InDeWaG Tool

for Water Flow Glazing Envelopes

Department of Applied Mathematics
School of Aeronautical and Space Engineering
Technical University of Madrid (UPM)

Part I

Graphical user interface

Chapter 1

General information

1.1 Software application overview

Water flow glazing facades are considered active envelopes able to react or to adapt to the external and internal conditions of the building. The use of the building, the orientation of the facade and the location of the project are relevant inputs to determine the glazing composition as well as the energy strategy involved in the facade. A proper design of the glazing composition and the energy strategy will allow huge energy savings in the whole project. The InDeWaG software tool implements a graphical user interface allowing engineers to design water flow glazing facades.

The software tool for engineers as a standalone application and the software library for developers is an open source project which has a documented application program interface to implement WFG modules in existing simulation codes. The software tool has different graphical screens such as:

1. Energy balance considerations associated to potential sun energy for specific locations.

2. Spectral properties of the glazing based on the selection of different glass layers or coatings.

3. Thermal simulator of the glazing module based on absorption or spectral properties.

4. Thermal simulator of simplified rooms including insulated opaque facades and water flow glazing facades.

The first phase of the design is to know the potential sun energy that the project can manage depending on location, orientation of different facades and usage. To accomplish this pre-design phase, basic geometrical aspects and location are needed to evaluate is the project can be seen as a nearly zero energy building. The tool implements real meteorological data to evaluate the total sun energy that can be absorbed in the envelope. It considers a full year energy balance to measure the energy demands and the absorbed energy. With a very intuitive graphical user, the engineer can modify the first draft of the project to fulfil the energy requirements. Energy balance considerations graphs are shown for the selected period of time: day, week, month or year. This also allows to select the amount of the thermal mass to flatten temperature oscillations of the selected period of time. Once the opaque and transparent surfaces are selected, the g factor is considered as an input at this stage to accomplish energy requirements. The output of this first phase is the transparent surfaces with their desirable g factor and thermal transmittance.

The purpose of the second phase is to select a glazing form the InDeWaG glazing catalogue to fulfil spectral properties obtained in the first phase. The tool determines spectral properties of a specific glazing composition. Transmittance, reflectance and absorptance can be presented versus wavelength of solar radiation and versus angle of incidence of solar beam radiation.

Once the glazing is chosen, the next phase is to simulate the thermal performance of the isolated glazing for given boundary conditions. This is done by imposing constant boundary conditions or variable boundary conditions. These boundary conditions comprise: exterior temperature, beam and diffuse solar radiation, interior temperature and inlet temperature. If real outdoor conditions are selected based on a specific location, season of period of time of the simulation can be selected. Movies time stories of spatial temperature profiles are shown to deeply understand the water flow glazing behavior. Outlet temperature is determined based on the selected flow rate of the module and water heat gain can be plotted versus time.

To complete the thermal study of water flow glazing facades, a simplified room can be simulated. The room is formed with opaque and water flow glazing facades. The dimensions of the room as well as the interior absorption properties of walls can be selected. In this case, the only boundary conditions are the outdoor temperature and beam and diffuse solar radiation. The interior temperature is simulated as a thermal property of the interior thermal mass.

1.2 Organization of the manual

This manual is organized in three parts:

I. Graphical user interface.

II. Application program interface.

III. software design document.

Part I describes the different graphical tools to allow project engineers to design water flow glazing facades. Understanding active facades as transparent thermal collectors are carried out by thermal simulation movies. This tool allows the project engineer to determine the water heat gain during the whole year. Part II constitutes a complete library to simulate water flow glazing envelopes written in modern Fortran. This library has a documented application program interface that allows developers to integrate water flow glazing envelopes in existing energy simulators for buildings. A complete benchmark for the isolated glazing and the insulated glazing in a simplified room is documented in part III.

Chapter 2

Getting started

2.1 Installation

The latest version of "InDeWaG tool to design water flow glazing envelopes" and the source code is available on GitHub (https://github.com/jahrWork). To install the software tool, proceed in the following way:

1. Download the rar file `Standalone InDeWaG tool.rar` from the website `https://github.com/jahrWork`

2. Unzip the downloaded file. A folder named `Standalone InDeWaG tool` will be created.

2.2 Standalone application

Once the downloaded file has been unzipped, the `Standalone InDeWaG tool` folder holds the standalone application. Besides the executable file, there are some material and climate folders and files. To execute the standalone application:

1. Open the `Standalone InDeWaG tool` folder and double-click the file `tools.exe`.

2. The simplified room screen will be opened.

3. Click the button `RUN Simulation`.

If decompressed files have been copied correctly, the scale widget will increase progressively showing the number of imges being simulated. After that, a story movie of the simulation will be shown (Figure 2.1) presenting the evolution of the inside temperature and temperatures profiles of the envelope: opaque walls and water flow glazing facade.

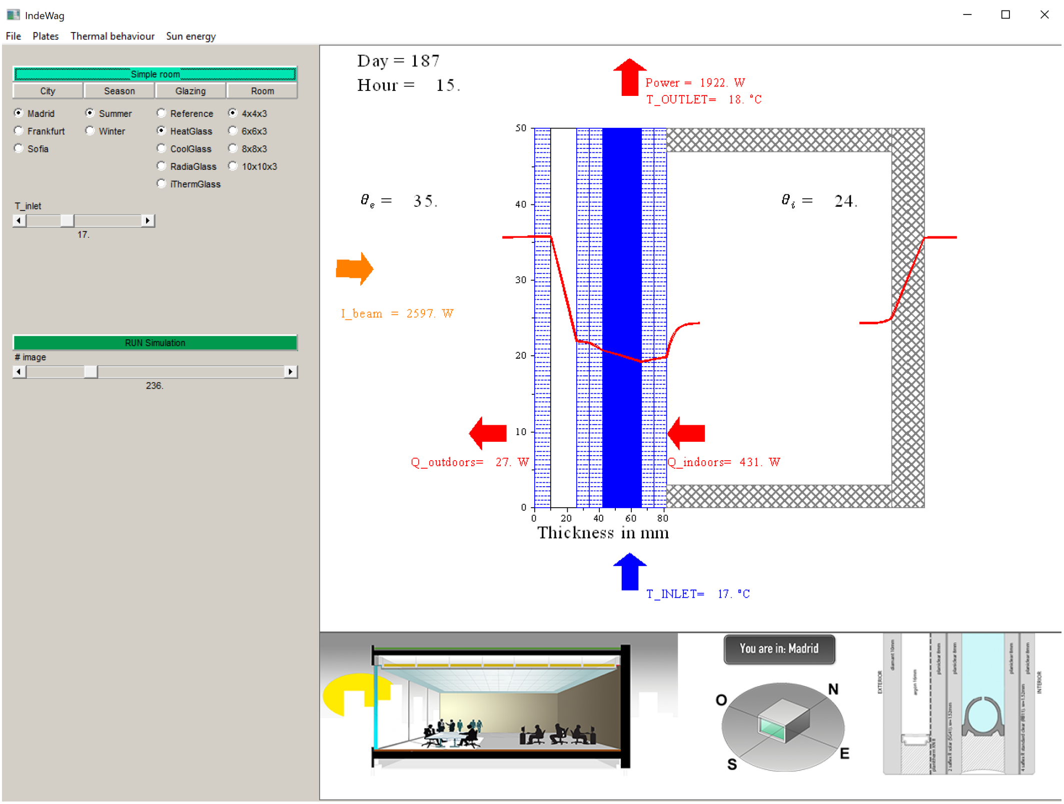

Table 2.1: Screen-shot of the simplified room simulation. Temperature profiles and parameters selection.

2.3 Software library for developers

In order to allow developers to integrate water flow glazing modules into existing energy simulation tools, the source codes of this module are available. They are located under the folder named `sources` which is inside the folder `Standalone InDeWaG tool`. There are 78 Fortran modules that comprehend the solution of the spectral and thermal problem of water flow glazing facades. Part II and part III of the manual are devoted to explain in detail how to work with this library to integrate this module in existing energy simulation tools for buildings. The software is written in modern Fortran based on object oriented paradigm. Software is structured attending to different physical models. The determination of spectral properties of the water flow glazing is encapsulated and based on different hierarchical levels of abstraction. The same is done for the solar distribution inside rooms or zones. A similar strategy was accomplished to deal with the thermal problem.

2.4 Exploring the start page: simplified room

Once the InDeWaG tool has been initiated, the application starts with the thermal simulation of a simplified room with water flow glazings as it is shown in Figure 2.1. It is considered an insulated room formed by opaque walls and one water flow glazing facade that can be located in different specific locations and dimensions. For all configurations the water flow glazing is facing south as it is shown in Figure 2.1. One of the main objectives of this screen is to show with a very simplified interface the thermal behavior of water flow glazing facades when they are integrated in buildings. The screen is divided into left part to control and to select different parameters of the simulation and right part to show graphical results such as story movies or schematics. The following parameters for the simplified room can be selected:

1. City: Madrid, Frankfurt and Sofia. Once any city of this list is selected, the simulation tool uses the standard EneryPlus EPW weather data file for the specific location.

2. Season: Summer and Winter. By selecting one of these two options the tool will simulate some period of time to show the influence of summer or winter conditions. The summer period goes from July 5th to July 5th. The winter period goes from January 15th to January 17th.

3. Glazing: Reference, HeatGlass, CoolGlass, RadiaGlass and iThermGlass. By selecting any of these glazings, the user can compare the adequacy to the different locations, sizes and orientations. These glazings are predefined in glazing catalog and are shown in detail in the lower right part of the screen (Figure 2.1).

4. Room dimensions. Different floor dimensions can be selected to understand the influence of the room dimensions.

Once these different parametes are selected, the user should click `RUN Simulation` to initiate the the simulation. The animation temperature movie is formed by 285 frames or images. By selecting an image number in the scale widget below the simulation control, temperature profiles are shown for this specific simulation time. The day and the hour of the day is shown in the upper right corner of the graphical window.

Chapter 3

Using the software tool

The InDeWaG tool will cover: *(i)* a thermal pre-design method based on energy balance, *(ii)* spectral characterization for a preselected glazing and *(iii)* thermal simulation of a simplified building in different orientations and locations. To navigate between different tools or options, the application menu has the following tabs:

- `File`. This tab allows to open or save different configurations or sessions and exit the actual session.
- `Plates`. By selecting this tab, the user could dimension the glass thickness to hold static loads such as hydrostatic pressure.
- `Thermal behaviour`. There are three sub-menus in this tab: `Spectral problem`, `Thermal problem` and `Simplified room`. By selecting one of these sub-menus, the user can determine spectral parameters or the simulate thermal performances.
- `Sun energy`. Under this tab, `Energy balance` graphs and `Cast Shadows` can be plotted.

The following protocol is established to help engineers in designing buildings with water flow glazing facades.

1. Understand the potential sun energy harvesting associated to some specific location and building orientation and dimension. This task can be accomplished with the `Energy balance` interface. Once it is understood if a zero energy project can be built, the following step is to decide the energy strategy and the specific glazing to implement this strategy.
2. The second step is to select a glazing to accomplish the energy strategy. The `Spectral problem` screen shows the absorption coefficient of each layer and global transmittance and reflectance of the glazing.
3. Once the selection of the glazing is validated, the final step is to check with a thermal simulation is the energy project fulfills the zero nergy expectation. This simulation process will give us feedback to iterate in previous design stages provoking changes in orientations, dimensions and glazing properties.

3.1 Energy balance interface

Select the `Sun energy` sub-menu in the `Energy balance` tab. Then, if `Evaluate whole year` is clicked, the sun energy curves associated to each facade will be shown (Figure 3.1). To create a building in a specific location, the buttons on the right part of the screen are used. These controls allows to show the potential absorbed sun energy associated to each facade. These controls are divided into the following parts:

- `EnergyPlus weather file`. This list allows to select the location and type of the incoming sun energy. Energy balance can be presented with beam and diffuse solar radiation, only diffuse or only beam radiation.

- `Building`. The building is formed by six surfaces: north, east ,south, west, roof, floor. These surfaces are defined by their dimension, thermal transmittance U, g factors, energy transmittances T, photo-voltaic proportion, indoor conditions, orientations and equipments.

- `Graphs`. This button list allows to select the type of graphic to show

  1. `Potential Solar Energy kWh.` It takes into account the integral impinging solar energy during some specific period of time. Parametric curves stand for different facades.
  2. `Potential Solar Power kW.` It takes into account the instantaneous impinging solar power during some specific period of time. Parametric curves stand for different facades.
  3. `Energy demand kWh.` According to thermal transmittances, surfaces and g factors or proportions of the incoming energy which is absorbed, these graphs show the energy demand to assure a comfort temperature.
  4. `Power demand kW.` According to thermal transmittances, surfaces and g factors, these graphs show the instantaneous power demand to assure a comfort temperature.
  5. `Int./Ext. temperatures.` These graphs show the indoor, outdoor and comfort temperature during some specific period of time.

  To the right of the graph selection, there two scale widgets: `Month` and `Day` that allows to show a specific time span. If zero is selected for the month, the whole year is presented. If some specific month from 1 to 12 is selected, the graph is presented from day 0 to day 30. The same is apppicable to the scale widget `Day`. If `Month` is greater than 0 and `Day` is greater than 0, the graph will show the time spane from 0 hours to 24 hours of this specific day and month. In Figure 3.2, parametric curves of impinging solar power for different facades is shown on July 5th.

- `Evaluate whole year`. This button allows to integrate energy balance equations during the whole year. It is important to notice that this integration is not numerical simulation it is just a detailed balance between incoming sun energy and outgoing heat losses or gain when the indoor temperature must be maintained close to the comfort temperature.

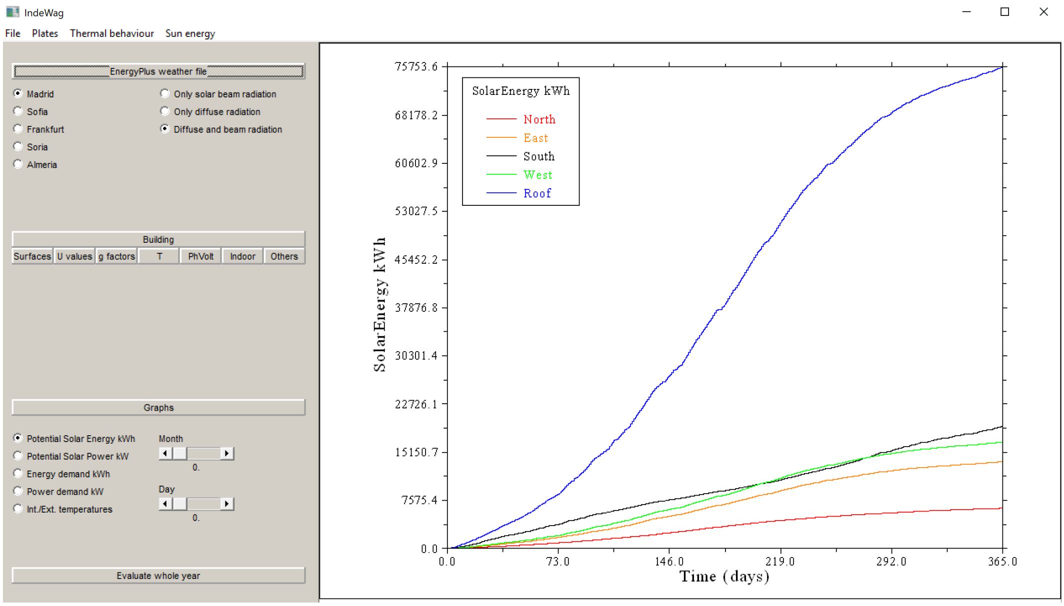

Figure 3.1: Potential sun energy associated to different building facadesfor the whole year

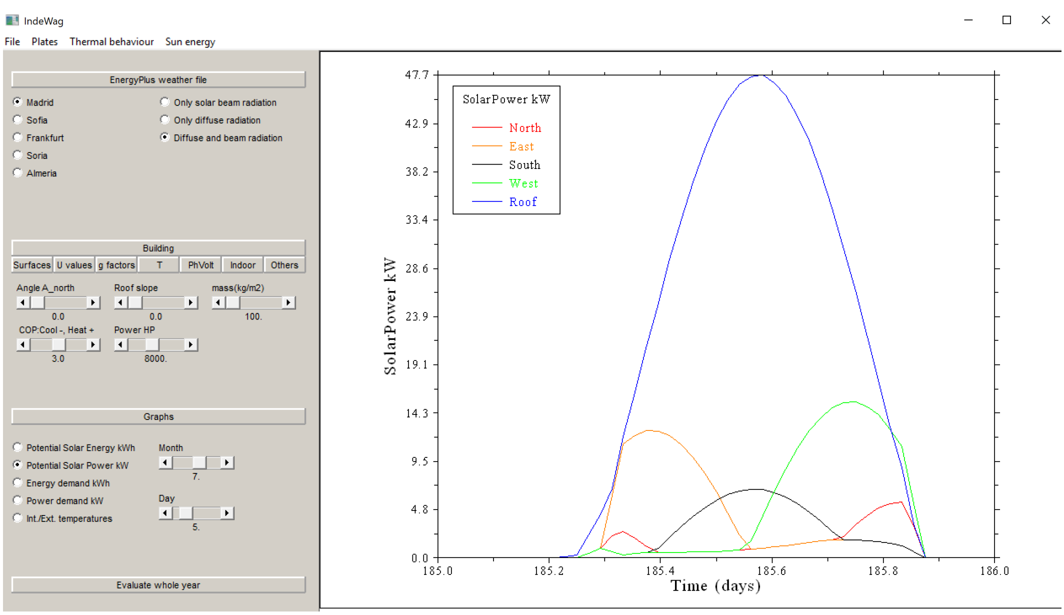

Figure 3.2: Sun power associated to different building facades on July 5th

3.2 Cast Shadows interface

The main idea of the `Cast Shadows` interface is to avoid solar obstructions on water flow glazing facades by exploring whole year simulations. Figure **??** represents a screen shot when selecting the `Cast Shadows` sub-menu in the `Energy balance` tab. There are blocks controls and ubication parameters to create a specific building. Once the building is created, `3D` visualization or `Floorplan` visulaization can be selected to draw cast shadows.

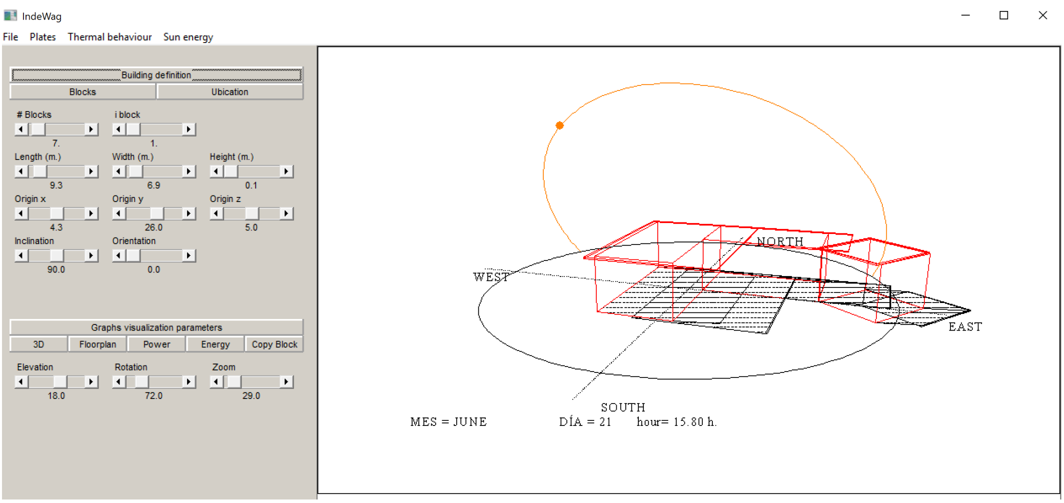

Figure 3.3: Cast shadows to avoid sun obstructions, June 21th at 15.8 h

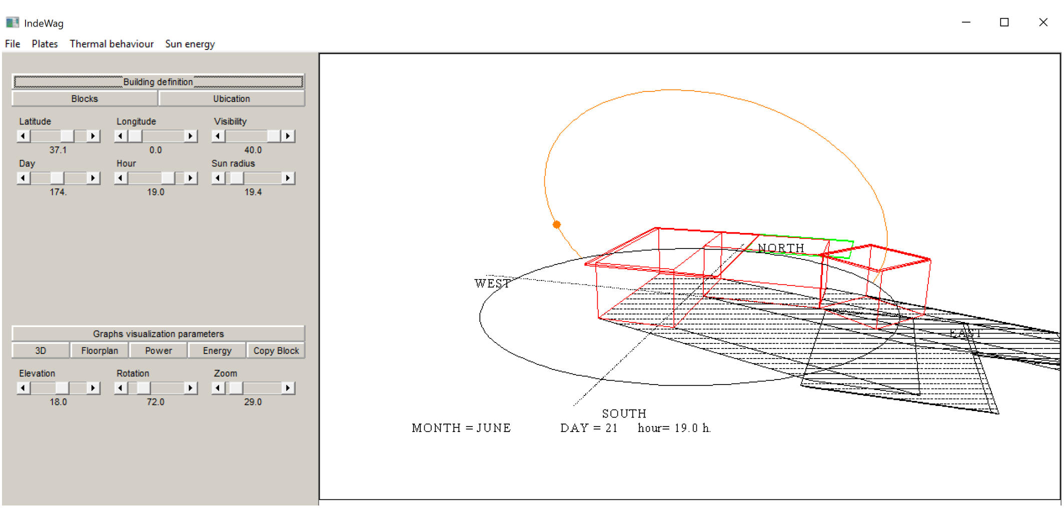

Figure 3.4: Cast shadows to avoid sun obstructions, June 21th at 19 h

3.3 Spectral problem interface

If `Spectral problem` sub-menu in the `Thermal behavior` tab is selected, an image similar to Figure 3.5 will be shown. The input parameters are:

- A list with the glazing catalog: `HeatGlass`, `CoolGlass`, `RadiaGlass`, `iThermGlass` and `RadiaWall`.
- A list of spectral properties to visualize:
  - `Angular properties`. Figure 3.6 shows the front and back reflectance R, transmittance T and absorptance A as a function of the angle of incidence.
  - `Spectral properties`. Figure 3.7 shows the front and back reflectance R, transmittance T and absorptance A as a function of the wavelength.
  - `Weighted values`. Figure 3.5 shows a complete description of different layers of the selected glazing. A detailed table of front and back layer absorptances for the beam and diffuse solar radiation are also shown. Besides, thermal and spectral global parameters are displayed. Since the water flow glazing is active, the g factor depends on the flow rate. If the water is flowing, the g factor is lower than when the water chamber is stopped, allowing energy entrance into the building. In the same manner, the thermal transmittance U is almost zero when the flow rate is the design flow rate because the water chamber isolates the building from outdoor conditions. When the flow is stopped, the thermal transmittance depends mainly on the composition of the air chamber.

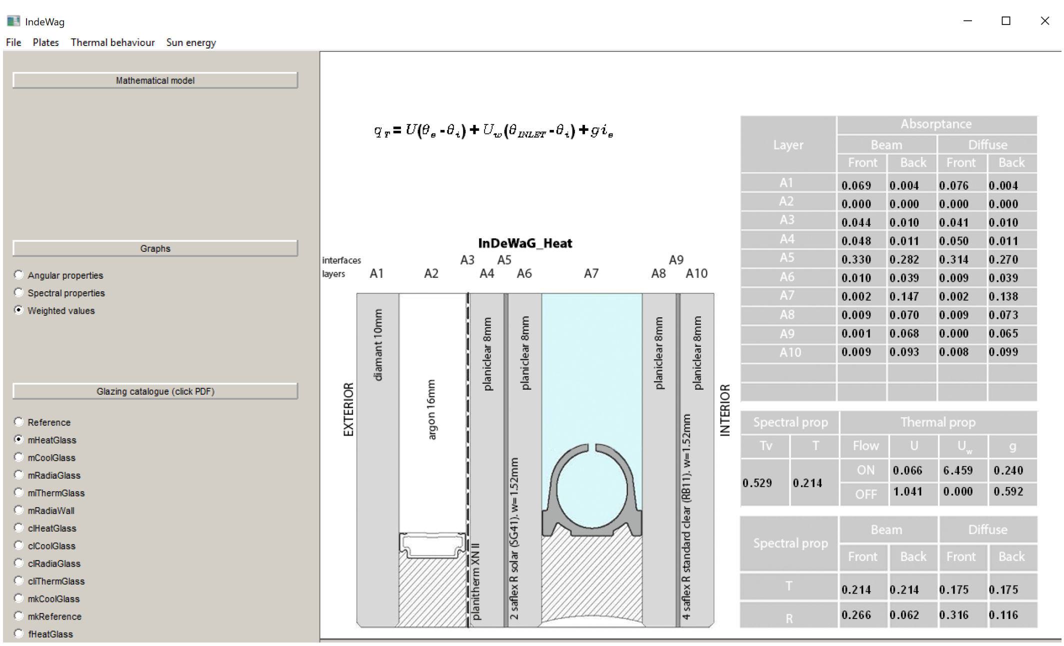

Figure 3.5: Layer composition and spectral properties of HeatGlass.

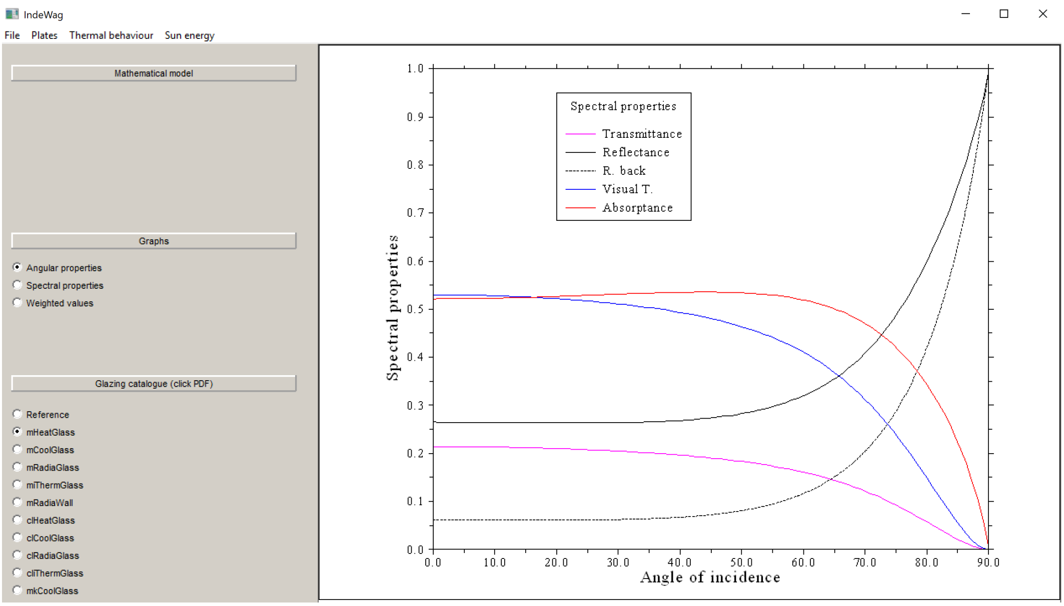

Figure 3.6: Spectral properties as a function of the angle of incidence.

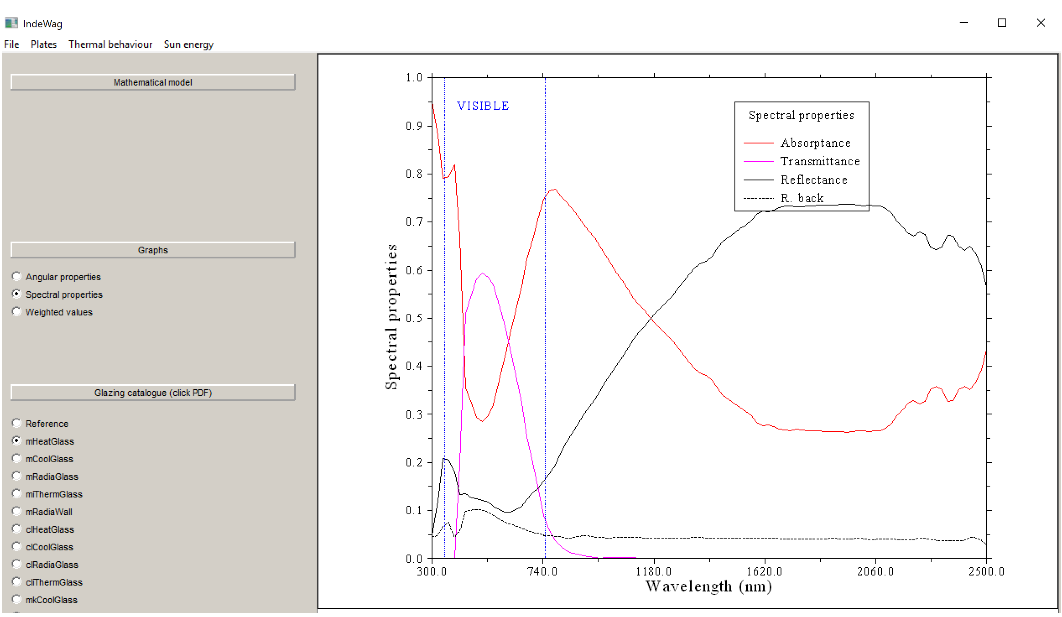

Figure 3.7: Spectral properties as a function of the solar wavelength.

3.4 Thermal problem interface

Select the `Thermal problem` sub-menu in the `Thermal behavior` tab. Then, if `RUN simulation` is clicked, simulation is initiated by creating different frames or images. The animation temperature movie is formed by 572 images for the default time span (Figure 3.8). By selecting an image number in the scale widget below the `MOVIE` button, temperature profiles are shown for this specific frame. The day and the hour of the day is shown in the upper right corner of the graphical window.

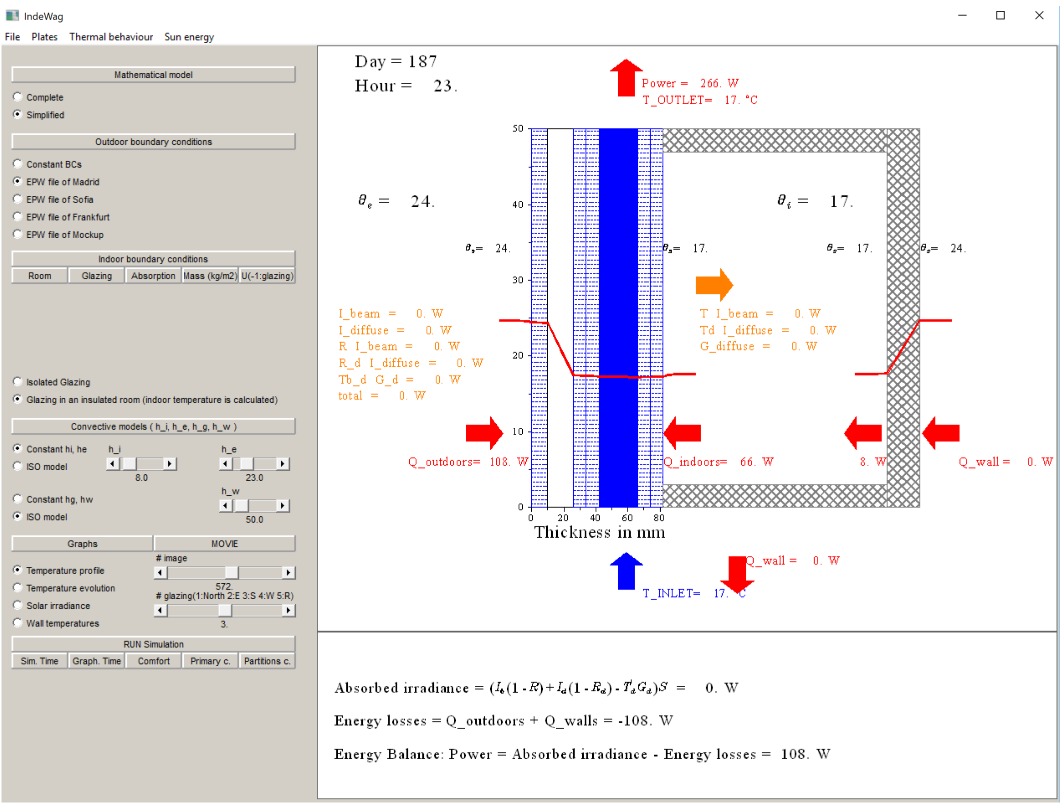

Figure 3.8: Thermal problem inputs

The thermal problem has many different parameters and models to be controlled or tested. The right part of the screen is divided into six different zones:

- `Mathematical model.`
- `Outdoor boundary conditions.`
- `Indoor boundary conditions.`
- `Convective models.`
- `Graphs.`
- `RUN Simulation.`

By default the `Mathematical model` is `Simplified` but `Complete` can be freely selected. The main difference between these two models is associated to thermal mass of the glazing. While the simplified model neglects the thermal mass of different layers and solves internally a system of algebraic equations, the complete model assumes no negligible thermal mass and integrates every layer as a partial differential equation. Then, it imposes matching conditions between different layers to solve the complete layer. Hence, it is clear that the complete model becomes much more demanding in terms of computational effort than the simplified model. However, the difference between the temperatures simulations results of these two models is insignificant. Only special test cases can show significant differences.

3.4.1 Outdoor boundary conditions

The `Outdoor boundary conditions` can be selected `Constant BCs` or the EnergyPlus weather file associated to some location. If they are selected as constant then, scale widgets for exterior temperature, beam and diffuse solar radiation should be chosen.

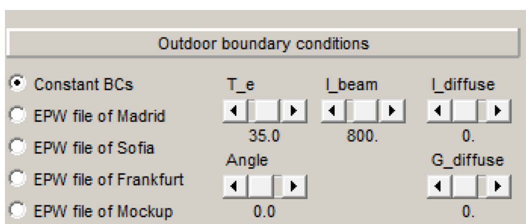

Figure 3.9: Constant outdoor conditions: exterior temperature, beam and diffuse radiation and angle of incidence.

3.4.2 Indoor boundary conditions

`Indoor boundary conditions` are related to interior room parameters. By default (Figure 3.10) and interior parameters should be defined. However, if `Isolated glazing` is selected in the same list (3.10), internal room parameters are not taken into account. When the `Glazing in an insulated room` is selected, the following thermal parameters that define the room and operational values for the water flow glazing facade should be given:

- `Room`.
- `Glazing`.
- `Absorption`.
- `Mass`.
- `U value`.

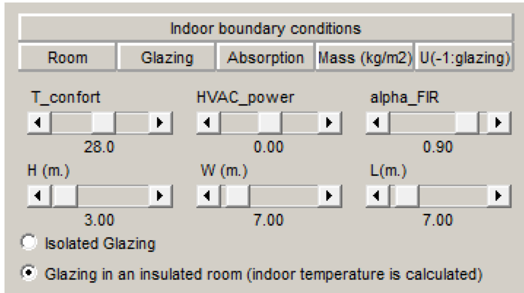

Figure 3.10: Indoor boundary conditions. Room dimensions and HVAC system.

When the `Room` button is clicked, selection parameters can be done by means of scale widgets as it is shown in Figure 3.10. These parameters include: `T_confort` which is the set-point of the indoor temperature, `HVAC_power` represents the extracted or injected heat flux associated to air renovation or HVAC equipments, `alpha_FIR` or the FIR absorption for the opaque walls of the room, `H` Height, `W` width and `L` length of the room measured in meters.

When the `Glazing` button is clicked, the following parameters of the secondary circuit can be configured (Figure 3.11):

- `c`: specific heat capacity of the secondary circuit fluid $[J/(KgK)]$.
- `Flow`: flow rate of secondary circuit $[l/(min.m^2)]$.
- `Orientation`: orientation of the glazing facade (180 means facing south).
- `Inclination`: inclination of the glazing facade (90 means vertical).

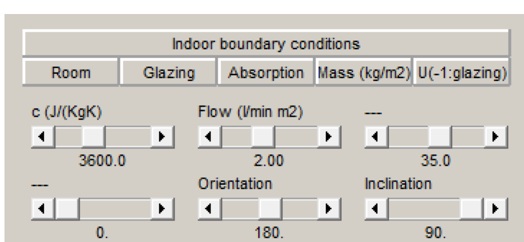

Figure 3.11: Indoor boundary conditions. Glazing position and secondary circuit parameters.

When the `Absorption` button is clicked, different NIR absorptions of the opaque walls is given (Figure 3.12).

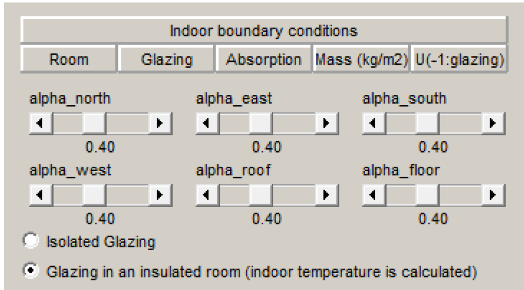

Figure 3.12: Indoor boundary conditions. Interior NIR absorption of opaque walls.

When the `Mass (kg/m2)` button is clicked, mass per square meter of different walls can be selected (Figure 3.13)

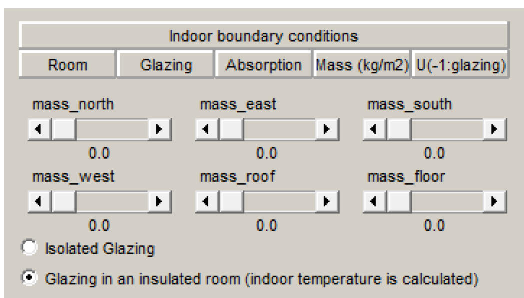

Figure 3.13: Indoor boundary conditions. Mass per square meter of opaque walls.

When the thermal transmittance button `U` button is clicked, thermal transmittances of walls can be modified by means of scale widgets (Figure 3.14). To assign a water flow glazing facade to any wall, `U` = `-1` is used.

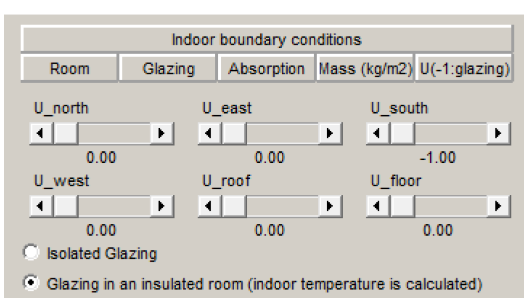

Figure 3.14: Indoor boundary conditions. Thermal transmittance of opaque walls.

3.4.3 Convective models

The interior or exterior convective heat transfer mechanism can be modeled by constants values or by more elaborated models given by the norm ISO 15099:2003. By default, constant values for `h_i` and `h_e` are given (Figure 3.15). However, by selecting `ISO model`, `h_i` and `h_e` are determined precisely.

Besides, the convective heat transfer mechanism inside the air chamber and the water chamber is modeled by the coeficients `h_g` and `h_w`. Once again, these coefficients can be considered constant or given by more precise models. By default, `ISO model` is selected for the air chamber of the glazing (Figure 3.15).

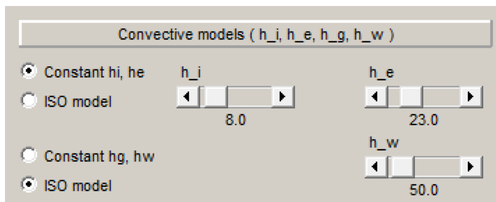

Figure 3.15: `Convective models` inputs

3.4.4 Graphs

Once `RUN simulation` is clicked and the complete simulation is finished, simulation results can be shown in four different ways (Figure 3.16).

- `Temperature evolution`. This $x - y$ plot shows exterior and interior temperatures of the air and the surfaces, inlet and outlet water temperature and buffer tank temperature.

- `Temperature profile`. This option presents simulation results as a temporal movie showing temperature profiles of the glazing and walls. It also shows heat fluxes through walls and glazing together with energy balance considerations.

- `Solar irradiance`. This $x - y$ plot shows exterior beam and diffuse solar radiation on a horizontal plane and their projections on the glazing surface. It also shows the internal diffuse radiation.

- `Wall temperatures`. This $x - y$ plot shows external and internal surface temperatures of walls and glazing.

By clicking button `MOVIE`, a consecutive sequence of images presenting temperature profiles is shown. Beneath this button there are two scale widgets, the first one is the for image number to be displayed, and the second one stands for the glazing to be drawn.

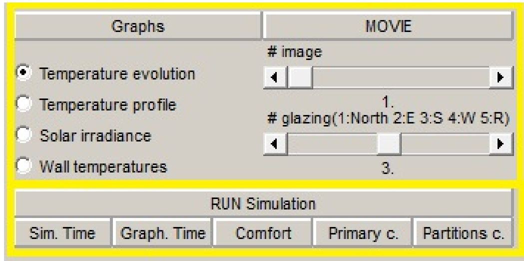

Figure 3.16: Different ways to present simulation results.

3.4.5 Simulations

There are five configuration buttons to control the simulation as it is shown in Figure 3.17.

- `Sim. Time`. By clicking this button, time span of the simulation can be set by selecting the starting time and finishing time of the simulation (Figure 3.17).

- `Graph. Time`. By clicking this button, axes limits of the graphs can be stated. This includes time axes, temperature and power (Figure 3.18).

- `Comfort`. By clicking this button, the initial room temperature, the comfort temperature, and the working hours can be set (Figure 3.19)).

- `Primary c.` By clicking this button, some primary circuit parameters can be set. These include the following parameters for the buffer tank: volume, the initial temperature, thermal transmittance, surface, flow rate and specific heat transfer of the primary circuit (Figure 3.20). Besides, there is a scale widget to set the constant free cooling power associated to the evaporation of a exterior open water surface. It is measured in W/m^2 and its surface is equal to the buffer tank surface.

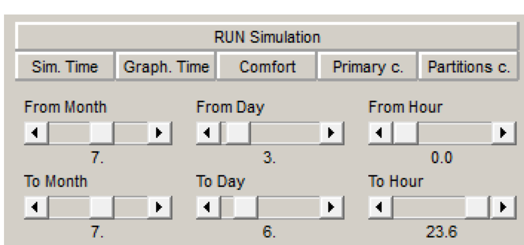

Figure 3.17: Simulation controls. Time span to be simulated.

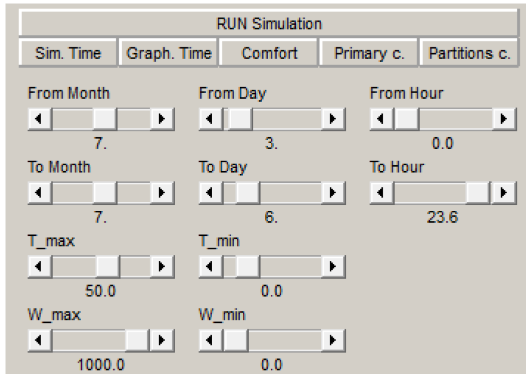

Figure 3.18: Simulation controls. Axis range to present results.

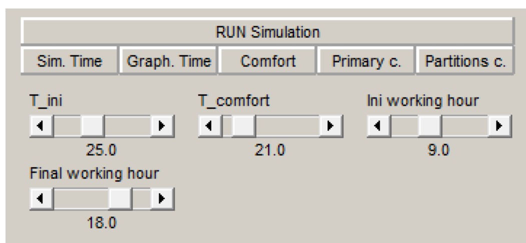

Figure 3.19: Simulation controls. Interior comfort parameters.

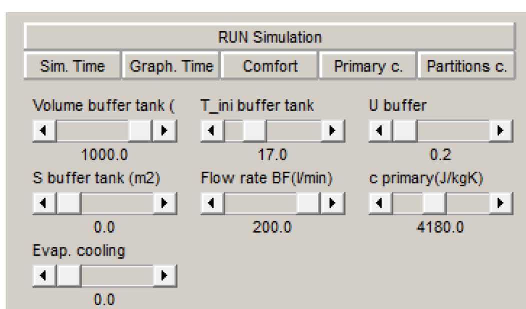

Figure 3.20: Simulation controls. Parameters of the primary circuit.

Part II

Application Program Interface

Chapter 1

Spectral problem

1.1 Overview

This library is designed to solve the spectral problem of a water flow glazing. These subroutines allow to obtain the optical properties of a multilayer glazing as a function of wavelength or averaged mean values for different angles of incidence. The API of the spectral problem comprises the following public subroutines:

```
module API_Spectral_Problem

 use GlazingM
 implicit none

  private
  public ::                          &
      Calculate_spectral_properties, & ! Spectral properties of a glazing
                                       ! Properties are stored in a database.
      Append_to_catalogue,           & ! Appends a new glazing to the catalog.
      Spectral_properties,           & ! As a function of wavelength (lambda).
      Averaged_spectral_properties,  & ! PVB are considered layers
      Glazing_Spectral_properties,   & ! As a function of wavelength (lambda)
                                       ! and angle of incidence.
      look_for,                      & ! Look for an existing glazing name.
      Averaged_absorptances,         & ! Averaged spectral properties.
      Averaged_diffuse_properties,   & ! Diffuse spectral properties.
      Glazing_database_query,        & ! Database queries
      FIR_emissivities                 ! Emissivities of all interfaces
```

Listing 1.1: `API_Spectral_problem.f90`

In order to facilitate the use of this API, a glazing catalog is predefined. To make use of this catalog, the subroutine `Calculate_spectral_properties` is called to calculate the spectral properties of this glazing once in a life. This subroutine creates internally an object to save this spectral properties for every angle of incidence.

It is also possible to define new elements aside from the glazing catalog predefined ones by means of the subroutine `Append_to_catalogue`. This new elements should be defined by means of existing layers of the folder `materials` or by including new layers from the International Glazing Database (IGDB). Optics program could be used to create new files for glass layers, coatings o PVB layers. This subroutine initializes a new glazing and allows the user to ask for subsequent spectral properties. Once the glazing has been calculated, the averaged spectral properties of a specific angle of incidence can be known by calling the subroutine `Averaged_Spectral_properties`. This subroutine performs a second order interpolation from the internal spectral data which have been calculated before.

To integrate a thermal problem, the `Averaged_absorptances` subroutine can be called. This subroutine does not consider layers that are thinner than 3 mm to avoid instability problems. The absorptance of these thin layer is added to the interface absorptance of neighbor layers. To obtain the spectral properties as a function of the wavelength, the subroutine `Spectral_properties` is called. In addition, the subroutines `Glazing_database_query` and `Averaged_diffuse_properties` are thought to be used in the thermal problem. The subroutine `Glazing_database_query` returns the layout of a glazing, but without taking in account the layers or interfaces whose thickness is less than 3 mm in order to avoid numerical instabilities in the non steady thermal problem.

In the same manner, the subroutine `Averaged_diffuse_properties`, which is similar to `Averaged_Spectral_properties`, gives the spectral mean diffuse properties. Once again, when a layer has a thickness less than 3 mm, it is considered as an interface to thermal problem with the same absorption properties as the considered layer.

This library is encapsulated in the module `API_Spectral_Problem`. To use this library, the sentence `use API_Spectral_Problem` should be included at the begining of the FORTRAN code.

1.2 Spectral properties as a function of wavelength and angle of incidence

In this first example, the spectral properties of a predefined glazing called `Radiaglass` are calculated. Once this glazing elements are calculated, the spectral properties are known for of normal angle of incidence in radians (`theta = 0`) and for a specific wavelength in nm (`lambda = 750`).

```
subroutine Example_API_Spectral_Problem

 ! maximum number of layers
   integer, parameter :: N = 7

   real :: absorptances(N), absorptances_back(N), absorptances_I(N)
   real :: absorptances_I_back(N)
 ! transmittance, reflectance: front and back
   real :: T, T_back, R, R_back, Tv, Tv_back
   real :: theta = 0.0, lambda = 750.0
 ! number of layers
   integer :: Nl

    call Calculate_spectral_properties("RadiaGlass")

    write(*,*) "Spectral properties (normal incidence) at lambda = ", lambda

    call Spectral_properties( "RadiaGlass", lambda,  Nl, absorptances,     &
                              absorptances_back, absorptances_I,           &
                              absorptances_I_back,   T, T_back, R, R_back)

    write(*,'(a10,15f8.3)') " A, T, R =",                                  &
                         sum(absorptances(1:Nl)+absorptances_I(1:Nl)), T, R
    write(*,'(a10,15f8.4)') " A =", absorptances(1:Nl)
    write(*,'(a10,15f8.3)') " A_I =", absorptances_I(1:Nl+1)
```

Listing 1.2: `API_Example_Spectral_problem.f90`

1.3 Create a new glazing

It is also possible to define new elements for the glazing catalog by means of the subroutine `Append_to_catalogue`. This new elements should be defined by means of existing layers of the folder `materials` or by including new layers from the International Glazing Database (IGDB). In this example a new glazing called `NewGlazing` is created by defining its layers, thicknesses, layer names and coatings or interfaces. Once this glazing is created, the above subroutines can de used to determine spectral properties of the new glazing.

```
    call Append_to_catalogue(                                              &
                   glazing_name = "NewGlazing",                            &
                   layer_name = [ character(len=100) ::                    &
                         "PLANILUX 10mm.SGG", "Air", "PLANILUX 10mm.SGG"], &
                   layer_thickness = [ 10., 16., 10.  ],                   &
                   interface_name  = [ "", "", "", "" ],                   &
                   substrate_name  = [ "", "", "", "" ]                    )
```

Listing 1.3: `API_Example_Spectral_problem.f90`

1.4 Averaged spectral properties

Generally, the spectral properties are averaged by means of a specific iluminant to give a averaged or mean value. These averaged values are only functions of the angle of incidence. They comprise front and back absorptance for each layer and each inteface and transmittance and reflectance of the glazing.

```
call Averaged_spectral_properties(                              &
                    "RadiaGlass", theta, Nl, absorptances,      &
                     absorptances_back, absorptances_I,         &
                     absorptances_I_back, T, T_back,            &
                     R, R_back, Tv, Tv_back                     )
```

Listing 1.4: `API_Example_Spectral_problem.f90`

1.5 Averaged absorptances

To solve the thermal problem, absorptances of each layer should be provided. When dealing with very thin layers like PVB, the resulting absorptance of these kind of layers is transferred to the closest interface. It allows to treat these thin layers as interfaces avoiding stiff problrm when integrating the thermal problem. This example allows to determine front and back absorptances of different layers with the above considerations.

```
call  Averaged_absorptances(                                            &
             "RadiaGlass", theta, Nl, absorptances,                     &
              absorptances_back,  absorptances_I, absorptances_I_back,  &
              T, T_back, R, R_back, Tv, Tv_back                         )
```

Listing 1.5: `API_Example_Spectral_problem.f90`

1.6 Application program interface

This chapter is about the module `API_Spectral_problem`. This module can calculate different spectral properties of a predefined glazing, like the absorptances of each layer or transmittance of the glazing. It contains the following subroutines:

- `Append_to_catalogue:` Adds a glazing to the catalogue and calculates its spectral properties once in a life.
- `Calculate_spectral_properties:` Calculates once in a life the spectral properties of a predefined glazing.
- `Spectral_properties:` Returns spectral properties for a given wavelength.
- `Averaged_spectral_properties:` Returns wavelength averaged spectral properties for a given angle of incidence.
- `Averaged_absorptances:` Returns wavelength averaged spectral properties for a given angle of incidence. Layers thinner than 3 *mm* are collapsed.
- `Averaged_diffuse_properties:` Returns wavelength and angle of incidence averaged spectral properties. Layers thinner than 3 *mm* are collapsed.
- `Glazing_database_query:` Returns the layout of defined glazing.
- `FIR_emissivities:` Returns resultant emissivities of each layer.

The predefined types of glazings are: CoolGlass, RadiaGlass, HeatGlass, iThermGlass and RadiaWall.

1.6.1 Create a new glazing

```
call Append_to_catalogue( glazing_name, layer_name, layer_thickness,  &
                          interface_name, substrate_name              )
```

The subroutine `Append_to_catalogue` defines the configuration of a new glazing and calculates once in a life its spectral properties, so it is not necessary to use `Calculate_spectral_properties` when it is created a new glazing. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Character string defined by the user to identify the new glazing element. Each glazing must have a different name.
layer_name	vector of characters	in	Array of layer names that comprises the glazing. Layers are numbered from outdoors (first layer) to indoors (last layer). These names must be in accordance with the names of the exported layers form the OPTICS program which manages the International Glazing Data Base (IGDB). Existing layers are located in the folder `materials`
layer_thickness	vector of reals	in	Thicknesses of each layer in millimeters. With the same order and dimension as layer_name.
interface_name	vector of characters	in	Array of interface names. If there is a coating between two layers (i-1 and i), write its name in the i vector position, otherwise write "". This array has the dimension of layer_name with plus one. The names of these coatings must be introduced according to the International Glazing Data Base (IGDB) nomenclature.
substrate_name	vector of characters	in	Coatings collected in the IGDB are always deposited in a substrate. This array holds the name of the substrates in which coatings are deposited.

Table 1.1: Arguments of Append_to_catalogue

1.6.2 Calculate spectral properties

```
call Calculate_spectral_properties( name )
```

The subroutine `Calculate_spectral_properties` calculates once in a life the spectral properties of a predefined glazing. This spectral properties depends on the angle of incidence and the wavelength of the solar radiation. The arguments of the subroutine are described in the following table. The compositions of the predefined glazings are defined in a document called `glazing catalog`.

Argument	Type	Intent	Description
name	character	in	Name of the predefined glazing. It creates an internal database where its spectral properties are stored.

Table 1.2: Description of Calculate_spectral_properties arguments

1.6.3 Spectral properties for a specific wavelength

```
call Spectral_properties( glazing_name, lambda, N, absorptances,      &
                          absorptances_back,absorptances_I,           &
                          absorptances_I_back, T, T_back, R, R_back )
```

The subroutine `Spectral_properties` returns the spectral properties of a glazing for a specific wavelength with normal incidence angle. This subroutine interpolates the spectral values from an internal database for a given wavelength. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Glazing name which has been already defined.
lambda	real	in	Given wavelength for the spectral properties.
Nl	integer	out	Number of layers of the glazing.
absorptances	vector of reals	out	Front absorptances of each layer.
absorptances_back	vector of reals	out	Back absorptances of each layer.
absorptances_I	vector of reals	out	Front absorptances of each interface.
absorptances_I_back	vector of reals	out	Back absorptances of each interface.
T	real	out	Front thermal transmittance of the glazing.
T_back	real	out	Back thermal transmittance of the glazing.
R	real	out	Front reflectance of the glazing.
R_back	real	out	Back reflectance of the glazing.

Table 1.3: Arguments of Spectral_properties

1.6.4 Averaged spectral properties

```
call Averaged_spectral_properties( glazing_name, theta, Nl, absorptances,    &
                                   absorptances_back, absorptances_I,        &
                                   absorptances_I_back, T,                   &
                                   T_back, R, R_back,Tv, T_back )
```

The subroutine `Averaged_Spectral_properties` returns the spectral properties of a glazing already defined and calculated. This subroutine interpolates the spectral values from an internal database for a given angle of incidence. The absorptances are wavelength averaged to determine the fraction of the impinging beam radiation which is absorbed in each layer. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Glazing name which has been already defined.
theta	real	in	Angle of incidence in radians. If the beam radiation is normal to he glazing, the angle should be zero.
Nl	integer	out	Number of layers of the glazing.
absorptances	vector of reals	out	Front absorptances of each layer.
absorptances_back	vector of reals	out	Back absorptances of each layer.
absorptances_I	vector of reals	out	Front absorptance of each interface.
absorptances_I_back	vector of reals	out	Back absorptance of each interface.
T	real	out	Front thermal transmittance of the glazing.
T_back	real	out	Back thermal transmittance of the glazing.
R	real	out	Front reflectance of the glazing.
R_back	real	out	Back reflectance of the glazing.
Tv	real	out	Front visual transmittance of the glazing.
Tv_back	real	out	Back visual transmittance of the glazing.

Table 1.4: Arguments of Averaged_Spectral_properties

1.6.5 Averaged absorptances

```
call Averaged_absorptances( glazing_name, theta, Nl, absorptances,    &
                            absorptances_back, absorpances_I,         &
                            absorptances_back_back, T,                &
                            T_back, R, R_back, Tv, Tv_back)
```

This subroutine can be used to be called by some thermal problem. The subroutine `Averaged_absorptances` returns the spectral properties of a glazing already defined and calculated.

The difference with the preceding subroutine `Averaged_Spectral_properties` is related to the treatment of thin layers. A layer whose thickness is less then 3 *mm* is considered as an interface for the thermal problem in order to avoid numerical instabilities in the non–steady simulation. In this case, it is assumed that the absorptance of the interface is equal to the absorptance of this layer. Generally, the layers which are considered as interfaces for the thermal problem are PVB layers with thicknesses varying from 380 μm to 1.5 *mm*.

This subroutine interpolates the spectral values from an internal database for a given angle of incidence. The absorptances are wavelength averaged to determine the fraction of the impinging radiation which is absorbed in each layer. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Predefined glazing name.
theta	real	in	Angle of incidence in radians. Zero if it is normal.
Nl	integer	out	Number of layers of the glazing.
absorptances	vector of reals	out	Front absorptances of each layer.
absorptances_back	vector of reals	out	Back absorptances of each layer.
absorptances_I	vector of reals	out	Front absorptances of each interface.
absorptances_I_back	vector of reals	out	Back absorptances of each interface.
T	real	out	Front thermal transmittance of the glazing.
T_back	real	out	Back thermal transmittance of the glazing.
R	real	out	Front reflectance of the glazing.
R_back	real	out	Back reflectance of the glazing.
Tv	real	out	Front visual transmittance.
Tv_back	real	out	Back visual transmittance.

Table 1.5: Arguments of Averaged_absorptances

1.6.6 Averaged diffuse properties

```
call Averaged_diffuse_properties( glazing_name, Nl, absorptances,          &
                                  absorptances_back, absorptances_I,       &
                                  absorptances_I_back, T, T_back, R, R_back )
```

This subroutine returns the diffuse spectral properties. The spectral properties are averaged for different wavelengths and for different angles of incidence to give the diffuse spectral properties of a glazing. This subroutine can be used to provide inputs for the thermal problem. The same considerations stated in the subroutine `Averaged_absorptances` related to thin layers apply. Hence, diffuse layers with thicknesses smaller than 3 *mm* are treated as interfaces.

The diffuse spectral properties are constant values which depend on the glazing composition. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Glazing name which has been already defined.
Nl	integer	out	Number of layers of the glazing.
absorptances	vector of reals	out	Front diffuse absorptances of each layer.
absorptances_back	vector of reals	out	Back diffuse absorptances of each layer.
absorptances_I	vector of reals	out	Front diffuse absorptances of each interface.
absorptances_I_back	vector of reals	out	Back diffuse absorptances of each interface.
T	real	out	Front thermal diffuse transmittance of the glazing.
T_back	real	out	Back thermal diffuse transmittance of the glazing.
R	real	out	Front diffuse reflectance of the glazing.
R_back	real	out	Back diffuse reflectance of the glazing.

Table 1.6: Arguments of Averaged_diffuse_absorptances

1.6.7 Glazing database query

```
call Glazing_database_query( glazing_name, Nl, layer_name, layer_thickness, &
                            Angle_of_incidence, Transmittance )
```

The subroutine `Glazing_database_query` returns the layout of a glazing already defined, returning its layers and the thickness of each layer. Layers or interfaces whose thickness is less than 3 *mm* are not taken into account. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Predefined glazing name.
Nl	integer	out	Number of layers of the glazing.
layer_name	vector of characters	out	Names of each layer. Exterior to interior.
layer_thickness	vector of reals	out	Thicknesses of each layer in millimeters.
Angle_of_incidence	real	in (optional)	Angle of sun incidence in radians. An angle of 0 means that the sun is normal to the glazing.
Transmittance	real	out (optional)	Front thermal transmittance of the glazing.

Table 1.7: Description of Glazing_database_query arguments

1.6.8 Far infrared emissivities

```
call FIR_emissivities( glazing_name, epsilon_FIR )
```

The subroutine `FIR_emissivities` returns the resultant emissivities of each layer. An air or gas chamber is delimited by two parallel glass panes with different coatings. The radiative interchange between these two glass panes are given by the following expression:

$$q_r = \epsilon_R \sigma (T_1^4 - T_2^4)$$

where T_1 and T_2 are the surface temperatures of each glass pane and the resulting emissivity is obtained by:

$$\frac{1}{\epsilon_R} = \frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1$$

with ϵ_1 and ϵ_2 being the emissivities corresponding the the specific coatings of these two glass panes. This subroutine calculates ϵ_R for each layer. If the layer is not a gas layer, this value is set to zero. The arguments of the subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Predefined glazing name.
epsilon_FIR	vector of reals	out	Resultant emissivities of each layer. From the outermost layer to the innermost layer.

Table 1.8: Description of FIR_emissivities arguments

Chapter 2

Thermal problem

2.1 Overview

This library can be used to solve the thermal problem of a water flow glazing. These subroutines allow to obtain the thermal profile of a multilayer glazing with glass panes, air and water chambers. There are two different models: complete and simplified. The complete model takes into account the thermal mass of every component. If thermal mass of the glazing is negligible, the simplified model involves an algebraic system of equations. This system allows to obtain the thermal profile of the glazing very quickly.

2.2 Complete model for WFG

The complete model is encapsulated in the module `API_Thermal_problem_active_wall`. The API of the thermal problem for the complete model comprises the following public subroutines:

```
    public ::  Create_active_wall            ! it creates an active wall
    public ::  Thermal_problem_active_wall   ! it evolves the thermal problem
```

Listing 2.1: `API_Thermal_problem_active_wall.f90`

Once initial condition, outdoor and indoor heat fluxes are given, this library yields the next temperature profile for a given time step. By means of successive calls of this routine, a temporal evolution of the temperatures of the glazing is obtained. The outdoor and indoor heat fluxes, the impinging solar radiation, the flow rate of the circulating water chamber and the inlet temperature are given as boundary conditions.

Once this subroutine is called, it returns the new temperature profile, the energy transmittance and the outlet temperature of the water chamber.

Before this subroutine is used, the subroutine `Create_WFG` must be called to create internally a new glazing. This subroutine is called with a predefined glazing name of the catalogue and it allocates the temperature vector of the glazing and the set of nodes or grid points where these temperatures are defined.

This library is encapsulated in the module `API_Thermal_Problem_WFG`. To use this library, the sentence `use API_Thermal_Problem` should be included at the beginning of the FORTRAN code.

```
Program Example

  use API_Thermal_problem_active_wall
  implicit none

     call Example_API_Thermal_Problem_WFG

end program
```

This example allows to reach the steady state from the zero initial condition (Temperature = 0). This subroutine integrates a set of partial differential equations taking into account solar absorptances in glass layers or water layers with no negligible thermal mass. To reach the steady state, the problem iterates $N = 100$ time steps of $Dt = 300$ seconds. Hence, the computational effort of this simulation is high.

```
subroutine  Example_API_Thermal_Problem_WFG

  integer :: Nv                                 ! number of grid points
  real, allocatable :: x(:), Temperature(:)     ! Temperature profile


  real ::  T,  R, Rb, T_d, Tb_d, R_d, Rb_d      ! spectral properties
                                                !(beam, diffuse, front and back)

  real :: T_OUT, Power                  ! outlet temperature and water heat gain

  integer :: i, N = 100                 ! 100 steps of Dt = 300 seconds
  real :: q_i, q_e, T_e = 35, T_i = 28, h_e = 23, h_i = 8

  call Create_active_wall(                                                      &
     glazing_name = "mHeatGlass", Temperature = Temperature, x = x,            &
     T_diffuse = T_d, Tb_diffuse = Tb_d, R_diffuse = R_d, Rb_diffuse = Rb_d )

  Nv = size(Temperature)
  Temperature = 0.


  do i=0, N
      write(*,'(a, f6.0, a, $)' ) " % of simulation = ",                       &
                                  i/real(N)*100.0, CHAR(13)

      q_e = h_e * ( T_e - Temperature(1) )
      q_i = h_i * ( Temperature(Nv) - T_i )

      call Thermal_problem_active_wall(                                         &
           glazing_name = "mHeatGlass",  Dt = 300.0,                            &
           I_beam = 800., I_diffuse = 0.0, J_beam = 0.0, J_diffuse = 0.0,       &
           Inclination = 90.0 , Angle_of_incidence = 0.0,                       &
           q_e = q_e , q_i = q_i, flow_rate = 2./60, T_inlet= 20.0,             &
           T_outlet = T_OUT, Water_heat_gain = Power,                           &
           Temperature = Temperature,  x = x,                                   &
           Transmittance = T, Reflectance = R, Reflectance_back = Rb )
  enddo

  write(*,*)
  write(*,'(a,2f5.1)') "Power = ", Power, T_OUT
  write(*,'(a,f5.1, i4)') "T1 = ", Temperature(1),  size(x)

end subroutine
```

Listing 2.2: API_Example_Thermal_problem.f90

2.3 Simplified model for WFG

The simplified model is encapsulated in the module `API_Thermal_problem_WFG_simplified`. The API of the thermal problem for the complete model comprises the following public subroutines:

```
   public ::  Thermal_problem_WFG_simplified
```

Listing 2.3: `API_Thermal_problem_WFG_simplified.f90`

To use this library, the sentence `use API_Thermal_Problem_simplified` should be included at the beginning of the FORTRAN code.

```
Program Example_simplified

   use API_Thermal_Problem_simplified
   implicit none

       call Example_API_Thermal_Problem_WFG_simplified

end program
```

```
 subroutine  Example_API_Thermal_Problem_WFG_simplified

   real, allocatable :: x(:), Temperature(:)  ! Temperature profile
   real :: T, R, Rb, T_d, Tb_d, R_d, Rb_d    ! spectral properties
   real :: T_OUT, Power               ! outlet temperature and water heat gain

   call Create_active_wall(                                             &
     glazing_name = "mHeatGlass", Temperature = Temperature, x = x,     &
     T_diffuse = T_d, Tb_diffuse = Tb_d, R_diffuse = R_d, Rb_diffuse = Rb_d )

   Temperature = 0.

   call Thermal_problem_WFG_simplified(                                 &
     glazing_name = "mHeatGlass",                                       &
     I_beam = 800., I_diffuse = 0.0, J_beam = 0.0, J_diffuse = 0.0,     &
     Inclination = 90.0 , Angle_of_incidence = 0.0,                     &
     q_e = 100.0 , q_i = 50.0, T_e = 35., T_i = 28., h_e = 23., h_i = 8.,  &
     flow_rate = 2./60, T_inlet= 20.0, T_outlet = T_OUT,                &
     Water_heat_gain = Power, Temperature = Temperature, x = x,         &
     Transmittance = T, Reflectance = R, Reflectance_back = Rb          )

   write(*,'(a,2f5.1)') "Power = ", Power, T_OUT
   write(*,'(a,f5.1, i4)') "T1 = ", Temperature(1), size(x)

 end subroutine
```

Listing 2.4: `API_Example_Thermal_problem.f90`

Given constant outdoor and indoor conditions, the steady state is reached by solving a set of energy balance equations. The steady temperature profile is saved in the variable `Temperature`.

2.4 Application program interface

This chapter is about the thermal problem of an isolated water flow glazing. There are two modules related to this:

- `API_Thermal_problem_WFG`.
- `API_Thermal_problem_WFG_simplified`.

The first one deals with complete problem and need initialization. It has three subroutines: `Convective_models`, `Create_WFG` and `Thermal_problem_WFG`.

The second module does not need initialization and only contains the subroutine `Thermal_problem_WFG_simplified`. Both treat thin layers as interfaces to avoid numerical instabilities.

- `API_Thermal_problem_WFG:` This module has been devised for the resolution of the non-steady thermal problem of WFGs. It solves a system of partial differential equations and takes into account solar absorptances in glass and water layers with no negligible thermal mass. The computational effort is high.
  - `Convective_models.` Defines the convection models for water and gas chambers.
  - `Create_WFG.` Initializes the thermal problem by creating it as an internal object.
  - `Thermal_problem_WFG.` Solves the thermal problem for the next time step.
- `API_Thermal_problem_WFG_simplified:` This module has been devised for the resolution of the steady thermal problem of WFGs, as it is quicker and less accurate in transitions. It does not take into account thermal mass of the layers, and for that reason an algebraic system of equations governs the problem.
  - `Thermal_problem_WFG_simplified.` Solves the simplified thermal problem for the next time step.

2.4.1 Convective models

```
call Convective_models( Convection_model , h_g, h_w )
```

This subroutine is intended to initialize the convective model for the water chamber and for the gas chamber. If "CONSTANT" model is selected, the convective coefficient must be given.

Argument	**Type**	**Intent**	**Description**
Convection_model	character	in	Convection model: `CONSTANT, ISO`.
h_g	real	in (optional)	Gas convective coefficient. $[W/K]$
h_w	real	in (optional)	Water convective coefficient. $[W/K]$

Table 2.1: Description of the Convective_models arguments

2.4.2 Create WFG

```
call Create_WFG( glazing_name, Temperature, x, T_diffuse,  &
                 Tb_diffuse, R_diffuse, Rb_diffuse )
```

The subroutine `Create_WFG` allocates internally a glazing by means of a predefined name of a glazing catalogue and returns an identifier to be simulated by means of the subroutine `Thermal_problem_WFG`. It also allocates the temperature vector of the glazing and the set of nodes or grid points where these temperatures are defined. The arguments of this subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Input name of a predefined glazing catalogue: ReferenceGlass, iThermGlass, HeatGlass, RadiaGlass, iThermGlassX, HeatGlassX, RadiaGlassX
Temperature	allocatable vector of reals	out	This subroutine allocates this vector with a dimension defined internally.
x	allocatable vector of reals	out	Grid points where these temperatures are defined. This subroutine allocates this vector and initializes their components in millimeters with the origin at the outermost glass pane.
T_diffuse	real	out	Front diffuse thermal transmittance of the glazing. It is a constant property which depends on the glazing composition.
Tb_diffuse	real	out	Back diffuse thermal transmittance of the glazing. It is a constant property which depends on the glazing composition.
R_diffuse	real	out	Front diffuse reflectance of the glazing.
Rb_diffuse	real	out	Back diffuse reflectance of the glazing.

Table 2.2: Description of Create_WFG arguments

The user can also define a new glazing using the subroutine `Append_to_catalogue` which is explained in the API Manual for the Spectral Problem.

2.4.3 Thermal problem WFG

```
call Thermal_problem_WFG( glazing_name, Dt, I_beam, I_diffuse, J_beam,     &
                          J_diffuse, Inclination,Angle_of_incidence,       &
                          q_e, q_i, Flow_rate, T_inlet,T_outlet,           &
                          Water_heat_gain, Temperature, x, Transmittance,  &
                          Reflectance, Reflectance_back                     )
```

Given an inital temperature profile and the required boundary conditions, this subroutine yields the non-steady temperature profile for the next time step. The arguments of this subroutine are described in the following table.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Predefined glazing name.
Dt	real	in	Time step in seconds.
I_beam	real	in	Outdoor beam solar radiation normal to the glazing. $[W/m^2]$
I_diffuse	real	in	Outdoor diffuse solar radiation. $[W/m^2]$
J_beam	real	in	Indoor beam radiation normal to the glazing. $[W/m^2]$
J_diffuse	real	in	Indoor diffuse irradiance.$[W/m^2]$
Inclination	real	in	Angle of inclination of the WFG (degrees).
Angle_of_incidence	real	in	Angle of sun incidence in radians. An angle of 0 means that the sun is normal to the glazing.
q_e	real	in	Outdoor heat flux.$[W/m^2]$
q_i	real	in	Indoor heat flux.$[W/m^2]$
Flow_rate	real	in	Flow rate in the water chamber. $[kg\ s^{-1}m^{-2}]$
T_inlet	real	in	Inlet water temperature. $[^oC]$
T_outlet	real	out	Outlet water temperature. $[^oC]$
Water_heat_gain	real	out	Water heat gain. $[W/m^2]$
Temperature	vector of reals	inout	Updated temperature profile. It should be previously allocated by Create_Glazing.
x	vector of reals	out	Grid points where temperatures are defined.
Transmittance	real	out	Transmittance of the glazing.
Reflectance	real	out	Front reflectance of the glazing.
Reflectance_back	real	out	Back reflectance of the glazing.

Table 2.3: Description of Thermal_problem_WFG arguments

2.4.4 Simplified thermal problem

```
call Thermal_problem_WFG_simplified(                                    &
        glazing_name, I_beam, I_diffuse, J_beam, J_diffuse, Inclination, &
        Angle_of_incidence,  q_e, q_i, T_e, T_i, h_e, h_i, flow_rate,    &
        T_inlet, T_outlet, Water_heat_gain, Temperature, x,              &
        Transmittance, Reflectance , Reflectance_back )
```

Given an initial temperature profile and the required boundary conditions, this subroutine yields the non-steady temperature profile for the next time step. For the use of this subroutine it is not needed to use the `Create_WFG` subroutine. This subroutine does not take into account the thermal mass of the layers.

Argument	**Type**	**Intent**	**Description**
glazing_name	character	in	Predefined glazing name.
I_beam	real	in	Outdoor beam solar radiation normal to the glazing. $[W/m^2]$
I_diffuse	real	in	Outdoor diffuse solar radiation. $[W/m^2]$
J_beam	real	in	Indoor beam radiation normal to the glazing. $[W/m^2]$
J_diffuse	real	in	Indoor diffuse irradiance.$[W/m^2]$
Inclination	real	in	Angle of inclination (degrees).
Angle_of_incidence	real	in	Angle of sun incidence (rad).
q_e	real	in	Outdoor heat flux.$[W/m^2]$
q_i	real	in	Indoor heat flux.$[W/m^2]$
T_e	real	in	Outdoor temperature.$[^oC]$
T_i	real	in	Indoor temperature.$[^oC]$
h_e	real	in	Indoor transport coefficient.$[W/m^2$ K$]$
h_i	real	in	Outdoor transport coefficient.$[W/m^2$ K$]$
Flow_rate	real	in	Flow rate in the water chamber. $[kg\ s^{-1}m^{-2}]$
T_inlet	real	in	Inlet water temperature. $[^oC]$
T_outlet	real	out	Outlet water temperature. $[^oC]$
Water_heat_gain	real	out	Water heat gain of the glazing. $[W/m^2]$
Temperature	vector of reals	inout	Updated temperature.
x	vector of reals	out	Grid points.
Transmittance	real	out	Transmittance of the glazing.
Reflectance	real	out	Front reflectance of the glazing.
Reflectance_back	real	out	Back reflectance of the glazing.

Table 2.4: Description of Thermal_problem_WFG_simplified arguments

Chapter 3

Outdoor conditions

3.1 Overview

The outdoor climate conditions can be obtained by using the subroutines encapsulated in this module. These allows you to obtain outdoor solar irradiance and temperature given a date and an hour. This subroutines are:

The outdoor climate conditions are encapsulated in the module `API_Outdoors`. The API comprises the following public subroutines:

```
    public :: Outdoor_initialization    ! select city and EPW weather file
    public :: Outdoor_conditions        ! given the actual time,
                                        ! it gives external weather conditions

    public :: Sun_beam_direction        ! it gives the unitary solar beam
    vector
    public :: Sun_Position              ! given the actual time,
                                        !  it gives zenith, azimuth of the sun
    public :: Sun_projection            ! it projects sun rays in a given plane
    public :: date2number, number2date ! it converts character date to seconds
```

Listing 3.1: `API_Outdoors.f90`

The first subroutine `Outdoor_initialization` allows to select the city to obtain the specific weather conditions as well as the external heat transport coefficients. Given the outdoor temperature and the surface temperature, `Outdoor_conditions` determines the outdoor heat flux.

3.2 Outdoor conditions

Once the city is selected and the `time`, measured in seconds from the first of January, is given, external temperature, solar irradiance heat fluxes projected in an specific surface given by their orientation and inclination are obtained. In the following example, weather conditions on February 1th, Madrid at 14:30 hours are calculated.

```fortran
subroutine  Example_Madrid_outdoor_conditions

  real :: PI = 4*atan(1.)
  real :: time, T_s = 20.0, Inclination = 90., Orientation = 180., T_e, T_ne
  real ::  q_e, h_e, I_beam, I_diffuse, Angle_of_incidence, I_b0, I_d0
  real :: Zenith, Azimuth


     call Outdoor_initialization(                                        &
                    emissivity = 0.9, length = 3.0,                      &
                    model = "ISO", h_e = 23.0, location = "Madrid"    )

     time = date2number("02-01T13:30", 2017)

     call Outdoor_conditions(                                                 &
          time, Inclination, Orientation, T_s, T_e, T_ne, h_e, q_e, I_beam,  &
          I_diffuse, Angle_of_incidence, I_b0, I_d0, Zenith, Azimuth         )


     write(*,*)  " time =", time, T_e, T_ne, q_e, h_e
     write(*,*)  " I_beam =", I_beam, I_diffuse, I_b0, I_d0
     write(*,*)  " Inclination = ",  Inclination, Orientation,               &
                                     180*Angle_of_incidence/PI
     write(*,*)  " Zenith= ", Zenith," Azimuth= ", Azimuth

end subroutine
```

Listing 3.2: API_Example_Thermal_problem.f90

To calculate the variable time in seconds, it is used the subroutine date2number which allows to calculate from a date string and the current year the time in seconds.

3.3 Sun beam direction

The solar beam direction should be known in order to determine indoor solar distribution. The beam solar direction can be obtained by using the following subroutine:

In the following example, latitude and longitude are given to calculate the direction of the beam solar radiation on February 1th at 14:30 hours.

```
subroutine  Example_sun_beam_direction

  real :: time, Latitude = 40.0, Longitude = 0.0, Beam_direction(3)
  integer :: Year = 2017, GMT = 0

     time = date2number("02-01T14:30", Year)

     call Sun_beam_direction( Year, time, Latitude,                   &
                              Longitude, GMT, Beam_direction          )

write(*,'(" Latitude =", F8.1," Longitude=", F8.1)' ) Latitude, Longitude
write(*,'(" V_beam=", 3F8.1)' ) Beam_direction

end subroutine
```

Listing 3.3: `API_Example_Thermal_problem.f90`

3.4 Sun position

Given latitude, longitude and actual time, elevation and azimuth of the sun can be obtained by calling the subroutine `Sun_position`. In the following example, approximate and accurate zenith and azimuth of the sun is calculated for three different cities.

```
 subroutine Test_accurate_sun_position

    character(len=11) :: idate = '02-22T12:00' ! GMT
    real :: Time, Longitude, Latitude
    integer:: Year = 1989
    character(len=20) :: city(3) =                                    &
                      [character(len=20) :: 'Madrid','Sofia','Frankfurt']

    real :: Zenith, Azimuth, desired_Zenith, desired_Azimuth
    integer :: i,GMT  = 0

    Time = date2number(idate, Year )
    write(*,*) '------------------------------------------------------- '
    write(*,*) 'Exact results at 12:00:00 on February 22nd of 1989      '
    write(*,*) '------------------------------------------------------- '

do i = 1, 3
        write(*,*) '------city =', city(i)

        if (i == 1) then                                  ! Madrid
            Longitude = -3.703790; Latitude = 40.416775
            desired_Zenith = 50.9413; desired_Azimuth = 171.0067

        else if (i == 2) then                             ! Sofia
            Longitude = 23.32601; Latitude = 42.696552
            desired_Zenith = 55.8602; desired_Azimuth = 203.9426

        else if (i == 3) then                             ! Frankfurt
            Longitude = 8.682127; Latitude = 50.110924
            desired_Zenith = 60.3908; desired_Azimuth = 186.0087
        end if

   write(*,'(a20,a10, f10.5, a10, f10.5)') 'Exact values:',             &
           ' Zenith =', desired_Zenith , ' Azimuth =', desired_Azimuth

   call Sun_position( Year, Time, Latitude, Longitude, Zenith, Azimuth, GMT )

   write(*,'(a20,a10, f10.5, a10, f10.5)') 'Accurate results:',         &
           ' Zenith =', Zenith,    ' Azimuth =', Azimuth

   call Sun_position( time, Latitude, Longitude, Zenith, Azimuth   )

   write(*,'(a20,a10, f10.5, a10, f10.5)') 'Basic results:',            &
           ' Zenith =', Zenith,    ' Azimuth =', Azimuth
enddo

end subroutine
```

Listing 3.4: `API_Example_Thermal_problem.f90`

3.5 Sun projection

Once the solar position is known, the beam solar radiation can be projected in any surface to obtain the inputs for the thermal problem. In the following example, different projections are calculated.

```
subroutine Test_Sun_projections

   real :: time, I_b0, I_d0,  I_b, I_d, Angle
   integer :: i, j, year = 1989, GMT = 0
   integer, parameter ::  Nt = 6
   real :: Zenith, Azimuth, Orientation(4) = [ 0., 90., 180., 270.]
   character(len=20) :: ctime(Nt) =                                    &
                       [ "02-22T12:29", "02-22T14:29",  "02-22T10:29",    &
                         "02-22T08:29", "02-22T06:29", "02-22T05:29" ]
   real :: Latitude = 40.416775, Longitude = -3.703790 ! Madrid
   character(len=40) :: cf = "'(a15, f10.2, a15, f10.2)'"

    write(*, cf ) "Floor Inclination = ", 180. , " Orientation =", 0.
    write(*, cf) " Zenith = ", 10., "Azimuth = ", 20

    call Sun_projection(                                              &
       Inclination = 180., Orientation = 0., Zenith = 10., Azimuth = 20., &
       Angle_of_incidence = Angle, I_b0 = 1000., I_d0 = 0.0,          &
       I_beam = I_b, I_diffuse = I_d                                  )

    write(*, cf ) " Angle = ", Angle, " I_beam =", I_b
    write(*, cf ) " Roof Inclination = ", 180. , " Orientation =", 0.
    write(*, cf ) " Zenith = ", 10., "Azimuthexample = ", 20.
```

Listing 3.5: `API_Example_Thermal_problem.f90`

3.6 Application program interface

This chapter comprises the `Outdoor` module, which output are the exterior boundary conditions for the thermal problem of the glazing, like exterior temperature in a city at a time of the year or the position of the Sun. It holds the following subroutines and functions:

- `Subroutine Outdoor_initialization:` Needed for the use of the Outdoor_conditions subroutine.
- `Subroutine Outdoor_conditions:` Returns the exterior boundary conditions for a location and a date.
- `Function date2number:` Converts date into a readable integer for the others subroutines.
- `Subroutine Sun_beam_direction:` Returns the beam direction of the Sun as a vector.

3.6.1 Outdoor initialization

```
call Outdoor_initialization( emissivity, length, model, h_e, location )
```

To use this subroutine you must use the `API_outdoors` module. This subroutine is necessary to use `Outdoor_conditions` subroutine.

Argument	Type	Intent	Description
emissivity	real	in	Emissivity is the ratio of the energy radiated from a material's surface to that radiated from a blackbody (a perfect emitter) at the same temperature and wavelength and under the same viewing conditions.
lenght	real	in	This is the height of the surface. $[m]$
model	character	in	Input name of the chosen convection model: `CONSTANT`, `ISO`, `EnergyPlus`.
h_e	real	in (optional)	Heat convective coefficient. It is necessary only if `CONSTANT` model is chosen.
location	character	in(optional)	Name of the city. You can choose among several predefined cities (Frankfurt, Las_Palmas, Madrid, Segovia, Sofia, Soria). You also can write the name of a new .epw weather file.*

*There is a folder named "./Climate/EPW" in which weather epw-files are located. To simulate different cities or new climate data, edit the file "list.txt" to include the new city together with its weather file name and copy this file to the folder "./Cimate/EPW".

3.6.2 Outdoor conditions

```
call Outdoor_conditions( time, Inclination, Orientation, T_s, T_e, T_ne,    &
                         h_e, q_e, I_beam, I_diffuse, Angle_of_incidence,   &
                         I_b0, I_d0, Zenith, Azimuth                        )
```

To call this subroutine you must to have called previously `Outdoor_initialization`. This is a subroutine that gives you external boundary conditions for the thermal and zone problems.* The arguments of this subroutine are described in the following table.

Argument	Type	Intent	Description
time	real	in	This call gives the resulted values at this time**. [*seconds*]
Inclination	real	in	Angle of inclination of surface measured from the ground to the indoor side of surface. (90 is vertical and 180 horizontal)
Orientation	real	in	Orientation of the surface. 0:North, 90:East, 180:South and 270:West.
T_s	real	in	Temperature of the exterior surface in touch with the air. [^oC]
T_e	real	out	outdoor temperature
T_ne	real	out	
h_e	real	out	Outdoor convection coefficient
q_e	real	out	Outdoor heat flux (positive to indoors)
I_Beam	real	out	Sun beam radiation projected to the surface
I_diffuse	real	out	Sun diffuse radiation impinging into the surface
Angle_of_incidence	real	out	Angle of incidence of the impinging beam radiation
I_b0	real	out	Direct beam radiation normal to the sun
I_d0	real	out	Diffuse radiation measured on horizontal plane
Zenith	real	out	Sun's zenith angle measured from vertical (0 is vertical and 90 is horizontal).
Azimuth	real	out	Sun's azimuth angle measured clockwise from North.

Table 3.1: Description of the Outdoor_conditions arguments

** The "time" variable must be the number of seconds of the year. Its value can be obtained using the "date2number" function as in the API example.

3.6.3 Data to time stamp number

```
date2number( time_string, year)
```

This is a function that convert the ”time string” to the corresponding number of seconds of the year. It results a float variable. The argument of this function is described bellow:

Argument	**Type**	**Intent**	**Description**
time_string	character	in	This variable is the date and time required. It must be written with the form 'month,day;hour:minute', each variable has to have to numbers. This character variable must be length 11. As in the example: '08/04T08:30'
year	integer	in	Year of the date.

Table 3.2: Description of date2number arguments

3.6.4 Climate data

```
call Climate_data( time, T_dry, T_dew, Direct_Normal_Radiation,
    Diffuse_Horizontal_Radiation, &
Horizontal_IR, W_speed, W_direction, Air_pressure, f_clr)
```

This subroutine provides the climate data information from the weather file: temperatures (dry and dew point), solar beam radiation and the diffuse one, horizontal infrared radiation and also the wind direction and velocity. All this values are given in a concrete date and time.

The arguments of this subroutine are described in the following table:

Argument	**Type**	**Intent**	**Description**
time	real	in	This call gives the resulted values at this time*. [*seconds*]
T_dry	real	out	Dry-bulb temperature. It is the temperature of air measured by a thermometer freely exposed to the air but shielded from radiation and moisture. [^oC]
T_dew	real	out	Dew point temperature. It is the temperature at which a given concentration of water vapor in air will form dew It could be seen as a measure of atmospheric moisture. [^oC]
Direct_Normal_Radiation	real	out	Direct normal beam radiation from the Sun. [W/m^2]
Diffuse_Horizontal_Radiation	real	out	Horizontal diffuse radiation from the exterior. [W/m^2]
Horizontal_IR	real	out	Horizontal infrared radiation from the exterior. [W/m^2]
W_speed	real	out	Wind speed value. [m/s]
W_direction	real	out	Wind direction. Taken the origin on the North direction and clockwise turning sense. [*degrees*]
Air_pressure	real	out	Outdoor air pressure. [Pa]
f_clr	real	out	Factor of the clearness of the sky. (1.0 means the clearest sky conditions)

Climate data are obtained from .EPW (EnergyPlus Weather) files. These data comprise weather records for a whole year in a precise location. The format of this EPW files is decribed below. The first eight lines of this file constitute the header with the following information:

- LOCATION: city, State Province Region, Country, Source, WMO (6digits), Latitude, Longitude, TimeZone (relative to GMT), Elevation (m).
- DESIGN CONDITIONS: Number of Design Conditions, Design Condition Source, Design Condition Type (HEATING), Design Condition Type (COOLING).
- TYPICAL/EXTREME PERIODS: Number of Typical/Extreme Periods, Typical/Extreme Period 1 Name, Typical/Extreme Period 1 Type, Period Start Day, Period 1 End Day
- GROUND TEMPERATURES: Number of Ground temperature Depths (m), Depth 1 Soil Conductivity (W m^{-1} k^{-1}), Depth 1 Soil Density (kg/m^3), Depth 1 Soil Specific Heat(J kg^{-1} k^{-1}), 12 x Depth 1 every month Average Ground Temperature (^{o}C) [This could be repeated for several depth measurements]
- HOLIDAYS/DAYLIGHT SAVING: LeapYear Observed (Yes/No), Daylight Saving Start Day, Daylight Saving End Day, Number of holidays,[Holiday 1 Name, Holiday 1 Day, Holiday 2 Name, Holiday 2 Day, ...]
- COMMENTS 1: [It may be .EPW file author's information, some clarification about the measurements, etc]
- COMMENTS 2: [Same as "COMMENTS 1".]
- DATA PERIODS: Number of Data Periods, Number of Records per hour,[Data Period 1 Name/Description, Data Period 1 Start Day of Week, Start Day, End Day, Data Period 2 Name/Description, ...]

Then, the following lines represent measurements every half an hour and their columns are organized as follows:

Column number	Data	Units
1	Year	-
2	Month	-
3	Day	-
4	Hour	-
5	Minute	-
6	Data Source and Uncertainty Flags	-
7	Dry Bulb Temperature	^oC
8	Dew Point temperature	^{o}C
9	Relative Humidity	-
10	Atmospheric Station Pressure	Pa
11	Extraterrestial Horizontal Radiation	Wh/m^2
12	Extraterrestial Direct Normal Radiation	Wh/m^2
13	Horizontal Infrarred Radiation from Sky	Wh/m^2
14	Global Horizontal Radiation	Wh/m^2
15	Direct Normal Radiation	Wh/m^2
16	Diffuse Horizontal Radiation	Wh/m^2
17	Global Horizontal Radiation	lux
18	Direct Normal Illuminance	lux
19	Diffuse Horizontal Illuminance	lux
20	Zenith Illuminance	Cd/m^2
21	Wind direction	deg
22	Wind speed	m/s
23	Total Sky Cover	-
24	Opaque Sky Cover	-
25	Visibility	km
26	Ceiling Height	m
27	Present Weather Observation	-
28	Present Weather Codes	-
29	Precipitable Water	mm
30	Aerosol Optical Depth	thousandths
31	Snow depth	cm
32	Days Since Last Snowfall	-

3.6.5 Sun beam direction

```
call Sun_beam_direction( Year, time, Latitude, Longitude, add_GMT,
     Beam_direction )
```

This subroutine allows to determine the sun position from the time of year, the latitude and the longitude of some specific location.

Argument	Type	Intent	Description
Year	integer	in	Year to calculate precisely to sun position
time	real	in	Time in seconds from January 1th
Latitude	real	in	Latitude of the location.
Longitude	real	in	longitude of the location.
add_GMT	integer	in (optional)	Time zone GMT
Beam_direction	vector of reals	out	Direction of the beam

Table 3.3: Description of the Sun_beam_direction arguments

3.6.6 Sun position

```
call Sun_position( Year, Time, Latitude, Longitude, Zenith, Azimuth, add_GMT )
```

This subroutine allows to determine the sun position from the time of year, the latitude and the longitude of some specific location.

Argument	Type	Intent	Description
Year	integer	in	Year to calculate precisely to sun position
time	real	in	Time in seconds from January 1th
Latitude	real	in	Latitude of the location.
Longitude	real	in	longitude of the location.
Zenith	real	out	Sun's zenith angle measured from vertical (0 is vertical and 90 is horizontal).
Azimuth	real	out	Sun's azimuth angle measured clockwise from North.
add_GMT	integer	in (optional)	Time zone GMT (Greenwich Mean Time)

Table 3.4: Description of the Sun_beam_direction arguments

Chapter 4

Thermal problem of insulated zones

4.1 Overview

Insulated zones are formed by opaque or transparent walls. This library can calculate interior parameters of a defined insulated zone like for example the interior temperature.

The indoor thermal problem is encapsulated in the module `API_Insulated_zone`. The API comprises the following public subroutines:

```
    public :: Create_Zone
    public :: Thermal_problem_zone
```

Listing 4.1: `API_Insulated_zone.f90`

The first subroutine `Create_zone` allows to define the dimensions of the zone as well as its spectral and thermal characteristics. Once the zone is defined, the thermal problem can be solved.

4.2 Zones

Each zone is defined by its dimmensions: height, length and width and six transparent or opaque walls. When the zone is created, a label for each wall allows to differentiate between “opaque” and transparent or glazing facades. If a transparent wall is considered, its label represents some name of the glazing catalog. The rest of opaque walls are characterized by their optical and thermal parameters.

The optical parameters are:

- ρ_{NIR}: reflectivity of inner surface of walls for near infrared wavelengths.
- ρ_{FIR}: reflectivity of inner surface of walls for far infrared wavelengths.

The thermal parameters are:

- U: thermal transmittance of each wall.
- σ: internal thermal mass per squere meter of each wall.
- c: specific heat of thermal mass. (J/(Kg K)).
- h_i: interior heat transfer coefficient (W/() m^2 K).

In the following example, a zone of $7 \times 7 \times 3$ meters is created with a “HeatGlass” glazing facing south.

```
subroutine Example_thermal_problem_zone

    real, allocatable ::  Temperature(:,:), Temp(:), x(:)
    real :: T(6), R(6), Rb(6), T_d, Tb_d, R_d, Rb_d
    real :: I_b0 = 800., I_d0 = 0.,  Dt, time
    real :: Q_i(6), Q_e(6), G_diffuse(6), T_a, T_inlet = 15., T_outlet, P
    character(len=30) :: o = "opaque", g = "mHeatGlass"
    integer :: Nv

     call Create_zone(                                                  &
             Height = 3.0, Length = 7.0, Width = 7.0,                   &
             south_orientation = 0., roof_inclination = 0.,             &
             r_NIR   = [ 0.6, 0.6, 0.6, 0.6, 0.6, 0.6 ],                &
             r_FIR   = [ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1 ],                &
             U_value = [ 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 ],                &
             mass_m2 = [ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 ],                &
             wall_name=[ o, o, g, o, o, o ],                            &
             c = 780. ,  transfer_coefficient = 8.0,                    &
             model = "CONSTANT", wall_model = "simplified"              )
```

Listing 4.2: `API_Example_Thermal_problem.f90`

4.3 Thermal problem of zones

If the outdoor conditions such as the outdoor temperature, beam and diffuse solar radiation are given, the thermal indoor problem is calculated by calling the subroutine `Thermal_problem_zone`.

```
    call Thermal_problem_zone(                                           &

            Dt = 100.0 , Q_air = 0.0, T_e = 10.0,                        &
            h_e = 8.0, I_b0 = I_b0, I_d0 = I_d0,                         &
            Zenith = 90., Azimuth = 180.0,                               &
            T_inlet = T_inlet,  flow_rate = 2/60.,                       &
            Temperature = Temperature, x = x,                            &
            Q_indoors = Q_i, Q_outdoors = Q_e,                           &
            G_diffuse = G_diffuse, T_a = T_a,                            &
            T_outlet = T_outlet, Power = P,                              &
            Transmittance = T, Reflectance = R, Reflectance_back = Rb    )
```

Listing 4.3: `API_Example_Thermal_problem.f90`

The inlet temperature is set to 15 Celsius degrees and the mass flow rate to 2 liters per square meter per minute. The outdoor conditions are constant and by calling `Thermal_problem_zone` the steady state is calculated in one iteration because no internal mass nor glazing mass are considered. It is considered a constant beam radiation impinging normally on south glazing.

4.4 Application program interface

This chapter comprises the `API_Insulated_zone` module, which holds the subroutines `Create_Zone` and `Thermal_problem_zone`. These are used for the thermal resolution of a zone surrounded with six walls or glazings.

- `Create_Zone`: Initializes the insulated zone to use Thermal_problem_zone subroutine.
- `Thermal_problem_zone`: Solves the insulated room problem, and calculates the interior temperature.

4.4.1 Create zone

```
call Create_Zone( Height, Length, Width, south_orientation,          &
                  roof_inclination, r_NIR, r_FIR, U_value, mass_m2,    &
                  wall_name, c, transfer_coefficient, model )
```

To use this subroutine you must to use the `API_Insulated_zone` module. This subroutine is necessary to use `Thermal_problem_zone` subroutine. It is supposed a zone with opaque or transparent walls.

Argument	**Type**	**Intent**	**Description**
Height	real	in	Glazing height $[m]$.
Length	real	in	Room length $[m]$.
Width	real	in	Room width $[m]$.
south_orientation	real	in	Orientation of the southern surface.
roof_inclination	real	in	Angle of inclination of the roof measured from the ground. $[degrees]$
r_NIR	vector of reals	in	NIR reflectance of each surface.
r_FIR	vector of reals	in	FIR reflectance of each surface.
U_value	vector of reals	in	Thermal transmittance coefficient of each surface $[W/(m^2K)]$.
mass_m2	vector of reals	in	Mass of each surface. $[W]$
wall_name	vector of characters	in	Names of the walls. It defines whether is ”opaque” or a glazing from the catalogue.
c	real	in	Specific heat coefficient of the envelope walls $[J/(KgK)]$.
transfer_coefficient	real	in	Indoor heat transfer coefficient (h) $[W/(m^2K)]$.
model	character	in	Convection model chosen among: `CONSTANT, ISO, EnergyPlus.`

4.4.2 Themal problem of some zone

```
call Thermal_problem_zone( Dt, Q_air, T_e, h_e, I_b0, I_d0, Zenith, Azimuth, &
                           T_inlet, flow_rate, Temperature, x, Q_indoors,    &
                           Q_outdoors, G_diffuse, T_a, T_outlet, Power,      &
                           Transmittance, Reflectance, Reflectance_back  )
```

Argument	Type	Intent	Description
Dt	real	in	Time step for the integration.
Q_air	real	in	Heat flux associated to air renovation or HVAC equipments.
T_e	real	in	Outdoor air temperature
h_e	real	in	Outdoor convection coefficient.
I_b0	real	in	Direct beam radiation normal to the sun
I_d0	real	in	Diffuse radiation measured on horizontal plane
Zenith	real	in	Sun zenith angle (0 is vertical)
Azimuth	real	in	Azimuth angle of the Sun measured clockwise from North.
T_inlet	real	in	Temperature
flow_rate	real	in	Flow rate in the water chamber. $[kg\ s^{-1}m^{-2}]$
Temperature	matrix of reals	inout	The input value is the initial condition. The output value is the updated temperature profile. The first index is related to the grid position x_i, the second index indicates the wall (1:North, 2:East, 3:South, 4:West, 5:Roof)
x	vector of reals	inout	Grid points where temperatures are defined
Q_indoors	vector of reals	out	Interior heat flux (6 surfaces)$[W]$
Q_outdoors	vector of reals	out	Exterior heat flux $[W]$
G_diffuse	vector of reals	out	Irradiance (6 surfaces) $[W/m^2]$
T_a	real	out	Air temperature of the room.
T_outlet	real	out	Outlet water temperature.
Power	real	out	Power heat gain $[W]$
Transmittance	vector of reals	out	NIR transmittance (6 surfaces)
Reflectance	vector of reals	out	Front reflectance (6 surfaces)
Reflectance_back	vector of reals	out	Back reflectance (6 surfaces)

Table 4.1: Description of the Thermal_problem_zone arguments

Part III

Software design document

Chapter 1

Spectral problem of WFG

The Spectral Problem is solved means of the Spectral library. This software library makes public to the user some subroutines which allow solving the spectral problem of a multilayer glazing. The structure of this Application Program Interface (API) is shown in Figure 1.1 where nodes represent software modules and edges represent file dependencies between modules.

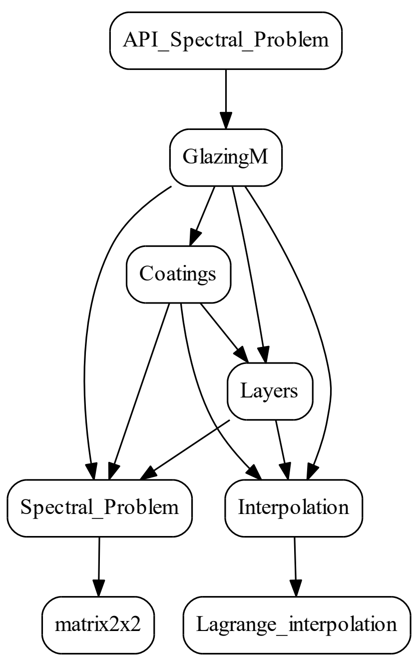

Figure 1.1: Modules dependencies of the Spectral Problem.

In Figure 1.2, a complete software structure with its object definition is shown. When a glazing is created, objects are internally instanced.

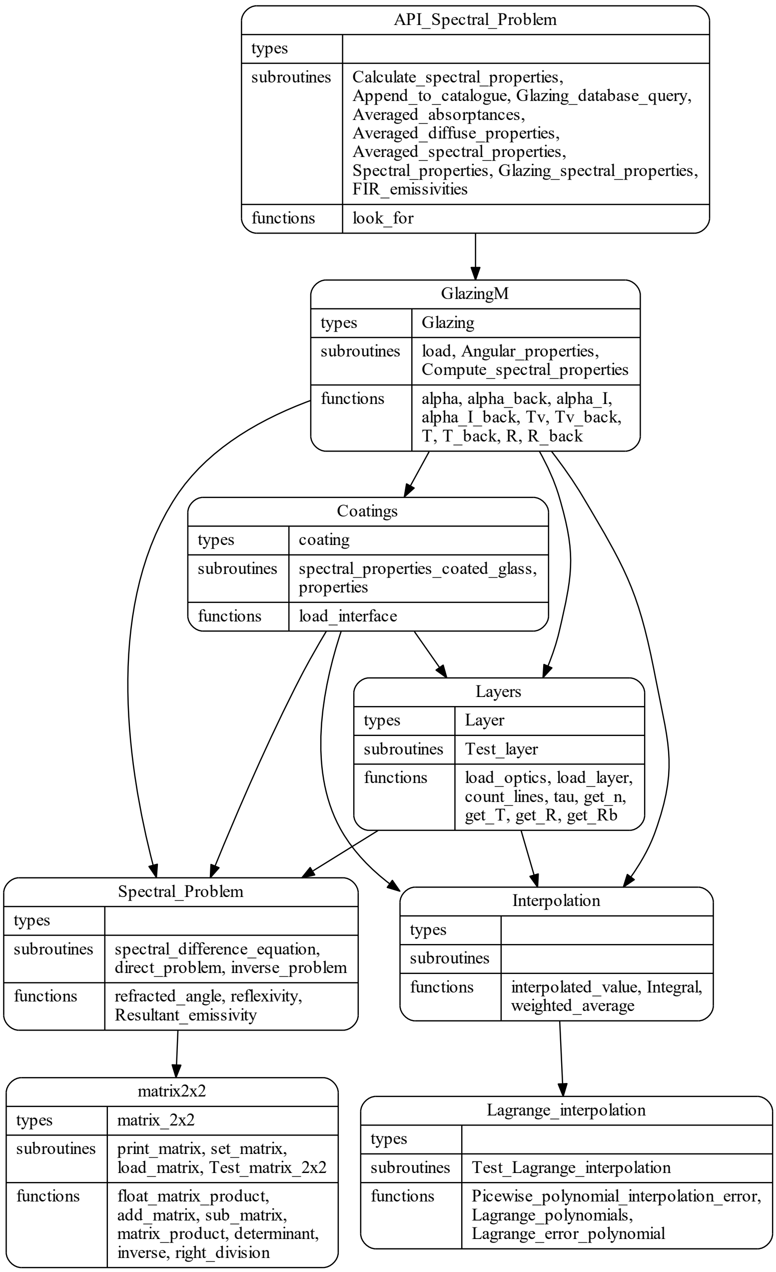

Figure 1.2: Complete use graph of the Spectral API

Both Coatings and Layers are different modules used by GlazingM. This main object is named Glazing and it holds all the properties and variables of the glazing itself along with the layer composition. The object definition is done in the module GlazingM, along with methods and functions for this object. These methods determine spectral properties such as Angular_properties or T (Transmittance). In turn, the object Glazing is composed of an array of layers an array of interfaces or coatings as it is shown in Figure 1.3.

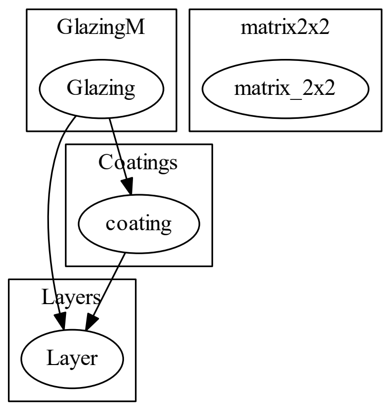

Figure 1.3: Classes or objects involved in the Spectral Problem

The Coatings module, apart from defining the coating type, contains subroutines that calculate the spectral properties of the coating as a function of the wavelength and the angle of incidence. Two subroutines: spectral_properties_coated_glass and properties are used to determine those properties.

Besides, the coating type definition involves the Layers module. That is, a coating is formed by a glass substrate and a coated glass which are layers by themselves. The layer object holds functions related to data handling and data reading from the International Glazing Database (IGDB).Once the spectral properties of each layer are read from the IGDB, the Spectral_Problem module deals with the problem of multiple reflections to finally determine the spectral properties of the whole glazing. A difference equation that governs the Spectral Problem (spectral_difference_equation) is solved to claculate this properties.

To facilitate the resolution of the difference equation, matrices together with their associated operations are defined. Finally, the Interpolation module gives interpolated values from discrete values. It allows to know the spectral properties for a specific angle of incidence interpolated from a small number of discrete angles.

1.1 GlazingM module

The `Glazing` object holds all the physical properties and variables of the glazing, and also the main functions or methods of the object. The main variables contained by the object `Glazing` are the following:

- `name`: It is a label string to identify the glazing.
- `Layer`:A glazing is a set of layers.
- `coating`: A glazing is a set of coatings or interfaces.
- Front and back transmittances, reflectances and absorptances for each layer.
- As the previous properties are in function of wavelength and impinging angle, there are defined a set of angles and a set of wavelengths.

The main functions and subroutines inside this module are the following:

- `subroutine Angular_properties:` Computes the absorption, transmittance and reflectance of the glazing for a given angle of incidence. The value is taken as an average over the available wavelength range.
- `subroutine Compute_spectral_properties:` Computes the absorption, transmittance and reflectance of the glazing for a given angle of incidence and a given wavelength.
- `function alpha and alpha_back:` Computes the front and back absorption array.
- `elemental function T, T_back, Tv and Tv_back:` Computes the front and back transmittance as a function of the angle of incidence.
- `elemental function R and R_back:` Computes the reflectance and back reflectance as a function of the angle of incidence.

1.2 Spectral_Problem module

This module deals with the spectral problem resolution. The resolution has two parts. The first one is called inverse problem and it consists on calculating reflexivity indexes of the interfaces and transmissivity of the layers (calculated in the `inverse_problem` subroutine). This output is the input for the subroutine `spectral_difference_equation` subroutine. Once the inverse problem is solved todetermine the reflectivity and transmissivity of each layer, the `spectral_difference_equation` subroutine calculates the absorption, transmittance and reflectance of the given structure of layers.

Chapter 2

Thermal problem of WFG

This library uses the previous calculated absorptions of each layer from the Spectral Library for a specific glazing. The aim of this module is to calculate the temperature profile of the glazing in a certain period of time. The general structure of this software library is shown in Figure 2.1 where each node represents a software module and each edge stands for uses between modules.

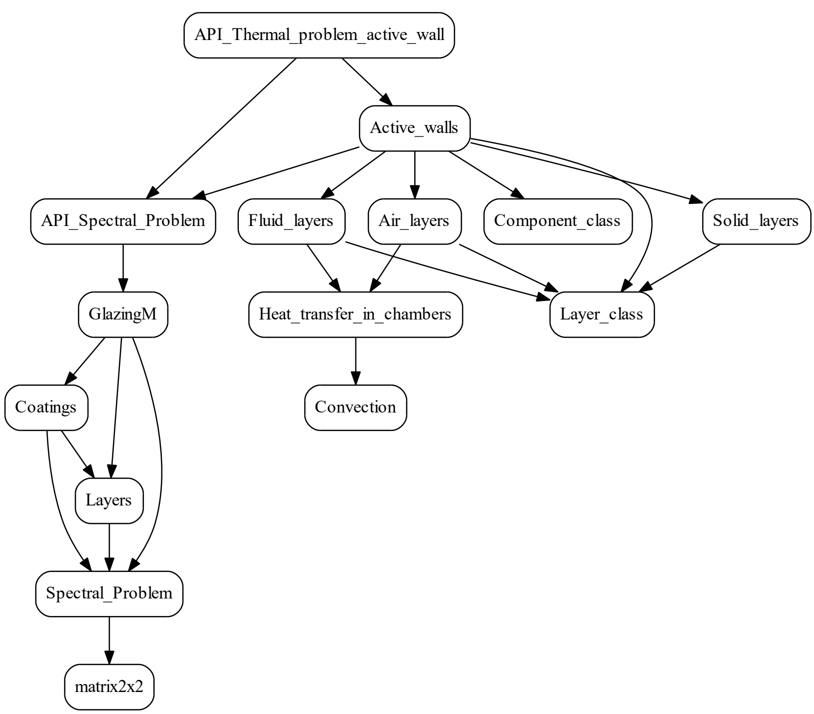

Figure 2.1: Software dependencies for the thermal problem.

Once the spectral problem is solved, the library creates an object named `Active_window` to calculate the thermal profile of the glazing. In Figure 2.2, a complete software structure with its object definition is shown. When a glazing is created, objects are internally instanced.

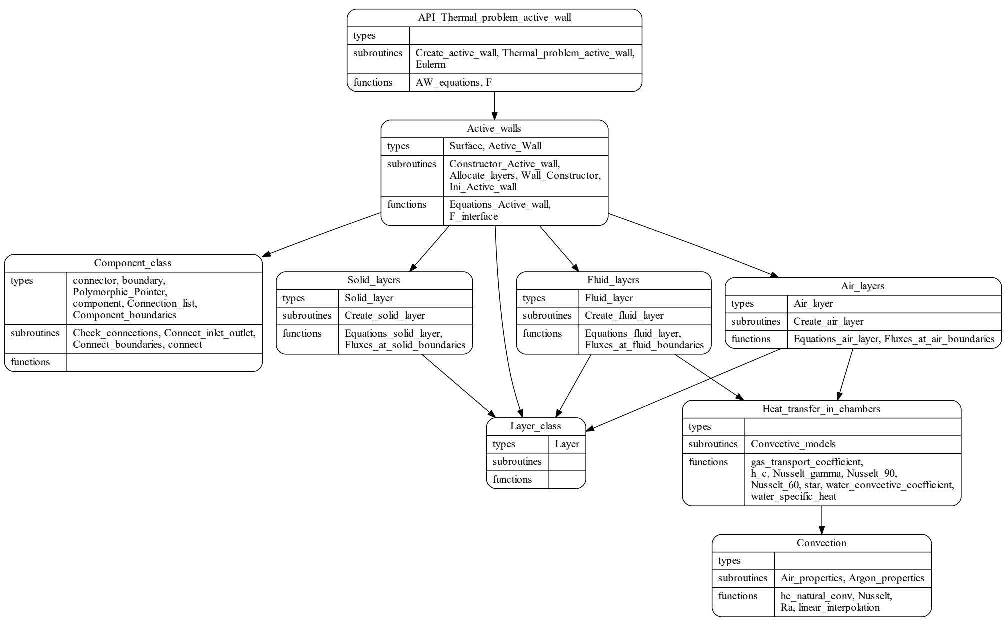

Figure 2.2: Complete use graph of the Spectral API

An `Active_wall` is a set of polymorphic layers that inherits the properties of the object `component`. The object `component` is shown in Figure 2.3 and comprises an inlet and outlet `connectors` and two `boundaries`. An inlet or outlet connector is an object defined by its inlet or outlet temperature and its flow rate. These two elements of any `connector` are pointers that can be linked to any other connector of the building such as primary or secondary circuit connectors. On the other hand, the object `boundary` is defined by its interface temperature and its heat flux. Hence, components can be considered one dimensional open systems that share heat through its boundary and interchange energy due to the difference of temperature between the inlet and outlet.

A `Connection_list` object is also defined. It allows to build complete energy circuits by assembling different components.

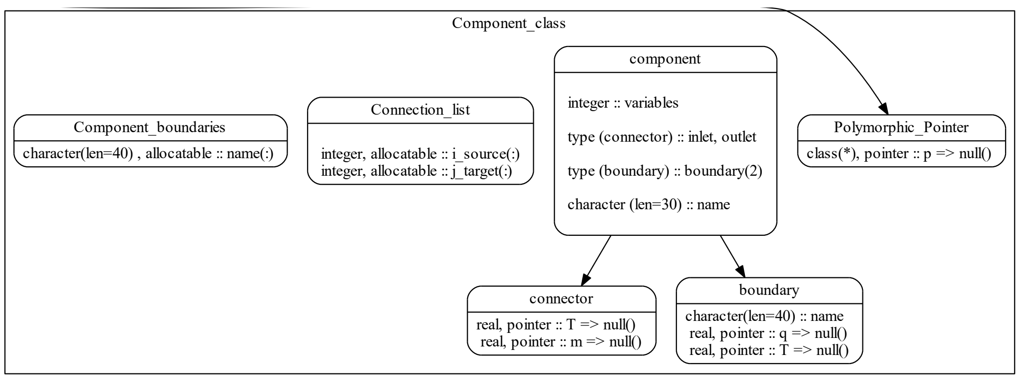

Figure 2.3: The component class comprises an inlet and outlet connector and two boundary conditions.

In Figure 2.4, the different layers involved in the thermal problem are shown. The modules: `Water_layers`, `Air_laters` and `Glass_layers` define new objects or classes with functions to create the layer, equations to determine the inner thermal behavior of the layer and boundary conditions to match interface values with other layers. Besides, these polymorphic layers inherit the `Layer` properties created in the `Layer_class` module. The spectral properties of layer class are shown in Figure 2.5.

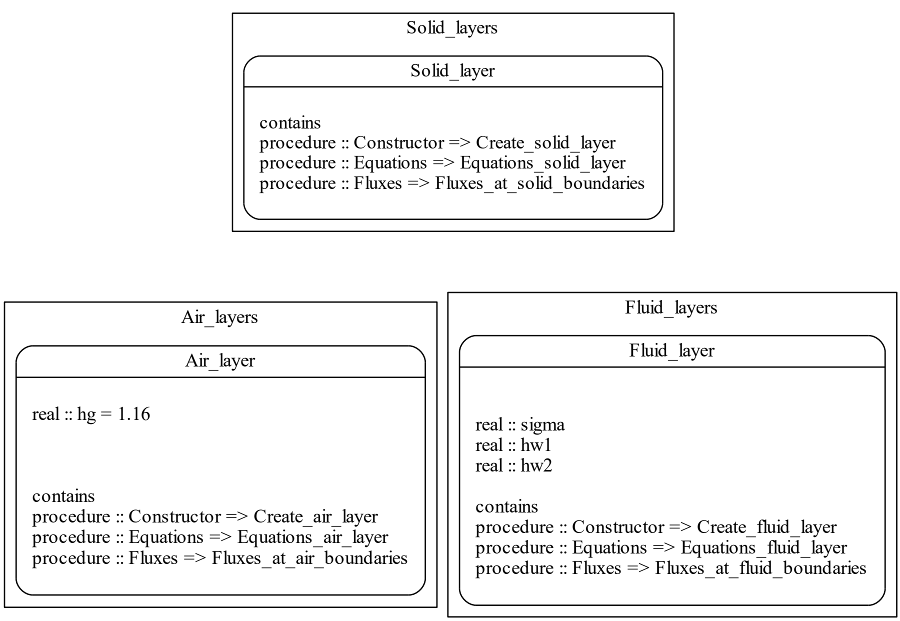

Figure 2.4: Objects and classes involved in the thermal problem.

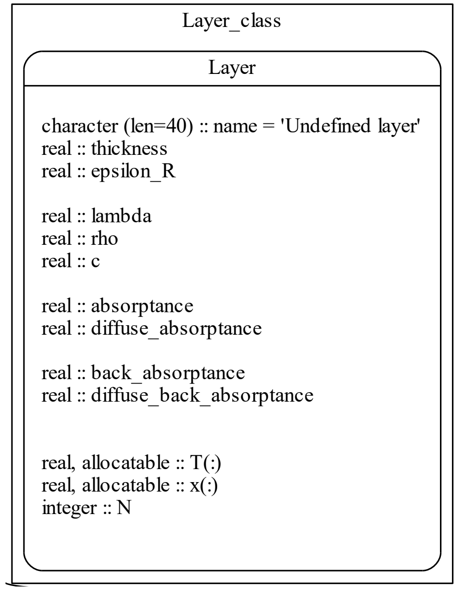

Figure 2.5: Layer class with its spectral properties.

2.1 Active windows

`Active_windows` is module which is used by the API Thermal problem module and builds the object `Active_Window` by using layer modules: `Glass_layers`, `Air_layers`, `Air_layers`.

This module contains methods or functions such as `Constructor_Active_Window` and `Equation_Active_window` to simulate the thermal behavior of the active glazing. These methods gather different equations of each layer and solve a nonlinear system of equations for the temperature and heat fluxes interfaces. The `API_Spectral_problem` is used to obtain the absorption of each layer which is a source term for the thermal equations of each layer.

2.2 Layers: glass, air and water

These modules are responsible of creating each type of layer of the glazing. Any specific layer is created by inheriting the properties of a `Layer`. More specifically, `Layer_class` holds the common characteristics of a layer such as thermal profile and spectral and thermal properties.

`Air_layers` and `Water_layers` are fluid layers and convection transfer mechanism is taken into account. This convection is modeled and calculated in the module `Heat_transfer_in_chambers` which is used by these layers.

2.3 Heat transfer in chambers

Three different convection models: ”CONSTANT”, ”ISO” and ”EnergyPlus” are used to determine the convection heat coefficients. Fluids properties are considered functions of temperature are calculated by the `Convection` module. Specific trigonometric maths functions are obtained by the `trigonometric` module.

2.4 Convection

This module is used to calculate fluid properties and coefficients related to convection. It contains two main functions and two subroutines:

- `Function hc_natural_convection`. This function is affected by the inclination of the fluid layer. Its output is the natural convection heat coefficient.
- `Function Nusselt`. Its output is the Nusselt number. This function contains the function `Ra` which calculates the Rayleigh number.
- `Subroutine Air_properties`. It returns air properties as a function of temperature and pressure.
- `Argon_properties`. It returns argon properties as a function of temperature and pressure.

Chapter 3

Outdoor conditions

Boundary conditions must be given to solve the thermal problem. In particular, external conditions have two ways of being set. The first one consists in constant temperatures and solar radiation. The second one is setting them from experimental database which gives you real temperatures and radiations as a function of time. The aim of this library is to provide real external conditions for the thermal problem of a particular location. In Figure 3.1, the software structure of this library is shown.

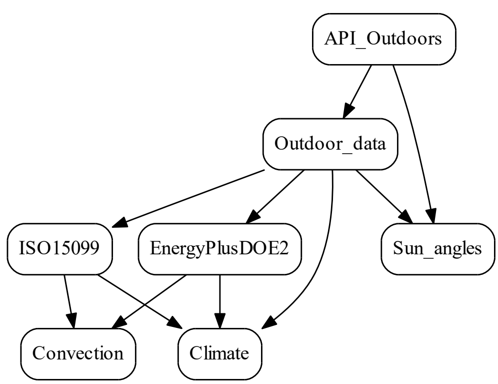

Figure 3.1: Software structure of outdoor conditions library

The main module use the module `Outdoor_data` and `Sun_angles`. The user by a proper initialization selects the specific convection model: `ISO15099` or `EnergyPlusDOE2`. A complete software structure showing details of different classes in each module is presented in Figure 3.2.

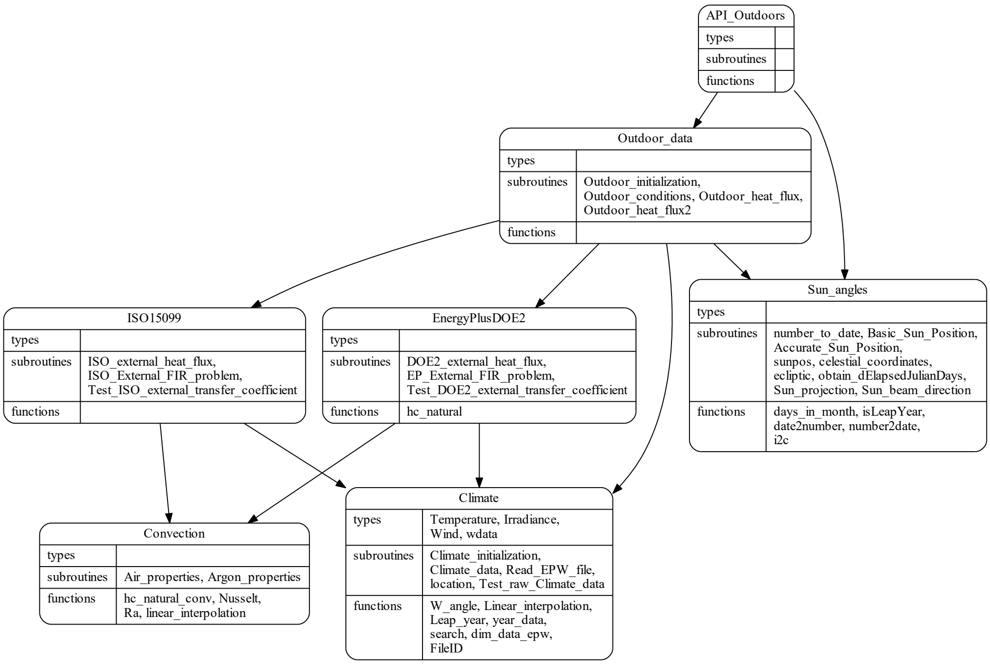

Figure 3.2: Complete softwaree structure of the Outdoorconditions module.

3.1 Outdoor data

This module holds the most important public subroutines for the outdoors problem (`Outdoor_initialization` and `Outdoor_conditions`). It also contains other subroutines that determines the heat flux of the exterior surface. These subroutines calls to subroutines contained in the modules `ISO15099`, `Sun_angles` and `Climate` depending on the convection model.

3.2 EnergyPlusDOE2 and ISO15099

The API has an option that allows the user to select the convection model that affects the external heat flux, with the `model` argument in the `Outdoor_initialization`. The different models available are: `CONSTANT`, `ISO` or `EnergyPlus`.

The ISO models uses the `ISO15099` module and the EnergyPlus model uses the `EnergyPlusDOE2` module to calculate the natural convection coefficient and the external flux. The `Climate` subroutine reads a standard EPW file to give the external conditions. Once this external data are determined, sun position angles such as zenith of azimuth are calculated to project the solar beam radiation into the glazing or walls.

3.3 Climate

The module Climate_data reads EPW files and extract climate data. This is done by the Read_EPW_file subroutine. It also contains types related to different climate variables such as Wind or wdata. A detailed image of this module is presented in Figure 3.3.

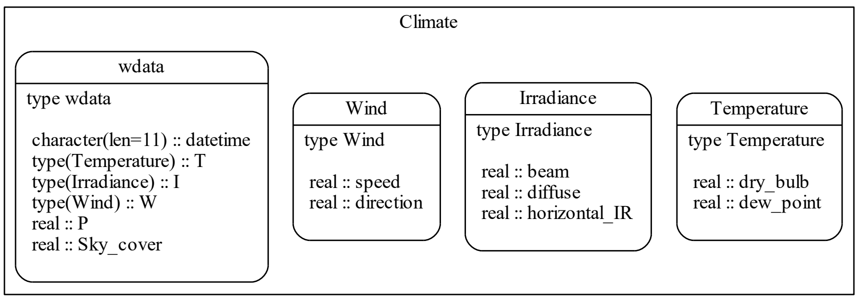

Figure 3.3: Climate data module with its types or classes.

3.4 Sun angles

This module comprises all functions and subroutines related to position of the sun and time to project the solar beam radiation on a surface by means of Sun_projection subroutine.

Chapter 4

Thermal problem of insulated zones

The `API insulated problem` allows the user to calculate the interior temperature of a simplified room with some opaque walls water flow glazings. Given the outdoor conditions for a previously initialized room, this library determines indoor temperature, water heat gain of the active glazings, heat fluxes and outlet temperature. To do that, this library uses the thermal problem library and a new library for the indoor solar distribution as it is shown in Figure 4.1.

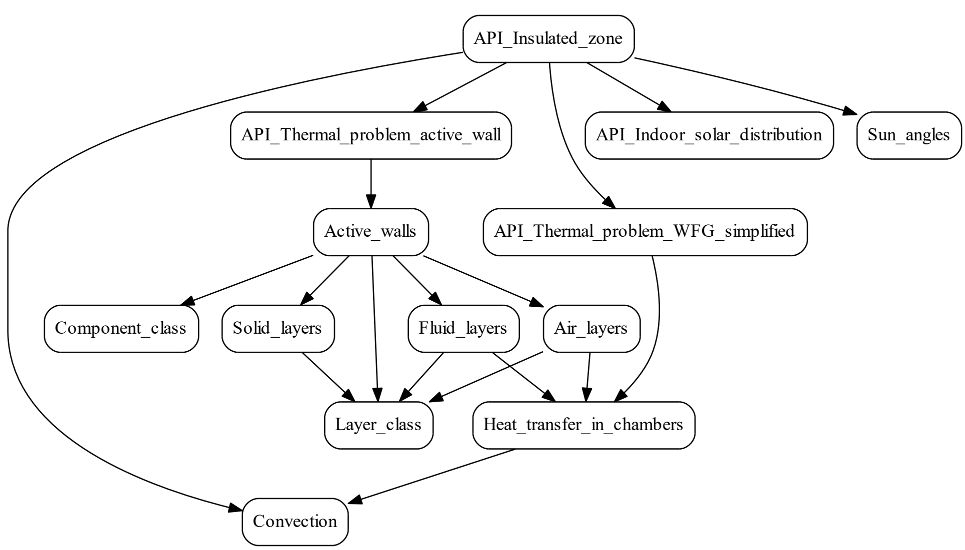

Figure 4.1: Software structure of insulated zones library.

The new library is called `Indoor_solar_distribution` and it is shown in Figure 4.2. Once the external beam and diffuse solar radiation are given by the outdoor conditions library, this library deals with the solar distribution of the incoming solar radition through glazings. When the beam solar radiation impinges in the interior surfaces of the room, some part of the radition is reflected diffusely. Later, this diffuse reflections creates a multiple reflection problem which is solved by the Radiosity Irradiation Method (RIM). This method calculates view factors between discrete interior surfaces and determines illuminated interior surfaces due to different angles of incidence.

API_Indoor_solar_distribution	
types	vertex, Plane_Surface
subroutines	RIM_initialization, RIM_solution, RIM_matrix_factorization, Beam_sources, filter, inner_points, common_points, intersection_points, Projection, adjacent_vertices, Test_clipping, Test_view_factor, Test_view_factor2
functions	View_factor_perpendicular_surfaces, View_factor_parallel_surfaces, Intersected_area, Plane_components, triangle_area, is_inside, segments_intersection, perpendicular, quadrilateral_area

Figure 4.2: Software structure of insulated zones library.

Once the solar incoming radiation is distributed, the thermal problem is solved. The simplified room is formed by six flat surfaces which can be opaque, transparent glazings or water flow glazings. This library gives the boundary conditions for each surface that constitutes the room in order to solve the thermal problem of each surface. Later, the indoor air temperature is determined as a weighted average temperature of the different interior surfaces.

Part IV

Benchmark

Chapter 1

Test cases for water flow glazings

To carry out a complete validation of the present simulation tool, each module or subroutine for each physical model should be validated incrementally. In this manner, the thermal problem of water flow glazing can be decoupled from the thermal problem of the rest of the components of the system.

Moreover, the solution of the thermal problem can be non-steady involving variable climate conditions and thermal mass of each layer or component. If boundary conditions are constants, the steady state does not depend on thermal mass and specific heat of components. Hence, a benchmark test cases based on constant boundary conditions is the easiest way to start with validation.

The benchmark comprises five different glazings: HeatGlass, Coolglass, iThermGlass, RadiaGlass and a Reference glazing. When the system is circulating, the flow rate is set to the design flow rate 2 l/min m^2 and the inlet temperature is set a constant value θ_{INLET}. When the system is stopped, the flow rate is set to zero and the outlet temperature of the water chamber θ_{OUTLET} is called the stagnation temperature.

The outputs of these test cases are the water heat gain or thermal power

$$P = \dot{m}\ c\ (\theta_{\text{OUTLET}} - \theta_{\text{INLET}}),$$

transported by the flow rate $\dot{m}$ and the room heat gain which is defined as the sum of the transmitted beam solar irradiance TI_b and the indoor thermal heat flux q_i or the heating/cooling power.

We consider that the outdoor heat flux is given by the following expression:

$$q_e = h_e(\theta_{ne} - \theta_{se}),$$

where h_e is the outdoor convective coefficient which could be consider a constant value or it can be calculated with more precise models, θ_{se} is the superficial temperature of the outermost glass pane and θ_{ne} is the environmental outdoor temperature which can be a given constant value or the real environmental outdoor temperature which takes into account the temperature of external radiating buildings, ground or sky. To complete the outdoor boundary conditions, beam solar irradiance, diffuse irradiance and the angle of incidence should be given.

Regarding indoor boundary conditions, the indoor heat flux is given by the following expression:

$$q_i = h_i(\theta_{si} - \theta_{ni}),$$

where h_i is the indoor convective coefficient which could be consider a constant value or it can be calculated with more precise models, θ_{si} is the superficial temperature of the innermost glass pane and θ_{ni} is the environmental indoor temperature which can be a given constant value or it can be calculated solving the indoor thermal problem. To complete the indoor boundary conditions, beam and diffuse indoor solar irradiance should be given or calculated.

The heat transfer inside gas chambers is also modeled by a Newton's law. The heat flux in a gas chamber between two parallel surfaces a temperature θ_1 and $theta_2$ is expressed by:

$$q_g = h_g(\theta_1 - \theta_2),$$

where h_g is the heat transfer coefficient of gas chambers. This coefficient takes into account the radiative heat transfer between the parallel glass panes and the natural convective transport. Once again, this value can be given as a constant value or can be calculated knowing the emissivities of the two glass planes and an experimental correlation for the natural convective transport.

In the water chambers, the situation is slightly different. Heat fluxes are proportional to the difference of temperature between the water temperature θ_w and the glass pane temperatures (θ_1 and θ_2). This coefficient takes into account the heat transport mechanism forced by the water flow inside the chamber. Since the water is opaque to far infrared, the radiative heat transfer mechanism is not present in the water chamber. Hence, the heat flux for the two glass panes is expressed by:

$$q_1 = h_w(\theta_1 - \theta_w), \qquad q_2 = h_w(\theta_w - \theta_2),$$

where h_w is the heat transfer coefficient for the water chamber.

The objective of this benchmark is twofold: *(i)* to define different test cases to compare numerical simulations with standard tools and *(ii)* to compare thermal performances of different glazings.

1.1 Isolated water flow glazing

In order to validate the simulation of the complete thermal problem, a simple isolated glazing is considered. Hence, the indoor temperature is considered a given value and the diffuse indoor irradiance is set to zero to decrease the difficulty of the problem. This is equivalent to impose indoor boundary conditions for an isolated glazing. Moreover, two different test cases are considered: steady and unsteady boundary conditions.

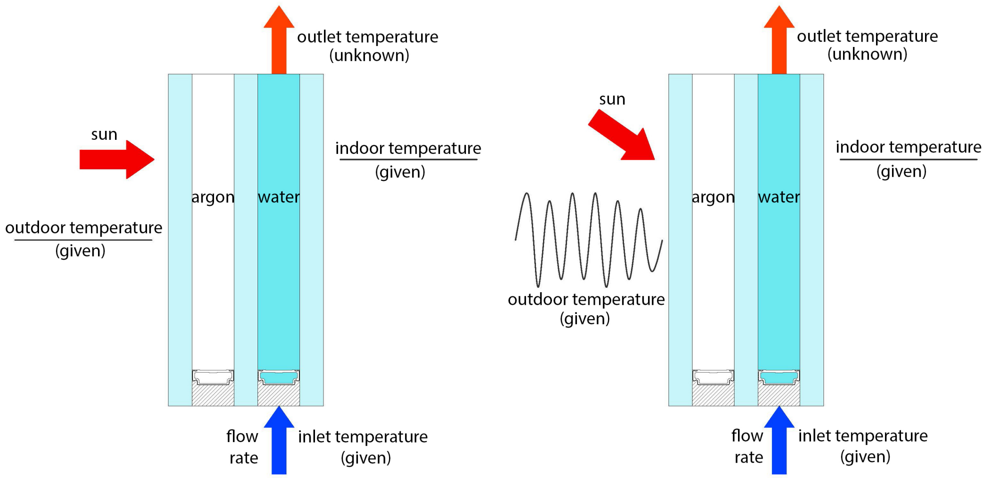

Table 1.1: Steady and unsteady test cases for an isolated water flow glazing

Steady test cases

If boundary conditions are constant, the solution becomes constant once the steady solution is reached. In this set of test cases, normal angle of incidence is considered to eliminate the uncertainties associated to the dependence of the absorptance of each layer with the angle of incidence.

Two steady states are considered: winter and summer. In the following table, outdoor and indoor boundary conditions which define these two seasons are fixed.

Season	θ_e	θ_i	$h_e(W/m^2K)$	$h_i(W/m^2K)$	$I_b(W/m^2)$	$I_d(W/m^2)$
Winter	0	21	23	8	600	0
Summer	35	28	23	8	800	0

Table 1.2: Indoor and outdoor boundary conditions for winter and summer test cases

Regarding the glazing, the inlet of the water chamber is also considered a boundary condition. In these test cases, the flow rate and the inlet temperature are considered constant.

Season	$\dot{m}(l/(min\ m^2))$	θ_{INLET}	$h_g(W/m^2K)$	$h_w(W/m^2K)$	$c(J/KgK)$
Winter	2	21	1.16	50	3600
Summer	2	17	1.16	50	3600

Table 1.3: Thermal and transport parameters for the water flow glazing

Once the glazing reaches the steady state in winter and in summer, thermal performances are determined. The following table shows water heat gain P, transmitted solar beam radiation TI_b and room thermal heat gain q_i for a winter and a summer condition for different glazings.

Glazing	P(W/m^2)	TI_b(W/m^2)	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	227.4	128.4	13.7	208.0	22.9	54.2	3.1	22.7
mCoolGlass	12.9	129.2	2.6	13.6	21.1	22.9	8.2	21.3
mRadiaGlass	9.0	146.5	2.7	5.6	21.1	21.6	17.9	21.3
miThermGlass	19.9	123.7	7.4	8.8	21.2	22.6	18.0	21.9
Reference	19.9	152.3	-7.8	-7.8	21.2	21.2	8.2	20.0

Table 1.4: Steady thermal performances for a standard winter condition

Glazing	P(W/m^2)	TI_b(W/m^2)	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	413.3	171.2	-44.4	308.7	20.4	77.3	37.1	22.4
mCoolGlass	127.2	172.3	-59.2	49.5	18.1	35.6	43.8	20.6
mRadiaGlass	571.4	195.3	-35.8	153.1	21.8	52.2	36.2	23.5
miThermGlass	556.1	164.9	3.5	43.2	21.6	62.7	36.2	28.4
Reference	556.1	203.1	23.1	23.1	21.6	21.6	44.5	30.9

Table 1.5: Steady thermal performances for a standard summer condition

Stagnation temperature θ_s, superficial outermost and innermost temperatures θ_{se} and θ_{si} are also shown in Tables 1.4– 1.5.

If energy management in winter is based on energy harvesting, **HeatGlass** has the best performance of all. Its water heat gain is ten times higher than **iThermGlass** which is the next competitor. On the other hand, if energy management in summer is based on energy harvesting, **RadiaGlass** or **iThermGlass** are the best elections. In summer the water heat gains of these glazings is 1.5 times larger than the one of **HeatGlass**.

In winter, heating capacity q_i associated to the sun energy is needed depending on location and usage of buildings. The above tables show that **HeatGlass** is the best glazing with a q_i value ten times the corresponding one associated to **iThermGlass** which is the next competitor.

In summer and depending on location and building usage, cooling capacity q_i could be a criterion to choose a glazing. The above tables show that **CoolGlass** is the best glazing which behaves more or less than **HeatGlass**. Notice that to achieve this cooling capacity, **CoolGlass** has to dissipate around 140 W/m^2, whereas **HeatGlass** has to dissipate around 500 W/m^2. This implies that **HeatGlass** will need more than double of power of **CoolGlass** to obtain a similar cooling capacity.

Simulations with more precise models for h_i, h_e h_g and h_w can be calculated. Namely, the ISO15099 norm and the EnergyPlus models have been implemented. In the following tables, numerical results with precise transport coefficients are shown.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	224.5	128.4	13.5	195.7	22.9	52.2	5.3	22.7
mCoolGlass	24.3	129.2	3.2	23.9	21.2	24.5	19.2	21.4
mRadiaGlass	180.3	146.5	11.6	95.8	22.5	36.1	25.8	22.4
miThermGlass	199.2	123.7	9.0	36.1	22.7	45.9	26.0	22.1
Reference	199.2	152.3	3.2	3.2	22.7	22.7	19.3	21.4

Table 1.6: Steady thermal performances for a standard winter condition with ISO models

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	233.5	128.4	14.0	204.1	22.9	53.6	12.0	22.7
mCoolGlass	28.8	129.2	3.4	27.9	21.2	25.2	23.2	21.4
mRadiaGlass	201.4	146.5	12.6	100.4	22.7	36.8	26.8	22.6
miThermGlass	220.8	123.7	9.2	36.6	22.8	46.3	27.0	22.2
Reference	220.8	152.3	7.5	7.5	22.8	22.8	23.3	21.9

Table 1.7: Steady thermal performances for a standard winter condition with EnergyPlus models

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	415.4	171.2	-44.3	299.6	20.5	75.8	37.7	22.5
mCoolGlass	146.2	172.3	-58.2	63.4	18.2	37.8	53.1	20.7
mRadiaGlass	567.0	195.3	-36.0	227.7	21.7	64.2	36.0	23.5
miThermGlass	552.4	164.9	2.6	82.9	21.6	84.3	36.0	28.3
Reference	552.4	203.1	38.7	38.7	21.6	21.6	54.9	32.8

Table 1.8: Steady thermal performances for a standard summer condition with ISO models

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	412.1	171.2	-44.5	295.9	20.4	75.2	35.0	22.4
mCoolGlass	140.8	172.3	-58.5	59.0	18.2	37.1	49.6	20.7
mRadiaGlass	550.2	195.3	-36.9	206.8	21.6	60.8	35.3	23.4
miThermGlass	534.9	164.9	2.4	73.2	21.5	77.7	35.2	28.3
Reference	534.9	203.1	33.8	33.8	21.5	21.5	51.2	32.2

Table 1.9: Steady thermal performances for a standard summer condition with EnergyPlus models

In the Figure 1.10, a comparison between the simplified and the complete model is done. It is shown the temperature profile of HeatGlass for a standard summer condition described in Table 1.2.

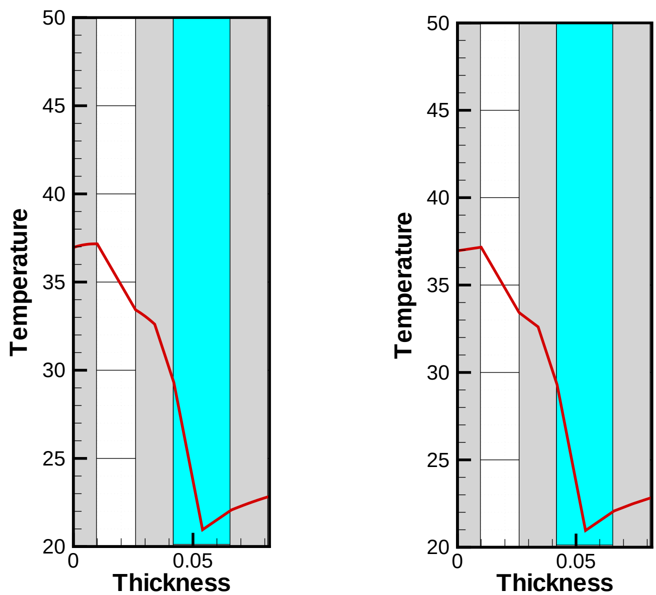

Table 1.10: Thermal profile of mHeatGlass for a standard summer condition simulated with the simplified and the complete model.

1.1.1 Unsteady test cases

When outdoor temperature and solar irradiance vary during the day, unsteady behavior is expected. In these following test cases, thermal performances are functions of time. Besides, the indoor temperature is a given indoor boundary condition and transport coefficients remained constants to avoid uncertainties in the validation process. These test cases are simulated in Madrid and the weather file is the standard EPW file (EnergyPlus Weather). Regarding the water flow glazing, the flow rate and the inlet temperature are constant values given by Table 1.3. Two simulations in winter and in summer are accomplished. The simulation winter period runs from January 12th to January 16th. The simulation summer period runs from July 3th to July 7th.

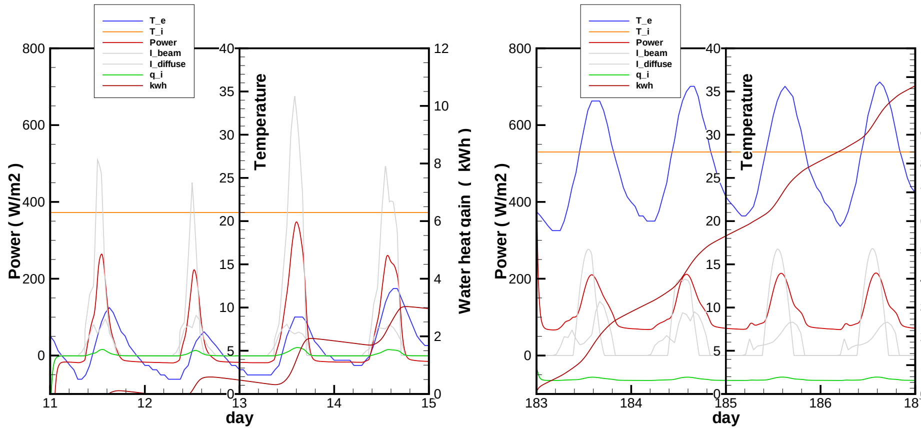

Table 1.11: Thermal behavior of HeatGlass in winter and summer period simulated with the complete model

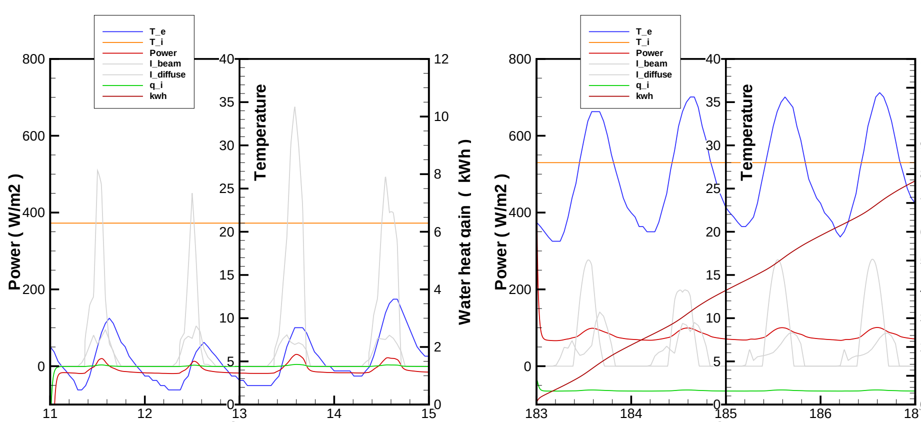

Table 1.12: Thermal behavior of CoolGlass in winter and summer period simulated with the complete model

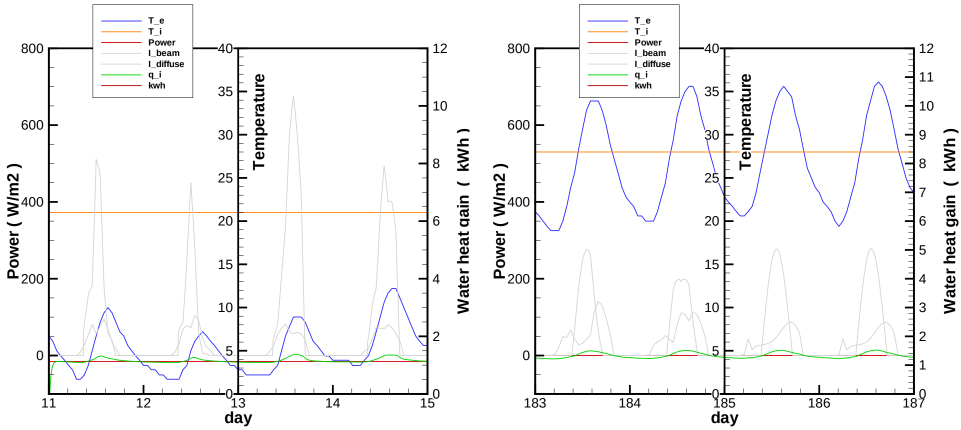

Table 1.13: Thermal behavior of the Reference glazing XTREME 60/28 in winter and summer period simulated with the complete model

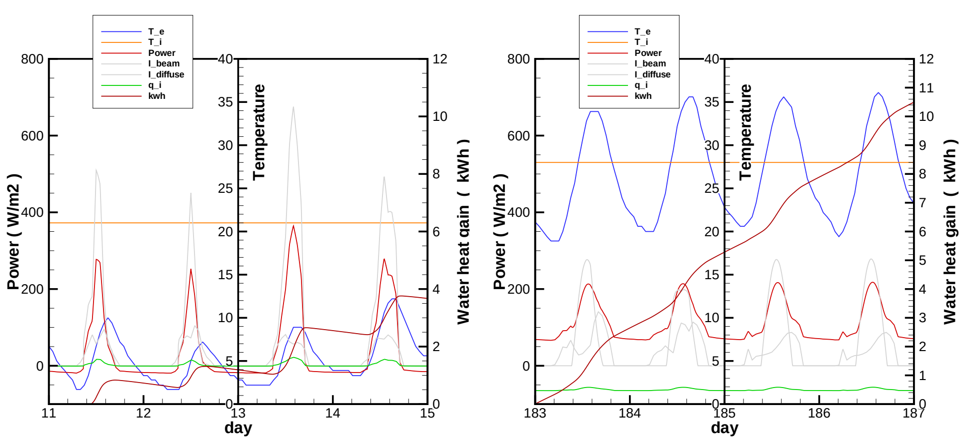

Table 1.14: Thermal behavior of HeatGlass in winter and summer period simulated with the simplified model

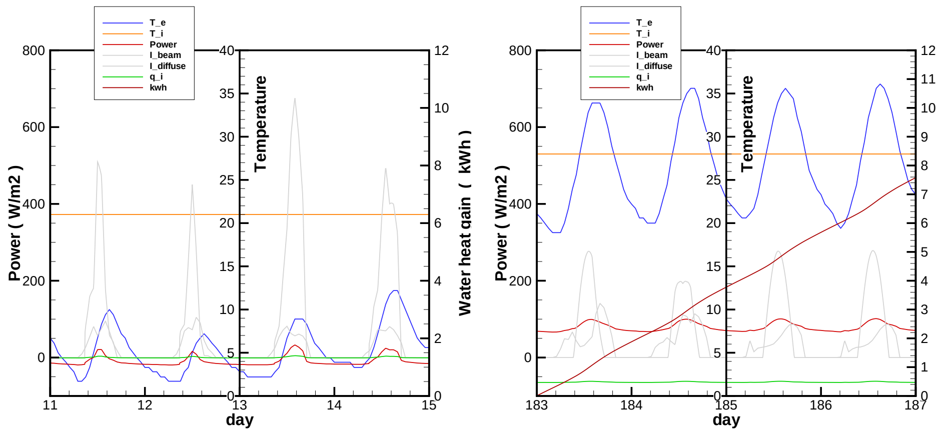

Table 1.15: Thermal behavior of CoolGlass in winter and summer simulated with the simplified model

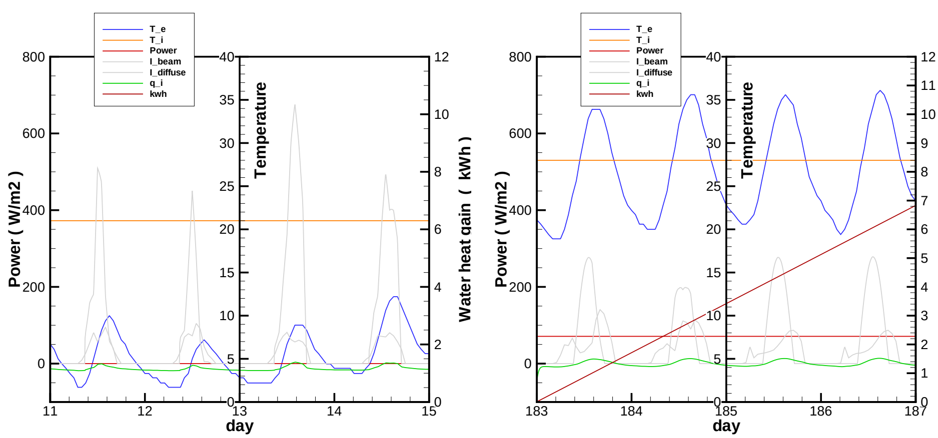

Table 1.16: Thermal behavior of the Reference glazing XTREME 60/28 in winter and summer period simulated with the simplified model

1.2 Water flow glazing in an insulated room

When the glazing is part of an insulated room, the thermal problem of the glazing is coupled with the thermal problem of the room and the indoor temperature should be determined. Hence, the complete problem becomes more difficult and the indoor boundary condition disappears to be part of the solution of the thermal problem.

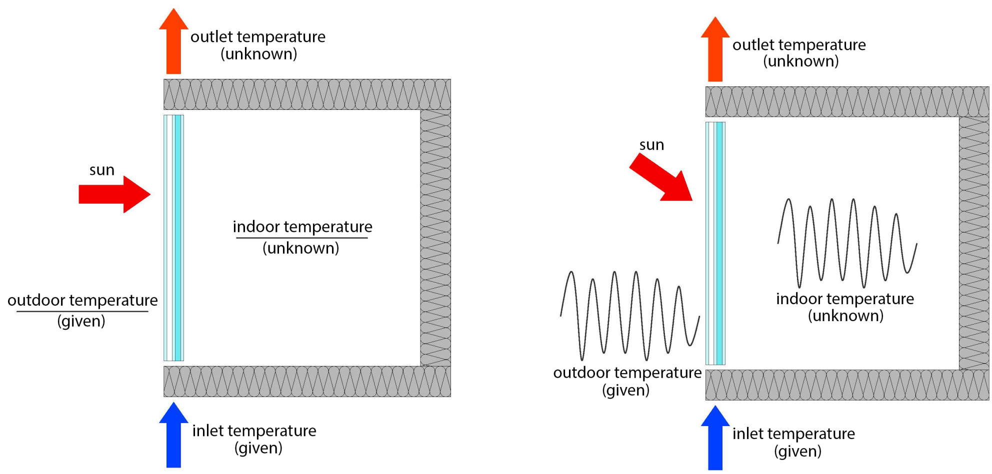

Table 1.17: Steady and unsteady test cases for a water flow glazing in an insulated room

To simplify this test case, a square room with a glazing facing south is considered. The dimensions of the room are defined in Table 1.18. The near and far infrared absorption (α_{NIR}, α_{FIR}) are uniform for all walls and showed in Table 1.18 together with the thermal transmittance of the opaque envelope.

Height(m)	Length(m)	Width(m)	α_{NIR}	α_{FIR}	$U_{wall}(W/m^2K)$
3	7	7	0.4	0.9	0.0

Table 1.18: Room dimensions, spectral and thermal properties of walls

1.2.1 Steady test cases

In these test cases, solar beam radiation is constant and normal to the glazing and diffuse radiation is set to zero (Table 1.2). Besides, transport heat coefficients are considered constants and given in Table 1.2. Regarding the water flow glazing, the same values set in Table 1.3 are used in this simulations.

When the thermal problem water flow glazing is coupled with the indoor thermal problem, the indoor temperature should be calculated at the same time the glazing temperature is determined. Besides, each wall has a different temperature due to the luminance of the direct beam sun radiation. Since the water flow glazing is facing south and the solar beam radiation is normal to the glazing, north facade is illuminated and some part of this impinging energy is absorbed. The rest of this energy is diffusely reflected and creates the indoor diffuse irradiance. Later, this irradiance is absorbed in each indoor surface. Hence, in this test case the water flow glazing absorbs extra energy form the indoor irradiance.

Two different mechanisms appear to modify the thermal performances of the water flow glazing: *(i)* the value of the indoor temperature which can be very high depending on the wall insulation and *(ii)* the abosorbed back irradiance. Depending on the insulation of the walls and the near infrared absorptances, the water heat gain can be very different when compared to the water heat gain of an isolated glazing.

In Tables 1.19 and 1.20, the thermal performances of these winter and summer test cases are shown.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	364.5	128.3	111.0	111.0	24.1	-	42.4
CoolGlass	137.9	129.2	113.0	113.0	22.1	-	40.8
iThermGlass	137.4	146.5	126.5	126.5	22.1	-	43.0
RadiaGlass	134.2	123.7	108.8	108.8	22.1	-	146.7
XTREME-60/28	0.0	152.3	139.8	139.8	0.0	0.0	161.8

Table 1.19: Steady thermal performances for a standard winter condition of a water flow glazing in an insulated room

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	534.0	171.1	148.0	44.1	21.4	-	45.8
CoolGlass	231.9	172.3	150.7	150.7	18.9	-	43.8
iThermGlass	702.0	164.9	145.1	145.1	22.9	-	189.0
RadiaGlass	706.0	195.3	168.6	168.6	23.0	-	50.8
XTREME-60/28	0.0	203.14	207.2	207.2	0.0	0.0	153.3

Table 1.20: Steady thermal performances for a standard summer condition of a water flow glazing in an insulated room

1.2.2 Unsteady test cases

In these following test cases, outdoor temperature and solar irradiance vary during the day and thermal performances are functions of time. The indoor temperature is unknown and it should be obtained at the same time than the glazing temperature profile.

These test cases are simulated in Madrid with the standard weather EPW file (EnergyPlus Weather). Regarding the water flow glazing, the flow rate and the inlet temperature are constant values given by Table 1.3. Two simulations in winter and in summer are accomplished. The simulation winter period runs from January 12th to January 16th. The simulation summer period runs from July 3th to July 7th.

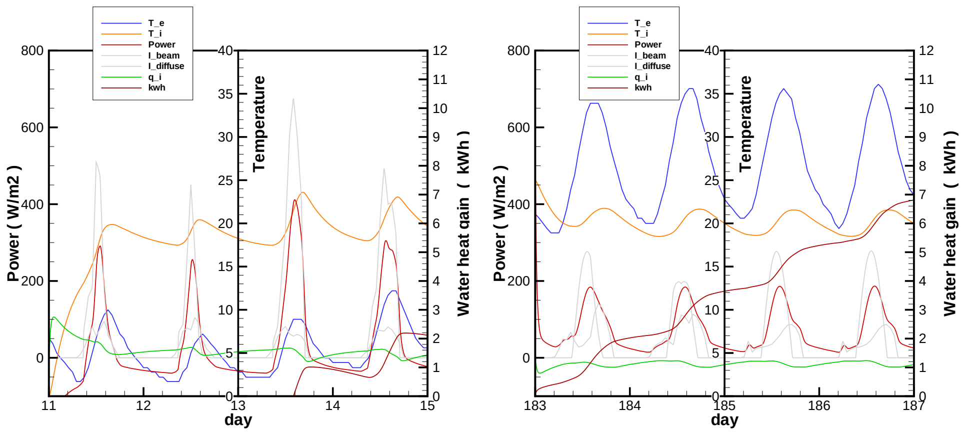

Table 1.21: Thermal behavior of HeatGlass in an insulated room in winter and summer period

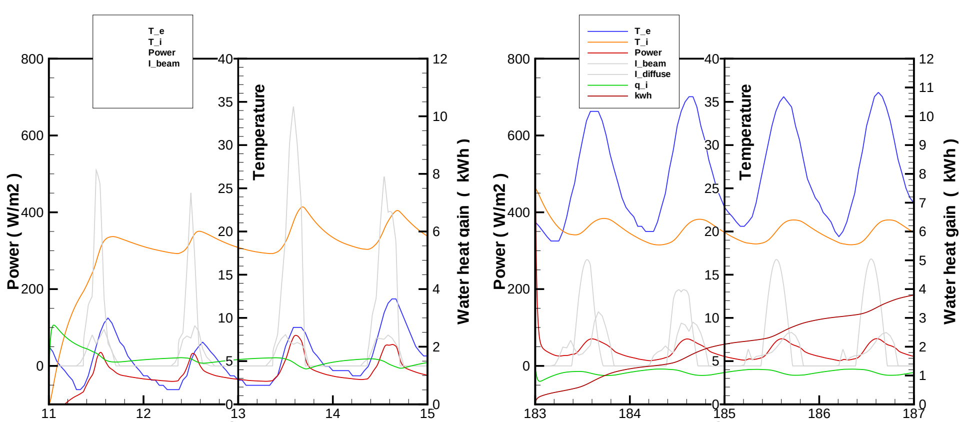

Table 1.22: Thermal behavior of CoolGlass in an insulated room in winter and summer period

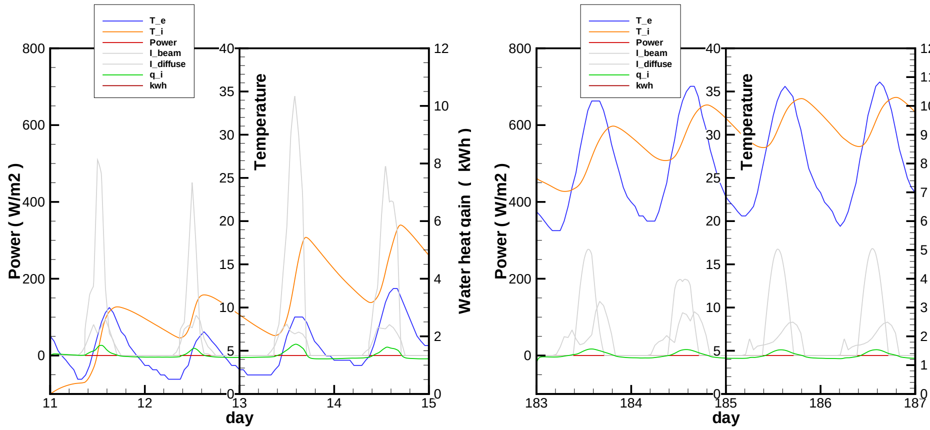

Table 1.23: Thermal behavior of the Reference glazing XTREME 60/28 in an insulated room in winter and summer period

Chapter 2

Validation with IDA-ICE

The same test cases, previously analyzed with the tool, are implemented in IDA-ICE. This projects are:

- Isolated water flow glazing
- Water flow glazing in an insulated room

Each project is developed in two different cases, which are:

- Steady test cases
- Unsteady test cases

2.1 Isolated water flow glazing

In order to validate the simulation of the complete thermal problem, a simple isolated glazing is considered. Hence, the indoor temperature is considered a given value and the diffuse indoor irradiance is set to zero to decrease the difficulty of the problem. This is equivalent to impose indoor boundary conditions for an isolated glazing. Moreover, two different test cases are considered: steady and unsteady boundary conditions.

2.1.1 Steady test cases

If boundary conditions are constant, the solution becomes constant once the steady solution is reached. In this set of test cases, normal angle of incidence is considered to eliminate the uncertainties associated to the dependence of the absorptance of each layer with the angle of incidence.

Two steady states are considered: winter and summer. In the following table, outdoor and indoor boundary conditions which define these two seasons are defined.

Once the glazing reaches the steady state in winter and in summer, thermal performances are determined. The following table shows water heat gain P, transmitted solar beam radiation TI_b and room thermal heat gain q_i for a winter and a summer condition for different glazings.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	227.4	128.4	13.7	208.0	22.9	54.2	3.1	22.7
mCoolGlass	12.9	129.2	2.6	13.6	21.1	22.9	8.2	21.3
mRadiaGlass	9.0	146.5	2.7	5.6	21.1	21.6	17.9	21.3
miThermGlass	19.9	123.7	7.4	8.8	21.2	22.6	18.0	21.9
Reference	19.9	152.3	-7.8	-7.8	21.2	21.2	8.2	20.0

Table 2.1: Steady thermal performances for a standard winter condition with the tool.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	227.4	128.4	-13.7	-208.2	22.9	54.2	3.1	22.7
mCoolGlass	12.9	129.2	-2.6	13.6	21.1	22.9	8.2	21.3
mRadiaGlass	8.95	146.5	-2.7	5.6	21.1	21.5	17.9	21.3
miThermGlass	19.9	123.7	-7.4	-8.8	21.2	22.6	18.0	21.9
Reference	0.0	152.3	7.8	7.8	0.0	0.0	8.2	20.0

Table 2.2: Steady thermal performances for a standard winter condition with IDA ICE.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	413.3	171.2	-44.4	308.7	20.4	77.3	37.1	22.4
mCoolGlass	127.2	172.3	-59.2	49.5	18.1	35.6	43.8	20.6
mRadiaGlass	571.4	195.3	-35.8	153.1	21.8	52.2	36.2	23.5
miThermGlass	556.1	164.9	3.5	43.2	21.6	62.7	36.2	28.4
Reference	556.1	203.1	23.1	23.1	21.6	21.6	44.5	30.9

Table 2.3: Steady thermal performances for a standard summer condition with the tool.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	θ_{se}	θ_{si}
mHeatGlass	413.3	171.2	44.4	-309.0	20.4	77.3	37.1	22.4
mCoolGlass	127.2	172.3	59.2	49.6	18.1	35.6	43.8	20.6
mRadiaGlass	571.3	195.3	35.8	-153.1	21.8	52.2	36.2	23.5
miThermGlass	556.1	164.9	-3.5	-43.2	21.6	62.7	36.2	28.4
Reference	0.0	203.1	-23.1	-23.1	0.0	0.0	44.5	30.9

Table 2.4: Steady thermal performances for a standard summer condition with IDA ICE.

2.1.2 Unsteady test cases

When outdoor temperature and solar irradiance vary during the day, unsteady behavior is expected. In these following test cases, thermal performances are functions of time. Besides, the indoor temperature is a given indoor boundary condition and transport coefficients remained constants to avoid uncertainties in the validation process. These test cases are simulated in Madrid and the weather file is the standard EPW file (EnergyPlus Weather).

Regarding the water flow glazing, the flow rate and the inlet temperature are constant values given by Table 1.3. Two simulations in winter and in summer are accomplished. The simulation winter period runs from January 12th to January 16th. The simulation summer period runs from July 3th to July 7th.

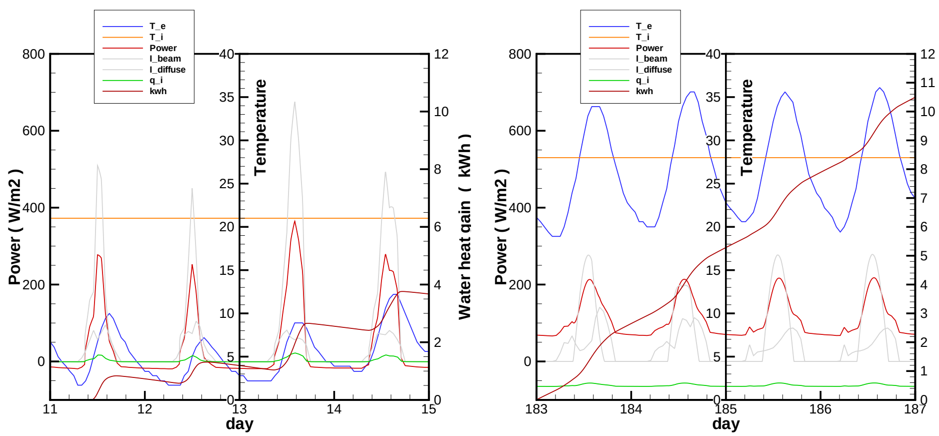

Table 2.5: Thermal behavior of HeatGlass in winter and summer period simulated with the tool.

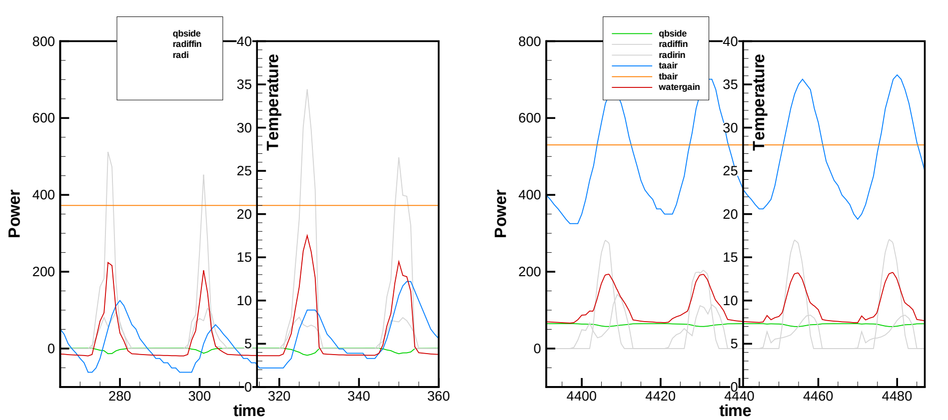

Table 2.6: Thermal behavior of HeatGlass in winter and summer period simulated with IDA-ICE.

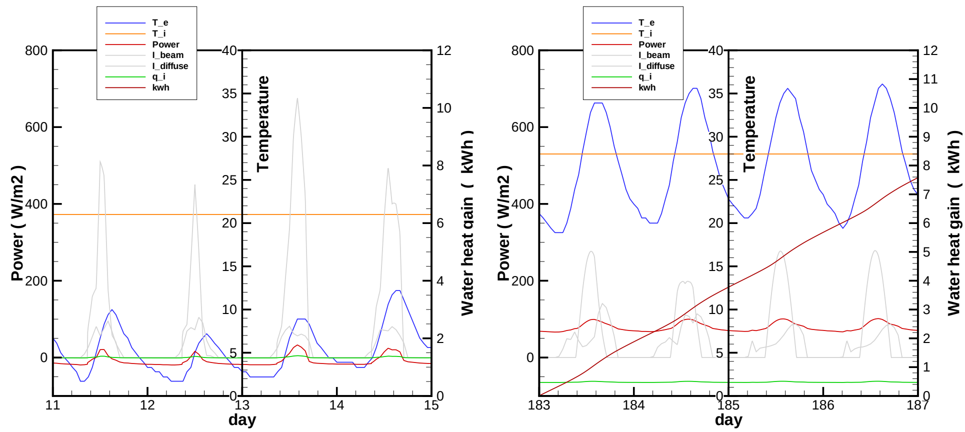

Table 2.7: Thermal behavior of CoolGlass in winter and summer simulated with the tool.

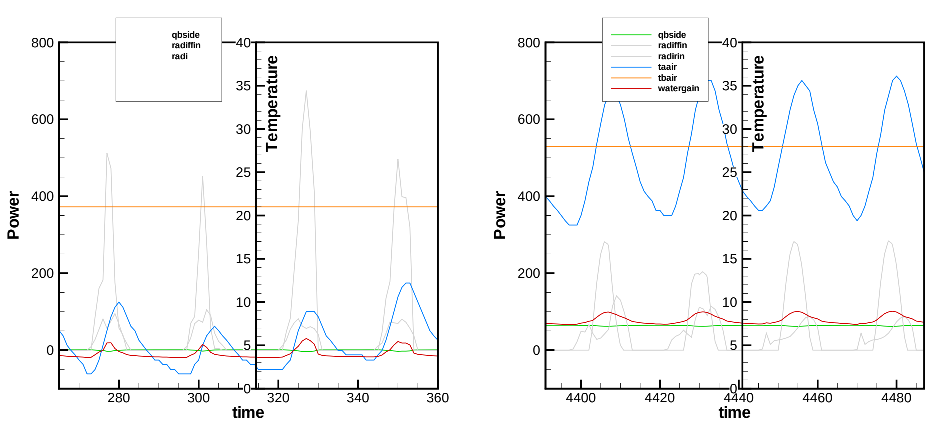

Table 2.8: Thermal behavior of CoolGlass in winter and summer simulated with IDA-ICE.

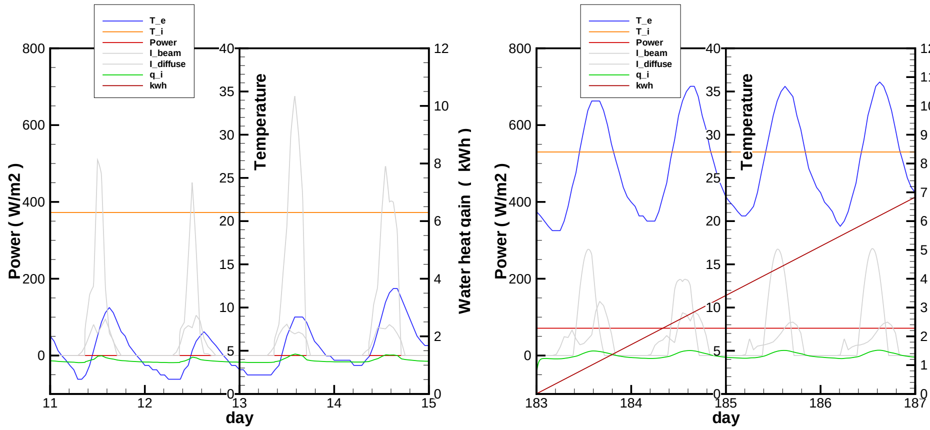

Table 2.9: Thermal behavior of the Reference glazing XTREME 60/28 in winter and summer period simulated with the tool.

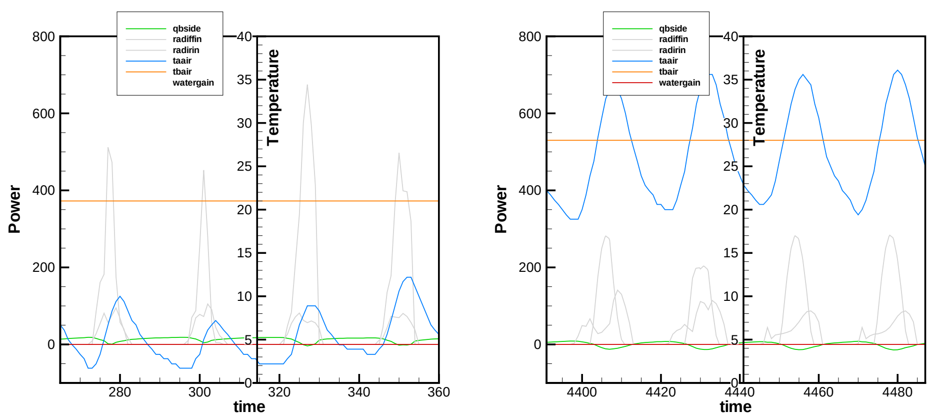

Table 2.10: Thermal behavior of the Reference glazing XTREME 60/28 in winter and summer period simulated with IDA-ICE.

2.2 Water flow glazing in an insulated room

When the glazing is part of an insulated room, the thermal problem of the glazing is coupled with the thermal problem of the room and the indoor temperature should be determined. Hence, the complete problem becomes more difficult and the indoor boundary condition disappears to be part of the solution of the thermal problem. To simplify this test case, a square room with a glazing facing south is considered.

2.2.1 Steady test cases

In these test cases, solar beam radiation is constant and normal to the glazing and diffuse radiation is set to zero. Besides, transport heat coefficients are considered constants. Regarding the water flow glazing, the same values as in the isolated water flow glazing cases are set.

When the thermal problem water flow glazing is coupled with the indoor thermal problem, the indoor temperature should be calculated at the same time the glazing temperature is determined. Besides, each wall has a different temperature due to the luminance of the direct beam sun radiation. Since the water flow glazing is facing south and the solar beam radiation is normal to the glazing, north facade is illuminated and some part of this impinging energy is absorbed. The rest of this energy is diffusely reflected and creates the indoor diffuse irradiance. Later, this irradiance is absorbed in each indoor surface. Hence, in this test case the water flow glazing absorbs extra energy form the indoor irradiance. In IDA-ICE, adiabatic walls are used.

Two different mechanisms appear to modify the thermal performances of the water flow glazing: *(i)* the value of the indoor temperature which can be very high depending on the wall insulation and *(ii)* the abosorbed back irradiance. Depending on the insulation of the walls and the near infrared absorptances, the water heat gain can be very different when compared to the water heat gain of an isolated glazing.

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	355.6	128.3	109.0	109.0	24.1	-	42.0
CoolGlass	133.4	129.2	109.8	109.8	22.1	-	40.2
iThermGlass	130.4	123.7	105.1	105.1	22.1	-	105.0
RadiaGlass	135.4	146.5	124.4	124.4	22.1	-	42.6
XTREME-60/28	0.0	151.3	128.5	128.5	0.0	0.0	107.2

Table 2.11: Steady thermal performances for a standard winter condition of a water flow glazing in an insulated room with the tool (ISO model and constant h_i and h_e).

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	354.2	128.4	111.1	111.1	23.9	286.6	40.6
CoolGlass	128.6	129.4	113.1	113.1	22.1	104.9	39.2
iThermGlass	95.0	123.7	108.9	108.9	21.8	29.2	101.3
RadiaGlass	127.6	146.5	126.7	126.7	22.1	31.7	41.2
XTREME-60/28	0.0	104.7	139.5	139.5	0.0	0.0	103.4

Table 2.12: Steady thermal performances for a standard winter condition of a water flow glazing in an insulated room with IDA-ICE (CESIMZON).

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	532.2	171.1	145.4	145.4	21.5	-	45.4
CoolGlass	232.0	172.3	146.4	146.4	18.9	-	43.1
iThermGlass	697.1	164.9	140.1	140.1	22.8	-	127.8
RadiaGlass	703.6	195.3	165.9	165.9	22.9	-	50.3
XTREME-60/28	0.0	201.7	171.4	171.4	0.0	0.0	175.7

Table 2.13: Steady thermal performances for a standard summer condition of a water flow glazing in an insulated room with the tool (ISO model and constant h_i and h_e).

Glazing	$P(\mathrm{W/m^2})$	$TI_b(\mathrm{W/m^2})$	q_i^{ON}	q_i^{OFF}	θ_w	θ_s	T_a
HeatGlass	520.9	171.2	148.2	148.2	21.3	405.3	43.6
CoolGlass	220.5	172.3	150.8	150.8	18.8	154.5	41.9
iThermGlass	642.7	164.9	145.1	145.1	22.4	72.2	119.5
RadiaGlass	691.6	195.3	168.9	168.9	22.8	74.5	47.8
XTREME-60/28	0.0	203.1	186.0	186.0	0.0	0.0	146.1

Table 2.14: Steady thermal performances for a standard summer condition of a water flow glazing in an insulated room with IDA-ICE (CESIMZON).

2.2.2 Unsteady test cases

In these following test cases, outdoor temperature and solar irradiance vary during the day and thermal performances are functions of time. The indoor temperature is unknown and it should be obtained at the same time than the glazing temperature profile.

These test cases are simulated in Madrid with the standard weather EPW file (EnergyPlus Weather). Regarding the water flow glazing, the flow rate and the inlet temperature are constant values. Two simulations in winter and in summer are accomplished. The simulation winter period runs from January 12th to January 16th. The simulation summer period runs from July 3th to July 7th.

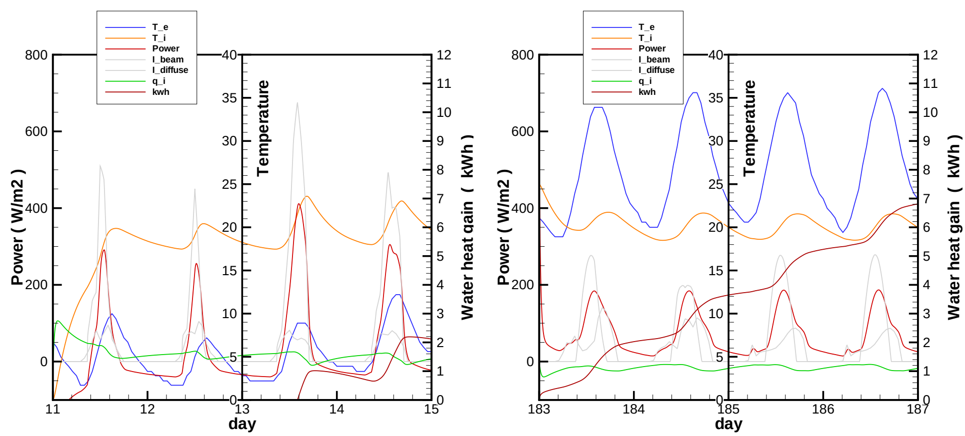

Table 2.15: Thermal behavior of HeatGlass in an insulated room in winter and summer period with the tool.

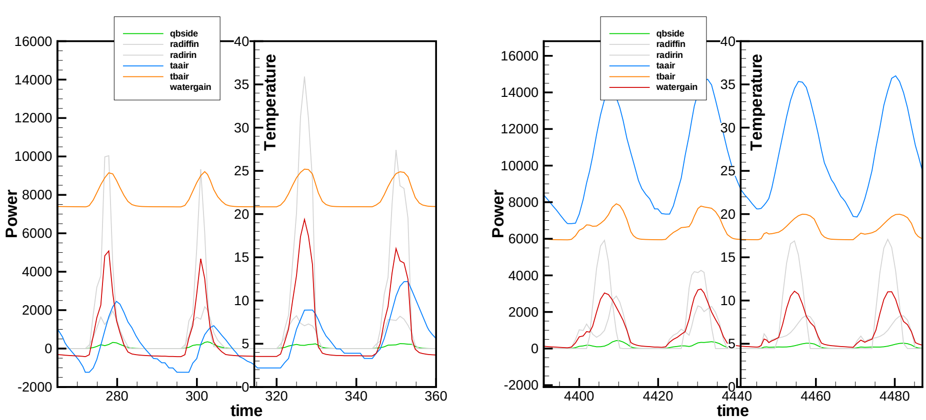

Table 2.16: Thermal behavior of HeatGlass in an insulated room in winter and summer period with IDA-ICE.

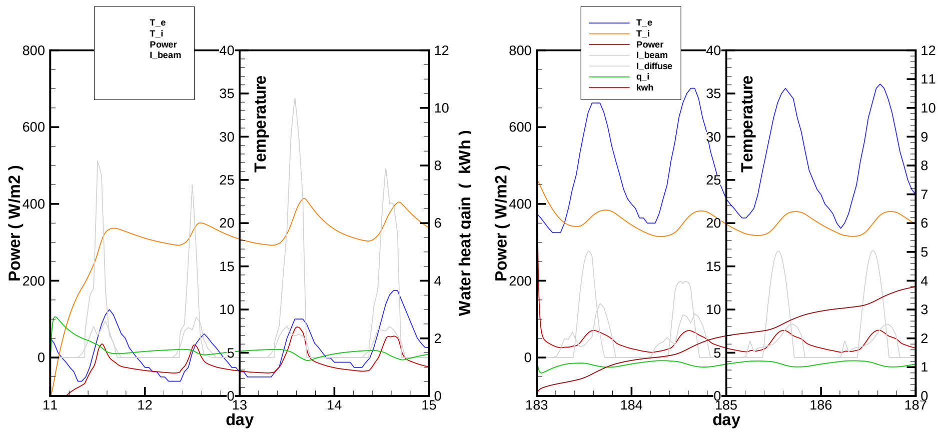

Table 2.17: Thermal behavior of CoolGlass in an insulated room in winter and summer period with the tool.

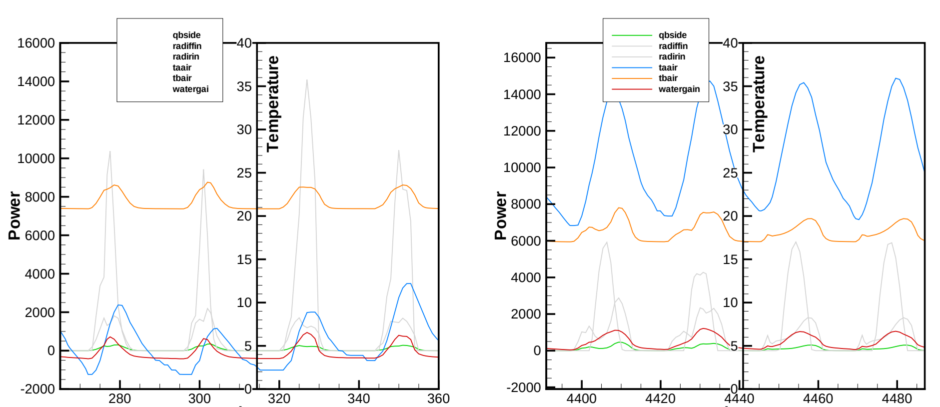

Table 2.18: Thermal behavior of CoolGlass in an insulated room in winter and summer period with IDA-ICE.

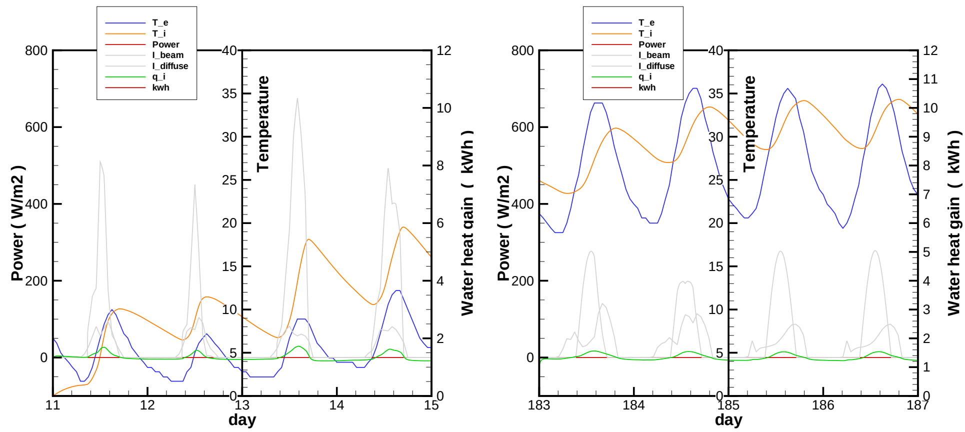

Table 2.19: Thermal behavior of the Reference glazing XTREME 60/28 in an insulated room in winter and summer period with the tool.

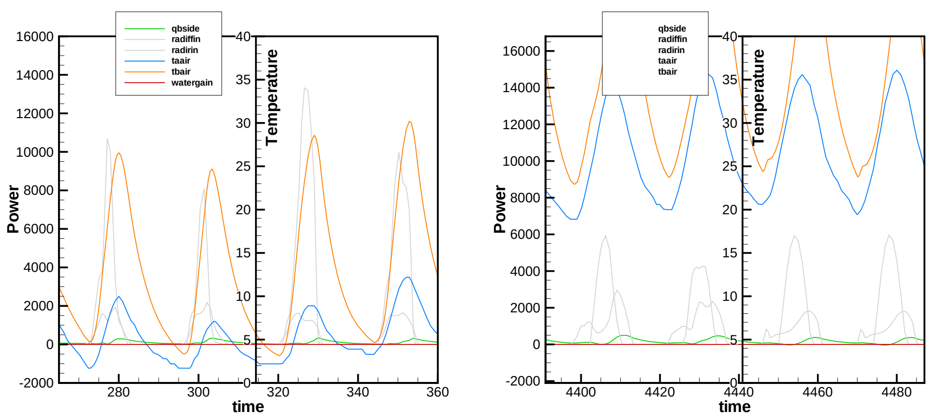

Table 2.20: Thermal behavior of the Reference glazing XTREME 60/28 in an insulated room in winter and summer period with IDA-ICE.

2.3 Reference Glazing

Detwind and Wintype models have been defined in IDA-ICE with the same layers and properties than the WFG Reference glazing. Simulation results have been compared with those in the tool. As in previous projects, two different test cases are considered: isolated glazing and glazing in a insulated room.

2.3.1 Isolated reference glazing

In order to validate the simulation of the complete thermal problem, a simple isolated glazing is considered. Hence, temperature and irradiance are considered given values . This is equivalent to impose indoor boundary conditions for an isolated glazing. Moreover, two different test cases are considered: steady and unsteady boundary conditions.

Steady test cases

If boundary conditions are constant, the solution becomes constant once the steady solution is reached. In this set of test cases, normal angle of incidence is considered to eliminate the uncertainties associated to the dependence of the absorptance of each layer with the angle of incidence.

Two steady states are considered: winter and summer. In the following table, outdoor and indoor boundary conditions which define these two seasons are fixed.

Season	θ_e	θ_i	$h_e(W/m^2K)$	$h_i(W/m^2K)$	$I_b(W/m^2)$	$I_d(W/m^2)$
Winter	5	21	23	8	0	75
Summer	35	28	23	8	200	0

Table 2.21: Outdoor boundary conditions for winter and summer test cases with diffuse radiation.

Once the glazing reaches the steady state in winter and in summer, thermal performances are determined. The following table shows water heat gain P, transmitted solar beam radiation TI_b and room thermal heat gain q_i for a winter and a summer condition for different glazings.

Glazing	$TI_b(\mathrm{W/m^2})$	$q_ia(\mathrm{W/m^2})$	$q_ib(\mathrm{W/m^2})$	θ_{se}	θ_{si}
Reference	16.6	37.3	13.7	6.5	19.5
HighSolarGain	37.1	23.7	9.6	6.0	20.0

Table 2.22: Steady thermal performances of a reference glazing for a standard winter condition with the tool.

Glazing	TI_b(W/m^2)	$q_i a$(W/m^2)	$q_i b$(W/m^2)	θ_{se}	θ_{si}
Reference	15.6	-38.9	14.4	6.7	19.2
HighSolarGain	37.0	-26.4	13.7	6.1	19.3

Table 2.23: Steady thermal performances of a reference glazing for a standard winter condition with IDA ICE.

Glazing	TI_b(W/m^2)	$q_i a$(W/m^2)	$q_i b$(W/m^2)	θ_{se}	θ_{si}
Reference	50.4	51.9	11.0	37.5	30.0
HighSolarGain	112.4	13.6	21.7	35.2	31.0

Table 2.24: Steady thermal performances of a reference glazing for a standard summer condition with the tool.

Glazing	TI_b(W/m^2)	$q_i a$(W/m^2)	$q_i b$(W/m^2)	θ_{se}	θ_{si}
Reference	49.7	-51.1	-13.8	37.2	29.7
HighSolarGain	116.1	-10.2	-21.0	35.4	30.6

Table 2.25: Steady thermal performances of a reference glazing for a standard summer condition with IDA ICE.

2.3.2 Insulated reference glazing

When the glazing is part of an insulated room, the thermal problem of the glazing is coupled with the thermal problem of the room and the indoor temperature should be determined. Hence, the complete problem becomes more difficult and the indoor boundary condition disappears to be part of the solution of the thermal problem. To simplify this test case, a square room with a glazing facing south is considered.

Height(m)	Length(m)	Width(m)	α_{NIR}	α_{FIR}	$U_{wall}(W/m^2K)$
1	1	1	0.4	0.9	0.3

Table 2.26: Room dimensions, spectral and thermal properties of walls for reference glazings validation.

Steady test cases

In these test cases, solar radiation is constant and normal to the glazing. Besides, transport heat coefficients are considered constants. Outside temperature is considered a constant value.

Once the glazing reaches the steady state in winter and in summer, thermal performances are determined. The following table shows water heat gain P, transmitted solar beam radiation TI_b and room thermal heat gain q_i for a winter and a summer condition for different glazings.

Glazing	TI_b(W/m^2)	G_d	q_ia(W/m^2)	q_ib(W/m^2)	T_a
Reference	16.6	3.8	28.3	3.6	11.0
HighSolarGain	37.1	8.5	22.6	6.5	19.3

Table 2.27: Steady thermal performances of a reference glazing in an insulated room for a standard winter condition with the tool.

Glazing	TI_b(W/m^2)	G_d	q_ia(W/m^2)	q_ib(W/m^2)	T_a
Reference	17.7	4.1	-29.3	3.6	11.7
HighSolarGain	40.0	9.2	-24.3	9.9	18.8

Table 2.28: Steady thermal performances of a reference glazing in an insulated room for a standard winter condition with IDA ICE.

Glazing	TI_b(W/m^2)	G_d	q_ia(W/m^2)	q_ib(W/m^2)	T_a
Reference	50.4	11.6	77.6	11.4	52.9
HighSolarGain	112.4	25.9	62.5	22.0	77.4

Table 2.29: Steady thermal performances of a reference glazing in an insulated room for a standard summer condition with the tool.

Glazing	TI_b(W/m^2)	G_d	q_ia(W/m^2)	q_ib(W/m^2)	T_a
Reference	49.7	13.0	-82.6	14.2	54.6
HighSolarGain	116.1	29.0	-76.1	39.6	72.9

Table 2.30: Steady thermal performances of a reference glazing in an insulated room for a standard summer condition with IDA ICE.