** 357-359.

Poissant Y., Couture L., Dignard-Bailey L., Thevenard D., (2003) Simple test methods for evaluating the energy ratings of PV modules under various environmental conditions, *ISES Solar World Congress 2003*, Sweden

Rodrigues A.M., et al., (2000) Modelling natural convection in a heated vertical channel for room ventilation, *Building and Environment* **35**, 455-469.

Rolle K., (2000) *Heat and Mass Transfer,* 1st edition, Prentice Hall, Upper Saddle River, NJ.

Saelens D., (2002) *Energy Performance Assessment of Single Storey Multiple-Skin Facades,* PhD Thesis, Katholieke Universiteit Leuven.

Sanberg M., (1999) Cooling of Building Integrated Photovoltaics by Ventilation Air, *HybVent Forum 1999,* Sydney.

Siegel R., Howell J. R. (1992) *Thermal Radiation Heat Transfer*, 3rd edn., Hemisphere Pub. Corp., Washington, D.C.

Tanimoto J., Kimura K-i., (1997) Simulation study on an air flow window system with an integrated roll screen, *Energy and Buildings* **26**, 317-325.

Vartiainen E., et al., (2000) Daylight Optimization of Multifunctional Facades, *Solar Energy* **68** 223-235.

Yang H.X. (1996) Validated Simulation for Thermal Regulation of Photovoltaic Wall Structures. *26th PVSC IEEE Conference*, 1453-1456.

Zhai Z., et al., (2002) On approaches to couple energy simulation and computational fluid dynamics programs, *Building and Environment* **37**, 857-864.

Zollner A., et al., (2002) Experimental studies of combined heat transfer in turbulent mixed convection fluid flows in double-skin facades. *Int. J. Heat and Mass Transfer 45,* 4401-4408.

APPENDIX A

Listing of HTcoef.F90

FORTRAN subroutine to calculate convective heat transfer coefficient in 2D model

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!! Subroutine name: HTcoef
!!
!!
!!
!! Description:    This subroutine is called from subroutine Funcv (EqnGen_wCond)
!!                 to calculate the convective heat transfer coefficients in the cavities.
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Subroutine HTcoef(Lgap,H_tot,Vel,G,T1,T2,Ta,div,hc1,hc2)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Variable declaration
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Implicit none
Integer, intent (in) :: div
Real, intent (in) :: Lgap, H_tot, Vel, G, T1(div),T2(div),Ta(div)
Real, intent (out) :: hc1(div),hc2(div)
Integer :: i

Real ::   T1_ave               !Variable to calculate ave temp of front surface
Real:: T2_ave                  !Variable to calculate ave temp of back surface
Real:: Ta_ave                  !Variable to calculate ave temp of air
Real :: dH                     !Height of elements
Real:: x, xturb                !Variables to determine if flow is turbulent
Real:: RaD                     !Rayleigh number for flow in a ciruclar duct
Real:: NuD                     !Nusselt number for flow in a circular duct
Real:: ReD                     !Reynold's number for flow in a circular duct
REal:: fric                    !Friction factor
Logical :: nat                 !Logical variable to state if flow is in natural convection
Logical:: mix                  !Logical variable to state if flow is in mixed convection
Logical:: force                !Logical variable to state if flow is in forced convection
Logical:: wide                 !Logical variable to state if the wide cavity approx. holds

!Lgap:         width of plate separation
!H_tot:        the total plate height
!Vel:          the velocity of the air flowing between the plates
!G:            incident solar radiation on the AFW
!T1:           elemental temperature of the plate facing the outdoor environment
!T2:           elemental temperature of plate opposite to T1
!Ta:           elemental temperature of the air flowing through the AFW
!hc1:          convective heat transfer coefficient along plate at T1
!hc2:          conevective heat transfer coefficient along plate at T2
!div:          number of elements the plate is sub-divided into

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following is an interface for functions used in this program
!!
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Interface
Function C(Tma)                 ! specific heat of air
implicit none
Real, Intent(IN) :: Tma
Real :: c
End function C

Function dens(Tma)              ! air density
implicit none
Real, Intent(IN) :: Tma
Real :: dens
End function dens

Function k(Tma)                 ! conductivity of air
implicit none
Real, Intent(IN) :: Tma
Real :: k
End function k

Function visc(Tma)              !air viscosity
implicit none
Real, Intent(IN) :: Tma
Real :: visc
End Function visc

Function kvisc(Tma)             !kinematic viscocity
implicit none
Real, Intent(IN) :: Tma
Real :: kvisc
End Function kvisc

Function Pr(Tma)                !Prandtl Number
implicit none
Real, Intent(IN) :: Tma
Real :: Pr
End Function Pr

Function Gr(Tsurf, Tma, dim)    !Grashoff number
implicit none
Real, Intent(in) :: Tsurf, Tma, dim
Real :: Gr
End Function Gr

Function h1(Tsurf, Tma,dH, i)
implicit none
Real, Intent(in) :: Tsurf, Tma, dH
Integer, Intent(in) :: i
Real :: h1
End Function h1

Function h2(Tma,vel,dH,i)
implicit none
Real, Intent(in) :: Tma, vel, dH
Integer, Intent (in) :: i
```

```
Real :: h2
End Function h2

Function h3(Tma, vel, dH, i)
implicit none
Real, Intent(in) :: Tma, vel, dH
Integer, Intent (in) :: i
Real :: h3
End Function h3

Function h4(Tma,vel,dH, i)
implicit none
Real, Intent(in) :: Tma, vel, dH
Integer, Intent (in) :: i
Real :: h4
End Function h4

Function h5(h1, h2)
implicit none
Real, Intent(in) :: h1,h2
Real :: h5
End Function h5

End interface

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!MAIN BLOCK
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

dH=H_tot/div

T1_ave=0; T2_ave=0; Ta_ave=0                    !initializing

!The following step will determine the average temperature of each surface based on elemental
temperatures

L1:     Do i=1,div
                T1_ave=T1_ave  + T1(i)
                T2_ave=T2_ave  + T2(i)          !finding the sum of the elemental temperatures
                Ta_ave=Ta_ave  + Ta(i)
        End Do L1

        T1_ave=T1_ave/div; T2_ave=T2_ave/div; Ta_ave=Ta_ave/div         !dividing sum by
                                                                         number of elements


!The following determines if the wide cavity approximation holds
!The expression is suggested by Saelens (2002)

If (((2*Lgap/H_tot - (Gr(T1_ave,Ta_ave,H_tot)*Pr(Ta_ave))**(-0.25)  -  &
    &(Gr(T2_ave,Ta_ave,H_tot)*Pr(Ta_ave))**(-0.25))>0) .or. &

        &  ((2*Lgap/H_tot -(Gr(T1_ave,Ta_ave,Lgap)*Pr(Ta_ave))**(-1)  &
        &  -(Gr(T2_ave,Ta_ave,Lgap)*Pr(Ta_ave))**(-1))>0))then
```

```
        wide=.true.                 !separate plate approximation is acceptable
!               print *,'wide'

        else
        wide=.false.

!               print *,'narrow'

        end If


        !The following determines if the heat transfer is either governed by forced, natural or mixed
convection (Rolle, 2000)

        nat=.false.; mix=.false. ; force=.false.;                                   !initializing

        If (wide .and. (Gr((T1_ave+T2_ave)/2,Ta_ave,H_tot)/(Vel*H_tot/kvisc(Ta_ave))**2)>4) then

        nat=.true.                  !Natural convection dominates

!               print *,'natural'

        Else If (wide .and.
(Gr((T1_ave+T2_ave)/2,Ta_ave,H_tot)/(Vel*H_tot/kvisc(Ta_ave))**2)>0.25)then

                mix=.true.          !Flow affected by both natural and forced convection (mixed)

!               print *,'mixed'

                Else If (wide) then
                        force=.true.                        !Forced convection dominates
!                       print *,'forced'
        End If

        If (.not.wide .and. (Gr((T1_ave+T2_ave)/2,Ta_ave,H_tot)/(Vel*H_tot/kvisc(Ta_ave))**2)>4) then
        nat=.true.
!       print *,'natural'
        Else If (.not.wide .and.
(Gr((T1_ave+T2_ave)/2,Ta_ave,H_tot)/(Vel*H_tot/kvisc(Ta_ave))**2)>0.25)then
                mix=.true.
!               print *,'mix'
                Else If (.not.wide) then
                        force=.true.                        !for a narrow front cavity
!                       print *,'forced'
        End If


    !initializing

If (wide .and. nat) then

L2:     Do i=1,div
        hc1(i)=h1(T1(i), Ta(i),dH,i)
        hc2(i)=h1(T2(i),Ta(i),dH,i)
```

```
        End Do L2

        Else if (wide .and. force) then
L3:     Do i=1,div
        xturb=(5*1e05)*kvisc(Ta(i))/Vel    !Height at which flow becomes turbulent
        x=dH*(i-0.5)
                        if (x<xturb .and. G<200)then          !Low incident solar radiation
                        hc1(i)=h2(Ta(i),Vel,dH,i)
                        hc2(i)=hc1(i)

                Else if (x<xturb .and. G>=200)then   !High incident solar radiation
                                hc1(i)=h3(Ta(i),Vel,dH,i)
                                hc2(i)=hc1(i)
                        Else if (x>xturb) then
                                hc1(i)=h4(Ta(i),Vel,dH,i)
                                hc2(i)=hc1(i)
                        End If
        End Do L3

        Else If (wide .and. mix) then
L4:     Do i=1,div
        xturb=(5*1e05)*kvisc(Ta(i))/Vel
        x=dH*(i-0.5)
                        if (x<xturb .and. G<200)then

                        hc1(i)=h5(h1(T1(i),Ta(i),dH,i), &
                                        &  h2(Ta(i),Vel,dH,i))

                        hc2(i)=h5(h1(T2(i),Ta(i),dH,i), &
                                        &  h2(Ta(i),Vel,dH,i))

                        Else if (x<xturb .and. G>=200)then

                                hc1(i)=h5(h1(T1(i),Ta(i),dH,i), &
                                        &  h3(Ta(i),Vel,dH,i))

                                hc2(i)=h5(h1(T2(i),Ta(i),dH,i), &
                                        &  h3(Ta(i),Vel,dH,i))

                        Else if (x>xturb) then

                                hc1(i)=h5(h1(T1(i),Ta(i),dH,i), &
                                        &  h4(Ta(i),Vel,dH,i))

                                hc2(i)=h5(h1(T2(i),Ta(i),dH,i), &
                                        &  h4(Ta(i),Vel,dH,i))
                        End If
        End Do L4

!The following calculations for narrow (.not.wide) cavities are done using average values !for convective heat transfer coefficient, that is, it assumes negligible effect from entrance !region

        Else If (.not.wide .and. mix) then
                RaD=Pr(Ta_ave)*(Gr(T1_ave,Ta_ave,2*Lgap)+ Gr(T2_ave,Ta_ave,2*Lgap))/2
                ReD=Vel*2*Lgap/kvisc(Ta_ave)
```

```
              NuD=(ABS((0.0357*ReD*Pr(Ta_ave)/(log(ReD/7)*(1+Pr(Ta_ave)**(-
0.8))**(5./6)))**3 &
                        & -
(0.15*RaD**(1./3)/(1+(0.492/Pr(Ta_ave))**(9./16))**(16./27))**3))**(1./3)

L5:     Do i=1,div
              hc1(i)=NuD*k(Ta_ave)/(2*Lgap)
              hc2(i)=hc1(i)
        End Do L5

        Else If (.not.wide .and. force)then
              ReD=Vel*2*Lgap/kvisc(Ta_ave)
              fric=(1.82*log10(ReD) -1.64)**(-2)
              NuD=((fric/8)*(ReD-1000)*Pr(Ta_ave)/(1+12.7*sqrt(fric/8)*(Pr(Ta_ave)**(2./3)-1))) &
                        & *(1+ (2*Lgap/H_tot)**(2./3))


L6:     Do i=1,div
              hc1(i)=NuD*k(Ta_ave)/(2*Lgap)
              hc2(i)=hc1(i)
        End Do L6

        Else If (.not.wide .and. nat)then
L7:     Do i=1,div
              hc1(i)=0.68*(Gr(T1_ave,Ta_ave,Lgap)*Pr(Ta_ave)*Lgap/H_tot)**0.25
*k(Ta_ave)/Lgap
              hc2(i)=0.68*(Gr(T2_ave,Ta_ave,Lgap)*Pr(Ta_ave)*Lgap/H_tot)**0.25
*k(Ta_ave)/Lgap
        End Do L7

End If

Return
End Subroutine HTcoef

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Function definitions
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Function C(Tma)
Implicit None
Real, INTENT(in) :: Tma
Real :: C
                        C= 1000*(1.0057 +0.000066*(Tma - 300.15))
Return
End Function C

Function dens(Tma)
Implicit None
Real, INTENT(in) :: Tma
Real :: dens
                        dens= 1.1774 - 0.00359*(Tma - 300.15)
Return
End Function dens
```

```
Function k(Tma)
Implicit None
Real, Intent(IN) :: Tma
Real :: k
                        k= 0.02624 + 0.0000758*(Tma - 300.15)
Return
End Function k

Function visc(Tma)
Implicit None
Real, INtent(in) :: Tma
Real :: visc
                        visc= (1.983 + 0.00184*(Tma - 300.15))*1e-5

Return
End Function visc

Function kvisc(Tma)
Implicit None
Real, Intent(in) :: Tma
Real :: kvisc, visc, dens

visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
Return
End Function kvisc

Function Pr(Tma)
Implicit None
Real, Intent(in) :: Tma
Real :: Pr, C, k, visc
visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
C= 1000*(1.0057 +0.000066*(Tma - 300.15))
k= 0.02624 + 0.0000758*(Tma - 300.15)

Pr = C*visc/k
Return
End Function Pr

Function Gr(Tsurf, Tma,dim)
Implicit None
Real, Intent(in) :: Tsurf, Tma, dim
Real :: Gr,visc, dens, kvisc
visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
Gr = (abs(Tsurf - Tma)*9.81 *Tma**(-1) *dim**3)/kvisc**2

Return
End Function Gr

Function h1(Tsurf, Tma,dH, i)
implicit none
Real, Intent(in) :: Tsurf, Tma,dH
```

```
Integer, Intent(in) :: i
Real :: h1, gPr, x, h1_lam, h1_turb, Xcrit, Pr, Gr, k, kvisc
Real :: visc, dens, C
x= dH*(i-0.5)

k=0.02624 + 0.0000758*(Tma - 300.15)
visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
C= 1000*(1.0057 +0.000066*(Tma - 300.15))
Pr = C*visc/k
Gr = (abs(Tsurf - Tma)*9.81 *Tma**(-1) *x**3)/kvisc**2


gPr= (0.75*Pr**0.5)/(0.609+ 1.221*Pr**0.5+ 1.238*Pr)**0.25
h1_lam=(Gr/4)**0.25 *gPr *k/x

h1_turb=0.02979*k* Pr**(1./15) *(((Pr/kvisc**2)*abs(Tsurf-Tma) &
                    & *9.81*Tma**(-1))**0.4 / (1+ 0.494*Pr**(2./3))**0.4)*x**0.2

Xcrit=(1e09*((abs(Tsurf-Tma)*Pr*9.81*Tma**(-1))/kvisc**2)**(-1))**(1./3)

if (x < Xcrit) then
        h1=h1_lam
Else
        h1=h1_turb
End If
Return
End Function h1

Function h2(Tma,vel,dH,i)
implicit none
Real, Intent(in) :: Tma, vel, dH
Integer, Intent (in) :: i
Real :: h2,x, k, Pr, kvisc, visc, dens, C

x= dH*(i-0.5)
k=0.02624 + 0.0000758*(Tma - 300.15)

visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
C= 1000*(1.0057 +0.000066*(Tma - 300.15))
Pr = C*visc/k

h2= 0.332*(vel*x/kvisc)**0.5 *Pr**(1./3) *k/x
return
End Function h2

Function h3(Tma, vel,dH, i)
implicit none
Real, Intent(in) :: Tma, vel,dH
Integer, Intent (in) :: i
Real :: h3,x, k, Pr, kvisc, visc, dens, C

x= dH*(i-0.5)
```

```
k=0.02624 + 0.0000758*(Tma - 300.15)

visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
C= 1000*(1.0057 +0.000066*(Tma - 300.15))
Pr = C*visc/k

h3= 0.453*(vel*x/kvisc)**0.5 *Pr**(1./3) *k/x
Return
End Function h3

Function h4(Tma,vel,dH,i)
implicit none
Real, Intent(in) :: Tma, vel,dH
Integer, Intent (in) :: i
Real :: h4,x, k, Pr, kvisc, visc, dens, C

x= dH*(i-0.5)
k=0.02624 + 0.0000758*(Tma - 300.15)

visc=(1.983 + 0.00184*(Tma - 300.15))*1e-5
dens= (1.1774 - 0.00359*(Tma - 300.15))
kvisc= visc/dens
C= 1000*(1.0057 +0.000066*(Tma - 300.15))
Pr = C*visc/k

h4= 0.0296*(vel*x/kvisc)**0.8 *Pr**(1./3) *k/x
Return
End Function h4

Function h5(h1, h2)
implicit none
Real, Intent(in) :: h1, h2
Real :: h5

h5= (h1**3 + h2**3)**(1./3)

Return
End Function h5
```

APPENDIX B

Listing of ViewFact.F90

FORTRAN subroutine to calculate view factors within ventilated façades

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!! Subroutine name: ViewFact (viewfac1pv_2w), this subroutine is called from subroutine
!!              funcv (EqnGen1pv_2w)
!!
!!
!!
!! Description:    This subroutine is used to calculate the view factors between various
!!                  elements within the function.
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Subroutine ViewFact(L,H_tot,w,N,Ff)
Implicit none

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Variable declaration
!!
!!Input and output are done with subroutine funcv (EqnGen1pv_2w)
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Integer, intent (in) :: N
Real, intent (in) :: L,H_tot, w
Real, intent (out) :: Ff(N*2+3,N*2+3)
Integer :: i, j
Real :: dH, mem,sum

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Interface Declaration
!! (defining functions used in the program)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Interface
Function Fpa(x,y,L)                 ! function for form factors of parallel rectangles
implicit none
Real, Intent(IN) :: x, y, L
Real :: Fpa
End function Fpa

Function Fpe(x,y,z)
implicit none
Real, Intent(in) :: x,y,z           !Function for from factors between perpendicular rectangles
Real :: Fpe
End Function Fpe
End interface

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!MAIN BLOCK
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
dH=H_tot/N

Ff=0     !initialize Ff

Ff(1,N +1)= Fpa(dH, w, L)
Ff(1,N +2)= Fpa(2*dH, w, L) - Ff(1, N+1)

!L3: Calculates the view factor between the first element to all the elements in
!the PV cover. These equations were found using an energy balance approach.

mem=0
L3:      Do i=3, N
         mem=mem+Ff(1,i-1+N)
         Ff(1,N +i)=(i/2.)*Fpa(i*dH,w,L)-0.5*Ff(1,N+1) -      &
                                  & 0.5*(i-1)*Fpa((i-1)*dH,w,L) - mem
         End Do L3

L5: Do i=1,N
         Ff(i+ N, 1)= Ff(1,i +N)              !By reciprocity, verified
         End Do L5

L6:      Do j=2,N
         Ff(j,N +1)= Ff(N+j,1)                !By symmetry, verified
         End Do L6

L7:      Do i=(N+2),2*N
L8:               Do j=2,N
                  Ff(j,i)=Ff(j-1, i-1)        !By Symmetry, verified
                  End Do L8
         End Do L7

L9: Do i=1,N
L10:     Do j=(N+1),2*N
                  Ff(j,i)=Ff(i,j)             !By reciprocity, verified
                  End do L10
         End Do L9

!Note, need to add extra surfaces to account for entry, exit, and side walls
!Surface of entry  2*N+1, Surface of exit is 2*N+2
!side walls are 2*N+3

Ff(1,2*N+1)=Fpe(w,dH,L)
Ff(2*N+1,1)=Ff(1,2*N+1)*(dH*w/(L*w))   !by reciprocity
Ff(N+1,2*N+1)=Ff(1,2*N+1)              !by symmetry
Ff(2*N+1,N+1)=Ff(2*N+1,1)              !by symmetry

L22:     Do i=2,N
                  Ff(i,2*N+1)=i*Fpe(w,i*dH,L)-(i-1)*Fpe(w,(i-1)*dH,L)
                  Ff(2*N+1,i)=Ff(i,2*N+1)*(dH*w/(L*w))          !reciprocity
                  Ff(N+i,2*N+1)=Ff(i,2*N+1)                     !symmetry
                  Ff(2*N+1,N+i)=Ff(2*N+1,i)                     !symmetry
         End Do L22

L23:     Do i=1,N
                  Ff(i,2*N+2)=Ff(N-(i-1),2*N+1)
```

```
                    Ff(2*N+2,i)=Ff(i,2*N+2)*(dH*w/(w*L))        !by reciprocity
                    Ff(N+i,2*N+2)=Ff(i,2*N+2)                   !by symmetry
                    Ff(2*N+2,N+i)=Ff(2*N+2,i)                   !by symmetry
            End Do L23

L26:    Do i=1,N
            sum=0
L27:        Do j=1,2*N+3
                    sum=sum+Ff(i,j)
                End Do L27
            Ff(i,2*N+3)=1-sum
            Ff(2*N+3,i)=Ff(i,2*N+3)*(dH*w/(2*L*H_tot))      !By reciprocity
            Ff(i+N,2*N+3)=Ff(i,2*N+3)
            Ff(2*N+3,i+N)=Ff(2*N+3,i)                       !by symmetry
        End Do L26

!View Factors between the sides, entrance and exit
Ff(2*N+3,2*N+3)=Fpa(H_tot,L,w)
Ff(2*N+1,2*N+3)=2*Fpe(L,w,H_tot)
Ff(2*N+3,2*N+1)=Ff(2*N+1,2*N+3)*(L*w/(2*L*H_tot))       !by reciprocity
Ff(2*N+3,2*N+2)=Ff(2*N+3,2*N+1)                         !by symmetry
Ff(2*N+1,2*N+2)=Fpa(L,w,H_tot)
Ff(2*N+2,2*N+1)=Ff(2*N+1,2*N+2)
Ff(2*N+2,2*N+3)=Ff(2*N+1,2*N+3)

Return
End Subroutine ViewFact

Function Fpa(x,y,L)                                     !This function is working, verified using
Mathcad
Implicit None
Real, INTENT(in) :: x,y,L
Real :: Fpa, xf,yf
xf=x/L
yf=y/L
Fpa=(2./(3.14159*xf*yf))*(log( ( (1+xf**2)*(1+yf**2)/(1+xf**2+yf**2) )**0.5) +      &
        & xf*(1+yf**2)**0.5 *atan(xf/(1+yf**2)**0.5) +yf*(1+xf**2)**0.5 *atan(yf/(1+xf**2)**0.5) &
        & - xf*atan(xf)    - yf*atan(yf))

Return
End Function Fpa

Function Fpe(x,y,z)
implicit none
Real, Intent(in) :: x,y,z
Real :: Fpe,H,W,pi
pi=3.14159
H=z/x; W=y/x

Fpe=(1./(pi*W))*(W*atan(1/W)+ H*atan(1/H)- (H**2 +W**2)**(0.5)*atan(1/(H**2+ W**2)**0.5) &
&+0.25*log(((1+W**2)*(1+H**2)/(1+W**2+H**2))*(W**2*(1+W**2+H**2)/((1+W**2)*(W**2+H**
2)))**(W**2) &
&   *(H**2*(1+H**2+W**2)/((1+H**2)*(H**2+W**2)))**(H**2)))

Return
End Function Fpe
```

APPENDIX C

Listing of EnvCond.F90

FORTRAN subroutine to calculate hourly environmental conditions for 1D model

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Subroutine name: EnvCond
!!
!!
!!
!! Description:   This is the subroutine where the environmental conditions are set.
!!                These include the beam and diffuse solar radiation, temperature,
!!                and the optical properties of elements based on solar position.
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

subroutine envcond(tod,dn,SA,Lat,LNG,STM,tilt,gnd_refl,kw1,kw2,tw1,tw2,kg1,tg1, &
& Gb,Gd,To, mth, tr_w1_bm, tr_w2_bm, rfl_w1_bm, rfl_w2_bm, tr_w1_df, &
& tr_w2_df, rfl_w1_df, rfl_w2_df,rfl_g1_bm,rfl_g1_df,tr_g1_bm,tr_g1_df)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Variable Declaration
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

implicit none
integer, intent(in):: dn, tod
integer, intent(out) :: mth
real, intent(in):: gnd_refl, Lat, tilt, SA, LNG,STM, kw1, kw2, tw1, tw2, kg1, tg1
real, intent(out):: rfl_g1_bm,rfl_g1_df,tr_g1_bm,tr_g1_df
real, intent(out):: Gb,Gd, To, tr_w1_bm, tr_w2_bm, rfl_w1_bm, rfl_w2_bm
real, intent(out):: tr_w1_df, tr_w2_df, rfl_w1_df, rfl_w2_df

real:: Ktm(12)          !Array with the mean monthly Kt values in Montreal
real:: Tm(12)           !Array with mean monthly temperatures in Montreal
real:: Tamp             !The amplitude of the temperature swing in Montreal
real:: dA               !Declination angle
real:: ts               !Sunset time
real:: ast              !apparent solar time
real:: ha               !hour angle
real:: sha              !sunset solar angle
real:: alt              !solar altitude
real:: solA             !Solar azimuth
real:: cosO             !dummy variable to calculate value of cosO
real:: aoi              !Angle of incidence
real:: esr              !The earth spin rate (rad/hr)
real:: A, B             !constants required to calculate hourly clearness indexes
real:: kt               !hourly clearness index
real:: kd               !hourly diffuse clearness index
real:: kb               !hourly beam clearness index
real:: Ion              !Incident radiation on the earth atmosphere
real:: Ibm              !Incident beam radiation
real:: Idif             !Incident diffuse radiation
real:: Ignd             !Reflected radiation from the ground
real:: deg              !dummy variable to change from radians to degrees
real:: SSA              !surface azimuth of system
real:: ET               !Equation of time
real:: value            !dummy variable
real:: n                !refractive index of the glass
```

```
real:: ag                !value needed to calculate optical properties
real:: r                 !component reflectivity
real:: Op                !angle using Snell's law
real:: Lp                !length of travel of beam in glazing

real:: pi                !pi
parameter (pi=3.14159)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Main block
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!open (96,file='G_DATA.dat')

!write(96,*) 'G value for ', dn, 'day'

!Finding the corresponding month to the day number
If (dn < 32) then
mth=1
        Else if ( dn<60) then
        mth=2
        Else if (dn<91) then
        mth=3
        Else if (dn<121)then
        mth=4
        Else if (dn<151)then
        mth=5
        Else if (dn<182)then
        mth=6
        Else if (dn<213)then
        mth=7
        Else if (dn<244)then
        mth=8
        Else if (dn<274)then
        mth=9
        Else if (dn<305)then
        mth=10
        Else if (dn<335)then
        mth=11
        Else
        mth=12
End If

!Defining the Mean monthly clearness index Kt values for Montreal
Ktm(1)=0.45
Ktm(2)=0.51
Ktm(3)=0.50
Ktm(4)=0.48
Ktm(5)=0.49
Ktm(6)=0.49
Ktm(7)=0.52
Ktm(8)=0.49
Ktm(9)=0.49
Ktm(10)=0.41
Ktm(11)=0.35
Ktm(12)=0.38
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Calculating solar properties
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

dA=23.45*sind(360.*((dn+284.)/365.))                    !declination angle

ET=(9.87*sin(4*pi*(dn-81)/364)-7.53*cos(2*pi*(dn-81)/364)-1.5*sin(2*pi*(dn-81)/364))/60  !eqn of time
in hours


ast=tod + ET+ (STM - LNG)/15                            !apparent solar time
ha= 15.*(ast-12.)                                       !hour angle from 0 to solar noon
sha=acosd(-tand(Lat)*tand(dA))                          !sunset solar angle
ts=acosd(-tand(Lat-tilt)*tand(dA))/15.                  !sunset time

if (ts > sha/15.) then
ts=sha/15.
end if



alt=asind(cosd(Lat)*cosd(dA)*cosd(ha)+ sind(Lat)*sind(dA))          !solar altitude

if (alt < 0) then
alt=0
end if

value=((sind(alt)*sind(lat)-sind(dA))/(cosd(alt)*cosd(lat)))*(ha/abs(ha))
if (abs(value) > 1.) then
solA=acosd(1.)
Else
solA=acosd((sind(alt)*sind(lat)-sind(dA))/(cosd(alt)*cosd(lat)))*(ha/abs(ha)) !solar azimuth
end if

SSA= solA - SA                  !surface azimuth taking into account solar azimuth

cosO= cosd(alt)*cosd(abs(SSA))

aoi=acosd((cosO + abs(cosO))/2.)                        !angle of incidence

esr=2.*pi/24.                                           !earths spin rate in rad/hr
deg=(1.047/pi)*180.
A=0.409+ 0.5016*sind(sha- deg)
B=0.6609- 0.4767*sind(sha- deg)

kt=(A+ B*cos(esr*(tod-11.99)))*Ktm(mth)  !hourly clearness index

if (kt < 0) then
kt=0
end if

if (kt < 0.35) then
kd=(1-0.249*kt)*kt                                      !hourly diffuse clearness index
```

```
        Else if (kt < 0.75) then
        kd=(1.557-1.84*kt)*kt
        Else
        kd=0.177*kt
End if

kb=kt-kd                                                    !hourly beam clearness index

Ion=1353.*(1.+0.033*cosd(360.*dn/365.))                     !irradiation outside atmosphere
Ibm=Ion*kb*cosd(aoi)                                        !beam irradiation
Idif=Ion*sind(alt)*kd*((1.+cosd(tilt))/2.)                  !diffuse irradiation
Ignd=Ion*sind(alt)*(kd+kb)*gnd_refl*((1.-cosd(tilt))/2.)  !rad'n. reflected off ground

Gd=Idif + Ignd
Gb=Ibm



!write(96,'(i4, f8.3)') tod, Ir(tod)
!write (96, *) 'total'

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Calculation of window properties
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

n=1.52              !refractive index of glass



!first window beam

Op=asind(sind(aoi)/n)                          !Snell's law

r=0.5*((sind(aoi-Op)/sind(aoi+Op))**2 + (tand(aoi-Op)/tand(aoi+Op))**2)        !component reflectivity
Lp=tw1/sqrt(1-(sind(aoi)/n)**2)                                                 !length of beam travel

ag=exp(-kw1*Lp)

tr_w1_bm=(ag*(1-r)**2)/(1- (r*ag)**2)

rfl_w1_bm=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)

!beam of second window's two glazings
Lp=tw2/sqrt(1-(sind(aoi)/n)**2)
ag=exp(-kw2*Lp)
tr_w2_bm=(ag*(1-r)**2)/(1- (r*ag)**2)
rfl_w2_bm=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)

!beam of PV cover

Lp=tg1/sqrt(1-(sind(aoi)/n)**2)
ag=exp(-kg1*Lp)
tr_g1_bm=(ag*(1-r)**2)/(1- (r*ag)**2)
rfl_g1_bm=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)
```

```
!first window diffuse assume aoi=60deg

Op=asind(sind(60.)/n)                    !Snell's law

r=0.5*((sind(60.-Op)/sind(60.+Op))**2 + (tand(60.-Op)/tand(60.+Op))**2)
Lp=tw1/sqrt(1-(sind(60.)/n)**2)

ag=exp(-kw1*Lp)

tr_w1_df=(ag*(1-r)**2)/(1- (r*ag)**2)
rfl_w1_df=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)

!second window diffuse assume aoi=60deg for each glazing
Lp=tw2/sqrt(1-(sind(60.)/n)**2)
ag=exp(-kw2*Lp)
tr_w2_df=(ag*(1-r)**2)/(1- (r*ag)**2)
rfl_w2_df=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)

!diffuse of PV cover
Lp=tg1/sqrt(1-(sind(60.)/n)**2)
ag=exp(-kg1*Lp)
tr_g1_df=(ag*(1-r)**2)/(1- (r*ag)**2)
rfl_g1_df=r + (r*(1-r)**2 * ag**2)/(1-(r*ag)**2)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Defining temperature
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!Mean monthly outdoor temperatures in Montreal
Tm(1)=-9.3
Tm(2)=-9.4
Tm(3)=-3.8
Tm(4)=4.6
Tm(5)=12.9
Tm(6)=17.9
Tm(7)=21.1
Tm(8)=19.7
Tm(9)=15
Tm(10)=7.7
Tm(11)=0.4
Tm(12)=-7.1

Tamp=7.
!Outdoor temperature assuming and daily amplitude in temperature of Tamp
To=Tm(mth) +Tamp*cos(tod*2*pi/24 – 5*pi/4)          !(Athienitis, 1997)

Return
End Subroutine EnvCond
```

APPENDIX D

Listing of Yearly.F90

FORTRAN program to calculate monthly performance of ventilated façades in Configuration 2 for 1D model

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!! Program name: Yearly
!!
!!
!!
!! Description:    This is the main Program that calls all the subroutines to model the
!!                 AFW. Here average monthly energy transfers are stored and sent to a
!!                 data file.
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

program yearly

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Variable Declaration
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

implicit none

real ::Etot_pv              !Total solar radiation on PV section
real ::Etot_w               !Total solar radiation on Vision Section
real ::Erefl_pv             !Radiation reflected out of the PV Section
real ::Erefl_w              !Radiation reflected out of the Vision section
real ::qair1f_pv            !Heat absorbed in air in front PV section
real ::qair1b_pv            !Heat absorbed in air flow in back PV section
real ::qair1f_w             !Heat absorbed in front Vision cavity
real ::qair1b_w             !Heat absorbed in back Vision cavity
real ::qo_pv                !Heat lost to outside in PV section
real :: qo_w                !Heat lost to outside in Vision section
real ::qroom_w              !Energy transferred to room from Vision section
real ::qroom_pv             !Energy transferred to room from PV section
real ::Etrans               !Energy transmitted through Vision section
real ::Egen                 !Generated electricity
real ::Et_pv(12)            !Array for monthly total solar radiation in PV section
real ::Et_w(12)             !Array for monthly total solar radiation in Vision section
real :: Erfl_pv(12)         !Array for monthly reflected radn from PV section
real ::Erfl_w(12)           !Array for monthly reflected radn from Vision section
real ::qaf_pv(12)           !Array for monthly energy absorbed by air in front PV section
real ::qab_pv(12)           !Array for monthly energy absorbed by air in back PV section
real ::qaf_w(12)            !Array for monthly energy abs in front Vision section
real ::qab_w(12)            !Array for monthly energy abs in back Vision section
real ::qopv(12)             !Array for monthly energy lost to outside from PV section
real :: qow(12)             !Array for monthly energy lost to outside from Vision section
real ::qrm_pv(12)           !Array for monthly energy transferred to room from PV Section
real ::qrm_w(12)            !Array for monthly energy transferred to room from Vision section
real ::Etr(12)              !Array for monthly energy transmitted through Vision section
real ::Eelec(12)            !Array for monthly electricity generation
real ::Tpv                  !Temperature of PV cells
real ::sum                  !dummy variable to add sum of elements
real ::Tave                 !dummy variable to take average temperature of elements
integer :: mth              !month number (1 to 12)
integer ::dn                !day number (1 to 365)
```

```
integer ::tod                       !Time of day in hours (1 to 24)
integer ::n                         !dummy variable to keep counts

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Main Block
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
n=0; sum=0
L1: Do dn=1, 365!amount of days for analysis (need 365 days for monthly averages)
print *,'day', dn
L2: Do tod=4, 23 !hours used in analysis

n=n+1

call oned_2pv2W(tod, dn, Etot_pv, Etot_w, Erefl_pv, Erefl_w, &
&                                   qair1f_pv,qair1b_pv, qair1f_w, qair1b_w, qo_pv, qo_w, qroom_w, &
&                                   qroom_pv, Etrans, Egen, mth, Tpv)
sum=sum+Tpv

!!!!!!!
!!Stores the energy in an array based on month
!!
!!!!!!
Et_pv(mth)=Et_pv(mth)+ Etot_pv
Et_w(mth)=Et_w(mth)+Etot_w
Erfl_pv(mth)=Erfl_pv(mth) + Erefl_pv
Erfl_w(mth)=Erfl_w(mth) + Erefl_w
qaf_pv(mth)=qaf_pv(mth) + qair1f_pv
qab_pv(mth)=qab_pv(mth)+qair1b_pv
qaf_w(mth)=qaf_w(mth) + qair1f_w
qab_w(mth)=qab_w(mth) + qair1b_w
qopv(mth)=qopv(mth) + qo_pv
qow(mth)=qow(mth) + qo_w
qrm_pv(mth)=qrm_pv(mth)+ qroom_pv
qrm_w(mth)=qrm_w(mth)+ qroom_w
Etr(mth)=Etr(mth) + Etrans
Eelec(mth)=Eelec(mth) + Egen
End Do L2

End Do L1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Can only find monthly averages, and print to file the monthly values if the
!!analysis is done over a full year.
!!
if (dn == 365) then
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!
!!Calculates the proportion of energy transfered by various mediums
!!!!!!!!!!!!!!!!!!!!!!!!!

L3: Do mth=1,12
```

```
Erfl_pv(mth)=Erfl_pv(mth)/Et_pv(mth)
Erfl_w(mth)=Erfl_w(mth)/Et_w(mth)
qaf_pv(mth)=qaf_pv(mth)/Et_pv(mth)
qab_pv(mth)=qab_pv(mth)/Et_pv(mth)
qaf_w(mth)=qaf_w(mth)/Et_w(mth)
qab_w(mth)=qab_w(mth)/Et_w(mth)
qopv(mth)=qopv(mth)/Et_pv(mth)
qow(mth)=qow(mth)/Et_w(mth)
qrm_pv(mth)=qrm_pv(mth)/Et_pv(mth)
qrm_w(mth)=qrm_w(mth)/Et_w(mth)
Etr(mth)=Etr(mth)/Et_w(mth)
Eelec(mth)=Eelec(mth)/Et_pv(mth)
End Do L3

Tave=sum/n

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following sends the energy data to a file
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

open (1,file='DATA.dat')

write(1,*) 'Results for PV Section'
write(1,'(a6,a14,6a10 )') 'Month','Tot E','Refl', 'Qairf','Qairb', 'Qout','Qroom','Elec'
L4: Do mth=1,12
        write(1,'(1x,i3,1x,f12.1,6x,6f10.6)') mth,Et_pv(mth),Erfl_pv(mth),qaf_pv(mth), &
&
qab_pv(mth),qopv(mth),qrm_pv(mth),Eelec(mth)

End Do L4

write(1,*) 'Average Daily PV temperatures:'
write(1,'(f10.6)') Tave


write(1,*) 'Results for Window Section'
write(1,'(a6,a14,6a10)') 'Month','Tot E','Refl', 'Qairf','qairb', 'Qout','Qroom','Etrans'
L5: Do mth=1,12
        write(1,'(1x,i3,1x,f12.1,6x,6f10.6)') mth,Et_w(mth),Erfl_w(mth),qaf_w(mth), qab_w(mth), &
&               qow(mth),qrm_w(mth),Etr(mth)

End Do L5

End if
Stop
End Program Yearly
```

APPENDIX E

Listing of 1D_2PV_2W.F90

FORTRAN subroutine used for system definition and solving called from Yearly.F90

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Subroutine name: 1D_2PV_2W
!!
!!
!!
!! Description:   This subroutine is the main subroutine where all the inputs are set,
!!                and the equations for the model are defined.
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Subroutine oned_2pv2W(tod,dn, Etot_pv, Etot_w, Erefl_pv, Erefl_w,qair1f_pv,qair1b_pv, &
&                                      qair1f_w, qair1b_w, qo_pv, qo_w, qroom_w, qroom_pv, &
&                                      Etrans, Egen, mth, Tpv)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Variable Declaration
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

implicit none
integer, intent (in) :: dn,tod
integer, intent (out) :: mth
real, intent (out) ::Etot_pv, Etot_w, Erefl_pv, Erefl_w, qair1f_pv, qair1f_w, Tpv, &
&                                      qo_pv, qo_w, qroom_w, qroom_pv, Etrans, Egen, qair1b_pv,
qair1b_w

real :: Tback          !Temperature of PV backplate
real ::Tw1             !Temperature of the single glazing in Vision section
real ::Tb              !Blind temperature
real ::Lgap_f          !Width of the front cavity (closest to outside)
real ::Lgap_b          !Width of the back cavity (closest to room)
real ::Hpv             !Height of the PV section
real ::Hw              !Height of the Vision section
real ::Vel_f           !Velocity of the front cavity (closest to outside)
real :: Vel_b          !Velocity of the air in back cavity
real :: Ebal_w         !Energy balance for Vision Section
real :: SA             !Surface azimuth
real :: Lat            !Latitude of the system
real :: tilt           !Tilt of the AFW (90 deg is vertical)
real :: gnd_refl       !The ground reflectance
real :: Gd             !The amount of diffuse solar radiation
real :: Gb             !The amount of beam solar radiation
real :: To             !The outdoor air temperature
real :: value          !Variable to keep track of largest Temperature change between iterations
real :: w              !Width of the AFW
real :: Aw             !Area of the vision Section
real :: Apv            !Area of the PV Section
real :: M_f            !Volumetric flow rate of air in front cavity
real :: M_b            !Volumetric flow rate of air in back cavity
real :: abs_pv         !absorptance of the PV cells
real :: refl_pv        !reflectance of the PV cells
real :: Troom          !room temperature
real :: Uwall          !Wall U-value
```

```
real :: hroom          !heat transfer coefficient between AFW and room
real :: ho             !heat transfer coefficient between AFW and outdoors
real :: ewall          !emissivity of the wall
real :: Uo_pv          !Thermal conductance to the outside from PV section
real :: Uo_w           !Thermal conductance to the outside from Vision section
real :: abs_b          !blind absorptance
real :: refl_b         !blind reflectance
real :: trans_b        !blind transmittance
real :: Tm             !Variable used to calculate the mean temperature
real :: eb             !emissivity of the blind
real :: ew1            !emissivity of the single glazing in Vision section
real :: ew2            !emissivity of the double glazed window
real :: Uroom_pv       !Thermal conductance between room and PV section
real :: Uroom_w        !Thermal conductance between room and Vision section
real :: Tpvg           !First pass guess of PV temperature
real :: Tbackg         !First pass guess of PV backplate
real :: Tw1g           !First pass guess of Tw1
real :: Tbg            !First pass guess of Tb
real :: Tw2_1g         !First pass guess of Tw2_1
real :: tol            !Tolerance to stop iterating
real :: Ucomb          !value used to sum two thermal conductance values
real :: Ebal_pv        !Energy balance for PV section
real :: Tw1_pvg        !First pass guess of Tw1_pv
real :: Taf_pvg        !First pass guess of Taf_pv
real ::   Tab_pvg      !First pass guess of Tab_pv
real ::   U1f_pv       !Thermal conductance between front glazing and PV front air cavity
real ::   U2f_pv       !Thermal conductance between PV and front front cavity
real ::   h1f_pv       !Conv. Heat Trans. Coef. between front glazing and PV front air cavity
real ::   h2f_pv       !Conv. Heat Trans. Coef. between PV and front air cavity
real ::   U1b_pv       !Thermal conductance between PV and back air cavity
real ::   h1b_pv       !Conv. Heat Trans. Coef. between PV and back air cavity
real ::   U2b_pv       !Thermal conductance between back wall and PV back air cavity
real ::   h2b_pv       !Conv. Heat Trans. Coef. between back wall and PV back air cavity
real ::   Qair2f_pv    !Energy absorbed in the front PV air cavity
real ::   Qair2b_pv    !Energy absorbed in the back PV air cavity
real ::   hrb_pv       !The radiative heat transfer coefficient in back PV cavity
real ::   qair2f_w     !Energy absorbed in the front Vision air cavity
real ::   qair2b_w     !Energy absorbed in the back Vision air cavity
real ::   hrf_pv       !Radiative heat transfer coefficient in front PV cavity
real ::   Urf_pv       !Radiative thermal conductance in front PV cavity
real ::   epv_f        !emissivity of the front of the PV
real ::   epv_b        !emissivity of the back of the PV
real ::   Urb_pv       !Radiative thermal conductance in back PV cavity
real ::   Taf_pv       !average air temperature in front PV cavity
real ::   Tab_pv       !average air temps in back PV cavity
real ::   Tw1_pv       !Temperature of the first glazing in PV section
real ::   Sw1_pv       !Radiation absorbed by first glazing in PV section
real :: Scov           !Radiation absorbed by PV cover
real ::   Spv          !Total Radiation absorbed by PV
real ::   Spv_heat     !Radiation absorbed in PV as heat
real ::   k1, k2, k3, k4, k5, k6, k7, k8          !Constants used to solve system of equations
real ::   k9, k10, k11, k12, k13, k14, k15, k16, k17, k18, k19, k20, k21
real ::   af, ab       !variables to solve system
real ::   sum, sum2    !dummy variables used to sum values
real ::   C            !thermal capacitance of air
real ::   Dens         !Air density
```

```
real :: Ip1, Ip2, Ip3, Ip4, Ip5, In1, In2, In3, In4, In5 !Variable used to calculate solar radiation abs.
real :: Sw1          !Radiation absorbed in single glazing in Vision section
real :: Sw2_1        !Radiation abs. in first glazing of double glazed window (cavity side)
real :: Sw2_2        !Radiation abs. in 2nd glazing of double glazed window (room side)
real :: Sblind       !Radiation abs. in blind
real :: hc_w2        !Convective heat transfer coeff. between glazings of W2
real :: Taf_wg       !First pass value of Average air temps. of front Vision section
real :: Tab_wg       !First pass value of average air temps of back vision section
real :: U1f_w        !Thermal conductance between W1 and front Vision air cavity
real :: U2f_w        !Thermal conductance between blind and front Vision air cavity
real :: h1f_w        !Conv. heat transfer coeff. between W1 and front Vision air cavity
real :: h2f_w        !Conv. heat transfer coeff. between blind and front Vision air cavity
real :: U1b_w        !Thermal conductance between blind and back Vision air cavity
real :: h1b_w        !Conv. heat transfer coeff. between blind and back Vision air cavity
real :: U2b_w        !Thermal conductance between W2 and back Vision air cavity
real :: h2b_w        !Conv. heat transfer coeff. between W2 and back Vision air cavity
real :: hr_w2        !Radiative heat transf. coeff. between glazings of W2
real :: hw2          !Combined rad. and conve. Heat trans. coeff. of W2
real :: Uw2          !Thermal conductance of W2
real :: Hr_w1b       !Radiative heat trans. coeff. between w1 and blind
real :: Urf          !Radiative thermal conductance between W1 and blind
real :: Urb          !Radiative thermal conductance between blind and W2
real :: hr_w2b       !Radiative heat trans. coeff. between blind and W2
real :: y(2)         !Temp. and its first derivative used to solve system of equations
real :: Tw2_2g       !First guess value of Tw2_2
real :: Sol_pv(2,101) !Solution found for temperatures using RungeK for PV section
real :: Sol(2,101)   !Solution found for temperatures using RungeK or Vision section
real :: LNG          !Longitude of the system
real :: STM          !Local standard time meridian of the system
real :: Taf          !Average air temperature in front Vision section
real :: Tab          !Average air temperature in back Vision section
real :: Tw2_1        !Temperature of first glazing of W2 (cavity side)
real :: Tw2_2        !Temperature of second glazing of W2 (room side)
real :: mem1,mem2, mem3      !Values used to keep stored memory between iterations
real :: rfl_w1_df    !diffuse reflectance of W1
real :: rfl_w2_df    !diffuse reflectance of W2
real :: tr_w1_bm     !beam transmittance of W1
real :: tr_w2_bm     !beam transmittance of W2
real :: rfl_w1_bm    !beam reflectance of W1
real :: rfl_w2_bm    !beam reflectance of W2
real :: tr_w1_df     !diffuse transmittance of W1
real :: tr_w2_df     !diffuse transmittance of W2
real :: tkw1         !thickness of w1
real :: tw2          !thickness of each glazing of W2
real :: kw1          !extinction coefficient of w1
real :: kw2          !extinction coeff. of w2
real :: abs_w1_bm    !beam absorptance of w1
real :: abs_w2_bm    !beam absorptance of w2
real :: abs_w1_df    !diffuse abs. of w1
real :: abs_w2_df    !diffuse abs of w2
real :: tkg1         !thickness of g1 (outside glazing in PV section)
real :: kg1          !extinction coeff. of g1
real :: tr_g1_bm     !beam trans. of g1
real :: rfl_g1_bm    !beam refl. of g1
real :: abs_g1_bm    !beam abs. of g1
real :: tr_g1_df     !diffuse trans. of g1
```

```
real ::    rfl_g1_df          !diffuse refl. of g1
real :: trans_g1              !average transmittance of g1
real :: refl_g1               !ave reflect. of g1
real :: trans_w1              !ave trans. of w1
real :: refl_w1               !ave refl. of w1
real ::    abs_g1_df          !diffuse abs. of g1
real ::    adj                !variable to adjust for fins
real ::    wadj               !variable to adjust width to account for fins
real ::    nfin               !number of fins used
real ::    mLc                !Value used in calculating fin effectiveness
real ::    Lc                 !Corrected length of fin to account of end effects
real ::    Lfin               !Length of the fin
real ::    tfin               !Fin thickness
real ::    sfin               !Spacing between each fin
real ::    Afin               !Surface area of the fin
real ::    kfin               !Fin material thermal conductivity

integer :: blind              !variable to allow to choose a blind from a selection of 4
integer ::i                   !dummy variable
integer ::Ni                  !number of divisions for air temperature in Vision section
integer ::Ni_pv               !number of divisions for air temp. in PV section
integer ::fin                 !variable to declare if fins are being used
integer ::yes, no  !dummy variables assigned 1 for yes and 0 for no
integer ::numf                !total number of fins used
integer ::count               !counter that keeps track of the number of iterations required

real :: sbc                   !stefen boltzman constant
Parameter (sbc=5.67*1e-8)!stefan boltzmann constant (W/m2K4)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Interface Declaration
!! (defining functions used in the program)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Interface
Function npv(T)                          ! PV efficiency
implicit none
Real, Intent(IN) :: T
Real :: npv
End function npv

End Interface

yes=1; no=0

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Model Inputs
!!
!INPUTS TO PROGRAM: The next section is where all the different parameters can be varied for the
program. The only other
!parameter that can be changed is in Yearly where you define the time span that you want the program
!!to run for
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
tol=0.5                 !tolerance to stop interating
Lat=45                  !Latitude
tilt=90                 !tilt angle from horizontal
LNG=74                  !longitude for Montreal region
STM=75                  !local standard time meridian
SA=0                    !Surface azimuth
gnd_refl=0.20           !ground reflection
kg1=6.96                !extinction coefficient of PV cover
kw1=6.96                !extinction coefficient for w1
kw2=6.96                !extinction coefficient for glazings in w2

tkg1=0.0003             !thickness of PV cover
tkw1=0.005              !thickness of w1
tw2=0.003               !thickness of glazings in w2
Hpv=1.12                !PV Section height
Hw=0.636                !Window section height
Vel_f=0.3               !Air velocity in the front cavity
Vel_b=0.6               !Air velocity in the back cavity
Lgap_f=0.035            !Gap width in front cavity
Lgap_b=0.055            !Gap width in back cavity
w=0.92                  !width of airflow window
abs_pv=0.92             !absorptivity of PV
refl_pv=0.08            !reflectivity of PV
epv_b=0.1               !emissivity of back of PV panel (oxidize aluminum)
epv_f=0.9               !emissivity of front of PV panel
Uwall=1                 !thermal conductance of wall (W/m2K)
ewall=0.9               !emissivity of wall
eb=0.9                  !emissivity of blind
ew1=0.9                 !emissivity of single glazing
ew2=0.9                 !emissivity of double glazing
hc_w2=2                 !convective heat transfer between 2 glazings in W2
hroom=6.5               !heat transfer coefficient to room
Troom=0.4               !room temperature
ho=12                   !heat transfer coefficient to outside

!!
!!The following sets the blind optical properties
!!
!There are four different blinds that can be used in this model. They have the following properties:
!blind 1: absorptivity=0.5, reflectivity=0.3, transmissivity=0.2
!blind 2: absorptivity=0.3, reflectivity=0.5, transmissivity=0.2
!blind 3: absorptivity=0.7, reflectivity=0.2, transmissivity=0.1
!blind 4: absorptivity=0.55, reflectivity=0.4, transmissivity=0.05

Blind=1                 !blind selection

If (Blind == 1) then
        abs_b=0.5
        trans_b=0.2
        refl_b=1-abs_b-trans_b
Else if (Blind == 2) then
        abs_b=0.3
        trans_b=0.2
        refl_b=1-abs_b-trans_b
Else if (blind==3) then
        abs_b=0.7
```

```
        trans_b=0.1
        refl_b=1-abs_b-trans_b
Else if (blind == 4) Then
        abs_b=0.55
        trans_b=0.05
        refl_b=1-abs_b-trans_b
Else
        abs_b=0.5
        trans_b=0.2
        refl_b=1-abs_b-trans_b
        Print *,'Improper blind selection, default (blind 1) used'
End If

Aw=w*Hw             !area of window section
Apv=w*Hpv           !area of pv section
M_f=Lgap_f*w*Vel_f  !Volumetric flow rate of air (m3/s) in front cavity
M_b=Lgap_b*w*Vel_b  !Volumetric flow rate of air (m3/s) in back cavity

Uroom_pv=hroom*Apv  !thermal conductance between wall and room for pv section
Uroom_w=hroom*Aw    ! "" for window section
Uo_pv=ho*Apv        !thermal conductance between pv and outside
Uo_w=ho*Aw          !thermal conductance between single glazing and outside

!FINS
fin=no                          !are fins included? yes or no
Lfin=0.015                      !fin length
tfin=0.002                      !fin thickness
Sfin=3.*tfin                    !fin spacing according to Rolle
Lc=Lfin + tfin/2.               !corrected fin length
Afin=2.*Hpv*Lc                  !fin area
Numf=w/(tfin+Sfin)              !number of fins
kfin=236                        !conductivity of al according to Rolle
adj=0.5                         !needed in energy balance to adjust

call envcond(tod,dn,SA,Lat,LNG,STM,tilt,gnd_refl,kw1,kw2,tkw1,tw2,kg1,tkg1, &
& Gb,Gd,To, mth, tr_w1_bm, tr_w2_bm, rfl_w1_bm, rfl_w2_bm, tr_w1_df, &
& tr_w2_df, rfl_w1_df, rfl_w2_df,rfl_g1_bm,rfl_g1_df,tr_g1_bm,tr_g1_df)

trans_w1= (Gd/(Gb+Gd))*tr_w1_df +(Gb/(Gb+Gd))*tr_w1_bm
refl_w1= (Gd/(Gb+Gd))*rfl_w1_df+(Gb/(Gb+Gd))*rfl_w1_bm
trans_g1= (Gd/(Gb+Gd))*tr_g1_df +(Gb/(Gb+Gd))*tr_g1_bm
refl_g1= (Gd/(Gb+Gd))*rfl_g1_df+(Gb/(Gb+Gd))*rfl_g1_bm

!print *,'trans_cov', trans_g1, 'refl_cov', refl_g1
!print *,'trans_w1', trans_w1, 'refl_w1', refl_w1

abs_w1_bm=1-tr_w1_bm-rfl_w1_bm      !beam abs. of w1
abs_w2_bm=1-tr_w2_bm-rfl_w2_bm      !beam abs. of each glazing of w2
abs_w1_df=1-tr_w1_df-rfl_w1_df      !diffuse abs. of w1
abs_w2_df=1-tr_w2_df-rfl_w2_df      !diffuse abs. of each glazing of w2
abs_g1_bm=1-tr_g1_bm-rfl_g1_bm      !beam abs. of PV cover
abs_g1_df=1-tr_g1_df-rfl_g1_df      !diffuse abs. of PV cover

!First guess values for Tw1_pv, Tpv, Tback, Tw1, Tb, Tw2, Taf_pv, Tab_pv, Taf_w, Tab_w
Tw1_pvg=21
Tpvg=25
```

```
Tbackg=22
Tw1g=19
Tbg=25
Tw2_1g=22
Tw2_2g=21
Taf_pvg=20
Tab_pvg=20
Taf_wg=24
Tab_wg=24


!END OF INPUTS TO PROGRAM

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following while loop calculates the temperatures in the PV section using an iterative
!!process.  It stops once convergence in the temperatures occurs.
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

value=10
mem2=0 ;mem1=0; count=0
Do While (value > tol)
count=count+1
h1f_pv=0; h2f_pv=0; h1b_pv=0; h2b_pv=0;
call htcoef1D(Lgap_f,Hpv,Vel_f,Tw1_pvg,Tpvg,Taf_pvg,(Gd+Gb),h1f_pv,h2f_pv)
call htcoef1D(Lgap_b,Hpv,Vel_b,Tpvg,Tbackg,Tab_pvg,(Gd+Gb),h1b_pv,h2b_pv)

!h1f_pv=4 !1.9*h1f_pv !4.1
!h2f_pv=7 !1.9*h2f_pv !8.1
!h1b_pv=7.5 !1.9*h1b_pv !10.1
!h2b_pv=4.5 !1.9*h2b_pv !7.4

print *,'h1f_pv', h1f_pv, 'h2f_pv', h2f_pv
print *,'h1b_pv', h1b_pv, 'h2b_pv', h2b_pv

mLc=(2.*h1b_pv/(kfin*Lc*tfin))**0.5 * Lc**1.5
nfin=tanh(mLc)/mlc
wadj=2.*numf*Lfin*nfin + 0.5*((1.+adj)*w + (1.-adj)*nfin*w)

U1f_pv=Apv*h1f_pv
U2f_pv=Apv*h2f_pv
U2b_pv=Apv*h2b_pv

if (fin==yes)then
U1b_pv=(Afin*numf*nfin + (Apv - numf*Hpv*tfin))*h1b_pv
else
U1b_pv=Apv*h1b_pv
End if


Ucomb=Uroom_pv*Uwall/(Uroom_pv+Uwall)

Tm=273.15+(Tw1_pvg+Tpvg)/2.

hrf_pv=4.*sbc*Tm**3/((1/epv_f)+(1/ewall) -1)
```

```
Urf_pv=Apv*hrf_pv

Tm=273.15+(Tpvg+Tbackg)/2.

hrb_pv=4.*sbc*Tm**3/((1/epv_b)+(1/ewall) -1)
Urb_pv=Apv*hrb_pv


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!Calculation of solar radiation absorbed in each layer
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Sw1_pv=0;Scov=0;Spv=0;Erefl_pv=0;Spv_heat=0; Etot_pv=0

Ip1=Gd + Gb
Ip2=(1- refl_w1*(refl_g1 + trans_g1**2 *refl_pv*(1-refl_pv*refl_g1)**(-1)))**(-1) *Ip1*trans_w1
Ip3=(1- refl_pv*refl_g1)**(-1) *trans_g1*Ip2
In3=Ip3* refl_pv
In2=Ip2* refl_g1 + In3*trans_g1
In1=In2* trans_w1 + Ip1*refl_w1

Sw1_pv   = w*Hpv*((Ip1+In2) - (Ip2+In1))
Scov  = w*Hpv*((Ip2+In3) - (Ip3+In2))
Spv         = w*Hpv*(Ip3-In3)

Etot_pv= Ip1*Apv
Erefl_pv = In1*Apv

Egen=Apv*Ip3*npv(Tpvg)
Spv_heat=Spv-Egen


!Print *,'Etot_pv', Etot_pv, 'Erefl_pv', Erefl_pv
!Print *, 'Sw1_pv', Sw1_pv, 'Scov', Scov, 'Spv_heat', Spv_heat, 'Egen', Egen

Spv_heat=Spv_heat + Scov            !Treating PV cover, cells and backplate as one


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Solving the energy balance equations for the PV section
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

af=M_f*C(Taf_pvg)*dens(Taf_pvg)/w
ab=M_b*C(Tab_pvg)*dens(Tab_pvg)/w

k1=Uo_pv + Urf_pv + U1f_pv
k2=Urf_pv + Urb_pv + U2f_pv + U1b_pv
k3=Urb_pv + U2b_pv + Ucomb
k4=(k2- Urf_pv**2/k1 - Urb_pv**2/k3)**(-1) * (U2f_pv + Urf_pv*U1f_pv/k1)
k5=(k2- Urf_pv**2/k1 - Urb_pv**2/k3)**(-1) * (U1b_pv + Urb_pv*U2b_pv/k3)
k6=(k2- Urf_pv**2/k1 - Urb_pv**2/k3)**(-1) * (Ucomb*Troom*Urb_pv/k3 + Spv_heat + &
&
Urf_pv*(Uo_pv*To +Sw1_pv)/k1)
```

```
k7=(Urf_pv*k4 + U1f_pv)/k1
k8=Urf_pv*k5/k1
k9=(Urf_pv*k6 + Uo_pv*To + Sw1_pv)/k1
k10=Urb_pv*k4/k3
k11=(Urb_pv*k5 + U2b_pv)/k3
k12=(Ucomb*Troom + Urb_pv*k6)/k3
k13=(h1f_pv*(k7-1) + h2f_pv*(k4-1))/af
k14=(h1f_pv*k8 + h2f_pv*k5)/af
k15=(h1f_pv*k9 + h2f_pv*k6)/af

if (fin == yes) then
k16=((wadj/w)*h1b_pv*k4 + h2b_pv*k10)/ab
k17=((wadj/w)*h1b_pv*(k5-1) + h2b_pv*(k11-1))/ab
k18=((wadj/w)*h1b_pv*k6 + h2b_pv*k12)/ab
Else
k16=(h1b_pv*k4 + h2b_pv*k10)/ab
k17=(h1b_pv*(k5-1) + h2b_pv*(k11-1))/ab
k18=(h1b_pv*k6 + h2b_pv*k12)/ab
End If


!!
!!Using the Runga Kutta method to solve for the air temperatures
!!
y(1)=To
y(2)=y(1)

Ni_pv=50                    !number of divisions in Hpv

call rkm(2,y,Hpv, Ni_pv, k13,k14,k15,k16,k17,k18,sol_pv)

sum=0; sum2=0
Do i=1,Ni_pv+1
sum=sum+Sol_pv(1,i)
sum2=sum2+Sol_pv(2,i)
End Do

Taf_pv=sum/(Ni_pv+1)
Tab_pv=sum2/(Ni_pv+1)

Tw1_pv=k7*Taf_pv + k8*Tab_pv + k9
Tpv=k4*Taf_pv + k5*Tab_pv + k6
Tback= k10*Taf_pv + k11*Tab_pv + k12

!!Sets tolerance values in memory to avoid getting stuck in a perpetual loop
mem3=mem2
mem2=mem1
mem1=value

!!
!!The following sets the tolerance of the loops based on the largest temperature difference
!!between the temperature calculated in the loops compared to previous iteration
!!

value=abs(Tw1_pv-Tw1_pvg)
If(abs(Tpv-Tpvg)>value) then
```

```
value=abs(Tpv-Tpvg)
End if

If(abs(Tback-Tbackg)>value) then
value=abs(Tback-Tbackg)
End If

If(abs(Taf_pv-Taf_pvg)>value) then
value=abs(Taf_pv-Taf_pvg)
End If

If(abs(Tab_pv-Tab_pvg)>value) then
value=abs(Tab_pv-Tab_pvg)
End If

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Sometimes the program gets stuck in a loop, and can't converge. The program detects this
!!and breaks the loop and prints the tolerance on convergence that was used for that loop
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
If ((mem3 == value) .or. mem2==value .or. count==20) then
print *, 'Temperatures not converging, used tolerance of:', value
value=tol-0.1
End If

!!
!!Setting the intial guesses for the temperatures as the last loops calculated values
!!

Twl_pvg=Twl_pv
Tpvg=Tpv
Taf_pvg=Taf_pv
Tab_pvg=Tab_pv
Tbackg=Tback
End Do

!!
!!Calculating the energy transferred in the system
!!

qair1f_pv=M_f*dens(Taf_pv)*C(Taf_pv)*(Sol_pv(1,Ni_pv+1)- To)
qair2f_pv=U1f_pv*(Twl_pv-Taf_pv) + U2f_pv*(Tpv-Taf_pv)
qair1b_pv=M_b*dens(Tab_pv)*C(Tab_pv)*(Sol_pv(2,Ni_pv+1)- To)
qair2b_pv=U1b_pv*(Tpv-Tab_pv) + U2b_pv*(Tback-Tab_pv)
qroom_pv=Ucomb*(Tback-Troom)
qo_pv=Uo_pv*(Twl_pv-To)
Ebal_pv=Etot_pv-Erefl_pv-qair1f_pv-qair1b_pv-qroom_pv-qo_pv-Egen

!print *,'Efficiency PV Section', (qair1f_pv+qair1b_pv+Egen)/Etot_pv
!print *,' Erefl', Erefl_pv, 'qo_pv', Qo_pv, 'Qroom_pv', qroom_pv
!print *,'Qairf', Qair1f_pv, 'Qairb', Qair1b_pv
!print *,'Gbeam', Gb, 'Gdiffuse', Gd
!print *,'Twl_pv', Twl_pv, 'Tpv', Tpv, 'Tback', Tback, 'Taf_pv', Taf_pv, 'Tab_pv', Tab_pv
!print *,'Tair exit front',Sol_pv(1,Ni_pv+1) , 'Tair exit back', Sol_pv(2,Ni_pv+1)
!print *, 'npv', npv(Tpv), 'Electricity', Egen
```

```
!print *, 'nfin', nfin
!print *, 'Ebalance PV Section', Ebal_pv

!Print *,'Front radiation exchange:', Urf_pv*(Tpv-Tw1_pv)
!Print *,'Back radiation exchange:', Urb_pv*(Tpv-Tback)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!
!CALCULATIONS DONE FOR THE VISION SECTION
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!The following is to determine the heat from solar radiation captured in each element of Vision Section
!!

Etot_w=0; Etrans=0; Erefl_w=0; Sw1=0; Sblind=0; Sw2_1=0;Sw2_2=0

!diffuse calculations

Ip1=Gd
In5=0              !assuming negligible radiation is leaving from room

k1=(1-rfl_w2_df)**(-1) * tr_w2_df
k2=(1-rfl_w2_df)**(-1) * rfl_w2_df*tr_w2_df
k3=tr_w2_df*rfl_w2_df*k1
k4=In5*tr_w2_df**2 + rfl_w2_df*k2*tr_w2_df
k5=(1-refl_b*k3)**(-1) * trans_b
k6=(1-refl_b*k3)**(-1) * k4*refl_b
k7=refl_b + trans_b*k3*k5

Ip2=(1-rfl_w1_df*k7)**(-1) * (Ip1*tr_w1_df + rfl_w1_df*trans_b*k4)
Ip3=k5*Ip2 + k6
Ip4=k1*Ip3 + k2
Ip5=tr_w2_df*Ip4 + In5

In2=k7*Ip2 +trans_b*k4
In3=k3*Ip3 +k4
In4=tr_w2_df*In5 + Ip4*rfl_w2_df
In1=In2*tr_w1_df + Ip1*rfl_w1_df

Etot_w=Gd*Aw
Etrans=Ip5*Aw
Erefl_w=In1*Aw

Sw1= Aw*((Ip1+In2)-(Ip2+In1))
Sblind= Aw*((Ip2+In3)-(Ip3+In2))
Sw2_1= Aw*((Ip3+In4)-(Ip4+In3))
Sw2_2= Aw*((Ip4+In5)-(Ip5+In4))

!For beam radiation

Ip1=Gb
In5=0              !assuming negligible radiation is leaving from room
```

```
k1=(1-rfl_w2_bm)**(-1) * tr_w2_bm
k2=(1-rfl_w2_bm)**(-1) * rfl_w2_bm*tr_w2_bm
k3=tr_w2_bm*rfl_w2_bm*k1
k4=In5*tr_w2_bm**2 + rfl_w2_bm*k2*tr_w2_bm
k5=(1-refl_b*k3)**(-1) * trans_b
k6=(1-refl_b*k3)**(-1) * k4*refl_b
k7=refl_b + trans_b*k3*k5

Ip2=(1-rfl_w1_bm*k7)**(-1) * (Ip1*tr_w1_bm + rfl_w1_bm*trans_b*k4)
Ip3=k5*Ip2 + k6
Ip4=k1*Ip3 + k2
Ip5=tr_w2_bm*Ip4 + In5

In2=k7*Ip2 +trans_b*k4
In3=k3*Ip3 +k4
In4=tr_w2_bm*In5 + Ip4*rfl_w2_bm
In1=In2*tr_w1_bm + Ip1*rfl_w1_bm

Etot_w=Etot_w+Gb*Aw
Etrans=Etrans+Ip5*Aw
Erefl_w=Erefl_w + In1*Aw

Sw1= Sw1+ Aw*((Ip1+In2)-(Ip2+In1))
Sblind= Sblind+ Aw*((Ip2+In3)-(Ip3+In2))
Sw2_1= Sw2_1+ Aw*((Ip3+In4)-(Ip4+In3))
Sw2_2=Sw2_2+ Aw*((Ip4+In5)-(Ip5+In4))


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following while loop calculates the temperatures in the Vision section using an iterative
!!process. It stops once convergence in the temperatures occurs.
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


value=10
mem1=0; mem2=0;count=0
Do While (value > tol)
count=count+1
call htcoef1D(Lgap_f,Hw,Vel_f,Tw1g,Tbg,Taf_wg,(Gd+Gb),h1f_w,h2f_w)
call htcoef1D(Lgap_b,Hw,Vel_b,Tbg,Tw2_1g,Tab_wg,(Gd+Gb),h1b_w,h2b_w)

U1f_w=Aw*h1f_w
U2f_w=Aw*h2f_w
U1b_w=Aw*h1b_w
U2b_w=Aw*h2b_w

Tm=273.15 + (Tw2_1g+Tw2_2g)/2.

hr_w2=4.*sbc*Tm**3/((1/ew2)+(1/0.9) -1)

hw2=hr_w2+hc_w2
Uw2=Aw*hw2
```

```
Tm=273.15 + (Tw1g+Tbg)/2.
hr_w1b=4.*sbc*Tm**3/((1/ew1)+(1/eb) -1)
Urf=hr_w1b*Aw

Tm=273.15 + (Tw2_1g+Tbg)/2.
hr_w2b=4.*sbc*Tm**3/((1/ew2)+(1/eb) -1)
Urb=hr_w2b*Aw

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Solving the energy balance equations for the Vision section
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

af=M_f*C(Taf_wg)*dens(Taf_wg)/w
ab=M_b*C(Tab_wg)*dens(Tab_wg)/w

k1=Uo_w + U1f_w + Urf
k2=Urf + Urb + U2f_w + U1b_w
k3=Urb + U2b_w + Uw2
k4=(k3- Uw2**2 /(Uroom_w + Uw2))**(-1) * Urb
k5=(k3- Uw2**2 /(Uroom_w + Uw2))**(-1) * U2b_w
k6=(k3- Uw2**2 /(Uroom_w + Uw2))**(-1) * (Sw2_1 + Uw2*((Sw2_2+Uroom_w*Troom)/(Uroom_w
+Uw2)))
k7=(k2- Urf**2 /k1 -Urb*k4)**(-1) * (Urf*U1f_w/k1 +U2f_w)
k8=(k2- Urf**2 /k1 -Urb*k4)**(-1) * (Urb*k5 + U1b_w)
k9=(k2- Urf**2 /k1 -Urb*k4)**(-1) * (Sblind +Urb*k6 +Urf*(Uo_w*To+Sw1)/k1)
k10=k4*k7
k11=k4*k8 + k5
k12=k4*k9 + k6
k13=(U1f_w + Urf*k7)/k1
k14=Urf*k8/k1
k15=(Uo_w*To + Sw1 + Urf*k9)/k1
k16=(h1f_w*(k13-1) + h2f_w*(k7-1))/af
k17=(h1f_w*k14 + h2f_w*k8)/af
k18=(h1f_w*k15 + h2f_w*k9)/af
k19=(h1b_w*k7 + h2b_w*k10)/ab
k20=(h1b_w*(k8-1) + h2b_w*(k11-1))/ab
k21=(h1b_w*k9 + h2b_w*k12)/ab

!!
!!Using the Runge Kutta method to solve air temperatures
!!

y(1)=Sol_pv(1,Ni_pv+1)
y(2)=Sol_pv(2,Ni_pv+1)

Ni=50
Call rkm(2,y,Hw, Ni, k16,k17,k18,k19,k20,k21,sol)

sum=0; sum2=0
L2: Do i=1,Ni+1

sum=sum+Sol(1,i)
sum2=sum2+Sol(2,i)
End Do L2
```

```
Taf=sum/(Ni+1.)
Tab=sum2/(Ni+1.)

Tb=k7*Taf + k8*Tab + k9
Tw2_1=k4*Tb + k5*Tab + k6
Tw1=k13*Taf + k14*Tab + K15
Tw2_2=(Uw2*Tw2_1 + Uroom_w*Troom + Sw2_2)/(Uroom_w + Uw2)

!!Sets tolerance values in memory to avoid getting stuck in a perpetual loop
mem3=mem2
mem2=mem1
mem1=value


!!
!!The following sets the tolerance of the loops based on the largest temperature difference
!!between the temperature calculated in the loops compared to previous iteration
!!
value=abs(Tw1-Tw1g)

if (abs(Tb-Tbg)>value) then
value=abs(Tb-Tbg)
End If

If(abs(Tw2_1-Tw2_1g)>value) then
value=abs(Tw2_1-Tw2_1g)
End If

If (abs(Taf - Taf_wg)>value) then
value=abs(Taf - Taf_wg)
End if

If (abs(Tab - Tab_wg)>value) then
value=abs(Tab - Tab_wg)
End if

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Sometimes the program gets stuck in a loop, and can't converge. The program detects this
!!and breaks the loop and prints the tolerance on convergence that was used for that loop
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
If (mem2 == value .or. mem3==value .or. count==20) then
print *, 'Temperatures not converging, used tolerance of:', value
value=tol-0.1
End If

!!
!!Setting the initial guesses for the temperatures as the last loops calculated values
!!
Tw1g=Tw1
Tbg=Tb
Tw2_1g=Tw2_1
Tw2_2g=Tw2_2
Taf_wg=Taf
```

```
Tab_wg=Tab
End Do

!!
!!Calculating the energy transferred in the system
!!


qair1f_w=M_f*dens(Taf)*C(Taf)*(Sol(1,Ni+1)-Sol_pv(1,Ni_pv+1))
qair2f_w=U1f_w*(Tw1-Taf) + U2f_w*(Tb-Taf)
qair1b_w=M_b*dens(Tab)*C(Tab)*(Sol(2,Ni+1)-Sol_pv(2,Ni_pv+1))
qair2b_w=U1b_w*(Tb-Tab) + U2b_w*(Tw2_1-Tab)

qroom_w=Uroom_w*(Tw2_2-Troom)
qo_w=Uo_w*(Tw1-To)
Ebal_w=Etot_w-Erefl_w-Etrans-qair1f_w-qair1b_w-qroom_w-qo_w


Return
End Subroutine oned_2pv2W

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Function Definition
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Function npv(T)                    ! Calculates PV efficiency based on Tpv
implicit none
Real, Intent(IN) :: T
Real :: npv

if (T>25) then
npv=0.062 - 0.0004*(T-25)
        Else
npv=0.062
End If

End function npv
```

APPENDIX F

Listing of Model.F90

FORTRAN program to calculate performance of ventilated façades

in Configuration 1 for 2D model

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!      Program: Model (2DModel1pv_2w)
!!
!!      Description: This is the Program that interfaces with all of the subroutines
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


PROGRAM model
implicit none

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Variable declaration and input number of elements per component
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


Integer:: i                     !dummy variable
Integer:: N                     !Total number of elements in system
Integer:: N1                    !Number of elements per component
Integer:: Npv                   !Total number of elements in PV section
Integer:: Nw                    !Total number of elements per component in Vision section

Integer:: div
Parameter (div=10)        !Input the number of elements per component (Max=700/14)

PARAMETER (N=div*(5+7)) !Total number of elements (Max 700), not that the (N=div*(a+b)) the div is
!the number of elements a section is divided into, and the a is the number of components in the PV section,
!while the b is the number of elements in the vision section

REAL:: T(N)                     !Array defining elemental temperatures
Real:: f(N)                     !Array defining energy balance equations for each element
LOGICAL :: check                !Variable used to check convergence


Npv=(N/12)*5                    !Number of elements in the PV Section
Nw=(N/12)*7                     !Number of elements in the Vision Section
N1=N/12                         !Number of elements per PV surface

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Input of temperatures used in the first iteration of the model
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


L1:     Do i=1,N1
        T(i)=277          +i*(4./N1)      !PV cover temperature
        T(i+N1)=279       +i*(4./N1)      !PV temperature
        T(i+2*N1)=280     +i*(4./N1)      !PV backplate temperature
        T(i+3*N1)=263.1   +i*(4./N1)      !air temperature in PV section
        T(i+4*N1)=271     +i*(2./N1)      !back wall temperature
```

```
          T(Npv+i)=269.5  +i*(2./N1)          !g1 in vision section
          T(Npv+i+N1)=265.15 +i*(2./N1)   !front air temperature in Vision section
          T(Npv+i+2*N1)=285 + i*(4./N1)    !front temperature of blind
          T(Npv+i+3*N1)=285.4 + i*(4./N1) !back temperature of blind
          T(Npv+i+4*N1)=264.1 + i*(4./N1) !back air temperature in Vision section
          T(Npv+i+5*N1)=285.5 + i*(4./N1) !first glazing temp in dbl glazed window
          T(Npv+i+6*N1)=290 + i*(4./N1)   !second glazing temp in dbl glazed window

          End Do L1

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Interface with subroutine broydn, and funcv (EqnGen1pv_2w)
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

call broydn(T,N,check)
call funcv(N,T,f)

open (1,file='DATA.dat')
          if (check) then
   write(1,*) 'Convergence problems.'
   end if


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following section sends the resulting temperatures to a file called Data
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

write(1,*) 'Results for PV Section'

write(1,'(10x,10a12)') 'Tcover', 'f', 'Tpv','f', 'Tback', 'f', &
& 'Tair', 'f', 'Twall', 'f'

L2:       do i=1,N1
     write(1,'(10x,10f12.6)') T(i),f(i),T(i+N1),f(i+N1), T(i+2*N1),f(i+2*N1),T(i+3*N1),f(i+3*N1), &
            & T(i+4*N1),f(i+4*N1)
          End do L2

write(1,*) 'Results for Vision Section'

write(1,'(14x, 14a12)') 'Tg1','f', 'Tair_f', 'f', 'Tblind_f', 'f', 'Tblind_b', 'f', 'Tair_b', 'f', &
& 'Tw2_1', 'f', 'Tw2_2', 'f'

L3:       do i=Npv+1,Npv+N1
     write(1,'(14x,14f12.6)') T(i),f(i),T(i+N1),f(i+N1), T(i+2*N1),f(i+2*N1),T(i+3*N1),f(i+3*N1), &
            & T(i+4*N1),f(i+4*N1), T(i+5*N1),f(i+5*N1), T(i+6*N1),f(i+6*N1)
          End do L3


END Program model
```

APPENDIX G

Listing of funcv.F90

FORTRAN subroutine used for system definition and solving called from Model.F90

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!! Subroutine name: funcv (EqnGen1pv_2w)
!!
!!
!!
!! Description:    This subroutine is the main subroutine where all the inputs are set,
!!          and the equations for the model are defined.
!!
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

SUBROUTINE funcv(n,T,f)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Variable Declaration
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

implicit none

INTEGER, intent(in):: n
Real, intent(in):: T(n)
Real, intent(out):: f(n)
Integer :: i,j                          !Dummy variables used for loop definitions
Integer :: N2                           !Number of elements in each component
Integer :: indx_pv(2*(N/12)+3), indx_w(2*(N/12)+3)      !dummy array used in LU decomposition
Integer :: Nw, NoPV                     !Number of elements in Vision and PV section, respectively
Integer :: indx_w2(2*(N/12)+3)     !Dummy array used in LU decomposition
Integer :: yes, no                 !Given values of 1 and 0 to set if fins, and low_e coating are used
Integer :: fin                          !Variable to state if fins are used yes or no (1 or 0)

Real:: H_tot_pv                         !The total height of the PV section
real ::   A(2*(N/12)+3,2*(N/12)+3)      !The radiosity matrix
real ::   abs_b                         !Blind absortance
real ::   Acov                          !Percent coverage of the PV cells
real ::   Check_1w                      !Dummy variable used to verify that solar rad abs. in each
                                        Vision element is well defined
real ::   dA_pv                         !The area of each PV element
real ::   dA_w                          !The area of elements in the Vision section
real ::   dAfin                         !The elemental fin area
real ::   dH_pv                         !The height of each PV element
real ::   dH_w                          !The height of elements in the Vision section
real ::   eb                            !The emissivity of the blind
real ::   eback                         !The emissivity of the PV back plate
real ::   Ebal_pv                       !The energy balance of the PV Section
real ::   Ebal_w                        !The energy balance of the Vision section
real ::   echan_b                       !The emissivity of the back cavity side walls of PV section
real ::   echan_f                       !The emissivity of the front cavity side walls of PV section
real ::   ecov                          !The emissivity of the PV cover
real ::   Eff_pv                        !PV Section efficiency
real ::   Eff_w                         !Efficiency of Vision section
real ::   egl                           !The emissivity of the front glazing
real ::   ein_b                         !The emissivity of the inlet of the back PV cavity
real ::   ein_f                         !The emissivity of the inlet of the front PV cavity
```

```
real ::  Eloss_pv                    !The amount of energy that is lost in the inlet and outlet sections
real ::  Eloss_w                     !The energy lost to inlet/outlet and side walls in vision section
real ::  eout_b                      !The emissivity of the outlet of the back PV cavity
real ::  eout_f                      !The emissivity of the outlet of the front PV cavity
real ::  Epv                         !The total amount of electricity generated
real ::  Erefl_pv                    !The amount of energy reflected in the PV section
real ::  Erefl_w                     !The energy reflected out of the Vision section
real ::  Etot_pv                     !The total amount of incident solar radiation available in PV section
real ::  Etot_w                      !The total available solar rad incident on Vision section
real ::  Etrans                      !The solar rad transmitted through the Vision Section
real ::  ew2                         !Emissivity of the outer sides of the double glazed window
real ::  ewall                           !The emissivity of the back wall
real ::  Ff(2*(N/12)+3, 2*(N/12)+3)      !Array used to calculate view factors
real ::  Ff_pv(((N/12)*5)+6,((N/12)*5)+6) !View factors for the PV section
real ::  Ff_w(((N/12)*7)+6,((N/12)*7)+6)  !View factors for the Vision section
real ::  G                               !The solar radiation incident on the system
real ::  H_tot_w                         !The height of the PV section
real ::  h1_pv(N/12)                     !CHTC in back of PV
real ::  h1b_w(N/12)                     !CHTC on front of blind
real ::  h1f_w(N/12)                     !CHTC on single glazing of vision section
real ::  h2_pv(N/12)                     !CHTC on back wall
real ::  h2b_w(N/12)                     !CHTC on cavity side of double glazed window
real ::  h2f_w(N/12)                     !CHTC on back of blind
real ::  ho                          !Combined convective and radn heat trans. coef. on external glazings
real ::  hroom                       !Combined convect. and radn heat trans. coef. from room side elements
real ::  hw2                             !The heat transfer coefficient between the two glazings of W2
real ::  Ip1, Ip2, Ip3, In1, In2, In3,Ip4, Ip5, In4, In5     !variables used to calculate solar rad
                                                             absorbed in ea layer
real ::  k1, k2, k3, k4, k5, k6      !Variables used in the calculation of absorbed solar radiation in ea layer
real ::  kb                              !The blind thermal conductivity
real ::  kback                           !The thermal conductivity of the PV back plate
real ::  kcov                            !The thermal conductivity of the PV cover
real ::  kfin                            !The thermal conductivity of the fins
real ::  kg                              !The thermal conductivity of the glass
real ::  kpv                             !The thermal conductivity of the PV cells
real ::  L_back                          !The (back) gap width between the PV and the back wall
real ::  L_front                         !The (front) gap width between the front glazing and the PV
real ::  Lc                              !The corrected fin length
real ::  Lfin                            !The length of fins
integer ::low_e   !An argument used to determine if there is low-e coating in double glazed window
integer :: blind                         !An argument used to determine if what blind type to use
real ::  Lpv                             !Width of the PV cavity
real ::  mLc(N/12)                       !A term needed to calculate energy transfer from fins
real ::  nfin(N/12)                      !A term needed to calculate energy transfer from fins
real ::  npv(N/12)                       !The array defining the elemental PV cell efficiency
real ::  num                             !Dummy variable used to store a number in memory
real ::  Numf                            !The total number of fins
real ::  P                               !Dummy variable used for the ludcmp subroutine
real ::  Pdif                            !Proportion of solar radiation that is diffuse
real ::  Q(2*(N/12)+3)                   !Array needed for the Lubksb subroutine
real ::  Qair_pv                         !The energy absorbed by the air flowing in PV section
real ::  Qairb_w                     !The solar rad absorbed in the air flowing in back Vision cavity
real ::  Qairf_w                         !The energy transfered to air flowing in front cavity
real ::  Qout_pv                         !The energy lost to the outside from the PV section
real ::  Qout_w                          !The energy lost to outside from Vision section
real ::  Qroom_pv                        !The energy transferred to the room from the PV section
```

```
real ::  Qroom_w                    !The energy transferred to the room from the vision section
real ::  refl_b                     !Reflectance of the blind
real ::  refl_b_lw                  !Long wave reflectance of blind
real ::  refl_back                  !The reflectance of the PV back plate
real ::  refl_back_lw               !The long wave reflectance of the PV back plate
real ::  refl_chanb          !The reflectance of the back cavity side walls of the PV section
real ::  refl_chanf                 !The reflectance of the front cavity side walls of PV section
real ::  refl_cov                   !The reflectance of the PV cover
real ::  refl_cov_lw                !The long wave refl. of the PV cover
real ::  refl_g1                    !The reflectance of the front single glazing
real ::  refl_g1_lw                 !Longwave refl. of single glazing
real ::  refl_g2                    !Reflectance of glazings in the double glazed window
real ::  refl_g2_lw                 !Long wave reflectance of glazings of double glazed window
real ::  refl_inb                   !The reflectance of the inlet of the back PV section
real ::  refl_inf                   !The reflectance of the inlet of front PV section
real ::  refl_outb                  !The reflectance of the outlet of the back PV Section
real ::  refl_outf                  !The reflectance of the outlet of front PV section
real ::  refl_pv                    !The reflectance of the PV cells
real ::  refl_wall                  !The reflectance of the back wall
real ::  refl_wall_lw               !The long wave refl. of the back wall
real ::  S(2*(N/12)+3)              !An array defining the radiation leaving each element for
                                    radiosity analysis
real ::  Sback                      !The amount of solar radiation absorbed in the PV back plate
real ::  Sblind                     !The elemental solar rad absorbed in blind
real ::  Scov                       !Solar radiation absorbed in each PV cover element
real ::  Sfin                       !The fin spacing
real ::  Sg1_w                      !The solar radiation absorbed in each elment of the single
                                    glazing in Vision section
real ::  Spv                        !Array defining the solar radiation + generated electricity in
                                    each PV element
real ::  Spv_h_tot                  !The total heat absorbed in the PV
real ::  Spv_heat(N/12)             !Array defining solar rad absorbed in each PV element
real ::  Sum                        !A dummy variable
real ::  sum1, sum2, sum3           !dummy variable
real ::  Sw2_1                      !The elemental solar rad absorbed in front glazing of W2
real ::  Sw2_2           !The elemental solar rad absorbed in back (room side) glazing of W2
real ::  Swall                      !The solar radiation absorbed in the back wall of PV section
real ::  t_b                        !The thickness of the blind
real ::  t_back                     !The thickness of the PV back plate
real ::  t_cov                      !The thickness of the PV covering
real ::  t_g1                       !The thickness of the front glazing
real ::  t_w2                       !thickness of each glazing of double glazed window
real ::  T1(N/12), T2(N/12), Ta(N/12)  !Arrays used to send on component elemental temps. to
                                    HTcoef subroutine
real ::  Tave                       !The approx. cavity wall temps in back PV section
real ::  Tchan_b_w                  !The approx. cavity wall temps. in front PV section
real ::  Tchan_f_w                  !The side wall temp. of front Vision cavity
real ::  Tchan_pv                   !The side wall temp. of the PV section
real ::  tfin                       !The approx. inlet temp. of inlet of back PV section
real ::  Tin_b_w                    !The approx. inlet temperature of inlet of front PV section
real ::  Tin_f_w                    !The inlet temperature of front Vision cavity
real ::  Tin_pv                     !The inlet temperature in the PV section
real ::  Tinlet                     !Inlet temperature
real ::  To                         !The approx. outlet temp. of outlet back PV section
real ::  Tout_b_w                   !The approx. outlet temp. of outlet front PV section
real ::  Tout_f_w                   !The outlet temp. of front Vision cavity
```

```
real ::    Tout_pv                    !The outlet temp. of the PV cavity
real ::    tpv                        !The thickness of the PV cells
real ::    trans_b                    !Transmittance of the blind
real ::    trans_back                 !Transmittance of the PV back plate
real :: trans_pv                      !Transmittance of the PV cells and air spaces combined
real ::    trans_cov                  !The transmittance of the PV cover
real ::    trans_g1                   !The transmittance of the front single glazing
real ::    trans_g2                   !Transmittance of glazing in the double glazed window (W2)
real ::    trans_wall                 !Transmittance of back wall in PV section
real ::    Troom                      !The room temperature
real ::    Upv(N/12)                  !Representative U-value of fins
real ::    Uwall                      !The back wall U-value
real ::    Vback                      !The air velocity in the back gap
real ::    Vfront                     !The air velocity in the front gap
real ::    Vo                         !The air velocity incident on the facade
real ::    Vpv                        !The velocity of the air flowing in the PV section
real ::    w                          !The width of the system

integer:: dn                          !Day number when test is being run (1 to 365)
integer::tod                          !Time of day in hours (1 to 24)

real:: Lat                            !Latitude of test site
real:: tilt                           !Tilt angle of the wall (90 is vertical)
real:: SA                             !Surface azimuth of the facade
real:: LNG                            !Longitude of the test site
real :: STM                           !Standard time meridian
real:: ekcov                          !extinction coefficient of PV cover
real:: ekback                         !extinction coefficient of PV backplate
real:: ekg2, ekg1                     !extinction coefficient of glazings g2 and g1

real ::    sbc
Parameter (sbc=5.67*1e-8)             !stefan boltzmann constant (W/m2K4)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Interface Declaration
!! (defining functions used in the program)
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

interface

Function Ch(Tma)                      ! specific heat of air
implicit none
Real, Intent(IN) :: Tma
Real :: Ch
End function Ch

Function density(Tma)                 ! air density
implicit none
Real, Intent(IN) :: Tma
Real :: density
End function density
end interface

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Main block
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

yes=1; no=0                         !Initialize, yes and no integers so they act as bolean type
N2=(N/12)                           !number of elements per surface
NoPV=N2*5                           !number of pv elements
Nw=N2*7                             !number of window elements
Eloss_pv=0                          !initialize
Eloss_w=0

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Model Inputs
!!
!INPUTS TO PROGRAM: The next section is where all the different parameters can be varied for the
!program.  The only other parameter that can be changed is the number of divisions that each component is
!divided into. This is changed in the top of the Model program.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

H_tot_pv=0.993                                  !height of PV Section (m)
H_tot_w=2.134-H_tot_pv                          !Height of vision section (m)
w=0.921                                         !width of AFW (m)
L_front=0.0455                                  !width of front cavity in Vision Section (m)
L_back=0.0455                                   !width of back cavity in meters in Vision Section (m)
Lpv=L_front+L_back                              !width of cavity in PV Section

tod=11                                          !time of day in hours
dn=26                                           !day number of the year
To=273.15  -17                                  !outdoor temperature  (K)
Troom=273.15 +0.4                               !room temperature (K)
Tinlet=273.15 -17                               !Inlet temperature into PV cavity

Vo=1                                            !Outside air velocity        (m/s)

!!!!
!!!Outdoor heat transfer calculation
!!!!

ho=8.6*(Vo**0.6/(h_tot_pv+h_tot_w)**0.4)        !outdoor HTC

if (ho<5.0) then
ho=5.0
end if

num=0;
L553:    Do i=1,N2                              !Loop to obtain average external glazing temper.
                    num=num+T(i)+T(NoPV+i)
               End Do L553

ho=ho + 0.9*sbc*4.*(0.5*(To+(num/(2*N2))))**3

ho=12                        !Input value to override calculate ho
print *,'ho=', ho


G=989                                           !solar radiation incident on the AFW (W/m^2)
Pdif=150/G                                      !Proportion of solar radiation that is diffuse
```

```
Vfront=0.6                                   !flow velocity in front cavity of Vision Section (m/s)
Vback=Vfront                                 !flow velocity in back cavity of Vision Section (m/s)

Vpv=(Vfront*L_front + Vback*L_back)/Lpv            !air velocity in PV Section (m/s), note, for
!conservation of mass to holed needs to be set to: Vpv=(Vfront*L_front + Vback*L_back)/Lpv

Acov=(2*36*0.10125**2)/(H_tot_pv*w)                !percent coverage of PV

print *,'Acov', Acov

tpv=0.0003                                   !PV thickness (m)
kpv=0.8                                      !PV thermal conductivity (W/mK)
Uwall=1                                      !Wall thermal conductance (W/Km^2)
hroom=6.5                                    !heat transfer coefficient to room (W/Km^2)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! HTcoef: convection heat transfer in cavity
!!
!! Front PV cavity

L101: Do i=1,N2
                T1(i)=T(i+2*N2)
                Ta(i)=T(i+3*N2)
                T2(i)=T(i+4*N2)
        End Do L101


Call HTcoef(Lpv,H_tot_pv,Vpv,G,T1,T2,Ta,N2,h1_pv,h2_pv)

L531:   Do i=1,N2
                        h1_pv(i)=2*h1_pv(i)
                        h2_pv(i)=2*h2_pv(i)
                End Do L531

!!Front Vision cavity

L103: Do i=1,N2
                T1(i)=T(i+NoPV)
                Ta(i)=T(i+N2+NoPV)
                T2(i)=T(i+2*N2+NoPV)
        End Do L103

Call HTcoef(L_front,H_tot_w,Vfront,G,T1,T2,Ta,N2,h1f_w,h2f_w)

!!Back Vision cavity

L104: Do i=1,N2
                T1(i)=T(i+3*N2+NoPV)
                Ta(i)=T(i+4*N2+NoPV)
                T2(i)=T(i+5*N2+NoPV)
        End Do L104

Call HTcoef(L_back,H_tot_w,Vback,G,T1,T2,Ta,N2,h1b_w,h2b_w)

!!Loop 683 to allow values of CHTC's to be input manually
```

```
L683:   Do i=1,N2
                    h1_pv(i)=h1_pv(i)
                    h2_pv(i)=h2_pv(i)
                    h1f_w(i)=h1f_w(i)
                    h2f_w(i)=h2f_w(i)
                    h1b_w(i)=h1b_w(i)
                    h2b_w(i)=h2b_w(i)
            End Do L683

!print *, 'h1_pv', h1_pv, 'h2_pv', h2_pv
!print *, 'h1f_w', h1f_w, 'h2f_w', h2f_w
!print *, 'h1b_w', h1b_w, 'h2b_w', h2b_w


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!!Loop 3 sets the efficiency of each PV element
!!
sum=0

L3:     Do i=1,N2
            If (T(N2+i)>298.15) then
                    npv(i)=0.0965 - 0.00038*(T(N2+i) - 298.15)  !factor in temperature dependence
            Else
!of the efficiency of the PV
                    npv(i)=0.0965
            End If
            sum=sum+npv(i)
        End Do L3

          print *,'Ave PV efficiency:', sum/N2
!!
!!



!Note the following 6 input temperatures used in the radiosity analysis even though, there is no concrete
!temperatures for the inlets and outlets. These temperatures are used to approximate the radiation heat
!transfer that occurs through the inlets and outlets.

Tin_pv=Tinlet                          !approx. inlet temperature for radiosity analysis in PV
Tout_pv=Tinlet                         !approx. outlet temperature for radiosity analysis in PV
Tout_f_w=(T(NoPV+N2) + T(NoPV+3*N2))/2.              !approx. inlet temperature for radiosity
analysis in front Vision
Tout_b_w=(T(NoPV+6*N2) + T(NoPV+4*N2))/2.   !approx. outlet temperature for radiosity analysis in
front Vision
Tin_b_w=(T(7*N2) + T(5*N2))/2.                  !approx. inlet temperature for radiosity analysis in
back Vision
Tin_f_w=(T(N2) + T(3*N2))/2.     !approx. outlet temperature for radiosity analysis in back Vision

kb=0.4                                               !blind thermal conductivity (W/mK)
t_b=0.002                                            !blind thickness (m)
eb=0.9                                               !emmissivity of blind
```

```
Lat=45                  !Latitude
LNG=74                  !Longitude
STM=75                  !local standard time meridian
SA=0                    !Surface azimuth (angle from south, east is neg)
tilt=90                 !tilt angle from horizontal

ekg1=6.96               !extinction coefficient of glass
ekg2=6.96               !                    "  "
ekcov=6.96
ekback=ekcov

t_g1=0.008                                  !exterior glazing thickness  (m)
t_w2=0.004                                  !thickness of glazings in W2
kg=0.8                                      !thermal conductivity of glass (W/mK)
t_cov=0.003                                 !PV cover thickness (m)
kcov=0.8                                    !PV cover thermal conductivity (W/mK)
t_back=0.003                                !PV backing thickness (m)
kback=0.8                                   !PV backing thermal conductivity (W/mK)


call envcond(Pdif,tod,dn,SA,Lat,LNG,STM,tilt,ekg1,ekg2,t_g1,t_w2,ekcov,t_cov, &
& ekback, t_back, trans_g1, trans_g2, refl_g1, refl_g2, trans_cov, trans_back, refl_cov, refl_back)

print *, 'trans_g1=', trans_g1                  !transmissivity of g1
print *, 'refl_g1=', refl_g1                    !reflectivity of g1
print *,'trans_cov=', trans_cov                 !transmissivity of PV cover
print *,'refl_cov=',refl_cov                    !reflectivity of PV cover
print *,'trans_back=', trans_back
print *,'refl_back=', refl_back
print *,'trans_g2=',trans_g2        !transmissivity of each glazing in the double glazed window
print *,'refl_g2=',refl_g2          !reflectivity of each glazing in the double glazed window

trans_wall=0

refl_wall=0.4

trans_pv=1-Acov                     !1 - % of area covered by pv
refl_pv=0.05*Acov/(H_tot_pv*w)      !reflectivity of PV cells (%area * refl)/Atot

eg1=0.9                             !emissivity of g1
ecov=0.9                            !emissivity of PV covering
eback=0.9                           !emissivity of PV backing
ewall=0.9                           !emissivity of back wall


low_e=yes                           !is there low_e coating on W2 (yes or no)
ew2=0.9                             !emissivity of W2 on cavity side

!emissivity of the inlets and outlets for radiosity analysis
ein_f=0.9999                                !inlet for front cavity
ein_b=0.9999                                !inlet for back cavity
eout_f=0.9999                               !outlet of front cavity
eout_b=0.9999                               !outlet of back cavity
echan_f=0.9999                              !two side walls in front channel
echan_b=0.9999                              !two side walls in back channel
```

```
!refl of a grey opaque surface (Saelens, 2002)
refl_inf=1-ein_f
refl_inb=1-ein_b
refl_outf=1-eout_f
refl_outb=1-eout_b
Refl_chanf=1-echan_f
refl_chanb=1-echan_b

!!
!!The following sets the blind optical properties
!!
!There are four different blinds that can be used in this model. They have the following properties:
!blind 1: absorptivity=0.5, reflectivity=0.3, transmissivity=0.2
!blind 2: absorptivity=0.3, reflectivity=0.5, transmissivity=0.2
!blind 3: absorptivity=0.7, reflectivity=0.2, transmissivity=0.1
!blind 4: absorptivity=0.55, reflectivity=0.4, transmissivity=0.05

Blind=1                                  !select which blind number to model (1 to 4)


!!!!!!!!!!!!!!!!!!!
!Input for FINS integrated to back plate
!!
fin=no                                   !are fins integrated into the PV back plate? yes or no
Lfin=0.015                               !fin length        (m)
tfin=0.002                               !fin thickness (m)
kfin=236                                 !fin conductivity (W/mK)


!END OF INPUTS TO PROGRAM

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!Parameter CALCULATIONS based on imputs
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!Following parameters are calculated based on inputed parameters
Etot_pv=G*w*H_tot_pv                     !Total incident solar radiation for PV section (W)
Etot_w=G*w*H_tot_w                       !Total incident solar radiation for vision section (W)
dH_pv=H_tot_pv/N2                        !height of each PV element        (m)
dH_w=H_tot_w/N2                          !height of each Vision element (m)
dA_pv=dH_pv*w                            !area of each PV element (m^2)
dA_w=dH_w*w                              !area of each Vision element (m^2)

If (blind==1)then

        abs_b=0.5
        refl_b=0.3
                Else If (blind==2)then
                        abs_b=0.3
                        refl_b=0.5
                Else If (Blind==3)then
                        abs_b=0.7
                        refl_b=0.2
```

```
                    Else If (Blind==4)then
                            abs_b=0.55
                            refl_b=0.4
                    Else
                            abs_b=0.5
                            refl_b=0.3
                            print *,'Improper blind selection, default (blind 1) used'
End If
trans_b=1-abs_b-refl_b

Sfin=3.*tfin                                                    !fin spacing (Rolle, 2000)
Lc=Lfin + tfin/2.                                               !corrected fin length
dAfin=2.*dH_pv*Lc                                               !fin area
Numf=w/(tfin+Sfin)                                              !number of fins

!Longwave reflectivity of surfaces for radiosity analysis
refl_g1_lw=1-eg1
refl_g2_lw=1-ew2
refl_b_lw=1-eb
refl_cov_lw=1-ecov
refl_back_lw=1-eback                                    !reflectivity of the PV Backing
        (Saelens, 2002)
refl_wall_lw=1-ewall                                    !reflectivity of the back wall


                    sum=0;num=0
L111:   Do i=1,N2
                            sum= sum+T(i+NoPV+5*N2)
                            sum= sum+T(i+NoPV+6*N2)
                    End Do L111

                    Tave=sum/(2.*N2)

if (low_e == yes) then
hw2=2+ (4*sbc*Tave**3)/(1/0.9 + 1/0.1 -1)

        else if (low_e ==no) then
        hw2=2+(4*sbc*Tave**3)/(1/0.9 + 1/0.9 -1)
        else
        hw2=2+ (4*sbc*Tave**3)/(1/0.9 + 1/0.1 -1)
        print *, 'low-e coating assumed for double-window'
end if


print *,'hw2', hw2

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!PV SECTION CALCULATIONS
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!The following two loops will determine what temperature should be used
!for the side walls of the channel in the radiation calculations
sum=0
L31: Do i=1,N2
```

```
            sum=sum+T(2*N2+i)+T(i+4*N2)
        End Do L31
Tchan_pv=sum/(2*N2)


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!The following is to determine the heat captured in each element of PV Section
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Ip1=G

k1=refl_back + (trans_back**2)*refl_wall*((1-refl_wall*refl_back)**(-1))
k2=refl_pv + (trans_pv**2) *k1*((1-refl_pv*k1)**(-1))

Ip2=(1-refl_cov*k2)**(-1) * trans_cov*Ip1
In2=Ip2*k2

In1=Ip1*refl_cov + In2*trans_cov
Ip3=(1-refl_pv*k1)**(-1) *trans_pv *Ip2
In3=k1*Ip3
Ip4=(1-refl_back*refl_wall)**(-1) *trans_back*Ip3
In4=Ip4*refl_wall
Ip5=Ip4*trans_wall
In5=0                           !assume no radiation coming from room, basis of above calculations

Scov  = dA_pv*((Ip1+In2) - (Ip2+In1))
Spv      = dA_pv*((Ip2+In3) - (Ip3+In2))
Sback  = dA_pv*((Ip3+In4) - (Ip4+In3))
Swall  = dA_pv*((Ip4+In5) - (Ip5+In4))

Erefl_pv  = In1*w*H_tot_pv          !Amount of energy reflected

Sum=0; Spv_h_tot=0
L4:     Do i=1,N2
            Spv_heat(i)= Spv - Ip1*(dA_pv)*npv(i)
            Sum=Sum+Ip1*(dA_pv)*npv(i)
            Spv_h_tot=Spv_h_tot+Spv_heat(i)
        End Do L4
        Epv=Sum

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!Print *,'Erefl_pv',Erefl_pv, 'Scov', Scov*N2, 'Spv_heat', Spv_h_tot,'Epv',Epv, 'Sback', Sback*N2, 'Swall',
Swall*N2,&
!& 'Etrans_pv', Ip5*w*H_tot_pv

!PV Check, the following is to verify above calculations (energy balance Input-Output=0)

Print *,'PV Check, zero=', Etot_pv-Erefl_pv-Scov*N2-Swall*N2-Sback*N2-Ip5*w*H_tot_pv-Spv_h_tot-
Epv

!Print *, 'reflected=', Erefl_pv/(dA_pv*N2), 'Apv=', dA_pv*N2
```

```
!!
!! Loop to determine heat sink parameters
!!
L777:   Do i=1,N2

                        mLc(i)=(2.*h1_pv(i)/(kfin*Lc*tfin))**0.5 * Lc**1.5
                        nfin(i)=tanh(mLc(i))/mlc(i)
!(Incropera, De Witt, 1990)

                        if (fin==yes)then
                        Upv(i)=(dAfin*numf*nfin(i) + (dA_pv - numf*dH_pv*tfin))*h1_pv(i)
                        else
                        Upv(i)=dA_pv*h1_pv(i)
                        End if
                End Do L777

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following section sets the view factors for the PV section. Note that the program
!!viewfac1pv_2w is does not produce outputs directly for Ff_pv and therefore we call
!!Ff and then input the value manually into Ff_pv
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Call viewfact(Lpv,H_tot_pv,w,N2,Ff)

Ff_pv=0

L55:    Do i=1,N2
L56:            Do j=1,N2
                                Ff_pv(2*N2+i, 4*N2+j)=Ff(i,j+N2)
                                Ff_pv(4*N2+j,2*N2+i)=Ff(j+N2,i)
                        End Do L56

L57:            Do j=1,3
                                Ff_pv(2*N2+i, NoPV+j)=Ff(i,2*N2+j)
                                Ff_pv(4*N2+i, NoPV+j)=Ff(i+N2,2*N2+j)
                                Ff_pv(NoPV+j, 2*N2+i)=Ff(2*N2+j,i)
                                Ff_pv(NoPV+j, 4*N2+i)=Ff(2*N2+j,i+N2)
                        End Do L57
                End Do L55

L58:    Do i=1,3
L59:            Do j=1,3
                                Ff_pv(NoPV+i, NoPV+j)=Ff(2*N2+i, 2*N2+j)
                        End Do L59
                End Do L58

!!
!! Loops to determine the radiosity matrix A for the PV cavity
!!

A=0;S=0
L28: Do i=1,N2
L29:    Do j=1,N2
                        A(i,j+N2)=-refl_back_lw*Ff_pv(i+2*N2,j+4*N2)
```

```
                    A(i+N2,j)=-refl_wall_lw*Ff_pv(i+4*N2,j+2*N2)
                    if (j==i) then
                    A(i,j)=1
                    A(i+N2,j+N2)=1
                    End if
                End Do L29
        End Do L28

L30: Do i=1,N2
                S(i)=sbc*eback*T(i+2*N2)**4
                S(i+N2)=sbc*ewall*T(i+4*N2)**4
        End Do L30

L36: Do i=1,N2
L37:    Do j=2*N2+1,2*N2+3
                    A(i,j)=     -refl_back_lw*Ff_pv(2*N2+i,NoPV+j-2*N2)
                    A(i+N2,j)= -refl_wall_lw*Ff_pv(4*N2+i,NoPV+j-2*N2)
                End Do L37
        End Do L36

L38: Do j=1,N2
                A(2*N2+1,j)= -refl_inb*Ff_pv(NoPV+1,j+2*N2)
                A(2*N2+2,j)= -refl_outb*Ff_pv(NoPV+2,j+2*N2)
                A(2*N2+3,j)= -refl_chanb*Ff_pv(NoPV+3,j+2*N2)

                A(2*N2+1,j+N2)= -refl_inb*Ff_pv(NoPV+1,j+4*N2)
                A(2*N2+2,j+N2)= -refl_outb*Ff_pv(NoPV+2,j+4*N2)
                A(2*N2+3,j+N2)= -refl_chanb*Ff_pv(NoPV+3,j+4*N2)
        End Do L38

L39: Do i=1,3
                A(2*N2+1,2*N2+i)=-refl_inb*Ff_pv(NoPV+1,NoPV+i)
                A(2*N2+2,2*N2+i)=-refl_outb*Ff_pv(NoPV+2,NoPV+i)
                A(2*N2+3,2*N2+i)=-refl_chanb*Ff_pv(NoPV+3,NoPV+i)
        End Do L39

        A(2*N2+1,2*N2+1)=A(2*N2+1,2*N2+1)+1
        A(2*N2+2,2*N2+2)=A(2*N2+2,2*N2+2)+1
        A(2*N2+3,2*N2+3)=A(2*N2+3,2*N2+3)+1

        S(2*N2+1)=sbc*ein_f*Tin_pv**4
        S(2*N2+2)=sbc*eout_f*Tout_pv**4
        S(2*N2+3)=sbc*echan_f*Tchan_pv**4

Q=S
P=0
indx_pv=0
i=2*N2+3
Call ludcmp(A, i, i, indx_pv,P)
Call lubksb(A, i, i, indx_pv,Q)

Eloss_pv=Eloss_pv-((Q(2*N2+1)*(refl_inf-1)+S(2*N2+1))/refl_inf)*w*Lpv
Eloss_pv=Eloss_pv-((Q(2*N2+2)*(refl_outf-1)+S(2*N2+2))/refl_outf)*w*Lpv
Eloss_pv=Eloss_pv-((Q(2*N2+3)*(refl_chanf-1)+S(2*N2+3))/refl_chanf)*2*H_tot_pv*Lpv

Sum=0
```

```
!!
!!Energy balance equation definition for PV Cover
!!


f(1)= Scov - dA_pv*ho*(T(1)- To)   &
          & - (kcov*w*t_cov/dH_pv)*(T(1)- T(2)) - &
               & (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(1)- T(N2+1))

Sum=Sum+dA_pv*ho*(T(1)- To)

f(N2)= Scov - dA_pv*ho*(T(N2)- To)  &
          & - (kcov*w*t_cov/dH_pv)*(T(N2)- T(N2-1)) - &
               & (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(N2)- T(2*N2))

Sum=Sum+dA_pv*ho*(T(N2)- To)

L12: Do i=2, N2-1
               f(i)=Scov - dA_pv*ho*(T(i)- To)  &
                    & - (kcov*w*t_cov/dH_pv)*(2*T(i)- T(i-1)-T(i+1)) - &
                    & (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(i)- T(i+N2))
               Sum=Sum+dA_pv*ho*(T(i)- To)
     End Do L12

Qout_pv=Sum

!!
!!Energy balance equation definition for PV cells
!!

f(N2+1)= Spv_heat(1)- (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(N2+1)- T(1))&
               & -(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(N2+1)-T(2*N2+1)) -
&
               & (kpv*w*tpv/dH_pv)*(T(N2+1)- T(N2+2))
f(2*N2)=Spv_heat(N2)- (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(2*N2)- T(N2))&
               & -(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(2*N2)-T(3*N2)) - &
               & (kpv*w*tpv/dH_pv)*(T(2*N2)- T(2*N2-1))

L14:   Do i=N2+2, 2*N2-1
     f(i)=Spv_heat(i-N2)- (0.5*t_cov/(kcov*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(i)- T(i-N2))&
               & -(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(i)-T(i+N2)) - &
               & (kpv*w*tpv/dH_pv)*(2*T(i)- T(i-1) - T(i+1))

     End Do L14

!!
!!Energy balance equation definition for PV backplate
!!

f(2*N2+1)= Sback-(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(2*N2+1)-T(N2+1))- &
               & ((Q(1)*(refl_back_lw-1)+S(1))/refl_back_lw)*dA_pv  -
(kback*w*t_back/dH_pv)*(T(2*N2+1)- T(2*N2+2)) - &
               & Upv(1)*(T(2*N2+1)-T(3*N2+1))

f(3*N2)=Sback-(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(3*N2)-T(2*N2))- &
```

```
                & ((Q(N2)*(refl_back_lw-1)+S(N2))/refl_back_lw)*dA_pv -
(kback*w*t_back/dH_pv)*(T(3*N2)- T(3*N2-1)) - &
                & Upv(N2)*(T(3*N2)-T(4*N2))

L17: Do i=2*N2+2,3*N2-1
        f(i)=Sback-(0.5*t_back/(kback*dA_pv) + 0.5*tpv/(kpv*dA_pv))**(-1)*(T(i)-T(i-N2))- &
                & ((Q(i-2*N2)*(refl_back_lw-1)+S(i-2*N2))/refl_back_lw)*dA_pv - &
                & (kback*w*t_back/dH_pv)*(2*T(i)- T(i-1)-T(i+1)) - &
                & Upv(i-2*N2)*(T(i)-T(i+N2))
   End Do L17
sum=0

!!
!!Energy balance equation definition for air cavity in PV section
!!

f(3*N2+1)=Vpv*Lpv*w*density(T(3*N2+1))*Ch(T(3*N2+1))*(T(3*N2+1)-Tinlet)- &
        !Includes BC
&               Upv(1)*(T(2*N2+1)-T(3*N2+1)) - h2_pv(1)*dA_pv*(T(4*N2+1)-T(3*N2+1))

Sum=Sum+Upv(1)*(T(2*N2+1)-T(3*N2+1)) + h2_pv(1)*dA_pv*(T(4*N2+1)-T(3*N2+1))

f(4*N2)=Vpv*Lpv*w*density(T(4*N2))*Ch(T(4*N2))*(T(4*N2)-T(4*N2-1)) - &
        !Includes BC
&         Upv(N2)*(T(3*N2)-T(4*N2)) - h2_pv(N2)*dA_pv*(T(5*N2)-T(4*N2))

Sum=Sum+Upv(N2)*(T(3*N2)-T(4*N2)) + h2_pv(N2)*dA_pv*(T(5*N2)-T(4*N2))

L19:Do i=3*N2+2, 4*N2-1
          f(i)=Vpv*Lpv*w*density(T(i))*Ch(T(i))*(T(i)-T(i-1)) - &
          &     Upv(i-3*N2)*(T(i-N2)-T(i)) - h2_pv(i-3*N2)*dA_pv*(T(i+N2)-T(i))
        Sum=Sum+Upv(i-3*N2)*(T(i-N2)-T(i)) + h2_pv(i-3*N2)*dA_pv*(T(i+N2)-T(i))
        End Do L19

Qair_pv=Vpv*Lpv*w*density(T(4*N2-N2/2))*Ch(T(4*N2-N2/2))*(T(4*N2)-Tinlet)

!Note the following is for verification purposes.  The energy transfered to the air by
!enthalpy calculations, should be the same as for convection heat transfer calculations

Print *,'Qair_pv-enth',Qair_pv
Qair_pv=Sum
Print *,'Qair_pv-conv',Qair_pv

SUm=0

!!
!!Energy balance equation definition for back wall in PV section
!!

L20:    Do i=4*N2+1, 5*N2                    !assume no conduction up or down the elements

f(i)=Swall-dA_pv*((1/Uwall + 1/hroom)**(-1))*(T(i)-Troom)- ((Q(i-3*N2)*(refl_wall_lw-1)+S(i-
3*N2))/refl_wall_lw)*dA_pv &
& - dA_pv*h2_pv(i-4*N2)*(T(i)-T(i-N2))
                Sum=Sum+dA_pv*((1/Uwall + 1/hroom)**(-1))*(T(i)-Troom)
                End Do L20
```

```
Qroom_pv=Sum

Ebal_pv=Etot_pv-Erefl_pv-Epv-Qroom_pv-Qair_pv-Qout_pv-Eloss_pv

Eff_pv=(Epv+Qroom_pv+Qair_pv)/Etot_pv

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!PV Section Printing
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Print *,'PV Section Energy Balance:',Ebal_pv
Print *,'PV Section Efficiency:', Eff_pv
Print *,'Etot_pv', Etot_pv, 'Erefl_pv', Erefl_pv, 'Epv', Epv, 'Qroom_pv', Qroom_pv
Print *,'Qair_pv', Qair_pv, 'Qout_pv', Qout_pv, 'Eloss_pv', Eloss_pv

!!
!!Loop to print the temperature of the PV backplate
!!

!Do i=1,N2
!print *, 'Height', dH_PV*(i-0.5),'Tpv_back', T(2*N2+i)-273.15
!End Do

num=T(2*N2+1); j=1; sum1=0; sum2=0; sum=0 ;sum3=0
print *,'PV Section Exit Temperature - Inlet Temperature',T(4*N2)-Tinlet
L163:   Do i=1,N2
                sum1=sum1+ T(i)
                sum2=sum2+ T(N2+i)
                sum=sum+      T(2*N2+i)
                sum3=sum3+T(4*N2+i)
                if (T(2*N2+i)>num) then
                j=i
                num=T(2*N2+i)
                end if


            End Do L163
Print *,'Maximum PV temperature = ', num-273.15, ' and occurs at height of:',(H_tot_pv/N2)*(j-0.5)

Print *,'Tpv_ave=', (sum1/N2 + sum2/N2+ sum/N2)/3 -273.15, 'Twall', sum3/N2 -273.15


Print *,'Tpv,bot',T(2*N2+1)-273.15,'Tpv,mid', T(2.5*N2)-273.15,'Tpv,top',T(3*N2)-273.15

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!CALCULATIONS FOR VISION SECTION
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!The following section sets the view factors for the Vision section. Note that the program
!!viewfac1pv_2w is does not produce outputs directly for Ff_w and therefore we call
```

```
!!Ff and then input the value "manually" into Ff_w
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!
!!Front cavity
!!
call viewfact(L_front,H_tot_w,w,N2,Ff)
Ff_w=0
L260:   Do i=1,N2
L261:           Do j=1,N2
                        Ff_w(i,j+2*N2)=Ff(i,j+N2)
                        Ff_w(j+2*N2,i)=Ff(j+N2,i)
                        End Do L261

L262:           Do j=1,3
                        Ff_w(i,Nw+j)=Ff(i,2*N2+j)
                        Ff_w(Nw+j,i)=Ff(2*N2+j,i)
                        Ff_w(i+2*N2,Nw+j)=Ff(N2+i,2*N2+j)
                        Ff_w(Nw+j,i+2*N2)=Ff(2*N2+j,N2+i)
                        End Do L262
                End Do L260

L263:   Do i=1,3
L264:           Do j=1,3
                        Ff_w(Nw+i,Nw+j)=Ff(2*N2+i,2*N2+j)
                        End Do L264
                End Do L263

!!
!!Back cavity
!!

call viewfact(L_back,H_tot_w,w,N2,Ff)

L265:   Do i=1,N2
L266:           Do j=1,N2
                        Ff_w(i+3*N2,j+5*N2)=Ff(i,j+N2)
                        Ff_w(j+5*N2,i+3*N2)=Ff(j+N2,i)
                        End Do L266

L267:           Do j=1,3
                        Ff_w(i+3*N2,Nw+j+3)=Ff(i,2*N2+j)
                        Ff_w(Nw+j+3,i+3*N2)=Ff(2*N2+j,i)
                        Ff_w(i+5*N2,Nw+j+3)=Ff(N2+i,2*N2+j)
                        Ff_w(Nw+j+3,i+5*N2)=Ff(2*N2+j,N2+i)
                        End Do L267
                End Do L265

L268:   Do i=1,3
L269:           Do j=1,3
                        Ff_w(Nw+i+3,Nw+j+3)=Ff(2*N2+i,2*N2+j)
                        End Do L269
                End Do L268

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!
!The following two loops will determine what temperature should be used
!for the side walls of the channel in the radiation calculations
!
sum=0
L200: Do i=NoPV+1,NoPV+N2
            sum=sum+T(i)+T(i+2*N2)
      End Do L200
Tchan_f_w=sum/(2.*N2)

sum=0
L201: Do i=NoPV+3*N2+1,NoPV+4*N2
            sum=sum+T(i)+T(i+2*N2)
      End Do L201
Tchan_b_w=sum/(2.*N2)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!The following is to determine the heat captured in each element of Vision Section
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Ip1=G
In5=0            !assuming neglibible radiation is leaving from room

!!
!!The k values are simplifications made when solving the system energy balance for Ip, In
!!
k1=(1- refl_b*refl_g2*(1 + trans_g2*trans_g1*(1-refl_g2**2)**(-1)))** (-1) * trans_b
k2=(k1/trans_b) * (trans_g2**2 *refl_b*(1+refl_g2**2))
k3=(1-refl_g2**2)**(-1) * trans_g1
k4=(k3/trans_g1)*trans_g2*refl_g2
k5=refl_b + trans_b*(refl_g2*k1 + trans_g2*refl_g2*k1*k3)
k6=trans_b*(refl_g2*k2 + trans_g2*(refl_g2*(k2*k3+k4) + trans_g2))

Ip2=(1-refl_g1*k5)**(-1) * (trans_g1*Ip1 + refl_g1*k6*In5)
In2=k5*Ip2 + k6*In5
Ip3=k1*Ip2+k2*In5
Ip4=k3*Ip3+k4*In5
Ip5=Ip4*trans_g2 + In5*refl_g2
In1=Ip1*refl_g1 + In2*trans_g1
In4=Ip4*refl_g2 + In5*trans_g2
In3=Ip3*refl_g2 + In4*trans_g2

Etrans=H_tot_w*w*Ip5              !Amount of transmitted solar radiation
Erefl_w= H_tot_w*w*In1            !Amount of energy reflected

Sg1_w= dA_w*((Ip1+In2)-(Ip2+In1))
Sblind= dA_w*((Ip2+In3)-(Ip3+In2))
Sw2_1= dA_w*((Ip3+In4)-(Ip4+In3))
Sw2_2= dA_w*((Ip4+In5)-(Ip5+In4))

!Print *,'Sg1_w', Sg1_w*N2, 'Sblind', Sblind*N2, 'Sw2_1', Sw2_1*N2, 'Sw2_2', Sw2_2*N2
Check_1w= Etot_w -(Sg1_w+Sblind+Sw2_1+Sw2_2)*N2 -Etrans -Erefl_w
!Print *,'Check_1w', 'zero=', Check_1w
```

```
!!
!!The following loops calculate the radiosity matrix A for front Vision section
!!

A=0;S=0

L202: Do i=1,N2
L203:    Do j=1,N2
                        A(i,j+N2)= -refl_g1_lw*Ff_w(i,2*N2+j)
                        A(i+N2,j)= -refl_b_lw*Ff_w(2*N2+i,j)
                        if (j==i) then
                        A(i,j)=1
                        A(i+N2,j+N2)=1
                        End if
                End Do L203
        End Do L202

L204: Do i=1,N2
L205:    Do j=2*N2+1,2*N2+3
                        A(i,j)=     -refl_g1_lw*Ff_w(i,Nw+j-2*N2)
                        A(i+N2,j)= -refl_b_lw*Ff_w(2*N2+i,Nw+j-2*N2)
                End Do L205
        End Do L204

L206: Do j=1,N2
                A(2*N2+1,j)= -refl_inf*Ff_w(Nw+1,j)
                A(2*N2+2,j)= -refl_outf*Ff_w(Nw+2,j)
                A(2*N2+3,j)= -refl_chanf*Ff_w(Nw+3,j)

                A(2*N2+1,j+N2)= -refl_inf*Ff_w(Nw+1,j+2*N2)
                A(2*N2+2,j+N2)= -refl_outf*Ff_w(Nw+2,j+2*N2)
                A(2*N2+3,j+N2)= -refl_chanf*Ff_w(Nw+3,j+2*N2)
        End Do L206

L207: Do i=1,3
                A(2*N2+1,2*N2+i)=-refl_inf*Ff_w(Nw+1,Nw+i)
                A(2*N2+2,2*N2+i)=-refl_outf*Ff_w(Nw+2,Nw+i)
                A(2*N2+3,2*N2+i)=-refl_chanf*Ff_w(Nw+3,Nw+i)
        End Do L207

        A(2*N2+1,2*N2+1)=A(2*N2+1,2*N2+1)+1
        A(2*N2+2,2*N2+2)=A(2*N2+2,2*N2+2)+1
        A(2*N2+3,2*N2+3)=A(2*N2+3,2*N2+3)+1

L208: Do i=1,N2
                S(i)=     sbc*eg1*T(NoPV+i)**4
                S(i+N2)= sbc*eb*T(NoPV+2*N2+i)**4
        End Do L208

        S(2*N2+1)=sbc*ein_f*Tin_f_w**4
        S(2*N2+2)=sbc*eout_f*Tout_f_w**4
        S(2*N2+3)=sbc*echan_f*Tchan_f_w**4

Q=S
P=0
indx_w=0
```

```
i=2*N2+3
Call ludcmp(A, i, i, indx_w,P)
Call lubksb(A, i, i, indx_w,Q)


Eloss_w=Eloss_w-((Q(2*N2+1)*(refl_inf-1)+S(2*N2+1))/refl_inf)*w*L_front
Eloss_w=Eloss_w-((Q(2*N2+2)*(refl_outf-1)+S(2*N2+2))/refl_outf)*w*L_front
Eloss_w=Eloss_w-((Q(2*N2+3)*(refl_chanf-1)+S(2*N2+3))/refl_chanf)*2*H_tot_w*L_front

!!
!!Energy balance equations for the First Glazing in Vision Section
!!

Sum=0
f(NoPV+1)= Sg1_w - ho*dA_w*(T(NoPV+1)-To) - h1f_w(1)*dA_w*(T(NoPV+1)-T(NoPV+N2+1)) &
& -(kg*t_g1*w/dH_w)*(T(NoPV+1)-T(NoPV+2))- ((Q(1)*(refl_g1_lw-1)+S(1))/refl_g1_lw)*dA_w

Sum=Sum+ho*dA_w*(T(NoPV+1)-To)

f(NoPV+N2)=Sg1_w - ho*dA_w*(T(NoPV+N2)-To) - h1f_w(N2)*dA_w*(T(NoPV+N2)-
T(NoPV+2*N2)) &
& -(kg*t_g1*w/dH_w)*(T(NoPV+N2)-T(NoPV+N2-1)) -((Q(N2)*(refl_g1_lw-
1)+S(N2))/refl_g1_lw)*dA_w

Sum=Sum+ho*dA_w*(T(NoPV+N2)-To)


L209: Do i= NoPV+2,NoPV+N2-1

   f(i)=Sg1_w - ho*dA_w*(T(i)-To) - h1f_w(i-NoPV)*dA_w*(T(i)-T(i+N2))- &
         & ((Q(i-NoPV)*(refl_g1_lw-1)+S(i-NoPV))/refl_g1_lw)*dA_w - &
         & (kg*t_g1*w/dH_w)*(2*T(i)-T(i-1)-T(i+1))

         Sum=Sum+ho*dA_w*(T(i)-To)

         End Do L209

Qout_w=Sum

Sum=0

!!
!!Front cavity in vision section

!!
f(NoPV+N2+1)=Vfront*L_front*w*density(T(NoPV+N2+1))*Ch(T(NoPV+N2+1))*(T(NoPV+N2+1)-
T(4*N2)) - &      !Includes BC
& h1f_w(1)*dA_w*(T(NoPV+1)-T(NoPV+N2+1)) - h2f_w(1)*dA_w*(T(NoPV+2*N2+1)-
T(NoPV+N2+1))

Sum=Sum+h1f_w(1)*dA_w*(T(NoPV+1)-T(NoPV+N2+1)) + h2f_w(1)*dA_w*(T(NoPV+2*N2+1)-
T(NoPV+N2+1))

f(NoPV+2*N2)=Vfront*L_front*w*density(T(NoPV+2*N2))*Ch(T(NoPV+2*N2))*(T(NoPV+2*N2)-
T(NoPV+2*N2-1)) & !Includes BC
```

```
& - h1f_w(N2)*dA_w*(T(NoPV+N2)-T(NoPV+2*N2)) - h2f_w(N2)*dA_w*(T(NoPV+3*N2)-
T(NoPV+2*N2))

Sum=Sum+h1f_w(N2)*dA_w*(T(NoPV+N2)-T(NoPV+2*N2)) + h2f_w(N2)*dA_w*(T(NoPV+3*N2)-
T(NoPV+2*N2))

L210:   Do i=NoPV+N2+2, NoPV+2*N2-1
          f(i)=Vfront*L_front*w*density(T(i))*Ch(T(i))*(T(i)-T(i-1))- &
        &                h1f_w(i-N2-NoPV)*dA_w*(T(i-N2)-T(i)) - h2f_w(i-NoPV-
N2)*dA_w*(T(i+N2)-T(i))

        Sum=Sum+h1f_w(i-N2-NoPV)*dA_w*(T(i-N2)-T(i)) + h2f_w(i-N2-NoPV)*dA_w*(T(i+N2)-
T(i))
        End Do L210

!!Energy balance verification, Qairf_w-enthalpy has to equal Qairf_w-conv
Qairf_w=Vfront*L_front*w*density(T(NoPV+(2*N2-N2/2)))*Ch(T(NoPV+(2*N2-
N2/2)))*(T(NoPV+2*N2)-T(4*N2))
!Print *,'Qairf_w-enth',Qairf_w
Qairf_w=Sum
!Print *,'Qairf_w-conv',Qairf_w

sum2=0
!!
!!Front of blind
!!
f(NoPV+2*N2+1)= Sblind/2. - dA_w*h2f_w(1)*(T(NoPV+2*N2+1)- T(NoPV+N2+1)) - &
& ((Q(1+N2)*(refl_b_lw-1)+S(1+N2))/refl_b_lw)*dA_w  - &
& (kb*dA_w/t_b)*(T(NoPV+2*N2+1)-T(NoPV+3*N2+1)) - &
& (kb*w*t_b/dH_w)*(T(NoPV+2*N2+1)- T(NoPV+2*N2+2))

sum2=((Q(1+N2)*(refl_b_lw-1)+S(1+N2))/refl_b_lw)*dA_w

f(NoPV+3*N2)= Sblind/2. - dA_w*h2f_w(N2)*(T(NoPV+3*N2)- T(NoPV+2*N2)) - &
& ((Q(2*N2)*(refl_b_lw-1)+S(2*N2))/refl_b_lw)*dA_w  - &
& (kb*dA_w/t_b)*(T(NoPV+3*N2)-T(NoPV+4*N2)) - &
& (kb*w*t_b/dH_w)*(T(NoPV+3*N2)- T(NoPV+3*N2-1))

sum2=sum2+((Q(2*N2)*(refl_b_lw-1)+S(2*N2))/refl_b_lw)*dA_w

L172: Do i=NoPV+2*N2+2, NoPV+3*N2-1
        f(i)=Sblind/2. - dA_w*h2f_w(i-NoPV-2*N2)*(T(i)- T(i-N2)) - &
                & ((Q(i-N2-NoPV)*(refl_b_lw-1)+S(i-N2-NoPV))/refl_b_lw)*dA_w - &
                & (kb*dA_w/t_b)*(T(i)-T(i+N2))- &
                & (kb*w*t_b/dH_w)*(2*T(i)- T(i-1)-T(i+1))

        sum2=sum2+((Q(2*N2)*(refl_b_lw-1)+S(2*N2))/refl_b_lw)*dA_w
        End Do L172
!Erad_f=sum2

A=0;S=0

!!
!!The following loops calculate the radiosity matrix for the back vision section
!!
L211: Do i=1,N2
```

```
L212:   Do j=1,N2
                    A(i,j+N2)= -refl_b_lw*Ff_w(i+3*N2,5*N2+j)
                    A(i+N2,j)= -refl_g2_lw*Ff_w(5*N2+i,j+3*N2)
                    if (j==i) then
                    A(i,j)=1
                    A(i+N2,j+N2)=1
                    End if
            End Do L212
        End Do L211

L213: Do i=1,N2
L214:   Do j=2*N2+1,2*N2+3
                    A(i,j)= -refl_b_lw*Ff_w(i+3*N2,Nw+j-2*N2+3)
                    A(i+N2,j)= -refl_g2_lw*Ff_w(5*N2+i,Nw+j-2*N2+3)
            End Do L214
        End Do L213

L215: Do j=1,N2
            A(2*N2+1,j)= -refl_inb*Ff_w(Nw+4,j+3*N2)
            A(2*N2+2,j)= -refl_outb*Ff_w(Nw+5,j+3*N2)
            A(2*N2+3,j)= -refl_chanb*Ff_w(Nw+6,j+3*N2)

            A(2*N2+1,j+N2)= -refl_inb*Ff_w(Nw+4,j+5*N2)
            A(2*N2+2,j+N2)= -refl_outb*Ff_w(Nw+5,j+5*N2)
            A(2*N2+3,j+N2)= -refl_chanb*Ff_w(Nw+6,j+5*N2)
        End Do L215

L216: Do i=1,3
            A(2*N2+1,2*N2+i)=-refl_inb*Ff_w(Nw+4,Nw+i+3)
            A(2*N2+2,2*N2+i)=-refl_outb*Ff_w(Nw+5,Nw+i+3)
            A(2*N2+3,2*N2+i)=-refl_chanb*Ff_w(Nw+6,Nw+i+3)
        End Do L216

        A(2*N2+1,2*N2+1)=A(2*N2+1,2*N2+1)+1
        A(2*N2+2,2*N2+2)=A(2*N2+2,2*N2+2)+1
        A(2*N2+3,2*N2+3)=A(2*N2+3,2*N2+3)+1

L217: Do i=1,N2
            S(i)=     sbc*eb*T(NoPV+3*N2+i)**4
            S(i+N2)= sbc*ew2*T(NoPV+5*N2+i)**4
        End Do L217

        S(2*N2+1)=sbc*ein_b*Tin_b_w**4
        S(2*N2+2)=sbc*eout_b*Tout_b_w**4
        S(2*N2+3)=sbc*echan_b*Tchan_b_w**4

Q=S

P=0
indx_w2=0
i=2*N2+3
Call ludcmp(A, i, i, indx_w2,P)
Call lubksb(A, i, i, indx_w2,Q)

Eloss_w=Eloss_w-((Q(2*N2+1)*(refl_inb-1)+S(2*N2+1))/refl_inb)*w*L_back
Eloss_w=Eloss_w-((Q(2*N2+2)*(refl_outb-1)+S(2*N2+2))/refl_outb)*w*L_back
```

```
Eloss_w=Eloss_w-((Q(2*N2+3)*(refl_chanb-1)+S(2*N2+3))/refl_chanb)*2*H_tot_w*L_back

sum2=0
!!
!!Back of blind
!!
f(NoPV+3*N2+1)= Sblind/2. - (kb*dA_w/t_b)*(T(NoPV+3*N2+1)-T(NoPV+2*N2+1)) - &
& ((Q(1)*(refl_b_lw-1)+S(1))/refl_b_lw)*dA_w - &
& (kb*w*t_b/dH_w)*(T(NoPV+3*N2+1)- T(NoPV+3*N2+2)) - &
& h1b_w(1)*(T(NoPV+3*N2+1) - T(NoPV+4*N2+1))*dA_w

sum2=((Q(1)*(refl_b_lw-1)+S(1))/refl_b_lw)*dA_w

f(NoPV+4*N2)= Sblind/2. -(kb*dA_w/t_b)*(T(NoPV+4*N2)-T(NoPV+3*N2)) - &
& ((Q(N2)*(refl_b_lw-1)+S(N2))/refl_b_lw)*dA_w - &
& (kb*w*t_b/dH_w)*(T(NoPV+4*N2)- T(NoPV+4*N2-1)) - &
& h1b_w(N2)*(T(NoPV+4*N2) - T(NoPV+5*N2))*dA_w

sum2=sum2+((Q(N2)*(refl_b_lw-1)+S(N2))/refl_b_lw)*dA_w

L218: Do i=NoPV+3*N2+2, NoPV+4*N2-1
        f(i)=Sblind/2. - (kb*dA_w/t_b)*(T(i)-T(i-N2)) - &
                & ((Q(i-3*N2-NoPV)*(refl_b_lw-1)+S(i-3*N2-NoPV))/refl_b_lw)*dA_w - &
                & (kb*w*t_b/dH_w)*(2*T(i)- T(i-1)-T(i+1)) - &
                & dA_w*h1b_w(i-NoPV-3*N2)*(T(i)-T(i+N2))

        sum2=sum2+((Q(i-3*N2-NoPV)*(refl_b_lw-1)+S(i-3*N2-NoPV))/refl_b_lw)*dA_w
       End Do L218
!Erad_b=sum2
sum=0
!!
!!Back air cavity

!!
f(NoPV+4*N2+1)=Vback*L_back*w*density(T(NoPV+4*N2+1))*Ch(T(NoPV+4*N2+1))*(T(NoPV+4*
N2+1)-T(4*N2))- &        !Includes BC
&  h1b_w(1)*dA_w*(T(NoPV+3*N2+1)-T(NoPV+4*N2+1))- h2b_w(1)*dA_w*(T(NoPV+5*N2+1)-
T(NoPV+4*N2+1))

Sum=Sum+h1b_w(1)*dA_w*(T(NoPV+3*N2+1)-T(NoPV+4*N2+1))+
h2b_w(1)*dA_w*(T(NoPV+5*N2+1)-T(NoPV+4*N2+1))

f(NoPV+5*N2)=Vback*L_back*w*density(T(NoPV+5*N2))*Ch(T(NoPV+5*N2))*(T(NoPV+5*N2)-
T(NoPV+5*N2-1)) - &     !Includes BC
& h1b_w(N2)*dA_w*(T(NoPV+4*N2)-T(NoPV+5*N2))- h2b_w(N2)*dA_w*(T(NoPV+6*N2)-
T(NoPV+5*N2))

Sum=Sum+h1b_w(N2)*dA_w*(T(NoPV+4*N2)-T(NoPV+5*N2))+ h2b_w(N2)*dA_w*(T(NoPV+6*N2)-
T(NoPV+5*N2))

L219:Do i=NoPV+4*N2+2, NoPV+5*N2-1
          f(i)=Vback*L_back*w*density(T(i))*Ch(T(i))*(T(i)-T(i-1)) - &
          &       h1b_w(i-4*N2-NoPV)*dA_w*(T(i-N2)-T(i)) - h2b_w(i-NoPV-4*N2)*dA_w*(T(i+N2)-
T(i))
        Sum=Sum+h1b_w(i-4*N2-NoPV)*dA_w*(T(i-N2)-T(i)) + h2b_w(i-NoPV-
4*N2)*dA_w*(T(i+N2)-T(i))
```

```
        End Do L219

Qairb_w=Vback*L_back*w*density(T(NoPV+(5*N2-N2/2)))*Ch(T(NoPV+(5*N2-
N2/2)))*(T(NoPV+5*N2)-T(4*N2))

!Energy balance verification, Qair by enthalpy has to equal Qair by convective heat transfer
!Print *,'Qairb_w-enth',Qairb_w
Qairb_w=Sum
!Print *,'Qairb_w-conv',Qairb_w

!!
!!First Glazing of W2
!!
f(NoPV+5*N2+1)= Sw2_1- dA_w*h2b_w(1)*(T(NoPV+5*N2+1)- T(NoPV+4*N2+1)) - &
                &       (kg*w*t_w2/dH_w)*(T(NoPV+5*N2+1)-T(NoPV+5*N2+2)) - &
                &       hw2*dA_w*(T(NoPV+5*N2+1)-T(NoPV+6*N2+1)) - &
                &       ((Q(1+N2)*(refl_g2_lw-1)+S(1+N2))/refl_g2_lw)*dA_w

f(NoPV+6*N2)=Sw2_1- dA_w*h2b_w(N2)*(T(NoPV+6*N2)- T(NoPV+5*N2)) - &
                &       (kg*w*t_w2/dH_w)*(T(NoPV+6*N2)-T(NoPV+6*N2-1)) - &
                &       hw2*dA_w*(T(NoPV+6*N2)-T(NoPV+7*N2)) - &
                &       ((Q(2*N2)*(refl_g2_lw-1)+S(2*N2))/refl_g2_lw)*dA_w


L220: Do i=NoPV+5*N2+2, NoPV+6*N2-1
f(i)=Sw2_1- dA_w*h2b_w(i-NoPV-5*N2)*(T(i)- T(i-N2)) - &
                &       (kg*w*t_w2/dH_w)*(2*T(i)-T(i-1)-T(i+1)) - &
                &       hw2*dA_w*(T(i)-T(i+N2)) - &
                &       ((Q(i-NoPV-4*N2)*(refl_g2_lw-1)+S(i-NoPV-4*N2))/refl_g2_lw)*dA_w
        End Do L220
sum=0
Qroom_w=0

!!
!!Second Glazing of W2
!!
f(NoPV+6*N2+1)=Sw2_2- hroom*dA_w*(T(NoPV+6*N2+1)-Troom) - &
                        hw2*dA_w*(T(NoPV+6*N2+1)-T(NoPV+5*N2+1)) - &
                        (kg*w*t_w2/dH_w)*(T(NoPV+6*N2+1)-T(NoPV+6*N2+2))


sum=sum+hroom*dA_w*(T(NoPV+6*N2+1)-Troom)

f(NoPV+7*N2)=Sw2_2- hroom*dA_w*(T(NoPV+7*N2)-Troom) - &
                        hw2*dA_w*(T(NoPV+7*N2)-T(NoPV+6*N2)) - &
                        (kg*w*t_w2/dH_w)*(T(NoPV+7*N2)-T(NoPV+7*N2-1))

sum=sum+hroom*dA_w*(T(NoPV+7*N2)-Troom)

L221: Do i=NoPV+6*n2+2, NoPV+7*N2-1
f(i)=Sw2_2- hroom*dA_w*(T(i)-Troom) - &
                        hw2*dA_w*(T(i)-T(i-N2)) - &
                        (kg*w*t_w2/dH_w)*(2*T(i)-T(i-1)-T(i+1))
sum= sum+hroom*dA_w*(T(i)-Troom)
        End Do L221
```

```
Qroom_w=sum

Ebal_w=Etot_w- Erefl_w- Qroom_w- Qairf_w- Qairb_w- Qout_w- Eloss_w- Etrans
Eff_w=(Qroom_w+Qairf_w+Qairb_w+Etrans)/Etot_w

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!!Printing of Vision section
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

Print *,'Energy balance for vision section', Ebal_w
Print *,'Vision Section Efficiency:', Eff_w
Print *,'Erefl_w', Erefl_w, 'Qroom_w', Qroom_w, 'Qout_w', Qout_w
print *,'Etrans', Etrans, 'Qairf_w', Qairf_w, 'Qairb_w', Qairb_w, 'Eloss_w', Eloss_w
print *,'Exit Temperature Front:', T(NoPV+2*N2), 'Exit Temperature Back:', T(NoPV+5*N2)
print *, 'Exit Temperature of PV Section:', T(4*N2)-273.15

!Loop 109 sums all energy balance equations.  This value should approach 0 as
!all the energy balance equations are defined to equal 0. Broyden's method will not
!always converge on the right answer, so this factor is an indicator of how close
!the convergence was
sum=0
L109:    Do i=1,N
                 sum=sum+abs(f(i))
                End Do L109

                Print *,'Non-convergence factor=', sum



return
END Subroutine funcv

Function Ch(Tma)
Implicit None
Real, INTENT(in) :: Tma
Real :: Ch
                        Ch= 1000*(1.0057 +0.000066*(Tma - 300.15))
Return
End Function Ch

Function density(Tma)
Implicit None
Real, INTENT(in) :: Tma
Real :: density
                        density= 1.1774 - 0.00359*(Tma - 300.15)
Return
End Function density
```