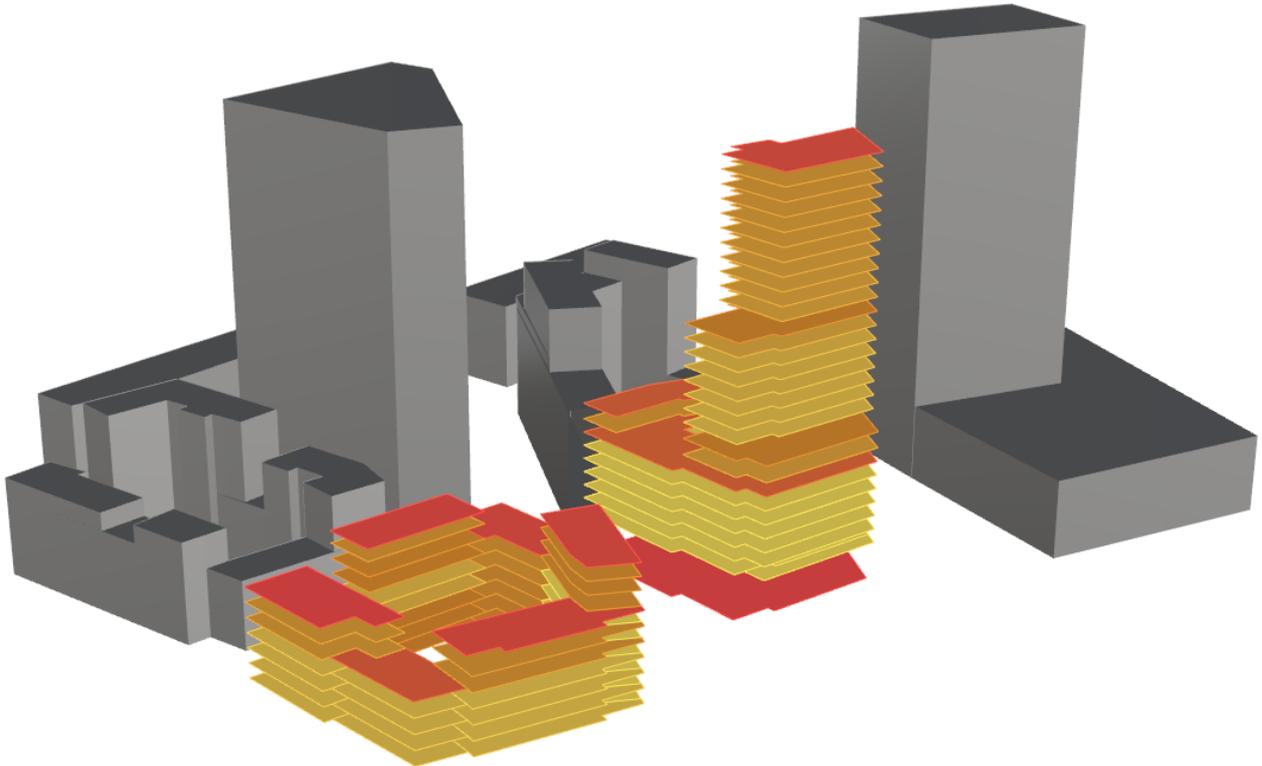




CHALMERS
UNIVERSITY OF TECHNOLOGY



Energy performance modelling

Introducing the Building Early-stage Design Optimization Tool (BeDOT)

Master's thesis in Structural Engineering and Building Technology

GIOVANA FANTIN DO AMARAL SILVA

RAMÓN BERGEL GÓMEZ

Department of Architecture and Civil Engineering

Division of Building Technology

Research Group Building Physics Modelling

CHALMERS UNIVERSITY OF TECHNOLOGY

Master's Thesis ACEX30-18-81

Gothenburg, Sweden 2018

MASTER'S THESIS ACEX30-18-81

Energy performance modelling

Introducing the Building Early-stage Design Optimization Tool (BeDOT)

Master's Thesis in the Master's Programme Structural Engineering and Building Technology

GIOVANA FANTIN DO AMARAL SILVA

RAMÓN BERGEL GÓMEZ

Department of Architecture and Civil Engineering

Division of Building Technology

Research Group Building Physics Modelling

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2018

Energy performance modelling

Introducing the Building Early-stage Design Optimization Tool (BeDOT)

Master's Thesis in the Master's Programme Structural Engineering and Building Technology

GIOVANA FANTIN DO AMARAL SILVA

RAMÓN BERGEL GÓMEZ

© GIOVANA FANTIN DO AMARAL SILVA & RAMÓN BERGEL GÓMEZ, 2018

Examensarbete ACEX30-18-81/ Institutionen för Arkitektur och samhällsbyggnadsteknik,
Chalmers tekniska högskola 2018

Department of Architecture and Civil Engineering
Division of Building Technology
Research Group Building Physics Modelling
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:

Cases studies A & B from the report.

Name of the publisher or Department of Architecture and Civil Engineering

Göteborg, Sweden, 2018

Energy performance modelling

Introducing the Building Early-stage Design Optimization Tool (BeDOT)

Master's thesis in the Master's Programme Structural Engineering Building Technology

GIOVANA FANTIN DO AMARAL SILVA

RAMÓN BERGEL GÓMEZ

Department of Architecture and Civil Engineering

Division of Building Technology

Research Group Building Physics Modelling

Chalmers University of Technology

ABSTRACT

The building stock is responsible for approximately 40% of energy consumption in Europe (Yang, et al., 2014). This has led to directives requiring that buildings perform ever more efficiently regarding energy consumption. In this context, greater is the importance of developing strategies for early design stages, especially improving the communication between architects and engineers. Decisions made early on in design have large impact over the project's final cost and performance, while design effort is still low. This project introduces the Building Early-stage Design Optimization Tool (BeDOT), a new set of tools for building early-stage design analysis, currently focused on energy performance modelling. It aims at being a broadly flexible and fast-running alternative in the process of decision-making based on building physics.

The model, programmed in Python and built upon ISO 13790:2008, has been implemented within Rhino/Grasshopper interface, which opens many possibilities for result visualization and optimization, besides easy geometry editing. BeDOT has been developed focusing on modularity, flexibility, fast processing and on the cooperation of professionals within the same environment.

BeDOT has been validated against IDA-ICE for over one thousand cases. Energy results lie within a range of 15% accuracy, with a slight tendency towards underestimation. The process of modelling a standard building from geometry to results usually takes a couple of hours, while calculations are performed in a matter of minutes. BeDOT's process and accuracy are suitable for early design stages. Further development is needed to improve the handling of data and the program's user-friendliness.

Key words: Energy modelling, Early-stage, Grasshopper, Building Efficiency

Contents

ABSTRACT	II
CONTENTS	III
PREFACE	VIII
GLOSSARY	IX
1 INTRODUCTION	1
1.1 Background	1
1.2 Goals	1
1.3 Thesis' boundaries	2
1.4 Methodology	3
2 BEDOT	4
2.1 Building energy modelling methods	4
2.2 Idealized features	5
3 BUILDING ENERGY MODEL	7
3.1 Modelling environment	7
3.2 Introduction to ISO, 5R1C model	10
3.3 6R1C-model	12
4 PRE-PROCESSING OF INPUT DATA	14
4.1 Data Structure	14
4.2 Building zoning and glazing	14
4.3 Internal gains	17
4.4 Ventilation	18
4.5 Solar radiation	19
4.6 Input data	20
5 SIMULATION	22
5.1 Heat balance calculation component	23
5.2 Heating Ventilation and Air Conditioning (HVAC)	30
5.2.1 Splitting the HVAC component	30
5.2.2 Hygienic Airflow component	31
5.2.3 Complementary Airflow component	36

6	VALIDATION	37
6.1	Verification simplified hourly method	37
6.2	Energy balance verification	39
7	PROCESS STUDY	40
7.1	Components' dependencies	40
7.2	Building process	42
7.3	Kunskapsdagen workshop	42
8	RESULTS	44
8.1	Solar gains nodal distribution in ISO 13790	44
8.2	Analysis on Air Handling Unit components	50
8.3	Sensitivity analysis of heat balance	57
8.3.1	Zone comparison	59
8.4	Case Studies	61
9	DISCUSSION	67
9.1	Strengths	67
9.2	Limitations	68
9.3	Further development	70
10	CONCLUSIONS	71
11	REFERENCES	73
	APPENDIX A INPUT DATA CASE STUDIES	75
A.1	Case Study A1	75
A.2	Case Study B1	77

Figure 1: Contextualization of current work and BeDOT.	2
Figure 2: BeDOT principles.....	6
Figure 3: Interface and its components.	7
Figure 4: Scheme of BeDOT – Energy.....	9
Figure 5: RC-model (5R1C). Source: ISO 13790:2008 (modified)	10
Figure 6: RC-model (5R1C)	10
Figure 7: 6R1C network (Nunes & Oliveira Panão, 2013).....	12
Figure 8: Difference between the 5R1C and the 6R1C.	13
Figure 9: Geometry in Rhino 3D	15
Figure 10: Pattern of zone naming	16
Figure 11: Geometry information from Rhino 3D.....	17
Figure 12: Obstacle modelling in one of the facades.....	20
Figure 13: Heat balance calculation split in two components.	22
Figure 14: Heat balance calculation component.....	22
Figure 15: Step calculation of mass temperature.....	26
Figure 16: Flowchart of heat balance calculation	28
Figure 17: Hygienic airflow component.....	31
Figure 18: Sketch on AHU components and naming pattern.....	32
Figure 19: Air heating in a Mollier diagram. Source: (Abel E., 2015).....	34
Figure 20: Air cooling in a Mollier diagram. Left: without condensation. Right: with condensation. Source: (Abel E., 2015)	35
Figure 21: Verification zone	37
Figure 22: Gantt chart of building energy calculations in BeDOT.....	40
Figure 23: Running times for a case with 100 zones.....	41
Figure 24: Kunskapsdagen workshop geometries.	43
Figure 25: Total airflow through the AHU for different nodal distributions.....	44
Figure 26: Case 1, temperature in a room facing south.	45
Figure 27: Case 1, Airflow through the AHU for different nodal distributions.....	46
Figure 28: Case 1, AHU Heating & Cooling for different nodal distributions.....	46
Figure 29: Case 2, temperature in a room facing south.	47
Figure 30: Case 2, Airflow through the AHU for different nodal distributions.....	47
Figure 31: Case 2, District Heating & Cooling for different nodal distributions.....	48
Figure 32: Case 2, transmissions through the different building elements.....	48
Figure 33: Comparison of different cases with solar gains distributed equally between surface and air nodes in BeDOT and IDA.....	49

Figure 34: Case 9, mean air temperature in a zone facing south.	51
Figure 35: Case 9, room heating and cooling power.	51
Figure 36: Case 10, mean air temperature in a zone facing south.	52
Figure 37: Case 10, room heating and cooling power	53
Figure 38: Case 10, AHU heating and cooling power	53
Figure 39: Case 11, AHU heating and cooling power.	54
Figure 40: Case 11, recovered heat.	54
Figure 41: Case 12, mean air temperature in a zone facing south.	55
Figure 42: Case 12, airflow rate.	55
Figure 43: Case 12, room heating power	55
Figure 44: Internal Heat Capacity analysis. Heating	56
Figure 45: Internal Heat Capacity analysis. Cooling	56
Figure 46: Results of sensitivity analysis of heat balance.	57
Figure 47: Sensitivity analysis of Room Heating.	58
Figure 48: Sensitivity analysis of Room Cooling.	58
Figure 49: Heating power errors. Comparison among zones.	59
Figure 50: Cooling power errors. Comparison among zones.	59
Figure 51: Spread analysis on heating power relative errors.	60
Figure 52: Spread analysis on cooling power relative errors.	60
Figure 53: Case study A, geometry.	61
Figure 54: Case study A, heat losses through the building envelope.	62
Figure 55: Case study A, energy results	63
Figure 56: Case study A, inlet temperature to the air cooler.	63
Figure 57: Case study B, geometry in IDA ICE.	64
Figure 58: Case study B, solar gains analysis.	65
Figure 59: Case study B, transmission through building elements.	65
Figure 60: Case study B, energy results.	66

Table 1: Variation in heating and cooling power results when comparing theoretic models and assumed room air temperatures.	13
Table 2: Input data gather in one envelope zone type.	20
Table 3: Input data gather in internal gains zone type.	21
Table 4: Input data gather in indoor climate zone type.	21
Table 5: Input data gather in ID zone type.	21
Table 6: Building classes. Internal capacity and mass area.	23
Table 7: Input data for the validation case according to ISO 13790:2008	37
Table 8: Results from verification model	38
Table 9: Steps for heat balance verification.....	39
Table 10: Nodal distribution analysis.	44
Table 11: Case 1, inputs for BeDOT and IDA.....	45
Table 12: Case 9, inputs for BeDOT and IDA ICE.	50
Table 13: Summary of main differences in cases 10 to 12.	52
Table 14: Sensitivity analysis. Combinations of parameters	57

Preface

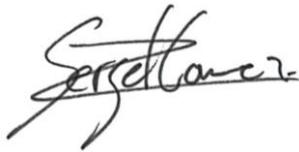
This project has been carried out at the Department of Building Technology, Chalmers University of Technology, Sweden, in collaboration with Bengt Dahlgren AB.

We would like to thank our examiner and supervisor Angela Sasic Kalagasidis for her guidance and support during this thesis work and all the help received during our 2-year Master Program.

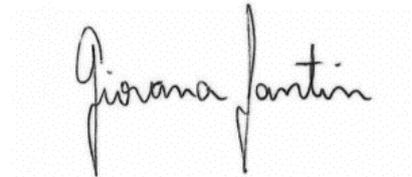
We would also like to thank Max Tillberg and Carl Molander for the constant ideas and feedbacks that helped pushing the project forward, and for valuable practical knowledge in the field of energy modelling.

Last but not least, our special thanks also to our families and friends, with whom we share the best laughs even in the darkest winters. Thank you for being with us all along the way supporting the creation of BeDOT.

Göteborg 2018



Ramón Bergel Gómez



Giovana Fantin do Amaral Silva

Glossary

Report	Description	Unit
A_f	Floor area of each thermal zone	m^2
A_m	Considered mass area that is exposed to each thermal zone	m^2
A_t	Area of all the surfaces facing the building zone	m^2
A_{roof}	Area of the roof	m^2
A_{wall}	Area of the external wall	m^2
A_{window}	Area of the window	m^2
b_{ek}	Adjustment factor to account for different airflow sources	-
c_a	specific heat capacity of air	J/kgK
C_m	Internal heat capacity of each thermal zone	J/K
density	Amount of people per square meter	m^2/p
$g_{skylight}$	Solar heat transmittance of skylights	-
g_{window}	Solar heat transmittance of windows	-
h_{is}	Heat transfer coefficient between the air and the surface node	W/m ² K
h_{ms}	Heat transfer coefficient between the mass and the air node	W/m ² K
$H_{tr,bridges}$	Conductance due to thermal bridges	W/K
$H_{tr,em}$	Conductance of the exterior wall, roof and thermal bridge	W/K
$H_{tr,is}$	Conductance between the air node and the surface node	W/K
$H_{tr,ms}$	Conductance between the mass node and the surface node	W/K
$H_{tr,w}$	Conductance of the window	W/K
$H_{tr,roof}$	Conductance of the roof	W/K
$H_{tr,wall}$	Conductance of exterior walls	W/K

H_{ve}	Ventilation conductance	W/K
$H_{ve,hyg}$	Hygienic airflow conductance	W/K
$H_{ve,leak}$	Infiltration airflow conductance	W/K
people	metabolic heat load (sensible)	W/p
P_{tot}	Barometric pressure	hPa
P_{ws}	Water vapor saturation pressure	hPa
T_{air}	Temperature of the air node	°C
$T_{air,0}$	Temperature of the air node in free floating	°C
$T_{air,10}$	Temperature of the air node when 10 W/m ² are applied	°C
T_{dew}	Dewpoint temperature	°C
T_m	Average mass temperature	°C
$T_{m,t}$	Mass temperature of the time step t	°C
$T_{m,t-1}$	Previous mass temperature of the time step t	°C
$T_{outside}$	Outdoor Temperature	°C
T_{rec}	Temperature after heat recovery	°C
T_{return}	Temperature of exhaust air	°C
T_s	Temperature of the surface node	°C
$T_{set,HC}$	Set point temperature for heating or cooling	°C
T_{sup}	Supply temperature	°C
U_{roof}	Thermal transmittance of the roof	W/m ² K
$U_{skylight}$	Thermal transmittance of the skylights	W/m ² K
U_{wall}	Thermal transmittance of the wall	W/m ² K
U_{window}	Thermal transmittance of the window	W/m ² K
ΔU_{roof}	Thermal bridge of the roof as function of the U_{roof}	W/m ² K

ΔU_{wall}	Thermal bridge of the walls as function of the U_{wall}	W/m ² K
η	Efficiency of the heat recovery	-
ρ_a	density of the air	m ³ /kg
\dot{v}	Hygienic and infiltration airflow sources	l/s
\dot{v}_{area}	Area component of hygienic airflow	l/sm ²
\dot{v}_{hyg}	Hygienic airflow	l/sm ²
\dot{v}_{leak}	Infiltration through the envelope	l/sm ² (façade)
\dot{v}_{people}	People component of hygienic airflow	l/sm ²
$\phi_{HC,nd}$	Heating or cooling power	W
$\phi_{HC,nd,un}$	Heating or cooling unrestricted power	W
ϕ_{ia}	Internal loads and solar gains applied in the air node	W
ϕ_{int}	Internal loads	W
ϕ_m	Internal loads and solar gains applied in the mass node	W
ϕ_{people}	Human load	W/m ²
ϕ_{sol}	Solar gains	W
ϕ_{st}	Internal loads and solar gains applied in the surface node	W

1 Introduction

1.1 Background

About 40% of the energy consumption in Europe is derived from Buildings (Yang, et al., 2014) where most of the stock was not designed based on efficiency principles (EU Commission, 2015). Bearing this in mind, the European Commission has recently released statements and directives to improve the energy efficiency of buildings. The last one, the European Performance Building Directive (EPBD) 2010/31/EU, embraces the member states to follow a strategy for more efficient buildings designs.

The EPBD supports a series of standards (Roulet & Anderson, 2006) and paths to accomplish with regulations and future goals of the EU on an energy framework (EU Commission & Council, 2014). These future goals can be gathered in the Energy strategy for 2020 and 2030, where the member states agreed on implementing measures to achieve energy savings and reduction in greenhouse emissions (EU Commission & Council, 2014)

The directives highlight the importance of calculation methodology, energy performance requirements and certification. In this project, the calculation methodology would be based on the standard ISO 13790:2008 Energy Performance of buildings – Calculation of energy use for space heating and cooling.

This project intends to give an understanding on early design stages simulations, where the main intention is to provide an idea on the energy consumption of a building. At this stage, decisions taken by designers or architects have a large impact on the final design of the building, while the cost of decisions and design effort are still very low (Li, 2017). This motivates the developing of tools for early design stages, where there is great potential for saving resources and time in the design process and make easier the collaboration among engineers, architects and energy specialists.

1.2 Goals

The main goal is to develop an energy performance model and thus create the Building Early-stage Design Optimization Tool (BeDOT). The energy model developed in this thesis work was the origin of BeDOT, and as for now both things are the same. Additional aims include:

1. To introduce the Building Early-stage Design Optimization Tool (BeDOT), a comprehensive project which is to include analyses of daylight, life cycle and emission systems.
2. To have this tool used by students, engineers and architects as an initial step in the pursuit of more effective buildings.
3. To obtain an accurate result in energy estimation, preferably tending to slight overestimation when compared to reference values.
4. To provide results within a computational timeframe of minutes, instead of the common hours or days that more detailed Building Energy Simulation (BES) tools often require.
5. To provide conditions for broad flexibility, both in the process of using the tool and within the model itself. The user should be able to easily replace

components or edit data at any desired point in the simulation. The model should be robust to work with a wide range of input data, climate characteristics, geometry and system settings.

6. To provide a user-friendly, intuitive interface, which does not require from the user a deep knowledge on BES tools.
7. To bear simulations containing a large number of zones (more than 100) and have equal calculation performance when handling a heavy set of data.
8. To have the tool integrated to a 3D modelling interface. In this way, it being able to both “read” information graphically provided by the user and return results visually onto the same model.
9. To consolidate the authors’ previous learning within the fields of building physics and building services.

1.3 Thesis’ boundaries

This report describes BeDOT, a tool that is envisioned to have a wide variety of analyses concerning building technology. However, the current work is mainly focused on the modelling of building energy performance, which represents a section of the whole project. Since this project’s energy model is also the origin of BeDOT, at this stage they correspond to the same thing, as seen in Figure 1. In the future, if the project continues, BeDOT could also comprise life-cost assessment, daylight and can also include optimization tools.

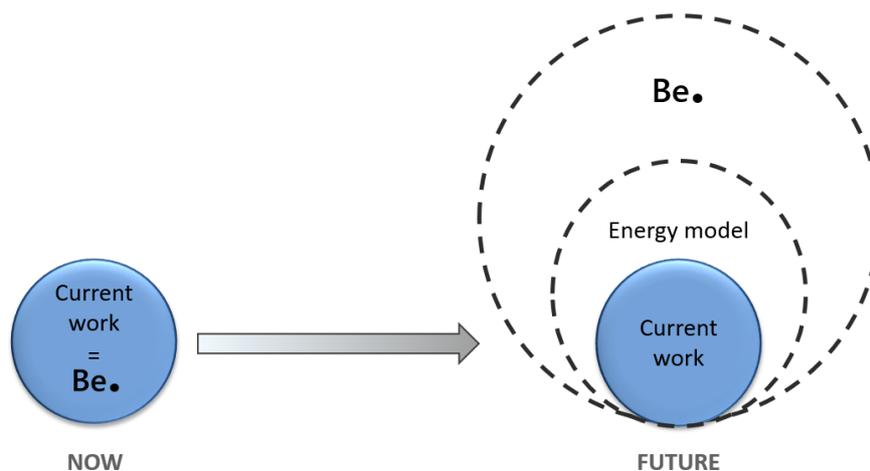


Figure 1: Contextualization of current work and BeDOT.

Although very relevant to BeDOT’s implementation and validation, some studies were not included in this report, namely:

- Comparison between available solar radiation models;
- Heat transfer to the ground;
- Thermal comfort;
- Daylight;
- Calculations of primary energy systems;
- Solar shadings control system;
- Uncommon or extreme design cases;
- Heat loss due to sky night radiation;
- Detailed windows design (frames, solar transmittance or emissivity)

1.4 Methodology

In this project, a building energy performance model based on ISO 13790/2008 simplified hourly method has been implemented in the CAD tool *Rhinoceros*. This software has been chosen for its wide range of possibilities for visualizing inputs and results in a graphic interface, and also for its compatibility to import and export files.

Rhinoceros also counts with a Visual Programming Language (VPL) plugin named *Grasshopper*. Within this interface, the model is written in *Python*, a programming language that was chosen for being easy to learn and friendly to beginners. Additionally, other open-source plug-ins have been used, such as *Honeybee*, *Ladybug* and *Daysim*. The program makes use of mass modelling combined with a façade model, also leaving possibilities for future implementation of more detailed analyses, such as representative rooms.

The model's heat balance has been verified against ISO's standard case. Then, it was validated multiple times against IDA-ICE, a comprehensive building energy simulation software very commonly used in Sweden. Sensitivity analyses were conducted for over one thousand cases. Finally, the model was applied on two study cases, of more varied geometry and input data, and both energy results and the modelling process were compared to equivalent projects carried out in IDA-ICE.

2 BeDOT

The Building Early-stage Design Optimization Tool (BeDOT) is a program comprised of a set of tools, aiming at being an alternative for architects and engineers in process of decision-making based on building physics.

BeDOT is built using a VPL named *Grasshopper*, which is a plugin of the CAD tool *Rhinoceros*. Within this interface, components are coded in the programming language Python or imported from available open-source plug-ins such as *Honeybee*, *Ladybug* or *Daysim*. Thus, *Grasshopper* allows to make all the calculations needed in an open-loop system, this means that the information and calculations are going downstream.

Building BeDOT in *Grasshopper* was decided since it is a platform that allows interaction with the CAD tool *Rhinoceros*, where broad visualization possibilities regarding inputs or outputs would help engineers and architects in this design phase.

Similarly to other early-stages tools, BeDOT makes use of mass modelling, sectioning large masses into smaller ones, called zones, which are independent of each other. A façade model is applied to take into account effects of solar radiation as it reaches the building surface. This procedure already accounts for the building's position and eventual shading obstacles. BeDOT's logic works from outside in. This means that, at this stage, zones do not necessarily correspond to rooms, which makes it unrealistic to try and estimate indoor parameters, such as thermal comfort.

However, BeDOT remains open for the implementation of more detailed analysis, for example using representative rooms, which would help verifying thermal comfort and daylight parameters.

2.1 Building energy modelling methods

Many studies on short-term building energy modelling have proposed methods to improve the design and operation of buildings and its installations. (Li & Wen, 2014) classify the used methods as white, black or grey box.

White boxes are methods which are strictly fundamented on physical equations. Among the most common are computational fluid dynamics (CFD), the zonal method and the multi-zone method, which can be found in software like COMSOL Multiphysics (CFD), EnergyPlus (multi-zone) and IDA-ICE (multi-zone). Although white boxes tend to be very accurate and effective, they come with three challenging drawbacks: the analyses are time-consuming, need detailed building parameters and the work of a higher-qualified designer.

On the other hand, methods that rely entirely on the processing of empirical data are called black boxes. In these, energy data is collected from existing buildings, and treated statistically to predict future behaviors. Some of the statistical methods used can be the conditional demand analysis (CDA), genetic algorithms (GA) and artificial neural networks (ANN). Black box models usually demand a long algorithm training period. They are able to adapt to abrupt pattern changes, but will only predict events within the limits of the empirical data.

Lastly, grey boxes are hybrid models that combine a simplified physical description, often in the form of thermal networks, and an optimization algorithm based on statistics. Taking as an example the work of (Braun & Chaturvedi, 2002), in which a

thermal network was used, the resistances and capacitances can be obtained by statistical treatment (i.e. regression analysis) of measured data. Then, this newly acquired coefficients are combined in the physical model, enabling the simulation of building operation with better accuracy. The simplification of the physical model allows the algorithm's training period to be reduced, therefore making calculations faster.

Focusing on optimizing building parameters in an early design stage, the designer's ambition is to obtain quick answers on which building parameters to prioritize or adjust, based on a non-detailed building geometry and materials. In this sense, white box methods might not be the most adequate, as such simulations often take long time and require the knowledge of construction details. Similarly, black box methods are also difficult to implement. They require an extensive database, which might not be available, or that has low reliability if the buildings' characteristics are not very similar to each other.

In early design stages, simplified white box methods could highly contribute to a fast processing of a simulation based on rough building characteristics. The simplified physical model is comprised of several types of analyses that could run in parallel: energy performance, daylight, indoor thermal comfort, primary energy systems, etc. which are supplied with inputs from outdoor conditions, system boundaries, desired indoor conditions and systems' characteristics. Assuming these analyses are independent and satisfactorily quick, it should be possible to perform them repetitively, having some of the building parameters altered at each time to evaluate the results. At this point, a statistical analysis can be implemented in the model, such as in a black box method, to manage the adjustment of parameters and conduct the post-processing of results. The optimization phase would therefore aim to achieve the most effective combination of parameters according to designer's priorities, which could be total cost, carbon footprint, energy consumption, PPD or other criteria.

This present study fits into the simplified white box phase, aiming to model building energy performance. In the future, BeDOT can integrate other white or grey models, and even implement black models of optimization processes, in this way consolidating itself as a grey box method.

2.2 Idealized features

The development of BeDOT followed 5 main principles (see Figure 2):

1. **Time & Simplicity:** The program constantly aimed at simplified solutions, making both the process and calculations clear. From choosing the physical method to assigning inputs, all steps were thought aiming to reduce computational time and modelling time.
2. **Modularity:** Both the model and the program itself are designed to be modular. On the program level, BeDOT intends to work with modules. This means that the different modules and components that form BeDOT can be easily replaced for other components, as long as the inputs and outputs are maintained in the same structure as the original modules. Regarding the building model, the zones are all independent of each other. The model can

simulate hundreds of zones at the same time and assign different input data to each of them.

3. Post-processing: Grasshopper, together with Rhinoceros scene, allow the user to have broad visualization opportunities. The user can visualize the inputs and design specifications in the 3D geometry in Rhinoceros, which allows a more visual approach. Results can be plotted in the same way, or even using diagrams. Zones can be coupled in different system boundaries, so results are displayed per building, building section, AHU, real stage, etc...
4. Accuracy: BeDOT aims to have an adequate accuracy, also compatible with early-stage designs. Therefore, it is not expected to achieve the same results as more comprehensive software. In this tool, a 15% error margin between the validation software IDA ICE and BeDOT was considered valid, considering that the model is not as detailed as IDA and BeDOT is developed for early-stage.
5. Collaboration: Clear and constant communication between Architects and Engineers is beneficial to any project. Especially during early-stage designs, it is expected that many changes in the future characteristics of the building will come up. A good communication among professionals guarantees that these changes will improve the building's energy performance, instead of making it worse, which is why everybody work in the same environment, Rhinoceros. Also, this is why Python language was chosen. It is accessible to everyone and easy to learn and edit.



Figure 2: BeDOT principles.

3 Building energy model

The standard ISO 13790:2008 proposed three different methods to calculate the heat balance of the building (CEN, 2008): a) a simple hourly method, b) a monthly or seasonal method and c) a more detailed dynamic simulation.

In this project, the calculation method that is going to be implemented is the simplified hourly method. This is mainly because the project is based on energy simulations in early stages, which means that the model intends to have a lower computational time than other detail software. A more detailed dynamic simulation would account for thermal coupling between zones or linear thermal bridges (CEN, 2008).

To reduce simulation time, implementing the monthly basis method could have been an option since it is able to handle much less data. However, this method has a lower accuracy and does not account for day and night conditions, thus being almost never used in practice. Furthermore, variations in loads, ventilation and mass temperature require a shorter time step, so nowadays energy simulations are mostly hourly based. Also, one of the major features is the possibility of calculating the indoor temperature hourly.

Additionally, what is said to consume more time in the simulation is the interaction between calculations and 3D model geometry (Michalak, 2014), not the size of the time step.

3.1 Modelling environment

The tools used in this project are displayed in Figure 3. The CAD tool Rhinoceros is the base for everything that follows. Inside Rhinoceros, a Visual Programming Language plugin called Grasshopper is the platform in which all the programming is done. Within Grasshopper, part of the analysis is done with the aid of several other open-source plugins, Ladybug, Honeybee and Daysim. Alternatively, Python scripts can be manually implemented by the user.

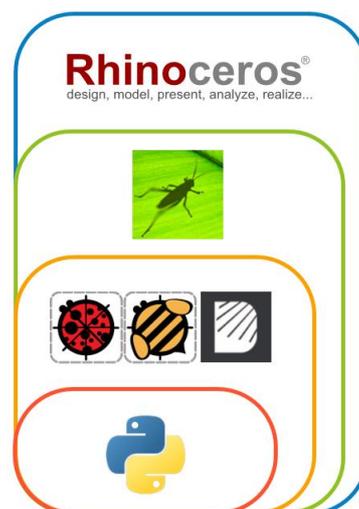


Figure 3: Interface and its components.

The outlines of the energy modelling part of BeDOT are shown in Figure 4. In the figure, it is possible to see the scheme of how the flow of information goes through the model. Then, it is shown that these modules can be implemented using *Grasshopper* components or python scripts.

For calculating energy and solar radiation, *Grasshopper* works along a series of plugins such as *Honeybee* and *Ladybug*. These plugins provide pre-made components that contain different calculation processes, focusing in energy simulation. Many of these components are of interest, for instance to read weather data, calculate solar radiation or split the building in thermal zones. The great availability of open-source plugins is one extra benefit of using *Grasshopper*, allowing the user to replace one method by another, if so desired.

The first two plugins used inside *Grasshopper*, *Ladybug* and *Honeybee*, were used for managing weather data and establishing the mass modelling, respectively. Regarding solar radiation, *Daysim* components were used. *Daysim* is a calculation module (Christoph Reinhart, u.d.) that provides illuminance or solar radiation in the different surfaces of a geometry. Further explanation on *Daysim* is provided in section 4.5.

The process of using *Daysim* to obtain the solar radiation falling over the zones' facade was carried out by Bengt Dahlgren, while the authors of this project took care of filtering the radiation per glazed area, and in an appropriate format. Respecting data format and structure is of great importance in this project, to guarantee the proper functioning of the program and support the principle of modularity that is shown in Figure 2.

Regarding the energy modelling, it has been developed in the outlines of ISO 13790:2008. This standard includes all the steps to calculate the heat balance of a building in a simplified manner. Although it also includes standardized input data, in this project data bases with more detailed input data are used. The ISO implementation has been done using compatible scripts, written in Python, and implemented by the authors in *Grasshopper* interface. All components are explained further on this report.

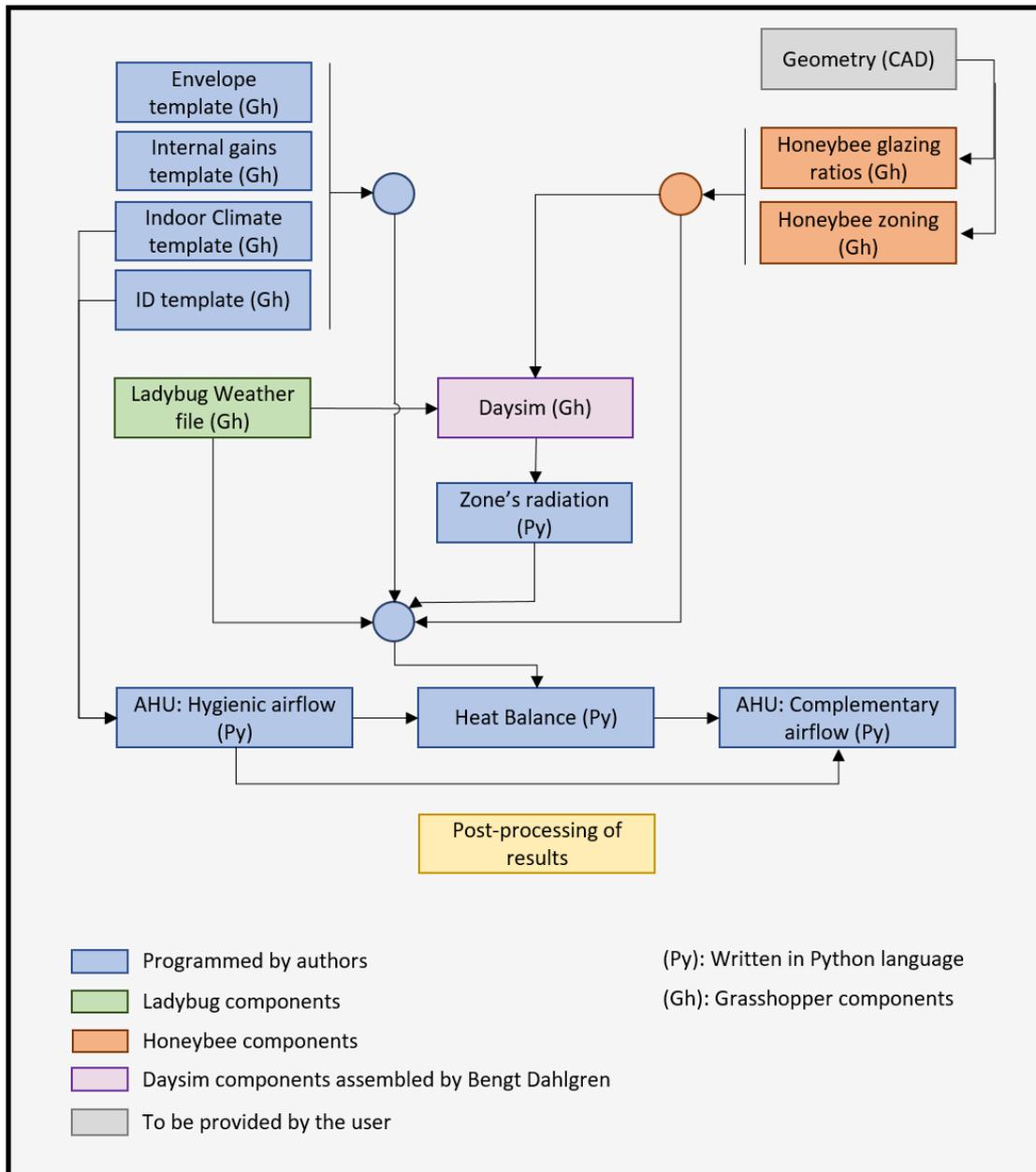


Figure 4: Scheme of BeDOT – Energy.

3.2 Introduction to ISO, 5R1C model

The calculation method established in ISO 13790:2008, called “simple hourly method” or 5R1C, is a simplified dynamic method expressed by the thermal network in Figure 5. The method is comprised of nodes, resistances and capacitances.

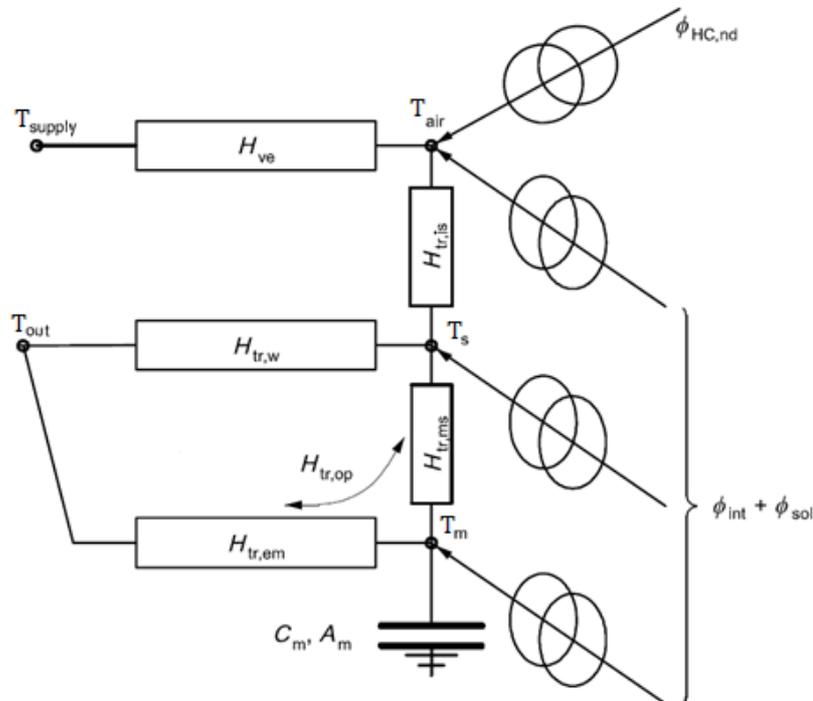


Figure 5: RC-model (5R1C). Source: ISO 13790:2008 (modified)

The five nodes represent the temperatures of the zone air (T_{air}), supply air (T_{supply}), outdoor air ($T_{outside}$), indoor surfaces (T_s) and thermal mass (T_m). The thermal mass node was included as an artifact to account for the building’s capacitance. It accounts for the simulation of the energy buffer effect that the internal mass has over the thermal balance, as a result of heat release and absorption. Half of internal gains (ϕ_{int}) are considered as being of convective nature, and so are directly assigned to the air node. The rest of internal gains, together with solar gains (ϕ_{sol}) are treated as radiative loads, divided between surface and mass nodes. The ratio at which this division occurs depends on the zone’s mass area and on the window resistance. A more visual RC-circuit is shown in Figure 6.

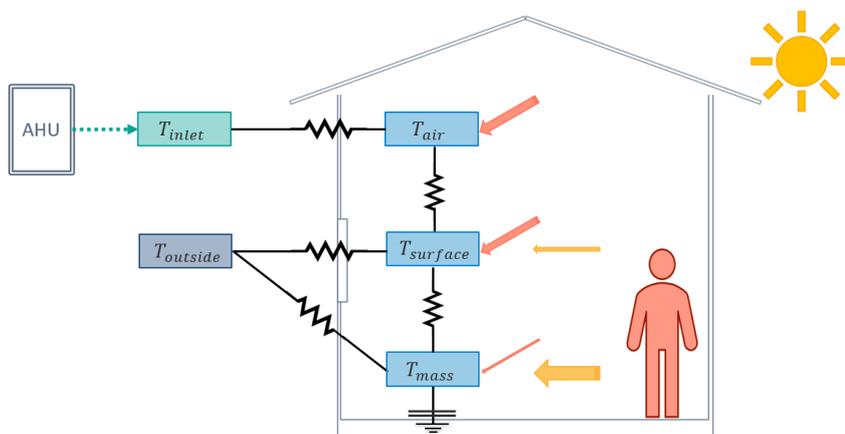


Figure 6: RC-model (5R1C)

In their work on the lumped model accuracy, (Kircher & Zhang, 2015) listed the following assumptions of the thermal network method:

1. Negligible conductive and radiative coupling between indoor surfaces
2. Convection between air and indoor surfaces behaves linearly
3. The continuous temperature distribution inside a wall can be approximated by a small number of lumps of uniform temperature.

Moreover, they claim the accuracy of the lumped model is adequate, as a rule of thumb, for elements whose Biot number is inferior to 0.1, which corresponds to materials that have a uniform temperature distribution. In this sense, the model assumes that, for a given layer, surface temperatures are close to the spatial average temperature. This simplification is indeed reasonable for materials placed between the insulation layer and the analyzed zone. Kircher and Zhang have tested lumped model's accuracy for a single material layer, through stationary and transient simulations. They concluded it is possible to maintain the error inferior to 5% even for cases where $Bi > 0.1$ on the surface that is exposed to temperature variations from outside, provided that the other surface's Biot number remains low. This corresponds to most cases of RC thermal networks in buildings, as the indoor conditions do not undergo many variations and convective heat transfer coefficient is often small.

The method's accuracy could be compromised by increased indoor convective coefficients or by very well-insulated building elements, comprised of many layers. For the 5R1C thermal network, a further source of error is the simplification of aggregating all thermal capacitances into one single element, regardless of their different boundary conditions.

ISO 52016:1, published in 2017, replaces and cancels ISO 13790. The method in ISO 52016:1 is based on knowing thermal properties of each construction element separately. This method would calculate nodal temperature through the different building elements, which are then included in the heat balance of the building. This method is close to more detailed methods since it calculates the surface temperature of each element of the envelope.

This allows each element to keep its own category of thermal capacity, instead of having the whole building roughly classified as one thermal mass. By defining each building element separately, each one can be assigned a different system boundary as well as other characteristics, such as solar absorption coefficients, materials or U-values

Although the new standard brings a more transparent and possibly more accurate analysis, ISO 13790's simple hourly method might still be used by designers for energy consumption estimation, especially in early design stages. The reason for this choice is indeed the required level of detailing, which is higher on the new standard and consequently not often compatible with a very early design phase. ISO 13790 method is therefore still well suited for analysis performed on simplified building geometry and unclear building element composition, in this way helping on the development of more refined studies, in which ISO 52016 is to be applied.

3.3 6R1C-model

In Figure 5, it is shown that ventilation conductance is in between supply temperature and air temperature in the room. This is then a limitation when airflows from other sources want to be included in the model, such as infiltration, which is a function of outdoor temperature. In a common case like this one, two airflows of different temperatures need to be implemented (ventilation and leakage), but the 5R1C model acknowledges only one ventilation resistance, and one temperature. The standard deals with this limitation by using an adjustment factor b_{ek} in clause 9.3.1 of ISO 13790:2008 equation 21. The factor is calculated according to the equation:

$$b_{ek} = (T_{set,HC} - T_{supply}) / (T_{set,HC} - T_{outside}) \quad (1)$$

Where,

$T_{set,HC}$ Is the heating or cooling set point in the zone in °C.

T_{supply} Is the supply temperature of the AHU in °C.

T_{out} Is the outdoor temperature in °C.

Since this factor is implemented to calculate the ventilation heat transfer coefficient (see section 5.1 for detailed explanation), it needs to be calculated before the heat balance of the room is performed. The first thing to notice is that the adjustment factor makes use of room setpoints as an estimation of the room air temperature, while it is unknown. This already brings an intrinsic inaccuracy, mostly for the hours in which the room air temperature is free-ranging between the two setpoints. Moreover, in a pre-calculation step, there is no way to accurately know which room setpoint, heating or cooling, should be used in equation (1) at a given time. Since the user is expected to choose, in an unspecified way, a setpoint value to calculate b_{ek} , it is likely that this factor varies a lot from user to user.

Another alternative for this problem was to adapt the model thermal network, dividing the ventilation branch into two parallel resistances, as done by (Węglarz & Narowski, 2011) and shown in Figure 7.

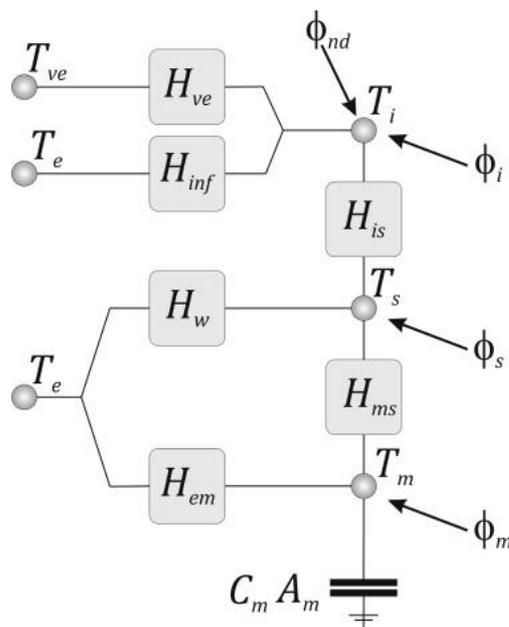


Figure 7: 6R1C network (Nunes & Oliveira Panão, 2013).

Including a sixth resistance would require a new balance, differing from that on the ISO 13790:2008. Their new set of equations allows the implementation of a second airflow source, which would dismiss the need for using the adjustment factor presented in equation (1).

To show how this factor affects the results, a series of test were simulated to evaluate the impact over the heating and cooling demands. For the 5R1C model, two different temperatures were used to calculate the adjustment factor: 20°C (assumed heating set-point), and 26°C (assumed cooling set-point). These two cases represent the range of temperatures that could be assumed. Results for the space heating/cooling demand are presented in Table 1.

Table 1: Variation in heating and cooling power results when comparing theoretic models and assumed room air temperatures.

Cases	Heating [kWh/m ²]	Cooling [kWh/m ²]
6R1C	3.09	-31.9
5R1C (20°C)	2.2	-49
5R1C (26°C)	2.4	-42

Results from *Table 1* show a discrepancy on the 5R1C compared to the 6R1C. The heating demand has an absolute error of 0.89 kWh/m² (28.8%) and in the cooling demand the error is 17.1 kWh/m² (53%). Within the 5R1C models with different assumed temperatures the variation is not that high. A plot showing the average air temperature in a zone is shown in Figure 8.

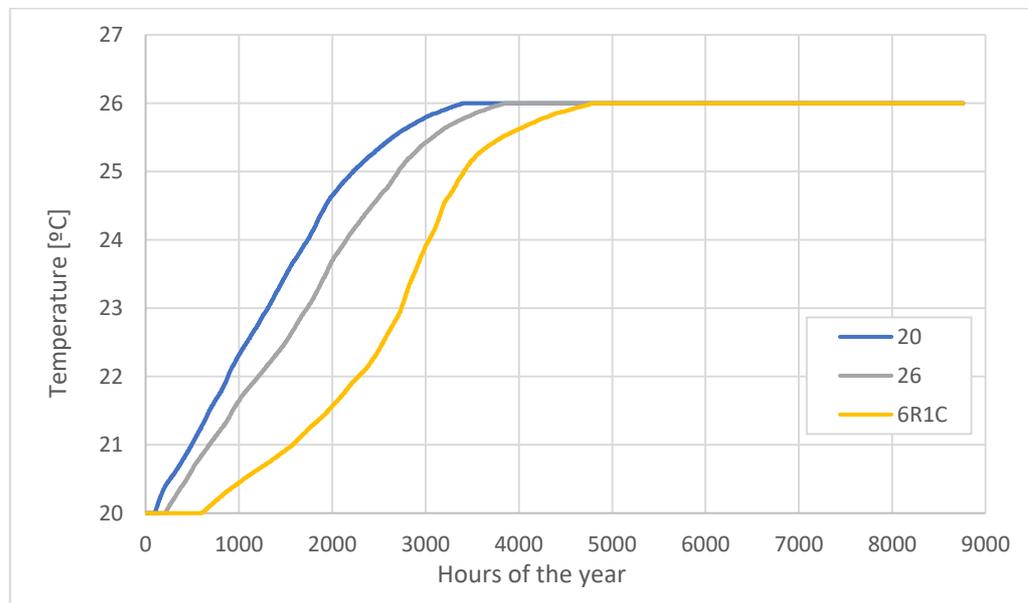


Figure 8: Difference between the 5R1C and the 6R1C.

The 5R1C's adjustment factor can have a determinant effect on energy estimation results. Therefore, it has been decided that avoiding additional assumptions is a fair reason to justify using the adapted model 6R1C, instead of carrying on with the one proposed in ISO 13790:2008.

4 Pre-processing of input data

4.1 Data Structure

Keeping an organized data structure is important in the development of this tool for multiple reasons. In the development of the model, it is expected that future modules will be integrated in the simulation, since modularity is one of the principles of BeDOT. To allow the connections between components, it is important that the data structure of inputs and outputs is kept consistent. Additionally, it makes the model more user-friendly if data is stored in a more intuitive manner.

The data structure of the model is shown in *BeDOT Manual* (an independent document), where it is explained how the input data must be provided and how this is implemented in the components. Also, the Manual includes some guidelines of how to use BeDOT.

The handling of input data has been an iterative process, where multiple structures have been tested. For instance, the original idea was to import/create all the inputs per hour and zone. This meant that all these values needed to be stored in the canvas, thus requiring a lot of memory. This might not be an issue for 100 zones, but it became a huge limitation when running 1000 zones, since the memory requirements increased beyond the computer specifications.

Therefore, the structure shown in *BeDOT Manual* reduced this amount of data by applying a filtering strategy inside the *Heat Balance* and *HVAC* components. In this way, only the types of different templates are imported into the canvas and by using a filtering list, the components will assign the right information for each zone at running time. This filtering strategy is implemented by a template called Retrieving index (further explained in *BeDOT Manual*).

Data management is an issue of high importance in future developments, since the model will be always limited by the stored data. In post-processing steps, the user must take care of how much information will be handled in the canvas, since it may slow down the model.

4.2 Building zoning and glazing

Creating zone characteristics lists

It was established that the model is required to have a broad flexibility regarding input data. In other words, it must be permitted to the user the free choosing of around 15 to 20 different parameters for each zone in a pre-processing stage. This can easily become a challenging step regarding user interface. For instance, assuming an average number of zones of 100, there will be over 1500 possibly parameters, where some of them are also changing hourly.

At the end, the method through which input lists are created remain a user's choice. The methodology regarding zone characteristics in this model is presented under this chapter and it is still under constant development. The method presented in this project is one possible solution, whereas there might be many other options. The list creation method might change partially or completely in the future, depending on the intended use of the model and user preferences. The example provided mainly illustrates the format in which data must be allocated so it can be correctly used as

input in further calculations. Any eventual changes in this stage are out of the scope of this project.

Integration with 3D geometry

One of the biggest challenges in the development of any engineering software is establishing a smooth integration between graphic interface and technical calculations. In this project, these two aspects are represented by *Rhinoceros* 3D interface and the visual programming plug-in *Grasshopper*, the latest also equipped with other plug-ins (i.e. *Honeybee* and *Ladybug*). No compatibility problem between the tools has been encountered during this project. However, as foreseen by (Michalak, 2014) the most frequently faced challenge was preventing the 3D geometry from drastically increasing computing time. This problem was soon identified in early attempts on editing 3D geometry via *Grasshopper* programming or using ready-to-use components from *Honeybee* plug-in that are automatically linked to the *Rhino* model. With the intent of keeping a low computation time, simplifications were sometimes introduced in the project, bypassing the need of inputs from geometry, as well as preventing previews of data in *Rhinoceros* 3D unless it is necessary.

The first step to run the model is to create a geometry in *Rhinoceros* 3D, displayed in Figure 9. This geometry is then selected in the graphic interface and defined as a building mass in a *Brep* (Boundary Representation) component in *grasshopper*.

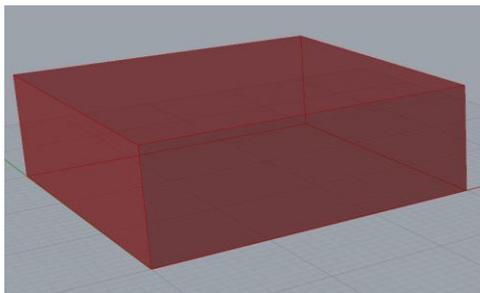


Figure 9: Geometry in Rhino 3D

Zoning

The *Brep* created in *Grasshopper*, is then divided in as many floors as the user decides, depending on the height of each floor. Then, each floor can be split in thermal zones according to Figure 10. The size of the zones will depend on the depth that the user sets for the zone. Note that if this depth is large enough the floor would be split only in 4 zones, since the core zone would disappear.

It is very important to notice that the zones are created in a fixed pattern, as shown in Figure 10.

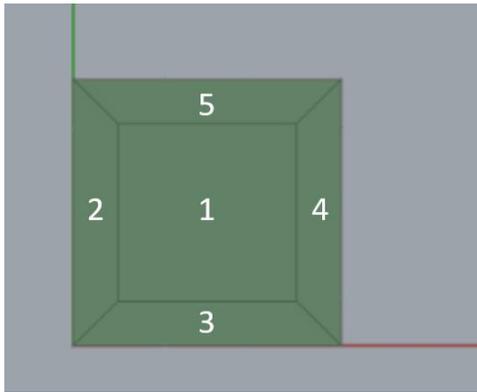


Figure 10: Pattern of zone naming

Taking as reference the X-Y axes (red and green lines respectively), zones are numbered in the order: core – left – down – right – up, starting in the bottom floor up to the top floor. The zoning sequence is important because it sets the pattern for all following vectors of data implemented in the program, so that a certain position in the vector will always refer to the same zone.

Weather data

One of the main inputs required from the user is a weather data file, which must contain standard hourly information on the geographical location the building is situated on. For the present project, the necessary hourly parameters are dry bulb temperature [$^{\circ}\text{C}$], direct normal radiation [W/m^2], diffuse normal radiation [W/m^2], relative humidity [%] and barometric pressure [Pa]. Provided that it includes the necessary parameters, any file can be imported to *Grasshopper* platform, regardless of the source. This grants the user the flexibility of using any preferred weather file, or even edit it in a pre-processing stage. However, the data needs to be provided always in the same structure. How data is imported is explained in the document *BeDOT Manual*.

Geometrical zone properties

The aim of these components is to gather, process and organize all parameters that characterize each of the building's zones, prior to the main energy balance calculation. Parameters comprise zone geometric characteristics and these are taken from the geometry using the component "*Honeybee_Descomposed Based On Type (decomposedByType)*", see Figure 11. From this component, information is obtained in the format of *brep* surfaces. Because of this, a cluster/component called "*Geometry input data*" was developed, which rearranges the information in the right format and structure. A cluster is not more than a set of different *Grasshopper* components combined in one block.

In the cases where one parameter, like windows or internal walls, stores more than one value per zone, this information needs to be regrouped. For example, internal walls in the heat balance are treated as the total sum of all the inner surfaces in a zone. A component was developed so the sum of all results per zone is calculated, making it just one value in the end.

The same accounts for windows, where the total area per zone is needed. All these rearrangements are performed inside the cluster "*Geometry input data*".

How Honeybee defines areas is important when it comes to extract data from databases or to compare and analysis results within different projects, since the areas could differ. BeDOT considers areas in the following format:

- All walls created by Honeybee components have thickness equal to zero.
- The internal floor areas are considered until the façade, disregarding internal walls.
- The envelope is considered as the external layer of the building, considering also the slabs thickness. This means that there will be small portions of the façade that are considered as external walls connected to interior environment when these are actually internal slabs.

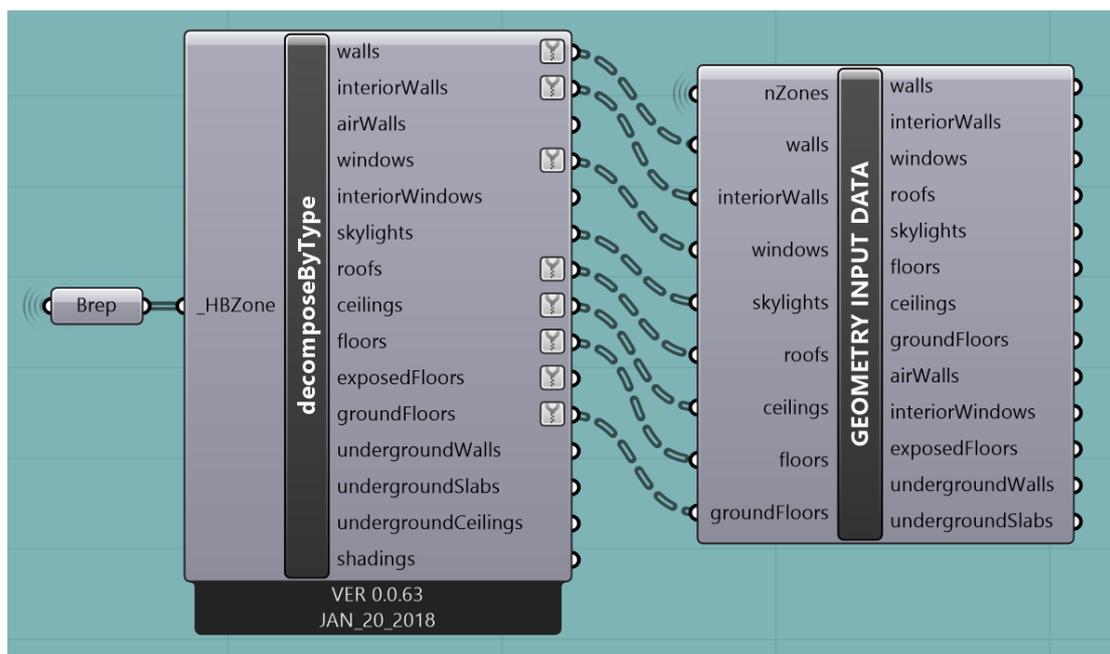


Figure 11: Geometry information from Rhino 3D.

4.3 Internal gains

Internal gains are composed of three different internal loads: Lighting, appliances and people. These loads are created using *Grasshopper* components, which build a list of hourly loads. Regarding the schedule, it is possible to choose when a weekday starts and finishes, whether different loads are applied in the weekends and how long the period should be (number of weeks). The default case is one year, so the list will have 8760 values by default. If there is a need to increase this list, a 2nd period should be included.

However, lists of values with the loads can be imported directly from other databases into lists in *Grasshopper*. It is up to the user to choose whatever is more convenient, as long as the structure of these lists is the appropriate (see *BeDOT Manual* for further explanation in data structure). It is really important to highlight this, otherwise the model would not work properly since the user could assign loads to different zones or hours.

Lighting and Appliances are given in W/m^2 , but internal loads due to human occupation need to be calculated in a previous step following equation (2).

$$\phi_{people} = \frac{people}{density} [W/m^2] \quad (2)$$

Internal loads and their characteristics can be coupled into collections of data, based on the zone end use, i.e. office, commercial, residential, etc. The user has the possibility to create data collections that are most often used, combining typical values of metabolic heat load [W/person], people density [m²/person], occupancy [-], lighting [W/m²], appliances [W/m²]. These values can either be obtained from reference literature (i.e. ISO, Sveby, ASHRAE, CIBSE) or from the user's own experience.

For the model, three different zone types for internal loads were created, each zone type includes a variety of these three internal loads, lighting, appliances and people. Depending on the final use of the building to be modelled, one or another zone type can be applied to each zone, which guarantees 100% flexibility when implementing internal loads in the zones. How this selection is done is explained in *BeDOT Manual*.

To keep the model flexible and to be able to insert other templates other than the ones provided as example, it is important to keep always the same structure of datatrees in *Grasshopper*. How the data structure is set is explained in *BeDOT Manual*.

4.4 Ventilation

Ventilation data is composed of 4 parameters: Hygienic airflow, infiltration, set-point for heating and set-point for cooling. How these parameters are handle by the model is presented in section 4.6 and *BeDOT Manual*.

The inlet temperature that goes into each room is handle by the Air Handling Units (AHU) and it is not include in the ventilation set of data. It is explained section 5.2 how the supply temperature is implemented.

Set-points for heating or cooling will vary depending on the final use of each zone and the same will happen with hygienic airflow. Infiltration is not that dependent on the use of the building but more on the envelope properties of each zone. Hygienic airflow is usually calculated considering two components: people and room area, see equation (3). Depending on use of the room the values for these two components will vary. For an office case, the room will need at least 7 l/s per person and 0.35 l/s per square meter (Sveby, 2013) for hygienic purposes.

$$\dot{v}_{hyg} = \frac{\dot{v}_{people}}{density} + \dot{v}_{area} \quad [l/sm^2] \quad (3)$$

There can be several ways to estimate the amount of leakage, \dot{v}_{leak} , through the thermal envelope. One method is to define leakage as a percentage of the infiltration that is achieved when carrying out a blower door test, applying an overpressure of 50Pa. At this pressure the leakage would be around 0.2-0.4 l/sm². The percentage recommended by Bengt Dahlgren is 7%. As presented in Table 4, the hygienic airflow units are per square meter, which means that people's component is transformed into this unit to further implementation in the *Heat Balance* and *HVAC* components.

It is explained later in the report that the hvac component requires as input a supply temperature in the case where an Air Handling Unit (AHU) is installed. This T_{supply} is chosen by the user. It can vary according to outdoor temperature or being set as a

constant temperature for the different seasons of the year. It is up to the user what supply temperature use.

4.5 Solar radiation

Solar radiation is calculated using *Honeybee* components and *Daysim*, which is a validated software that calculates the effect of sun over surfaces. It is possible to choose whether this component supplies solar radiation values, in W/m^2 , or illuminance, in lux. (further details are presented in *BeDOT Manual*). The use and assembly of *Daysim* components was mainly done by Bengt Dahlgren, and its results were later combined and adapted by us into the correct format.

From assembly of *Daysim*, the first step is to create a mesh around the building and have solar radiation calculated for every point of the mesh. Then, Grasshopper components associate the middle point of each window or skylight to their closest mesh point value. After that, a script in python, developed by the authors, connects these points and values, multiplies by their respective glazed areas and applies g-value coefficients (both due to glass type and shading). The result is the total amount of solar irradiance, in Watts, that every zone gets hourly from its glazed surfaces. In this project, the effect of radiation over opaque surfaces is neglected, thus only glazed surfaces are accounted for heat gains through radiation.

The mesh does not need to have points in the ground, so this surface is removed from the *Brep* for the purpose of saving time.

Some of the advantages of using *Daysim* and meshing the building's surface are a simplified calculation and therefore lower computational time. In addition, this component also considers shading obstacles in the surroundings and the possibility to have a shading control for the windows. It was decided, based on Bengt Dahlgren's experience, that a reasonable assumption is to consider that all the windows in the same thermal zone will have the same g-value. However, shadings can be controlled at window level to account for an increased accuracy when having obstacles. Even if a zone has many window units, applying on them distinct characteristics means adding complexity, at times unnecessary, in an early design stage.

Since obstacles are considered, results will depend directly on the number of windows and mesh points. If one window per zone is simulated, what would make the simulation faster, the shade coming from the obstacle would be averaged along the whole window area on the zone. On the other hand, if a larger number of windows is modelled, obstacles will affect only some of them, as shown in Figure 12, which increases accuracy.

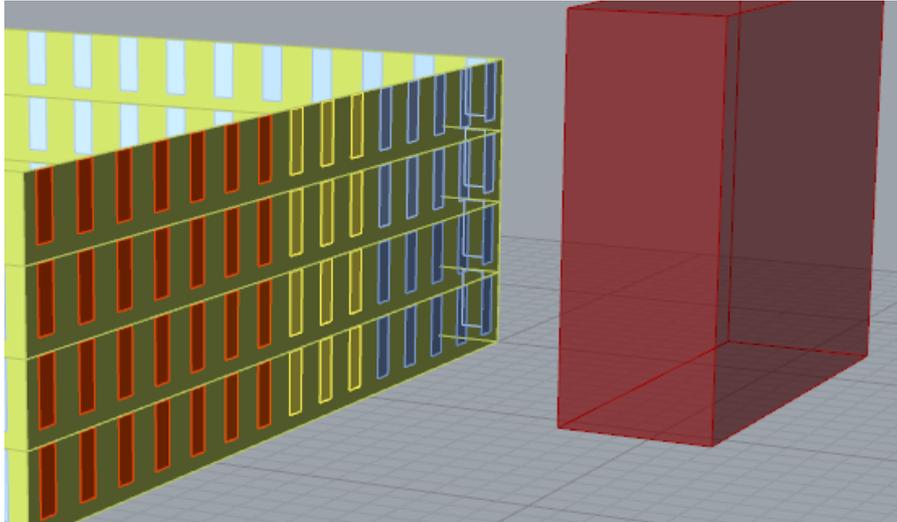


Figure 12: Obstacle modelling in one of the facades.

4.6 Input data

To organize the different input data needed for the heat balance and HVAC components, the parameters were organized in types of lists. These types are combined into templates such as internal gains, envelope, indoor climate and ID. The format of these templates, as well as how they are created is presented in *BeDOT Manual*.

The zone types of the envelope template are not made hourly since these properties are fixed in time, which results in a structure that is different from the other templates. However, this variation is compensated by saving computational time, since the amount of information handled by the components is much smaller. Inputs included in these zone types are shown in Table 2, as well as the indexation that the python code used for each of them.

Table 2: Input data gather in one envelope zone type.

Index	Parameter	Unit
0	u-value wall	[W/m ² K]
1	$\Delta U_{wall} + \Delta U_{roof}$ (thermal bridges)	[W/m ² K]
2	u-value window	[W/m ² K]
3	g-value window	-
4	u-value skylight	[W/m ² K]
5	g-value skylight	-
6	u-value roof	[W/m ² K]
7	Internal Heat Capacity	[J/m ² K]
8	Internal walls inside each zone	[m ² (wall)/m ² (floor)]
9	Shading factor	[-]

The rest of lists will be created hourly. For internal gains, the zone type is presented in Table 3, together with the units and indexation used in python for each parameter.

Table 3: Input data gather in internal gains zone type.

Index	Parameter	Unit
0	Lighting	[W/m ² (floor)]
1	Appliances	[W/m ² (floor)]
2	Human Occupancy	[W/m ² (floor)]

Other zone type will be the one related to indoor climate in the zones. This zone type will be variable per zone and hour, following the data structure presented in *BeDOT Manual*.

This list includes hygienic airflow, infiltration and the set-points for cooling and heating as it is shown in Table 4, as well as the indexation used in the python code. Leakage is given per square meter of envelope, which includes the external wall, windows and roof areas.

Table 4: Input data gather in indoor climate zone type.

Index	Parameter	Unit
0	Hygienic airflow	[l/sm ² (floor)]
1	Infiltration	[l/sm ² (envelope)]
2	Set-point Heating	[°C]
3	Set-point Cooling	[°C]

The last set of input that is the ID zone types. In this type different aspects of the model can be added. Right now, the zone type includes two parameters that are presented in Table 5.

Table 5: Input data gather in ID zone type.

Index	Parameter	Unit
0	AHU_id	[-]
1	Building_id	[-]

It is important to point out the AHU_id is the only one needed for the calculation process. However, Building_id is just an example of the post-processing that is possible in the model. Other parameters could be attached in the type regarding future post-processing needs.

5 Simulation

How to structure the model has been a process of many changes. The main issue faced was how to implement hourly data in the heat balance. This was first solved by using python libraries. In those libraries we could save the balance calculations of each zone applying *Heat balance calculation: ZONES*, and then called them every hour using the component *Zones simulation: HOURLY* (see Figure 13).

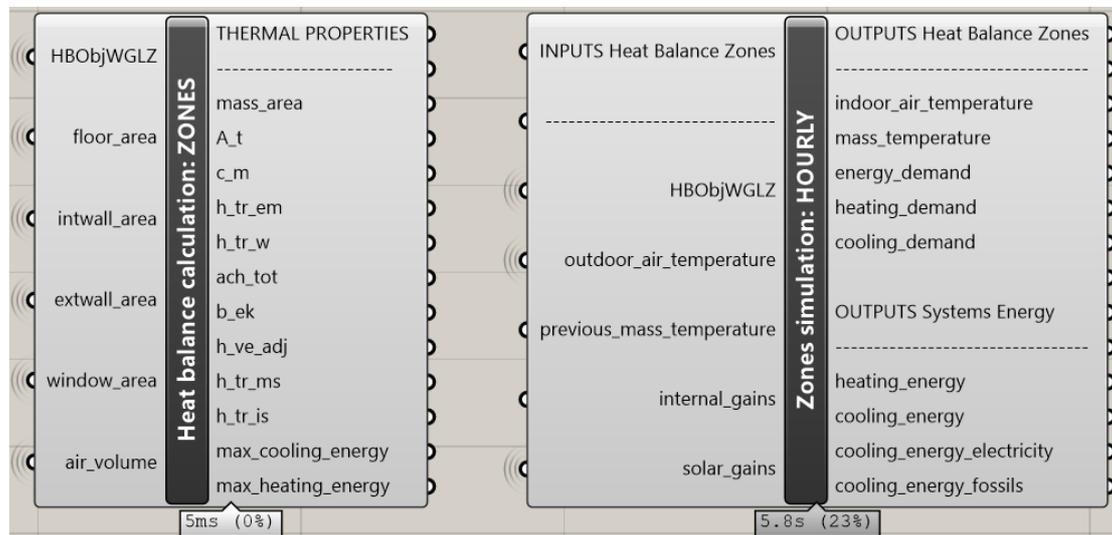


Figure 13: Heat balance calculation split in two components.

However, in a middle stage of the project, skipping the use of these libraries was in fact considered a better option for two reasons. Firstly, it meant saving substantial computational time. Secondly, it made the model easier to understand, so the flow of data and calculations was more transparent to the user. Consequently, these two components were combined into only one: *Heat balance* (see Figure 14).

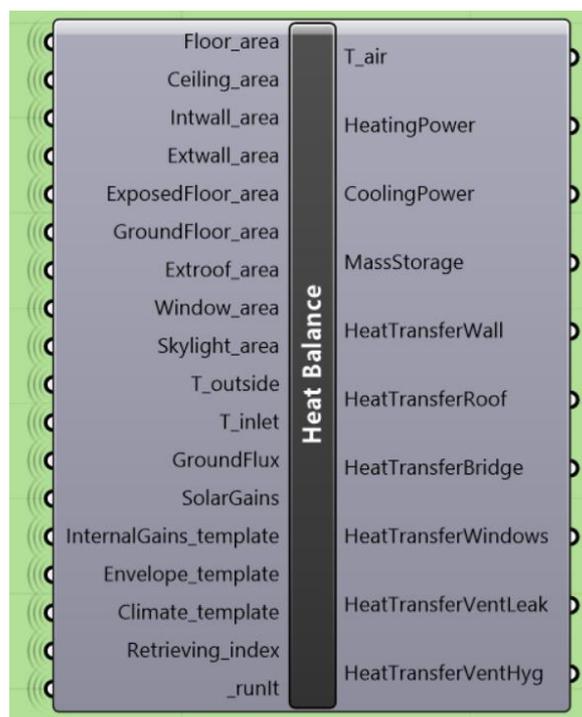


Figure 14: Heat balance calculation component.

In Figure 14 are displayed all the inputs that are needed for the building heat balance. Inputs can be subdivided in two groups. The first group of inputs would be the ones related to the geometry of the building (all the areas) and the envelope properties (u-values, g-values and internal heat capacity). The second group of inputs, such as heat gains, setpoints and supply temperatures, are defined in hourly schedules.

How these different inputs are implemented in the component is presented in sections 4.3, 4.4, 4.5 and 4.6.

5.1 Heat balance calculation component

Heat balance component from Figure 14 is the one containing all the calculations regarding the heat balance of the zones, based on ISO 13790:2008. The scripts, written in Python language, allow us to get multiple outputs. By feeding the component with thermal parameters needed for heat balance calculation (transmittances, areas, ventilation design, etc.), it is then possible to obtain results that are the major interest, such as:

1. Indoor air temperature per zone and hour.
2. Heating power per zone and hour.
3. Cooling power per zone and hour.

As it is explained before, there is a need for defining two kinds of inputs. The first one, geometrical properties, can be obtained directly as data lists from pre-processed components in Geometry. The second type of inputs are those that will come as lists of data from templates, namely:

1. Internal loads coming from lighting, people and appliances.
2. Envelope properties.
3. Climate settings.
4. HVAC information.
5. Solar gains.

Both inputs came as lists of data, according to specified templates.

Following Figure 7, all the formulation, still based on ISO 13790:2008, is defined in the coming paragraphs.

To calculate the internal heat capacity (C_m) of each zone, ISO's clause 12.3.1 is being followed, where the heat capacity of the building is simply defined as an assumed value, classifying the building from very light to very heavy (see Table 6). From this table we get C_m and the mass area, A_m .

Table 6: Building classes. Internal capacity and mass area.

Class	Internal heat capacity C_m [J/K]	Mass area A_m [m²]
Very light	$80000 \cdot A_f$	$2.5 \cdot A_f$
Light	$110000 \cdot A_f$	$2.5 \cdot A_f$
Medium	$165000 \cdot A_f$	$2.5 \cdot A_f$
Heavy	$260000 \cdot A_f$	$3 \cdot A_f$
Very Heavy	$370000 \cdot A_f$	$3.5 \cdot A_f$

Heat transfer by transmission coefficients are calculated according eq. 18 in ISO:

$$H_{tr,em} = (U_{wall} + \Delta U_{wall}) \cdot A_{extwall} + (U_{roof} + \Delta U_{roof}) \cdot A_{roof} \quad [W/K] \quad (4)$$

$$H_{tr,w} = U_{window} \cdot A_{window} \quad [W/K] \quad (5)$$

Where,

U_{wall}	Is the thermal transmittance of the wall in W/m^2K
U_{window}	Is the thermal transmittance of the window in W/m^2K
ΔU_{wall}	It accounts for the thermal bridge in the external façade in W/m^2K
ΔU_{roof}	It accounts for the thermal bridge in the roof in W/m^2K
U_{roof}	Is the thermal transmittance of the roof in W/m^2K
A_{window}	Is the area of the window in m^2
A_{wall}	Is the area of the wall in m^2
A_{roof}	Is the area of the roof in m^2

In BeDOT, the values of thermal bridges are considered the same both for external walls and roof, for each zone. Consequently, for simplification, only one parameter representing thermal bridges is defined.

Heat transmission to adjacent spaces is neglected since there is no thermal coupling between zones. Another limitation is that no model of thermal coupling to the ground has been yet implemented. Estimating the ground boundary conditions is a process of considerable complexity, that requires much longer time steps (often of months), due to the thermal pillow phenomenon. Additionally, it would require an adaptation of the 5R1C model, inserting a new temperature node (T_{ground}) and a new resistance value. This problem is often also encountered when using renowned *softwares* such as IDA-ICE or Energy+, and it is solved by using other models for pre-processing the ground coupling. The same solution is applied in the current project. Heat transfer to the ground can be pre-processed outside the program, and later inserted in the heat balance directly as losses. In this project, heat losses to the ground are calculated by a model in *KIVA*, provided by Bengt Dahlgren.

The input *HeatTransferGround*, shown in Figure 14, is dummy signal containing heat values, in W/m^2 , to show how transfers to the ground can be implemented into the balance component.

Heat transfer by ventilation is calculated according to eq. 21 in Clause 9.3.1 in ISO:

$$H_{ve} = \rho_a \cdot c_a \cdot (b_{ve} \cdot \dot{v}) \quad in \quad [W/K] \quad (6)$$

Where,

$\rho_a \cdot c_a$	Is the heat capacity of air per volume, $1200 J/(m^3K)$
b_{ve}	Is the temperature adjustment factor.
\dot{v}	Is the airflow rate in m^3/s

It is explained in section 3.3 that the adjustment factor is not included in the calculations, since it was preferred to include another resistance in the 5R1C network

that accounts for another ventilation heat transfer, in this case infiltration. In this way possible errors are evaded and results are more accurate.

Two ventilation heat transfers are then calculated: $H_{ve,hyg}$ and $H_{ve,leak}$, and both adjustment factors in these equations are 1. The infiltration accounts for the leakage through the envelope including the roof area but not the ground, which is considered air thigh in this case.

From Figure 7, it is seen that there are two resistances which represent the thermal coupling between the surface node and the air or mass, $H_{tr,is}$ and $H_{tr,ms}$ respectively.

$$H_{tr,ms} = h_{ms} \cdot A_m [W/K] \quad (7)$$

$$H_{tr,is} = h_{is} \cdot A_t [W/K] \quad (8)$$

Where,

h_{ms}	Is the heat transfer coefficient between the mass node and the surface node. It is a fixed value of 9.1 W/m ² K (eq. 64 in ISO 13790)
h_{is}	Is the heat transfer coefficient between the air node and the surface node. It is a fixed value of 3,45 W/m ² K (eq. 9 in ISO 13790)
A_t	Is the area of all the surfaces facing the building zone in m ²

Once all transmittances are obtained, the heat balance is calculated following the steps in Annex C of ISO 13790:2008. Some of the inputs needed are internal gains, (ϕ_{int}) and solar gains (ϕ_{sol}), both expressed in Watts. How this data is collected is explained in section 4.3, 4.4, 4.5 and 4.6.

Still referring to Figure 7, the scheme shows heat flows going from internal loads and solar gains to the air, mass and surface nodes. The expressions implemented are:

$$\phi_{ia} = 0.5 \cdot \phi_{int} [W] \quad (9)$$

$$\phi_m = \frac{A_m}{A_t} \cdot (0.5 \cdot \phi_{int} + \phi_{sol}) [W] \quad (10)$$

$$\phi_{st} = \left(1 - \frac{A_m}{A_t} - \frac{H_{tr,w}}{9.1 \cdot A_t}\right) (0.5 \cdot \phi_{int} + \phi_{sol}) [W] \quad (11)$$

Nodal temperatures are calculated based on Crack-Nicolson, which is a finite different method developed to solve partial differential equations. A scheme of this procedure is shown in Figure 15 and Figure 16.

For this calculation component, temperatures and energy demand are calculated in an hourly basis, so the step size is one hour. In turn, mass, surface and air temperatures are averaged over one hour (Annex C in ISO 13790).

The equations needed for the heat balance are originally based on ISO 13790, but following the needed modifications (Węglarz & Narowski, 2011), a new set of equations are implemented to follow the 6R1C network model:

$$T_{m,t} = \frac{\left\{T_{m,t-1} \left[\frac{C_m}{3600} - 0.5 \cdot (H_5 + H_{tr,em}) \right] + \phi_{mtot} \right\}}{\left[\frac{C_m}{3600} + 0.5 \cdot (H_5 + H_{tr,em}) \right]} \quad (12)$$

Where,

$T_{m,t-1}$ Is the mass temperature from the previous step.

$$\begin{aligned} \phi_{mtot} &= \phi_m + H_{tr,em}T_{outside} \\ &+ H_5 \frac{\phi_{st} + (H_{tr,w} + H_2)T_{outside} + H_1 \left[\frac{\phi_{ia} + \phi_{HC,nd}}{H_{ve}} + T_{supply} \right]}{H_4} \end{aligned} \quad (13)$$

$$H_1 = \frac{H_{tr,is} \cdot H_{ve}}{H_{tr,is} + H_{ve} + H_{vi}} \quad (14)$$

$$H_2 = \frac{H_{tr,is} \cdot H_{vi}}{H_{tr,is} + H_{ve} + H_{vi}} \quad (15)$$

$$H_3 = H_1 + H_2 \quad (16)$$

$$H_4 = H_3 + H_{tr,w} \quad (17)$$

$$H_5 = \frac{H_{tr,ms} \cdot H_4}{H_{tr,ms} + H_4} \quad (18)$$

$T_{m,t-1}$ is assumed to be 20 degrees for t_0 . A flowchart explaining how the mass temperatures are assigned for each time step is shown in Figure 15.

Equation (13) includes the parameter $\phi_{HC,nd}$, which is the heating or cooling power needed in the zone at every time step, expressed in Watts.

As previously mentioned, nodal temperatures are averaged and adapted (Węglarz & Narowski, 2011) to follow the 6R1C model.

$$T_m = (T_{m,t} + T_{m,t-1})/2 \quad (19)$$

$$T_s = \frac{H_{tr,ms}T_m + \phi_{st} + H_{tr,w}T_{outside} + H_{tr,1}[T_{supply} + (\phi_{ia} + \phi_{HC,nd})]/H_{ve}}{H_{tr,ms} + H_{tr,w} + H_{tr,1}} \quad (20)$$

$$T_{air} = (H_{tr,is}T_s + H_{ve}T_{supply} + \phi_{ia} + \phi_{HC,nd})/(H_{tr,is} + H_{ve}) \quad (21)$$

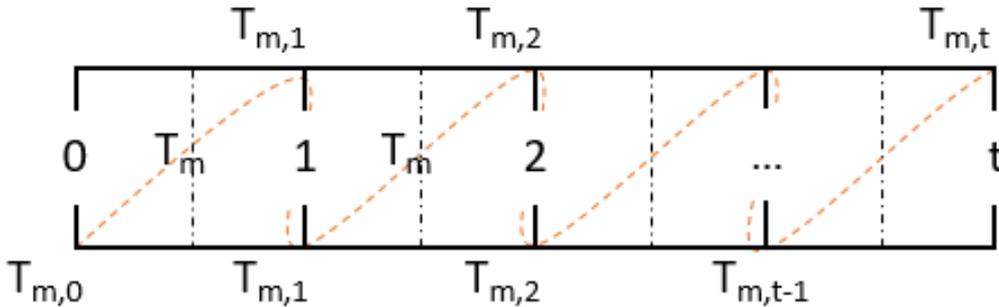


Figure 15: Step calculation of mass temperature.

All the parameters used to calculate the average temperatures, T_m , T_s and T_{air} are given for each time step, t .

This balance gathers constant coefficients in H_1 , H_2 , H_3 , H_4 and H_5 to simplified as much as possible the main equation. In this format, a mathematical limitation appears when one of the ventilation heat transfers, either infiltration or mechanical airflow, is

set to zero. In this case a singularity appears, so a filtering function is needed to replace zeros in ventilation for a negligible value. This function was written in the main balance component.

It is important to realize that equations (13), (20) and (21) may vary depending on the chosen radiation system. Depending on where is the heat emitted to, the term $+\phi_{HC,nd}$ must be summed up to the corresponding node (ϕ_{ia} , ϕ_{st} or ϕ_m). However, for this project it is fixed that energy is always implemented in the air node, thus being convective.

Equations (12) to (21) allow the calculation of temperature nodes of the zone for a given value of $\phi_{HC,nd}$. These equations are then used to calculate the internal temperature and the required energy demand to keep the zone temperature within the limits of the defined set-points, varying according to user's preferences. These equations are needed to calculate the energy demand for a time step. How to proceed for the 8760 time steps that are needed to simulate a year is explained in the following lines.

Three different cases may occur in a zone depending on the air temperature:

1. The heating power is sufficient to keep the set point temperature for heating. The air temperature is equal to the set-point temperature.
2. The air temperature is in between the set-points so the zone does not require either heating or cooling. In this case the air temperature varies freely within the limits.
3. The cooling power is sufficient to keep the set-point temperature for cooling. The air temperature is equal to the set-point temperature.

The ISO Standard contemplates another two cases where a limitation of power is included and the air temperature will be outside the comfort range. However, in this project this limitation is disregarded.

The calculation procedure is based on Annex C in ISO 13790:2008. The first needed step is to check if there is an actual need of heating or cooling in the zone.

STEP 1:

The parameter $\phi_{HC,nd}$ is set to zero, and then equations (12) to (21) are applied.

Once we have the air temperature for this step, $T_{air,0}$, it is compared with the set-points. If $T_{air,0} > \theta_{set,C}$ cooling is needed; If $T_{air,0} < T_{set,H}$ there is need for heating. Otherwise, neither heating or cooling is required. In this case, energy demand is set to zero for both heating and cooling and the calculation is performed again for the following hour.

STEP 2:

If this step is called in the model, it means that there is a need to heat up or cool down the zone. If there is a need for heating, $T_{air,set} = T_{set,H}$ and if there is a need for cooling, $T_{air,set} = T_{set,C}$.

To calculate the energy demand either for cooling or heating, a linear interpolation approach is adopted. The air temperature has already been calculated for $\phi_{HC,nd} = 0$, so now it must be calculated again for $\phi_{HC,nd10} = 10 \text{ W/m}^2$ applying equations from (9) to (21).

Once $T_{air,10}$ is calculated, we can calculate the energy demand $\phi_{HC,nd,un}$, assuming that there is no restriction in power.

$$\phi_{HC,nd,un} = \phi_{HC,nd10} \cdot \frac{T_{air,set} - T_{air,0}}{T_{air,10} - T_{air,0}} \quad (22)$$

STEP 3:

Once $\phi_{HC,nd,un}$ is calculated, the availability of power that is provided to the building needs to be checked. However, the model was developed not considering any limitation in power, hence STEP 4 in Annex C of the standard is not required in this project. This unrestricted power is used to recalculate equation (12) to (21), so the actual air temperature T_{air} is obtained.

A flowchart of the process is shown in Figure 16.

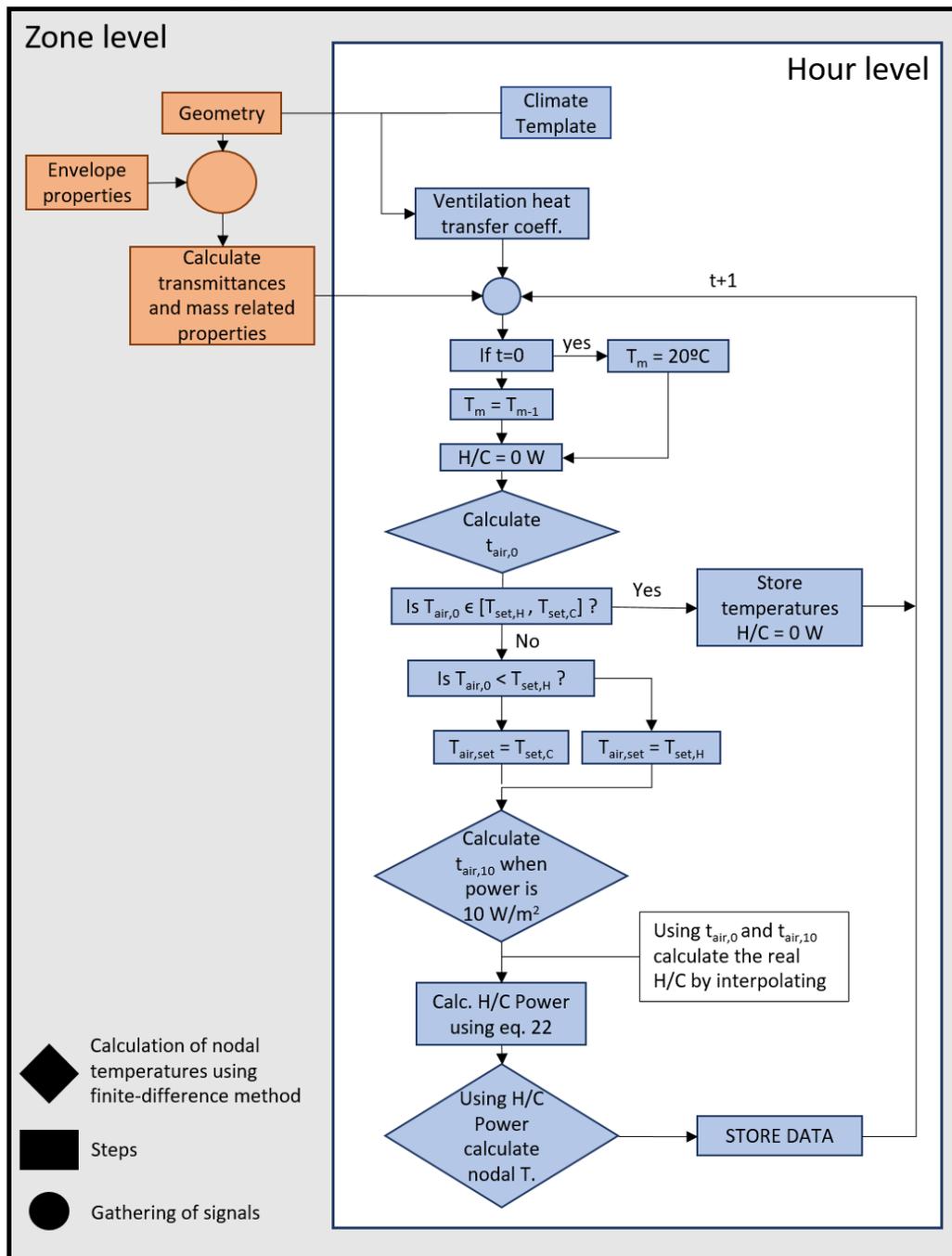


Figure 16: Flowchart of heat balance calculation

Once the heat balance is done, the different outputs, shown in Figure 14, are stored. The air temperature as well as the heating or cooling power are taken straightforwardly from the heat balance. On the other hand, the mass storage and the transmission losses need to be calculated for each time step by using equations (23) to (29).

Following Figure 7, one can easily understand the transmission losses that the model has at zone level. Transmission losses between internal nodes (air to surface and surface to mass) were not calculated since they occur inside the zone and are not considered relevant.

To export the transmissions between the different building elements, the envelope heat transfer coefficient $H_{tr,em}$ was split into the different elements: external wall, roof and bridges following the expression shown in (4).

$$HeatTransferWall = H_{tr,wall} \cdot (T_m - T_{outside}) [W] \quad (23)$$

$$HeatTransferRoof = H_{tr,roof} \cdot (T_m - T_{outside}) [W] \quad (24)$$

$$HeatTransferBridges = H_{tr,bridges} \cdot (T_m - T_{outside}) [W] \quad (25)$$

$$HeatTransferWindow = H_{tr,w} \cdot (T_s - T_{outside}) [W] \quad (26)$$

$$HeatTransferVentHyg = H_{ve,hyg} \cdot (T_{supply} - T_{air}) [W] \quad (27)$$

$$HeatTransferVentLeak = H_{ve,leak} \cdot (T_{outside} - T_{air}) [W] \quad (28)$$

$$MassStorage = C_m \cdot (T_{m,t-1} - T_{m,t}) [W] \quad (29)$$

This energy model considers the leakage through the entire envelope, so windows, external walls and roof are included. The same happens with the thermal bridges.

The mass storage and the transmission losses are used in section 6.2 to validate the equations in the heat balance.

5.2 Heating Ventilation and Air Conditioning (HVAC)

Most buildings in Sweden nowadays use mechanical systems to control and maintain indoor climate, usually called Heating, Ventilation and Air Conditioning (HVAC). Therefore, estimating energy consumption of a building without taking HVAC systems into account would be unrealistic.

Many challenges have arisen from the need to incorporate HVAC systems into the model. One of the most evident was guaranteeing that the model would work for any system configuration. HVAC systems can vary substantially from one building to another, either regarding which pieces of equipment that are included, their types or characteristics.

The relation between hygienic and additional ventilation also plays an important role, eventually demanding that the HVAC calculation be split into two components, instead of only one.

5.2.1 Splitting the HVAC component

The first attempt on implementing HVAC consisted in a single component, right after the zone heat balance. This way, the zone heat balance component was still being fed with the input of outdoor temperature, so that ventilation losses were dependent on hygienic air flows and outdoor conditions. Then, the resulting required heating and cooling powers were inserted in HVAC calculations. This component would then calculate the ventilation needed to fulfil the heating or cooling demands, and then decompose it, through a balance, into two parts: hygienic and complementary ventilation.

This method was soon discarded, for it resulted in concurrent heating and cooling demands. When calculating cooling and heating power due to hygienic ventilation after the heat balance, it often happens that the hygienic flow is lower than the ventilation that is effectively needed.

To better explain this problem, let's assume, for instance, a zone with a certain cooling demand. By knowing the desirable supply air temperature, it is possible to obtain the air flow needed to cool the zone. If it happens that this needed air flow is lower than the hygienic flow, it means that the hygienic ventilation will cool the zone too much. In that case, we need to decide between two options, both with undesirable outcomes:

- Changing supply temperature, thus lowering the air cooling capacity until needed air flow coincides with hygienic airflow. This could result in cases with unacceptably high supply air temperatures.
- Keeping supply air temperature unchanged and adding a heating power at zone level. This means cooling air down to supply temperature, only for later having to heat it up again inside the zone.

Once this problem came to light, it became clear the impossibility of having only one component for HVAC calculation. It was then decided to split this calculation into two, according to the volume of air to be heated up or cooled down.

The first component, positioned before the zone heat balance, calculates the hygienic ventilation supply. Its output, hygienic air flow at supply temperature, is then the starting point for the zone heat balance. Consequently, all heating or cooling demands

calculated in the zone heat balance component represent additional demands. These are then addressed in the second HVAC calculation block.

5.2.2 Hygienic Airflow component

The first HVAC component addresses the changes in hygienic airflow temperature, essentially promoted by the air handling unit (AHU). One important simplification in the model concerns the assumption that the zone will always be kept within the indoor setpoints, either because of the mechanical ventilation system or other methods, at zone level. This is needed in order to make assumptions on return temperatures, which are needed in heat recovery calculations but are in reality still unknown. The alternative for this simplification would mean having to run the first HVAC component along with the zone heat balance, in an iterative process until a common zone air temperature is reached. Although there has been no attempt to implement this method, it is believed that it would severely increase running time, reason why a simplification was preferred. Figure 17 displays the component, along with its input and outputs.

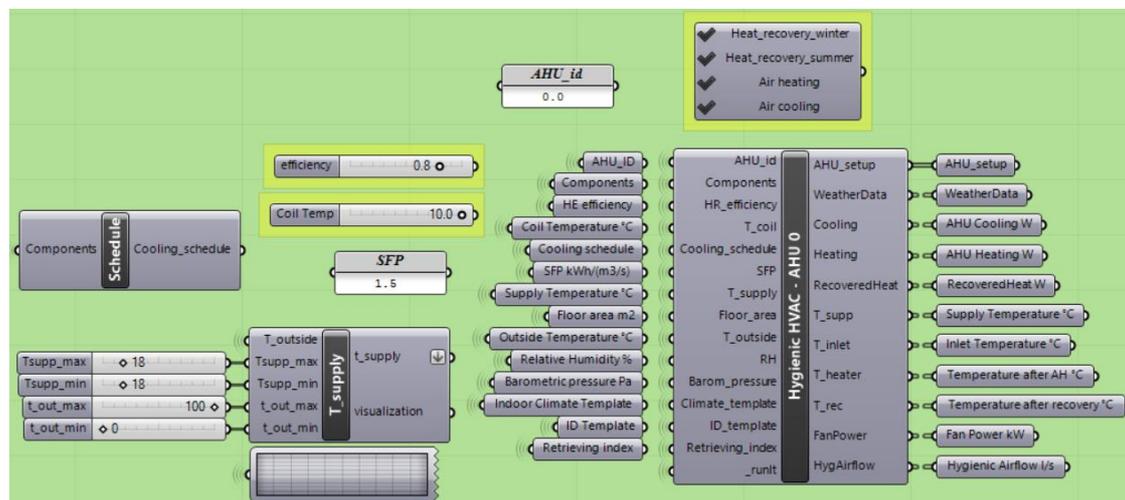


Figure 17: Hygienic airflow component.

A checklist of pieces of equipment that could possibly be included in the calculations is provided to the user previously to the calculation. All components that are not checked are bypassed in the calculations, meaning that they will not induce any changes in temperature. Then the user is required to specify an efficiency to the heat recovery system. The input T_{coil} represents the average cooling coil temperature and is necessary for estimating the amount of condensation happening during an air cooling process. This parameter will not play any role if the system does not include an air cooling coil, in which case it might be left blank.

The user is free to set the supply air temperature as desired, which could be a constant value or a function of outdoor temperature. Additionally, a cooling schedule can also be provided, forcing the cooling coil to turn off at specific days/ hours.

Then the Indoor Climate template is responsible for feeding the AHU component with hygienic air flow, cooling and heating setpoints and supply temperature. The ID template is responsible for assigning the right zones to each AHU.

Figure 18 shows a sketch on the order in which the components are calculated and the temperature naming pattern adopted at each step.

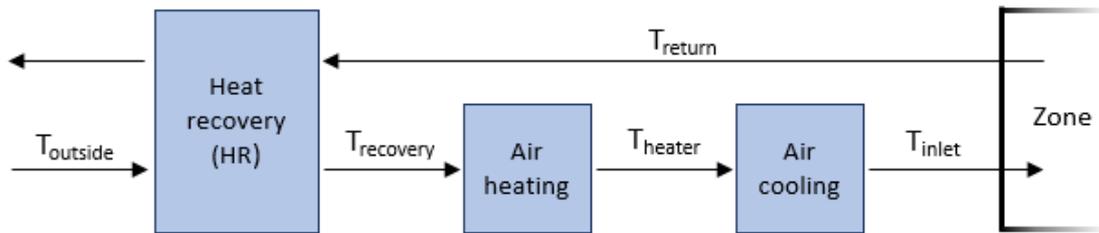


Figure 18: Sketch on AHU components and naming pattern.

Summarizing the whole process, air comes into the system at outdoor temperature $T_{outside}$, and leaves the system at return temperature T_{return} . T_{rec} is how the output from heat recovery unit is called, combining the thermal conditions of $T_{outside}$ and T_{return} , and depending on the heat recovery's efficiency. The air heating coil receives T_{rec} as input and gives T_{heater} as output. To maintain the process continuity, T_{heater} is inserted into the air cooling component. The final outcome of this system is air being supplied to the zone at temperature T_{inlet} .

At this stage, air heating and cooling units are considered ideal (100% efficiency) and no power limitations apply. It is important to notice that, although HVAC calculations are performed trying to achieve the desired T_{supply} , T_{inlet} might not actually correspond to T_{supply} . This could happen if the air heating and cooling units had limited power or were turned off.

In the end, the result of this component will be a volume of air, corresponding to hygienic requirements, at temperature T_{inlet} , being supplied to the zone heat balance component. Additionally, the heating and cooling power applied in the process are also obtained.

The outdoor temperature, as well as relative humidity and barometric pressure, are obtained directly from weather data.

The very first step in this component is to calculate the amount of air corresponding to hygienic ventilation in each zone, in m^3/s . This will be the amount of air that will have its temperature changed in every step of this component. The following steps are more thoroughly explained below.

Heat recovery unit

The heat recovery unit calculation is essentially based upon equation (30):

$$T_{rec} = T_{outside} - \eta(T_{return} - T_{outside}) \quad [^{\circ}C] \quad (30)$$

The difficulty lies in estimating the efficiency η , which varies significantly as function of T_{return} , $T_{outside}$ and T_{supply} . For this efficiency to be obtained, the heat recovery calculation has to be broken up in three main different cases, which are subjected to different assumptions.

The first one is if $T_{outside}$ is lower than the desired T_{supply} at that hour, case in which air needs to be heated. This raise in temperature can be partially or totally done in the heat recovery unit. Due to the aforementioned simplification, it can be assumed that

the return air temperature in this case will be the zone heating setpoint, which corresponds to the zone's inferior temperature limit. Because T_{supply} might not be a fixed constant value, and can instead vary hourly, the heat recovery's efficiency calculation is performed according to equation (31).

$$\eta_{calc} = \frac{T_{supply} - T_{outside}}{T_{return} - T_{outside}}, \text{ where } \text{if } \begin{cases} \eta_{calc} > \eta_{max} \rightarrow \eta_{calc} = \eta_{max} \\ 0 < \eta_{calc} < \eta_{max} \rightarrow \eta_{calc} = \eta_{calc} \\ \eta_{calc} < 0 \rightarrow \eta_{calc} = 0 \end{cases} \quad (31)$$

In which η_{max} is the equipment's maximum efficiency and η_{calc} is an idealistic efficiency value, which might not correspond to realistic values. It is necessary to further evaluate η_{calc} to make sure it will always be a value between 0 and η_{max} .

The second case is when $T_{outside}$ exceeds the desired T_{supply} , but is lower than the cooling setpoint, which corresponds to the zone's maximum temperature limit. This means that air would need to be cooled down to T_{supply} after passing the heat recovery, and that the heat recovery efficiency will be zero. Consequently, T_{rec} will be actually $T_{outside}$ unchangeably passing through. Although it can be stated that the return temperature does not exceed the zone heating setpoint, in reality its value is unknown. This is not a problem at this stage, since the return temperature is not needed while the heat recovery unit is inactive.

The third and final case deals with very high outside temperatures, exceeding T_{return} , which allows the activation of the heat recovery system again. This will only apply to warm places, so it is recommended to leave *Heat_recovery_summer* unchecked from the list if that is not the case. Calculations are similar to those explained on the first case.

It is important to clarify the impacts of a possible design situation in which the user wants only a heating setpoint, but not a cooling setpoint. So, it is assumed that air temperature will have an inferior limit, but can range freely with no upper temperature limit. The user has the option of simulating such situation by adopting very high (even possibly unrealistic) values of cooling setpoints, which will never be achieved normally in a building. When this happens, we can only have the first two logic cases, since the third will not exist. It is of utmost importance that a cooling setpoint value, however unrealistically high, is still defined, otherwise the logic inside the HVAC component will not work properly.

Air heating and cooling

Following the heat recovery unit comes the air heating coil. The heating power needed to heat up an air volume in the heating coil is obtained directly from equation (32):

$$Q = \dot{v} \rho_a C_a \Delta T \quad [W] \quad (32)$$

Where:

- Q Is the amount of heat, in W
- \dot{v} Is the air flow, in m^3/s
- ρ_a Is the air density, assumed as a constant value of 1.2 kg/m^3

regardless of temperature of pressure variations

C_a

Is the air heat capacity, assumed as a constant value of 1000 J/(kg.K) regardless of temperature of pressure variations

ΔT

Is the temperature difference between the air before and after the coil.

The process air heating can be visualized in a Mollier diagram, as seen in Figure 19. In the figure, air is heated from point 1 to point 2. Despite going through a change in temperature, air remains with the same vapour ratio.

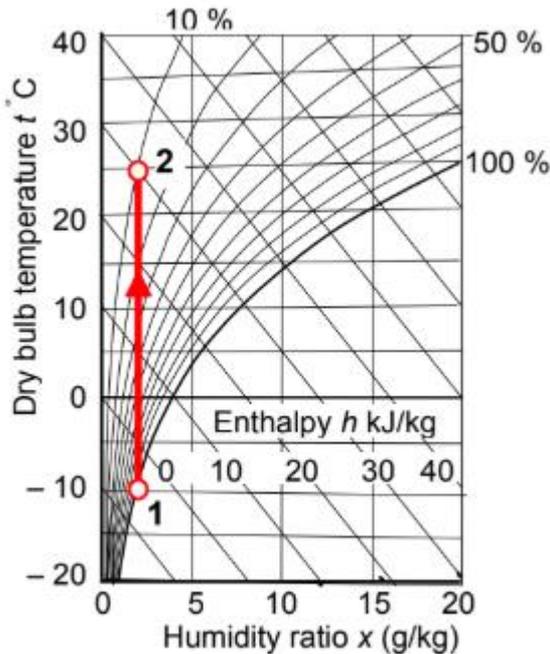


Figure 19: Air heating in a Mollier diagram. Source: (Abel E., 2015)

In the case of a working heating coil, air is assumed to have achieved T_{supply} after the coil, so the amount of heat is calculated with a temperature difference between T_{supply} and T_{rec} . Otherwise, if the heating is inactive or inexistent, air temperature after this step will still be T_{rec} and the heating power will be zero. According to the previously explained convention, the air temperature leaving the heating coil is called T_{heater} .

After the air heating step comes air cooling. This process requires more attention, once there is risk of condensation if the coil temperature is lower than the incoming air temperature. If condensation occurs, a significant amount of extra energy will be needed to cool down the air volume. The user has the option of choosing a cooling coil temperature, so it can be adapted to values commonly used locally.

Figure 20 displays the two possible processes for air cooling. The figure on the left represents cooling without condensation, which happens when the cooling coil temperature is higher than the dew point of air. On the other hand, if the coil temperature is lower than the dew point of the air, vapour condenses over the cooler surface. This is the case represented by figure on the right, in which condensation happens.

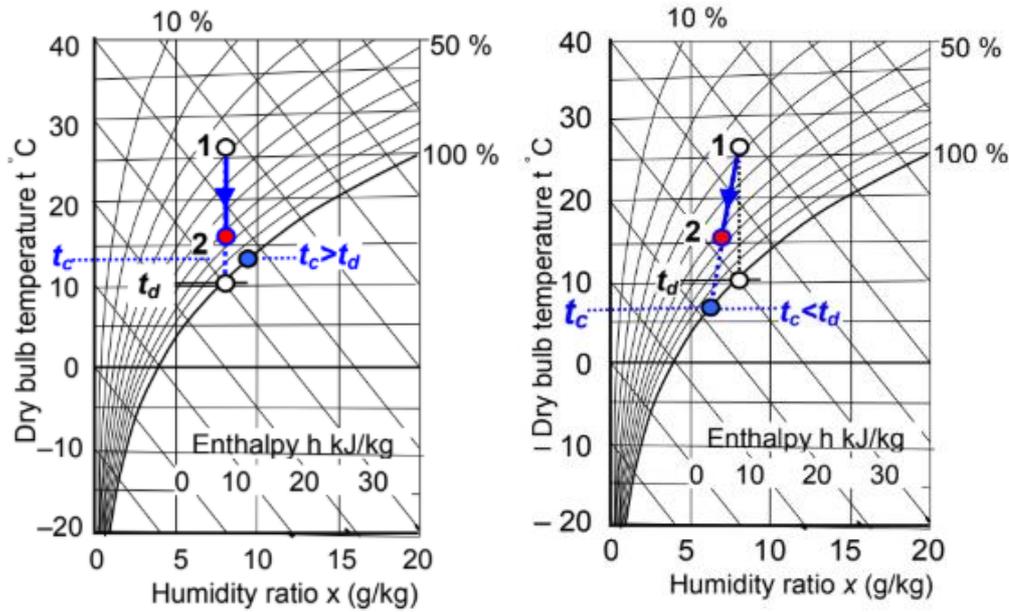


Figure 20: Air cooling in a Mollier diagram. Left: without condensation. Right: with condensation. Source: (Abel E., 2015)

The air cooling process displayed in Figure 20 are then translated into equations. The occurrence of condensation is evaluated by calculating the dew point temperature of the humid air entering the cooling coil. That is done according to equations (33) and (34), which are based on (Alduchov & Eskridge, 1996):

$$T_{dew} = \frac{273.3 \cdot B}{1 - B} \quad [^{\circ}\text{C}] \quad (33)$$

Where:

$$B = \frac{1}{17.27} \left[\ln \left(\frac{RH}{100} \right) + \frac{(17.27 \cdot T)}{(237.3 + T)} \right] \quad (34)$$

RH Is the air relative humidity, in %

T_{dew} Is the dew point temperature for a mass of humid air, in $^{\circ}\text{C}$

T Is the temperature that goes into the air cooler unit in $^{\circ}\text{C}$.

If the cooling coil temperature is lower than the dew point temperature, condensation will occur. In this case, the amount of energy needed to cool the mass of air is calculated with enthalpy values, as shown in equation (35):

$$Q = \dot{v} \rho \Delta h \quad [W] \quad (35)$$

Where:

Δh Is the difference in enthalpy values, in J/kg

The enthalpy of air is calculated using psychrometric equation (36) (Vaisala Oyj, 2013), which also require values of mixing ratio (37) and vapor pressure (38).

$$h = T (1.01 + 0.00189x) + 2.5x \quad [k]/\text{kg} \quad (36)$$

Where:

x Is the mixing ratio between mass of water vapor and mass of dry gas, being equation (37) already adapted to common air composition, given in g/kg

$$x = 621.991 \cdot \frac{P_w}{(P_{tot} - P_w)} \quad [g/kg] \quad (37)$$

P_{tot} Is the total ambient pressure, also called barometric pressure, in hPa

P_{ws} Is the water vapor saturation pressure, in hPa

$$P_{ws} = 6.116 \cdot 10^{\left(\frac{7.591 \cdot T}{T+240.726}\right)} \quad [hPa] \quad (38)$$

The values of temperature, relative humidity and barometric pressure can often be obtained directly from the weather file.

In case the cooling coil temperature not below dew point temperature, condensation will not occur, so the process does not require extra energy.

5.2.3 Complementary Airflow component

The zone heat balance, fed with temperature results from the first HVAC component, calculates the extra cooling and heating power needed to achieve the desired air temperature inside the zone. This extra cooling or heating can be provided either locally, with equipment such as radiators and chilled beams, or through the air handling unit.

Although it remains possible to establish components to estimate power from local heating or cooling strategies, or even heating with air, none of these options have been thoroughly developed.

Focusing on the cooling case, the second HVAC component, *Complementary Airflow*, deals with an extra volume of air going through the air handling unit. Currently, the option is restricted to *air cooling*. In the future, however, other options of *cooling at zone level* can exist, branching into several local cooling components (i.e. chilled beams, heat pumps, chilled panels, etc.). The idea is to allow the user to define which percentage of the demand that is fulfilled by each strategy.

Aiming now the attention to air cooling, most of the calculation is very similar to the one presented in the *Hygienic Airflow* component. It is important to guarantee the coherence of equipment and temperatures between the two HVAC components, provided that the AHU and its operation should be exactly the same for both airflows. For this reason, the temperatures at each step (T_{rec} , T_{heater} and T_{inlet}) are not calculated, but imported from the *Hygienic Airflow* component.

Estimating the needed additional air volume is the only different thing in this component. It is again based on the heat equation (32), using the cooling demand calculated in the *Heat balance* component and a ΔT corresponding to the temperature difference between the air temperature in the zone and T_{inlet} . The resulting air volume is then used on the following cooling power calculations.

6 Validation

6.1 Verification simplified hourly method

ISO 13790:2008 already includes a verification case for the simply hourly method. For this validation the reference building is a zone (see Figure 21) which dimensions are 3.6 x 5.5 x 2.8 meters.

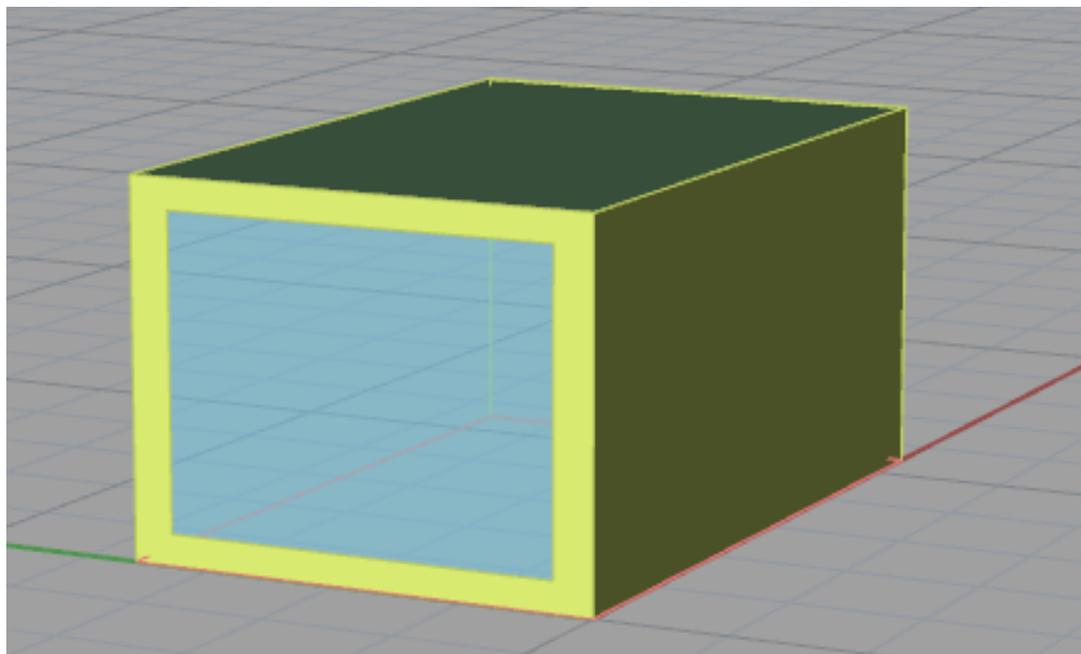


Figure 21: Verification zone

The zone facade is facing West, so solar radiation is calculated only in this direction. Input data for this particular model is found in ISO 13790:2008, however the test case is extracted from CEN/TC 89. From this standard, weather data such as solar radiation and outdoor temperature is exported.

Input data needed to run the model is presented in Table 7 regarding geometry, gains, materials and ventilation.

Table 7: Input data for the validation case according to ISO 13790:2008

Geometry	
	External Wall 10.08 m ²
	Opaque west façade 3.08 m ²
	Window 7 m ²
	Floor and Ceiling 19.8 m ²
	Internal walls 15.4 m ²
Transmissions	
	Window: 16.6 W/K
	Opaque façade: 1.51 W/K

Ventilation	
	Air changes: 1 [1/h]
	Schedule 08:00 to 18:00 weekdays
	$T_{supply} = T_{out}$
	Leakage: 0 [1/h]
Solar Gains	
	g-value window: 0.2
	Absorption opaque façade: 0.6
	External surface resistance: 0.04 m ² K/W
	u-value façade: 0.496 W/m ² K
	Radiation West [W/m ²]: CEN/TC 89
Internal Gains	
	Load: 20 W/m ²
	Schedule: 08:00 to 18:00 weekdays
Building time constant	
	Internal heat capacity 355000 J/m ² K
Use	
	Heating set point: 20 °C
	Cooling set point: 26 °C

Solar radiation is taken from CEN/TC 89, where we have either normal direct and diffuse horizontal radiation or global horizontal radiation for west façade. For this verification we have taken global data as an input.

Following table J.3 in ISO 13790:2008, inputs can be checked to ensure that we introduced the same values into the model. Our input data, a ventilation of 55.44 m³/h and 1033.56 kWh of internal gains, coincide with the data presented in the standard.

The expected output from this case is 537 kWh for heating power and -177 kWh for cooling power (ISO 13790:2008). Results from the simulation of 8760 hours are presented in the following table.

Table 8: Results from verification model

	Reference [kWh]	Calculated [kWh]	Error [%]
Heating	537	550	2,53
Cooling	-177	-168	4,71

Results from the model are very promising, since these are close to the values obtained in the standard. Standard CEN/TC 89 includes three different levels of accuracy: Level A, B and C. since our results are below 5%, the level obtained would be A, the most accurate of all three.

6.2 Energy balance verification

To further verify the model regarding its physical behavior, heat balance calculations have also been performed. According to the principle of conservation of energy, there should be a balance between the energy coming into and leaving the system.

Being each zone an independent system, it should be possible to verify energy balance for every zone, at every moment in time – in this case, hourly. The energy balance performed follows the following expression:

$$Q_{tr} + Q_{vent} + Q_{mass} + Q_{rad} + Q_{ig} + Q_{ed} = 0 \quad (39)$$

In which Q_{tr} is the energy transmitted through the system's envelope; Q_{vent} is transferred by ventilation; Q_{mass} is storage in the building mass; Q_{rad} are solar radiation gains; Q_{ig} are internal gains of people, lighting and appliances and Q_{ed} is the energy demand that needs to be supplied to the system in order to keep it within the acceptable temperatures.

The verification has been performed in steps, as summarized in Table 9. Regarding zone types, tests have been performed firstly for a core zone in the middle of the building, therefore only under the influence of Q_{vent} , Q_{mass} , Q_{ig} and Q_{ed} . Then, for a core zone in the ground or top floors, which will additionally have influence of Q_{tr} . Finally, for a non-core zone, which will also have windows and consequently, Q_{rad} . For each of these three zone types other two cases were simulated: selected hours in which there is no need for extra energy, so the zone temperature is within the limits without any external influence, and hours in which the zone has an energy demand.

Table 9: Steps for heat balance verification

	Description	Q_{tr}	Q_{vent}	Q_{mass}	Q_{rad}	Q_{ig}	Q_{ed}
1	Core zone, 2 nd floor		X	X		X	
2	Core zone, 2 nd floor w/ <i>ed</i>		X	X		X	X
3	Core zone, top floor	X	X	X		X	
4	Core zone, top floor, w/ <i>ed</i>	X	X	X		X	X
5	South zone, top floor	X	X	X	X	X	
6	South zone, top floor, w/ <i>ed</i>	X	X	X	X	X	X

Through simulating the aforementioned cases, it was possible to isolate the effects of each heat balance component, making it possible to pinpoint anomalies more easily. This method was particularly useful also to check the calculation of how much energy is stored in the building mass, which is based on the temperature difference between two consecutive time steps.

Results from this verification process showed that there is a consistent balance in the system. The difference between ingoing and outgoing energy, mainly due to mathematical imprecision, is always inferior to 0.03W for all zones and times tested.

7 Process study

One of the most important things to consider when designing a tool for early stages is how the process is build up. Since in early stages many changes are made and very quickly, it is of utmost importance that the tool is easily editable and provides fast feedback. In this chapter we write about the process of modelling using BeDOT and how it is compared to IDA-ICE.

7.1 Components' dependencies

Figure 22 shows the steps included in a building energy performance analysis, summarizing the process explained in chapters 4 and 5. The lateral position of a step represents the chronological order in which it happens, when compared to others. Different levels, combined with arrows, establish a relation of dependency between specified steps.

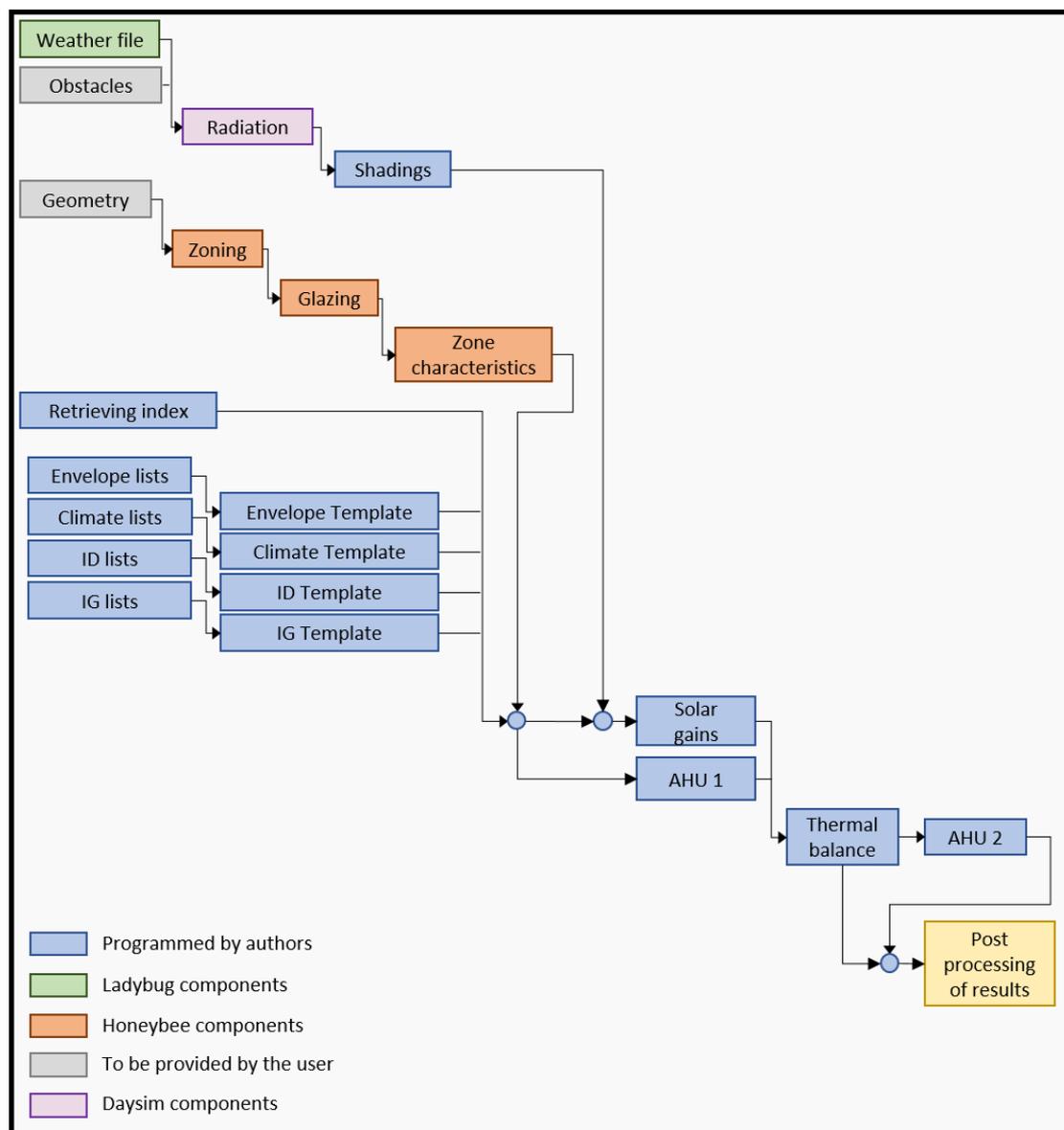


Figure 22: Gantt chart of building energy calculations in BeDOT.

One can see in the chart that after *Solar gains* and *AHU 1* there is a linear relation of dependency up to *AHU 2*. However, before these components, the flow of information follows three independent branches, related to input lists, geometry and solar radiation.

Regarding the process of building a simulation from scratch, any of these three branches could integrate the critical path, depending on the time it takes the user to implement the settings. A critical path represents the sequence of stages that determines the time a process takes to be performed. On the other hand, when regarding running time, the input lists will often be finished before the other two branches.

Since the branches are independent, the editing of one branch does not affect either the result or the computational time of the others. This can be an important advantage when dealing with a larger number of zones or more complex solar radiation meshes.

In Figure 22, since we are focusing on the three main branches, the four types of input templates are considered all at once. However, each of these templates actually affects a different part of the following process. To illustrate that, consider a simple case, of a model consisting of 100 zones, and the diagram in Figure 23.

When simulating this, the total running time was 10.4', which is represented in the figure by the largest module (square in red). Within the whole simulation, each section was responsible for a certain amount of time. Solar Gains calculations took 5.2', which corresponds to 50% of the total running time. Then the AHU and Thermal Balance calculations took 2.5' and 1.4', which correspond to 24% and 14% of the total running time, respectively.

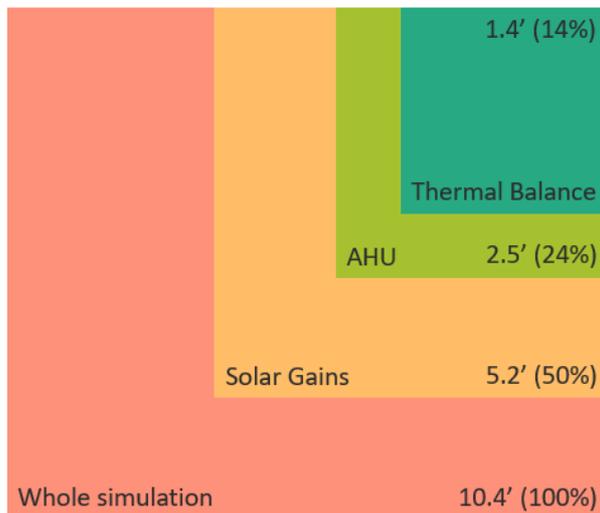


Figure 23: Running times for a case with 100 zones.

The module' position display relations of dependency in terms of time. If a module includes another, it means that any change on the first will also affect the second, but not the other way around. In other words, the Thermal Balance, being the last in the chain, is also the easiest to modify. On the other hand, if something is changed in the Solar Gains' step, both AHU and Thermal Balance will consequently have to be run again too.

This evidences that changes can be made locally, affecting strictly the parts that need to be modified and avoiding that other parts, unrelated to the change, are forced to run again. This modularity concept is adopted so that changes in the model can be performed quicker.

In practise, the Thermal Balance is directly affected by Internal Loads and Envelope Properties templates. The AHU depends on Indoor Climate template and on the AHU ID, which is a list included in the ID template. Solar Gains are directly affected by shading control. Lastly, changes made in geometry will cause the entire program to run again.

7.2 Building process

The Building process in BeDOT is rather easy compared to other softwares, such as IDA ICE. Because it fits the purpose for an early stage design, BeDOT adopts a mass modelling, in which masses can be stacked together to build up a geometry. This process is, according to the experience of the authors in this thesis work, faster than creating the geometry in IDA ICE, which is significantly more detailed.

While building geometries in IDA ICE may take several hours or even days, it is expected that a similar process in BeDOT, from geometry to results, only takes around two hours. This, of course, is expected for users that are already familiar with the tool.

An important aspect while creating the geometry in BeDOT is to consider the future use of each floors and its heights. Sometimes is better to subdivide the building in different *Breps* that are then combined to form the entire building. Thus, different *Breps* can be created for different building uses or heights. How these *Breps* can be put together is explained in *BeDOT Manual*.

7.3 Kunskapsdagen workshop

At the end of April, a series of workshops were held in Bengt Dahlgren, as part of the company's *Kunskapsdagen* activities. One of these workshops, attended by the authors together with 4 people from Energi och Inomhusmiljö division, was focused on using BeDOT. The workshop aims were: to model two buildings simultaneously, experimenting geometries and shading obstacles; to evaluate the buildings' energy performance; to identify flaws in the program or things that needed improvement and to evaluate user-friendliness.

The buildings modelled in this task are displayed in Figure 24. The first building is a 3-storey high office building, with a commercial ground floor and a technical room on the roof. The second building is entirely residential. The task of modelling two buildings at once, i.e. in the same model, was in itself already enough to bring to surface a few important issues. Firstly, since each part of the buildings had a different glazing ratio, limitation in Honeybee components made it necessary to have the geometry and zoning of each part modelled separately. Consequently, there was a redundancy of many components, which made the canvas look very messy.

The second problem concerned the relation between zones and buildings (or building sections). Since the model dissolves the geometry into zones and merges all zones together, at the time of the workshop there was still no way of splitting up this list of

zones back into two separate buildings. This experiment showed us the need for implementing a pre-processing indexation list that links the zone to its corresponding building, so it is possible to obtain energy results per building in a post-processing stage.

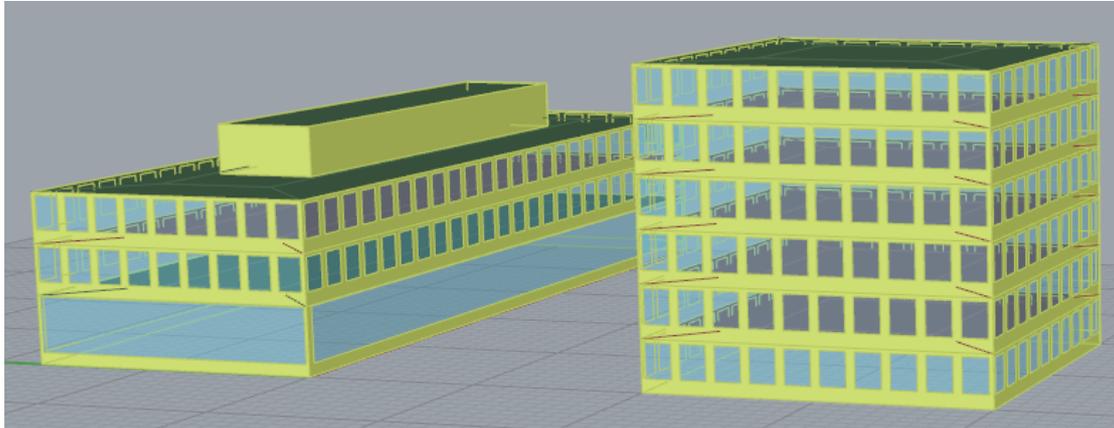


Figure 24: Kunskapsdagen workshop geometries.

Another discussed topic was how to implement a control system for internal and external shadings. It has been decided to include external shadings as a reduction of solar radiation incident over each window, and that they should vary as a function of time, irradiance and external temperature.

Balconies have been tested as shading obstacles in the building and apparently perform as expected, provided that the mesh is fine enough. The possibility of implementing shading control was tested at window level, however this level of detail in the control could end up in a much higher computational time.

The input from two new users have been a large contribution to the project, mostly regarding practicalities when inserting input data. One suggestion was to adapt some inputs to the format they are provided in BEN and Skanska database. For instance, lighting heat gains are provided as a single energy value per year, so in this case we should need to go backwards and estimate a working schedule from in value, assuming a certain percentage of this energy is applied at every hour of the day.

It was also discussed the importance and the need for preparing good filtering functions that are able to connect Rhino's geometry with input lists, making it easier to visualize and edit the lists information. Additionally, the program should count with a nice visualization method that displays clearly its results and major conclusions, preferably also in the 3D interface.

8 Results

BeDOT model was compared against IDA ICE, a software commonly used in building energy analysis in Sweden. Different cases were simulated in both models to benchmark BeDOT.

8.1 Solar gains nodal distribution in ISO 13790

During the model validation, a series of discrepancies were found by running different cases. Trying different scenarios under extreme conditions, such as high loads with low ventilation or vice versa, it was found that the distribution of the solar radiation in the standard does not behave as IDA ICE. More cases were tested, this time with no extreme values, and the issue with solar distribution still remained. How ISO 13790 distributes solar gains is shown in equations (9) to (11).

Different distributions of solar gains were tested to analyse how the BeDOT behaves. Also, in order to make the comparison, both models were tested with the same solar gains. These distributions are shown in Table 10 and the different results are plotted in Figure 25.

Table 10: Nodal distribution analysis.

BeDOT ISO 1 (Case 1)	ISO 13790
BeDOT AIR	Solar gains placed in air node.
BeDOT Equally dist.	Solar gains distributed as the internal loads.
BeDOT 50/50 (Case 4)	50% solar gains to air node and 50% to the surface node.

As a result of this analysis, it can be seen that when solar gains are distributed equally between the surface node and the air node BeDOT follows the trend closer than other distributions. Detailed results for when BeDOT has ISO nodal distribution, and when solar radiation is distributed between surface and air node are shown in this section.

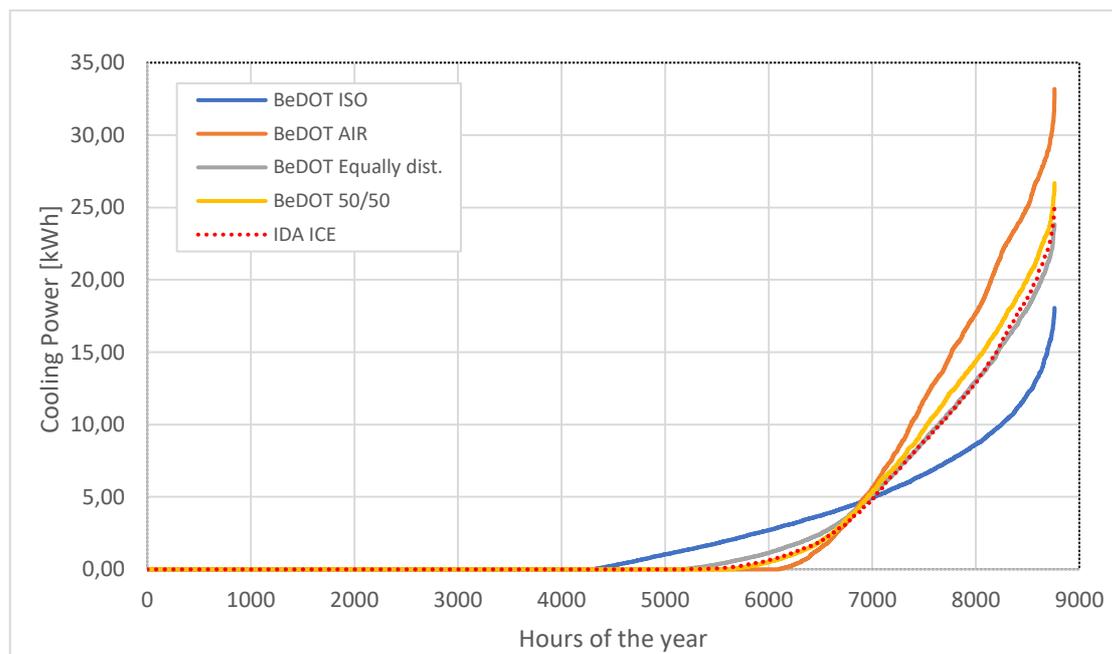


Figure 25: Total airflow through the AHU for different nodal distributions.

The case shown in Table 11 was run when solar radiation was distributed following equations (9) to (11) and it shows that both models have different tendencies (see Figure 26).

Table 11: Case 1, inputs for BeDOT and IDA.

Geometry		
	Windows u-value	1,9 W/m ² K
	Window g-value	0.68
	Windows ratio	50% each side
	U-value wall	0.2 W/m ² K
	U-value roof	0.172 W/m ² K
	Floor area	600 m ²
Air Handling Unit		
	Heat Recovery	OFF
	Air Heater	ON
	Air Cooler	ON
	Ventilation System	VAV – Air cooling
	SFP	1.5
Other inputs – constant for all zones		
	Internal loads	15 W/m ²
	Schedule loads	8 to 18h
	Hygienic airflow	0.35 l/sm ²
	Infiltration	0.02 l/sm ²
	Thermal bridges	0.05 W/m ² K
	set point heating	20 °C
	set point cooling	24 °C

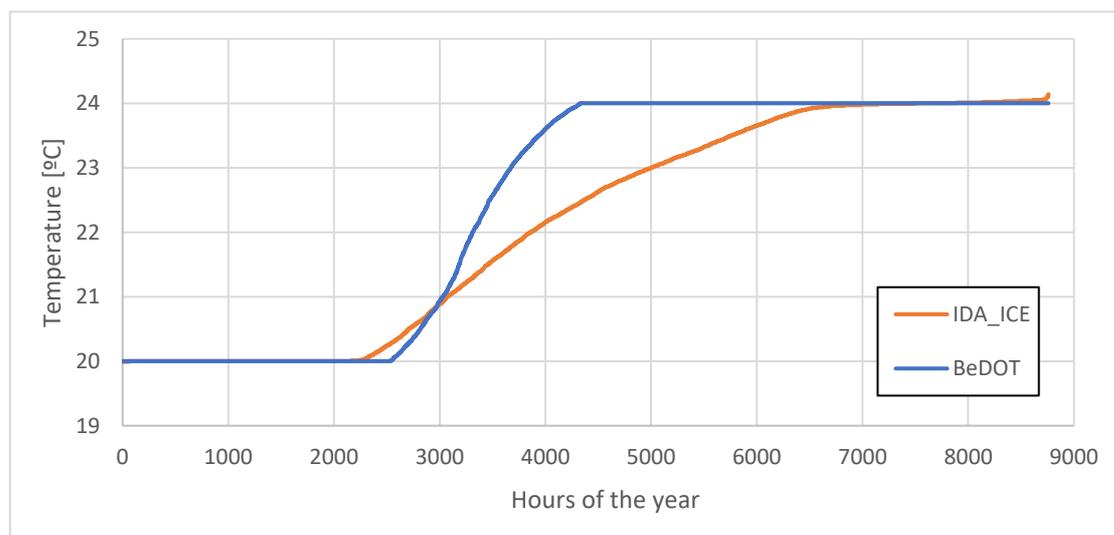


Figure 26: Case 1, temperature in a room facing south.

In Figure 26 one can see that BeDOT has higher temperatures, where the temperature in the zone is 24 degrees around 2000 hours more than IDA.

In Figure 27 the airflow through the AHU is plotted. In this figure IDA model requires more airflow, apart from hygienic, to cool down the zones. That is easily compared by looking at the areas: the area where IDA requires more airflow than BeDOT is larger than the hours where BeDOT's demand is higher. These differences in the airflow explain why in Figure 28 AHU cooling has this peak. AHU cooling in IDA has a demand of 7.1 kWh/m^2 while BeDOT requires 4.6 kWh/m^2 . The absolute difference in this parameter is 2.5 kWh/m^2 (35%). On the other hand, AHU heating in IDA has a demand of 65.1 kWh/m^2 and in BeDOT 67.2 kWh/m^2 . In this case the error is 2.1 kWh/m^2 (3.3%).

It is important to look at both errors. Here, district heating or cooling depend on the same airflow that is going through the AHU, but the relative errors differ a lot. However, by looking at the absolute value, the difference is not very significant.

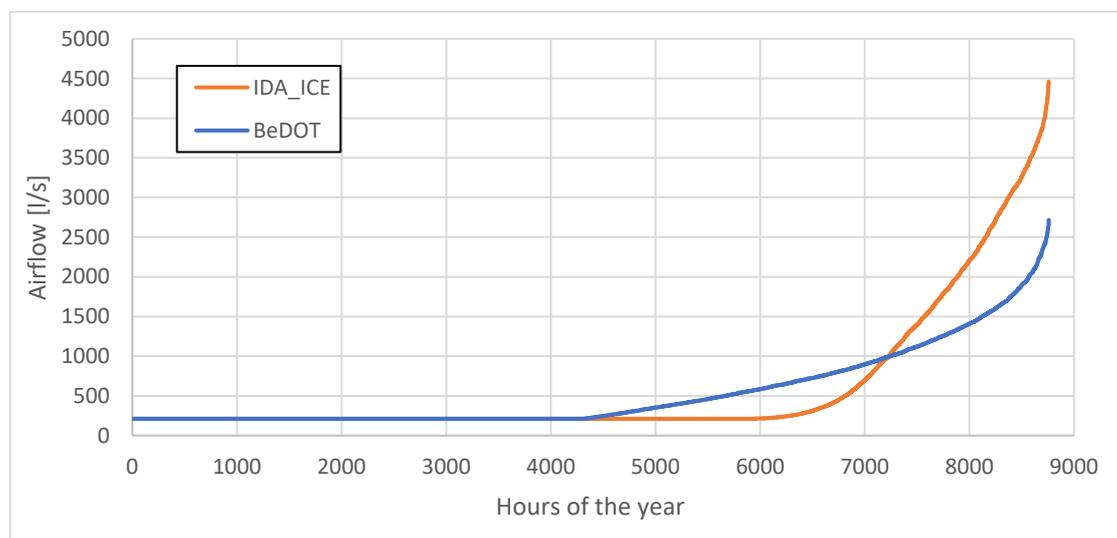


Figure 27: Case 1, Airflow through the AHU for different nodal distributions.

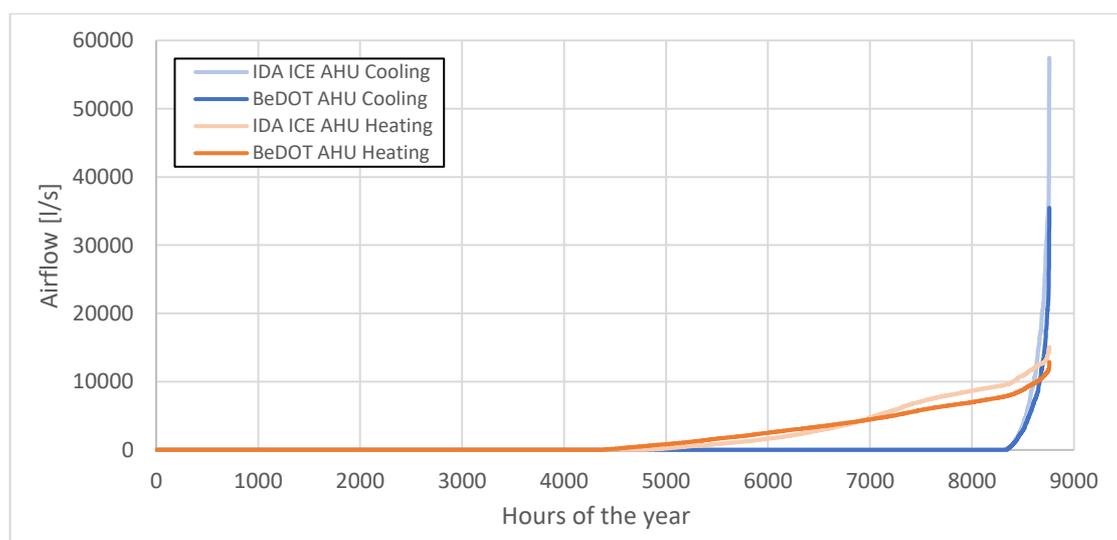


Figure 28: Case 1, AHU Heating & Cooling for different nodal distributions.

Therefore, after running all the cases in Figure 25, the one that distributes the solar gains equally between the surface node and the air node is the case following IDA's trend the most. This case is explained in detailed in the following figures.

The inputs for the model are still the same as the ones in Table 11, but now with a different distribution.

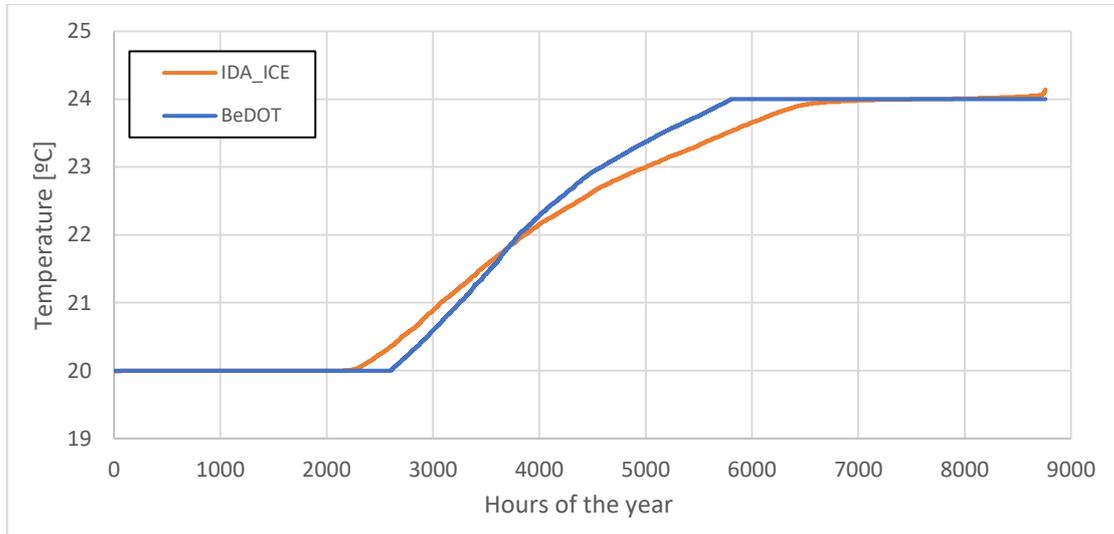


Figure 29: Case 2, temperature in a room facing south.

The temperature distribution of Figure 29 is following the path of IDA much closer than the one shown in Figure 26, which is a good indicator that this nodal distribution of solar gains is behaving well.

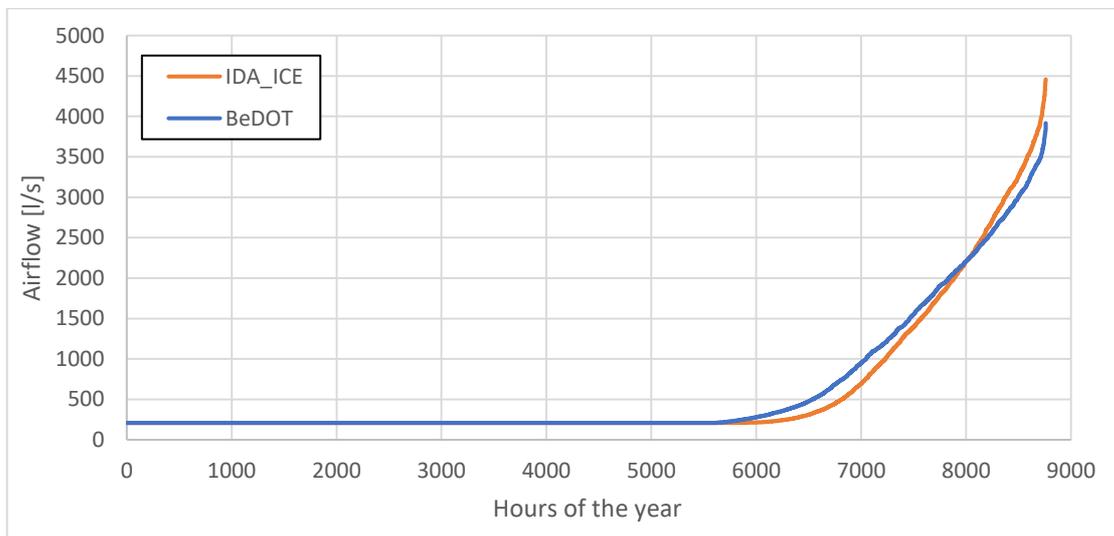


Figure 30: Case 2, Airflow through the AHU for different nodal distributions.

The improvement in the airflow is clear, the curves in Figure 30 are much closer than the ones in Figure 27. Now that the airflow is more accurate, the difference between BeDOT and IDA for district heating and cooling would be lower, as it is presented in Figure 31.

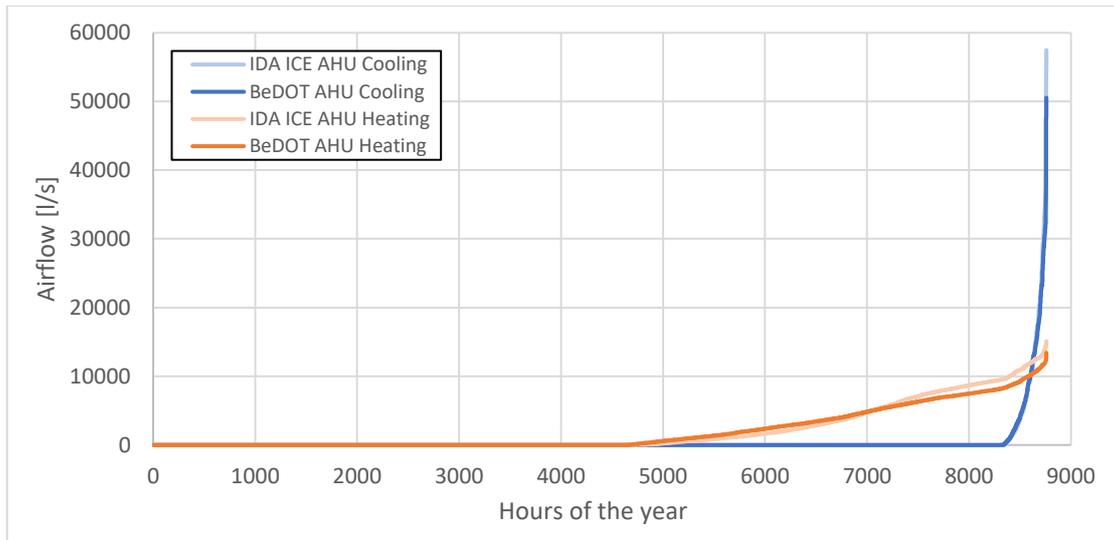


Figure 31: Case 2, District Heating & Cooling for different nodal distributions.

For Case 2, IDA is still giving the same results as Case 1, but there are differences in BeDOT. AHU heating is now 69.8 kWh/m^2 which gives an absolute error of 4.7 kWh/m^2 (7.3%). AHU cooling demand is now 6.8 kWh/m^2 and its error is 0.3 kWh/m^2 (3.6%).

By doing this, the absolute error in AHU heating is larger but is still in between our limits when comes to relative errors (less than 15%). The agreement in AHU cooling is now much better than using ISO 13790 solar distribution between the air, surface and mass node.

Transmissions through the building envelope are presented in Figure 32, and the correspondence between IDA and BeDOT has a relative error below 5% for all cases.

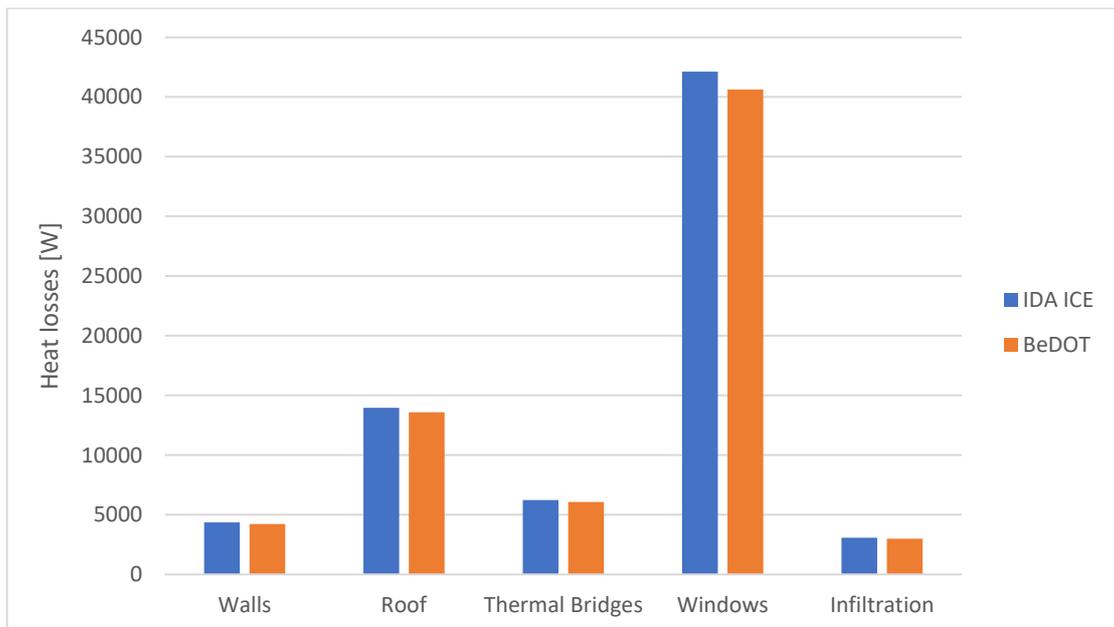
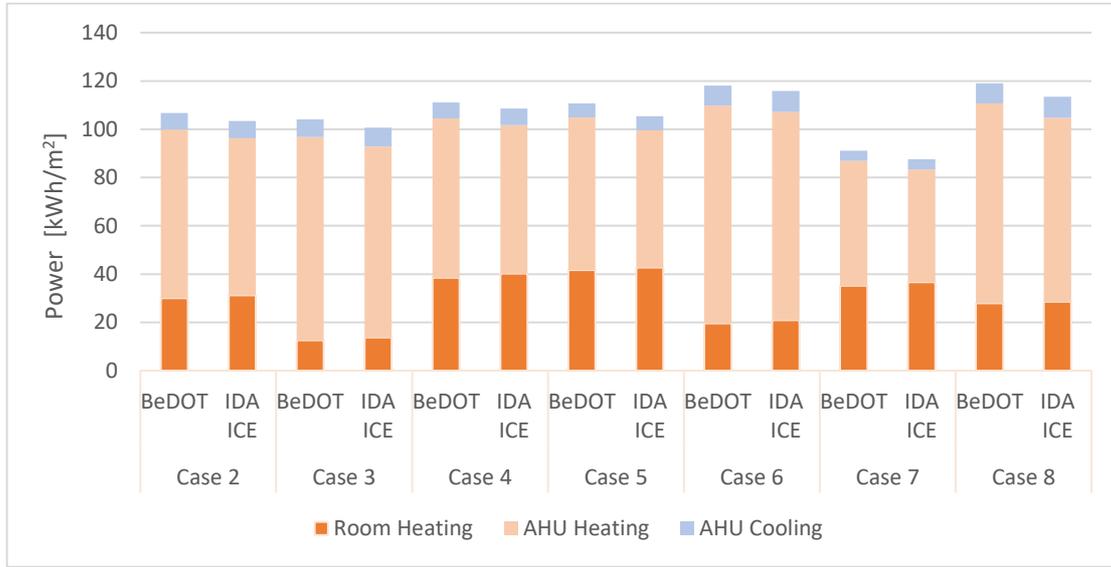


Figure 32: Case 2, transmissions through the different building elements.

One may realize that the transmission losses due to thermal bridges are high. BeDOT's energy module approach thermal bridges as a function of the total envelope

area (external walls, windows and roof) in the same way IDA ICE does, and in this case this area is 930 m², which explains why the transmission losses have this value.

However, changing the solar distribution of BeDOT, and by doing this modifying the ISO, is a step that must be taken with caution. Thus, a few cases were simulated to motivate the change of the solar gains distribution. These cases are based on input parameters listed in Table 11 with some differences (see Figure 33).



Case 2: Default case for solar gains analysis.

Case 3: U-value of the windows was reduced to 0.9 W/m²K.

Case 4: Infiltration was increased to 0.08 l/sm².

Case 5: Internal gains were reduced to 7.5 W/m².

Case 6: Internal gains were increased to 25 W/m².

Case 7: G-value of windows was reduced to 0.4.

Case 8: G-value of windows was increased to 0.85.

Figure 33: Comparison of different cases with solar gains distributed equally between surface and air nodes in BeDOT and IDA.

From this analysis it is confirmed that this distribution of solar gains is accurate and verified with IDA, so it has been decided to adapt the heat balance to this distribution by changing equations (9) to (11) to the three equations presented below.

$$\phi_{ia} = 0.5 \cdot \phi_{int} + 0.5 \cdot \phi_{sol} [W] \quad (40)$$

$$\phi_m = \frac{A_m}{A_t} \cdot 0.5 \cdot \phi_{int} [W] \quad (41)$$

$$\phi_{st} = \left(1 - \frac{A_m}{A_t} - \frac{H_{tr,w}}{9.1 \cdot A_t}\right) \cdot 0.5 \cdot \phi_{int} + 0.5 \cdot \phi_{sol} [W] \quad (42)$$

8.2 Analysis on Air Handling Unit components

Proceeding with the comparison between the results from BeDOT and the reference IDA-ICE, this section focuses on the effect of air handling unit components. This study was done by selecting a base case (Case 9), changing gradually its AHU operation and analysing the consequences and inaccuracies of each step.

Case 9 consists on a 5-zone building of 600m² of floor area, at first containing no AHU component besides a fan. It is supposed in this case that the building is ventilated with air at outdoor conditions. The case was modelled both in BeDOT and IDA-ICE using the input data shown in Table 12.

Table 12: Case 9, inputs for BeDOT and IDA ICE.

Geometry		
	Windows U-value	0.9 W/m ² K
	Window g-value	0.25
	Windows ratio	50% each side
	Roof U-value	0.172 W/m ² K
	Floor area	600 m ²
Air Handling Unit		
	Heat Recovery	OFF
	Air Heating	OFF
	Air Cooling	OFF
	Hygienic airflow	VAV (compatible to load schedule)
	Zone cooling with air	OFF
	SFP	1.5
	Supply temperature	Outdoor temperature
Other inputs – constant for all zones		
	Internal loads	15 W/m ²
	Load schedule	8 to 18h
	Hygienic airflow	0.35 – 0.70 l/sm ²
	Airflow schedule	8 to 18h
	Infiltration	0.04 l/sm ²
	Thermal bridges	0.05 W/m ² K
	Set point heating	20 °C
	Set point cooling	24 °C

Figure 34 contains the mean air temperature duration diagram for a zone located facing South. On the secondary Y-axis it is plotted the absolute difference between BeDOT and the reference. From the figure, it is possible to perceive that BeDOT's

room temperature is higher when free-ranging between the setpoints, although both curves increase at the same ratio. Observing how the absolute difference varies through the year, most of the overestimation happens during the summer. The figure also shows that both methods have approximately the same amount of heating and cooling hours.

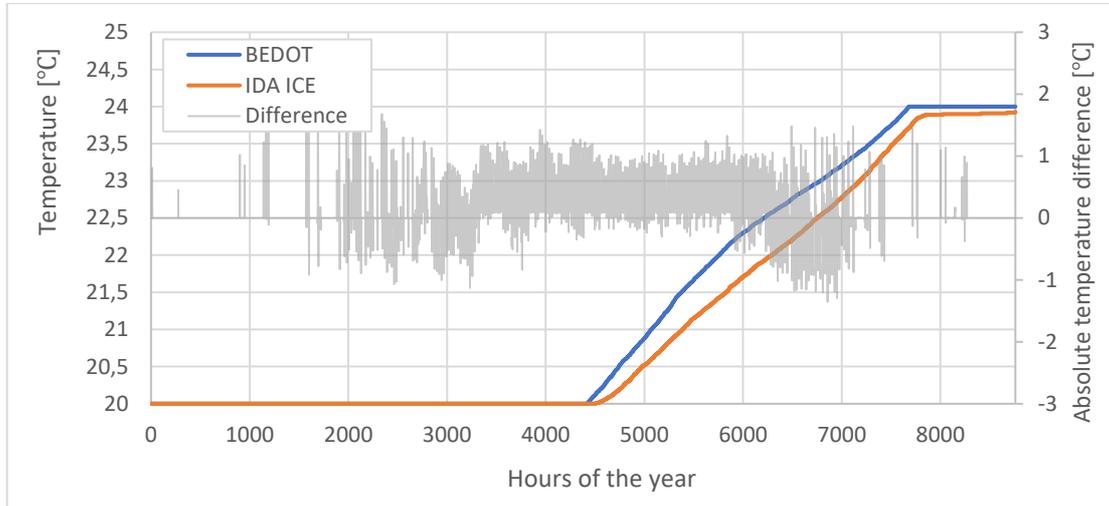


Figure 34: Case 9, mean air temperature in a zone facing south.

The room heating and cooling powers can be seen in Figure 35. The cooling power curves look almost identical, indicating a similar demand, during the same amount of hours. Regarding room heating, both methods find it necessary to provide heating power during approximately 5700 hours of the year. Although BeDOT tends to underestimate it in 3700 of such hours, it is clear that both curves follow the same pattern.

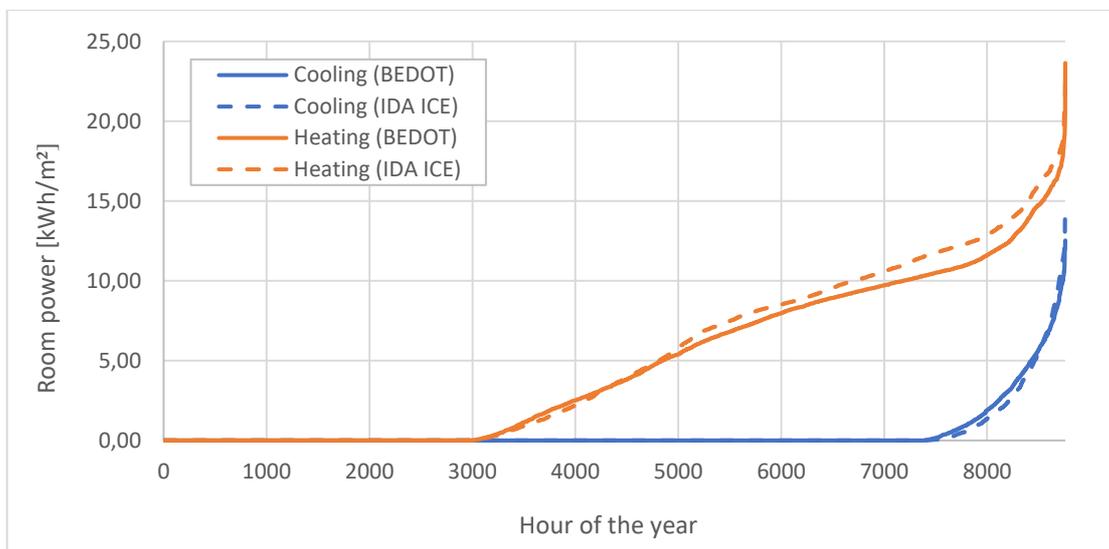


Figure 35: Case 9, room heating and cooling power.

After Case 9, three cases are introduced as comparison, mainly differing on the AHU operation. Table 13 summarizes the peculiarities of each case.

Table 13: Summary of main differences in cases 10 to 12.

Case 10	Case 9 + AHU cooling and heating for hygienic airflow + constant supply temperature
Case 11	Case 10 + Heat recovery unit
Case 12	Case 10 + room air cooling and varying supply temperature scheme

Case 10 is characterized by the addition of air heating and cooling coils in the AHU, to modify the temperature of hygienic air volume up to desired supply temperature (in this case, constant 18 °C). Figure 36 shows the new mean air temperature distribution in the zone, which has shifted when compared to case 9. Now both methods show more hours of required cooling power, but less hours of heating power. Also, different from the previous case is the fact that BeDOT requires a longer heating period, and for around 1000 hours its temperature is inferior to IDA-ICE's. The former pattern is soon recovered, and for the following free-ranging hours BeDOT resumes reaching a higher mean temperature.

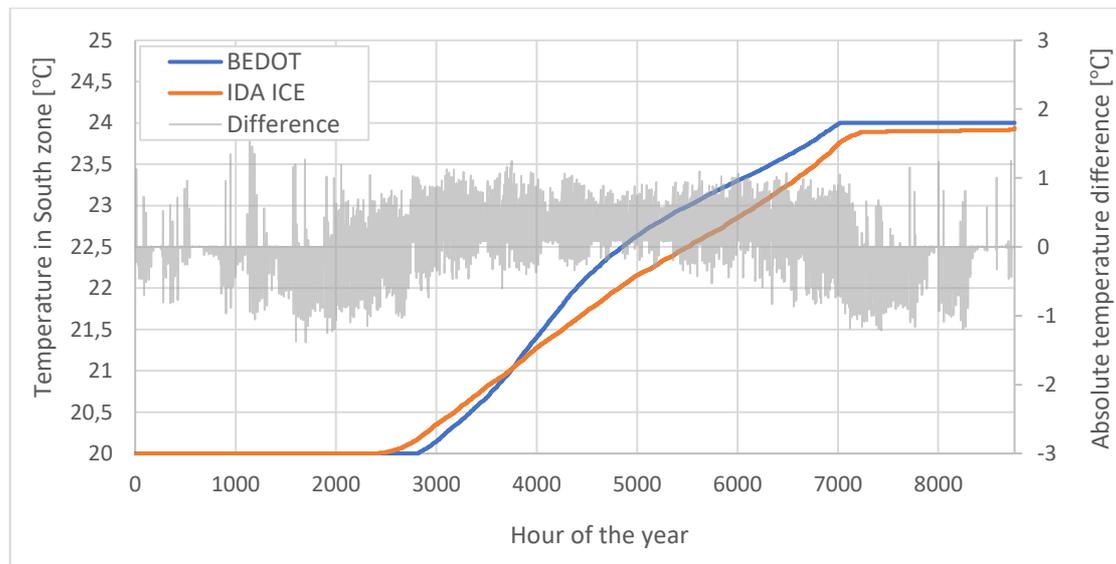


Figure 36: Case 10, mean air temperature in a zone facing south.

Figure 37 and Figure 38 display room and AHU heating and cooling powers for Case 10. As the previous case, room heating power differs more than the cooling, which almost completely coincides. However, the establishment of a constant supply temperature has decreased room heating power to approximately half the original value. Another consequence is that power needs to be supplied to the AHU's heating and cooling coils. In Figure 38, BeDOT's and IDA-ICE's curves match throughout the whole year, which is why only two lines can be identified straight away. This last figure leads to the conclusion that the AHU's heater and cooler are working as intended, and are apparently not introducing any significant inaccuracies to BeDOT model.

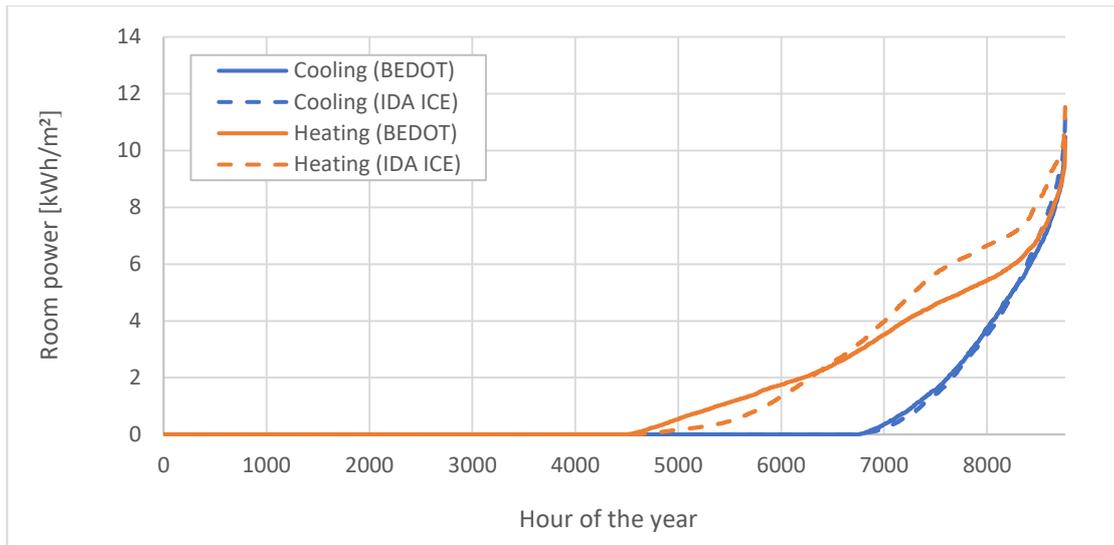


Figure 37: Case 10, room heating and cooling power

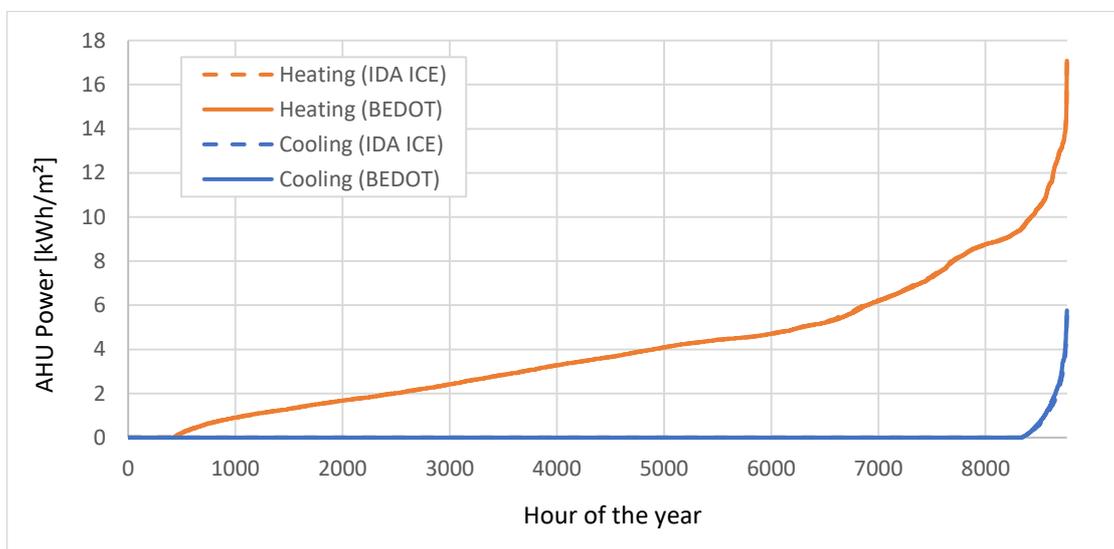


Figure 38: Case 10, AHU heating and cooling power

The following case, Case 11, differs from the previous one only because it has an operating heat recovery unit. Since this system, in this case, only causes changes on the AHU power results, mean air temperatures and room power remain the same as in Case 10.

The effect of a heat recovery unit of efficiency 70% over the AHU power is displayed in Figure 39. Although the heat recovery unit is set to work all year long, it generates more significant during cold periods, by preheating the air before it reaches the heating coil. The result is a dramatic reduction in AHU heating power. Since the return temperature in BeDOT is estimated, instead of calculated, its curve is differing from IDA-ICE's on approximately 6000 hours. Also because of return temperature is estimated, BeDOT tends to underestimate the heat recovery unit's capacity, as shown in Figure 40. The tendency for underestimation is good to the analysis, as it is a conservative approach. In both figures the two models follow the same pattern, which also indicates that the heat recovery unit operation imposes no obstacle to energy performance calculations.

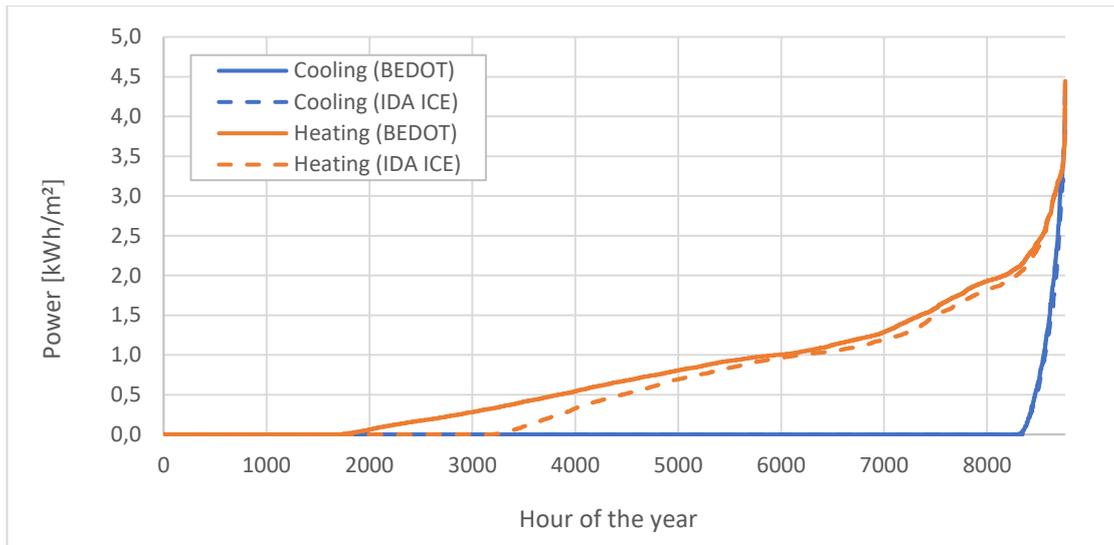


Figure 39: Case 11, AHU heating and cooling power.

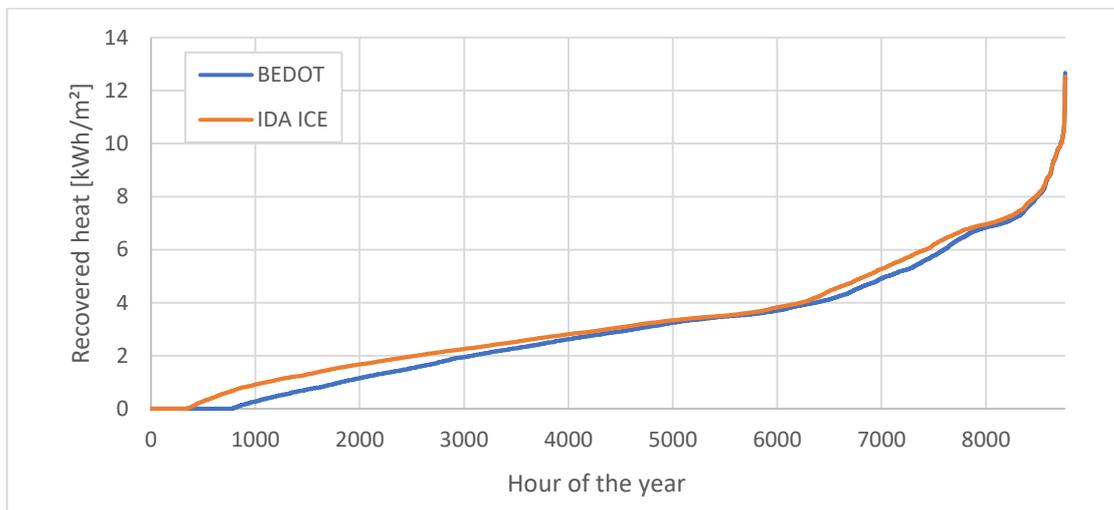


Figure 40: Case 11, recovered heat.

Finally, Case 12 introduces the second AHU component. As explained in chapter 5.2.2, this AHU component represents the process of zone cooling with air, which means a higher airflow needs to be provided to the zones. In order to simulate a more realistic case, the supply temperature scheme was modified to a function of outdoor temperature.

From Figure 41, it can be seen that even though the supply temperature is no longer constant, the mean air temperature behavior changes very little when compared to Cases 10 and 11. The introduction of extra air volumes, with the purpose of cooling, is apparently not influencing accuracy much either. Figure 42 shows that the magnitude of airflow differences between the models can be sometimes very large. However, the duration diagrams display the same tendencies, which means that the differences are likely compensating each other over time.

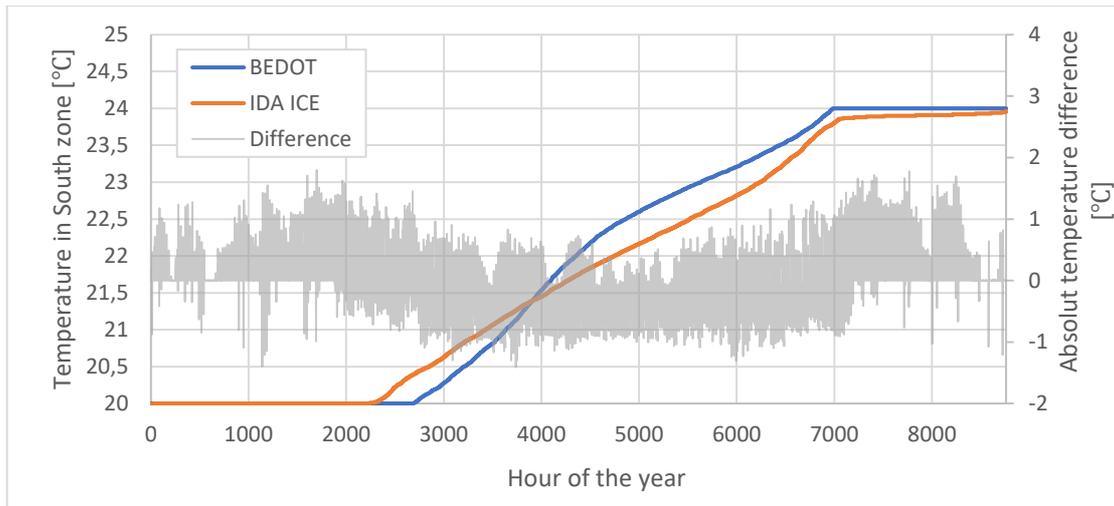


Figure 41: Case 12, mean air temperature in a zone facing south.

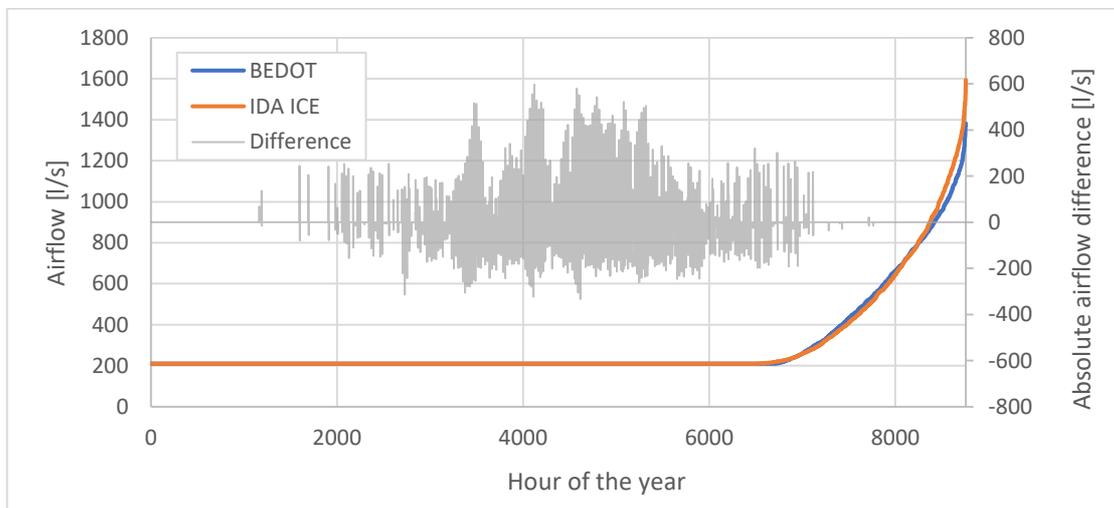


Figure 42: Case 12, airflow rate.

As a result of introducing an ideal air cooling, the room required cooling power is fulfilled. The only room power now is heating, as shown in Figure 43. The room heating power again much resembles the result from Cases 10 and 11.

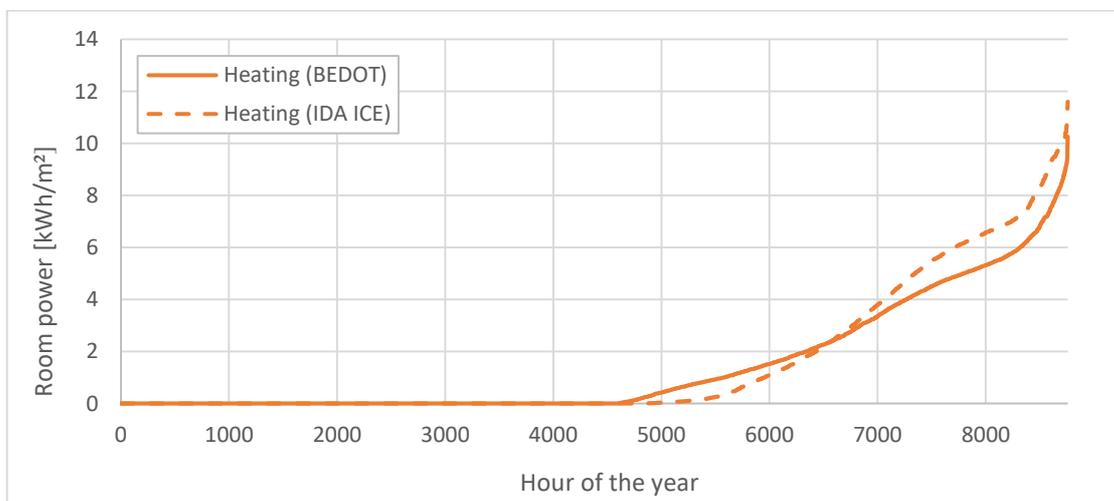


Figure 43: Case 12, room heating power

Internal Heat Capacity analysis

Thermal mass in BeDOT is given per square meter of floor area. This parameter is tabulated according to the standard (see Table 6) and it is an important step in the process of designing a building with BeDOT.

When validating the model against IDA, it was pointed out that defining this parameter is not straightforward. BeDOT does not require material properties of the wall, differently from IDA, to account for the thermal capacitance of the zone. Then, this parameter needs to be specified by the user, whether the building is light or heavy.

To analyze how BeDOT models are affected by this parameter, Figure 44 synthesizes the behavior of a 5-zone model when different loads are tested against different internal heat capacities.

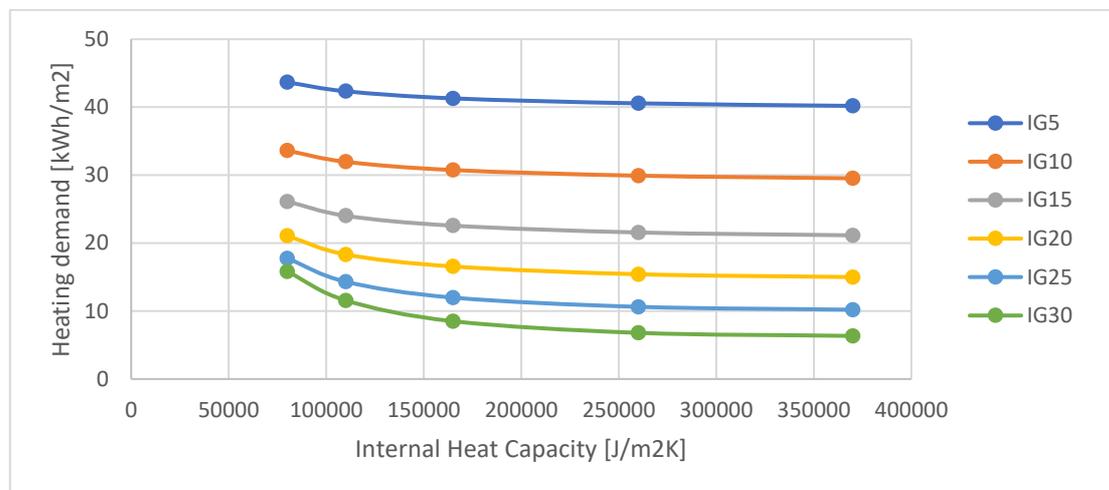


Figure 44: Internal Heat Capacity analysis. Heating

For the cases shown in Figure 44, different thermal capacitances do not show a large effect when the loads are small. However, if the loads are increased to 30 W/m^2 , the heating demand could be decreased around 60%. Following the same test case, the cooling load could be reduced around 23% for high internal loads (see Figure 45).

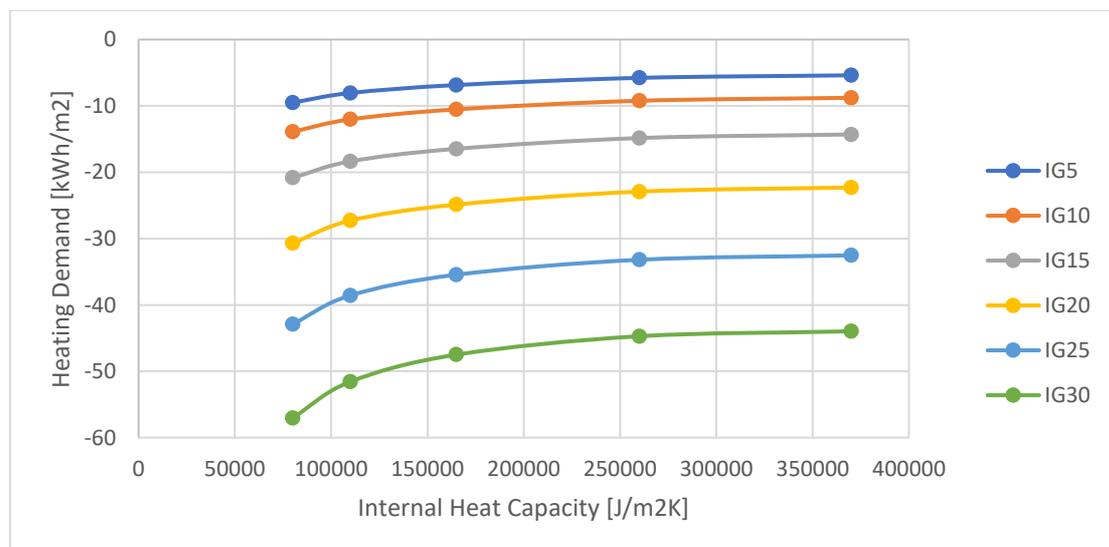


Figure 45: Internal Heat Capacity analysis. Cooling

8.3 Sensitivity analysis of heat balance

After the different AHU components were analysed and their performance validated, a sensitivity analysis of different designs was carried out to evaluate the effectiveness of the heat balance component. This analysis was conducted with a wide range of combinations. These combinations were based on four main parameters: U-value of windows, hygienic airflow, infiltration and internal gains. An overall view of the result is shown in Figure 46, and in Table 14 the variation of this parameters is presented.

Table 14: Sensitivity analysis. Combinations of parameters

Parameters	Min	Max	Step	Unit
U-value windows	0.8	1	0.2	W/m ² K
Hygienic airflow	0.1	1	0.1	l/sm ² (floor)
Infiltration	0.01	0.1	0.01	l/sm ² (façade)
Internal gains	10	25	5	W/m ²

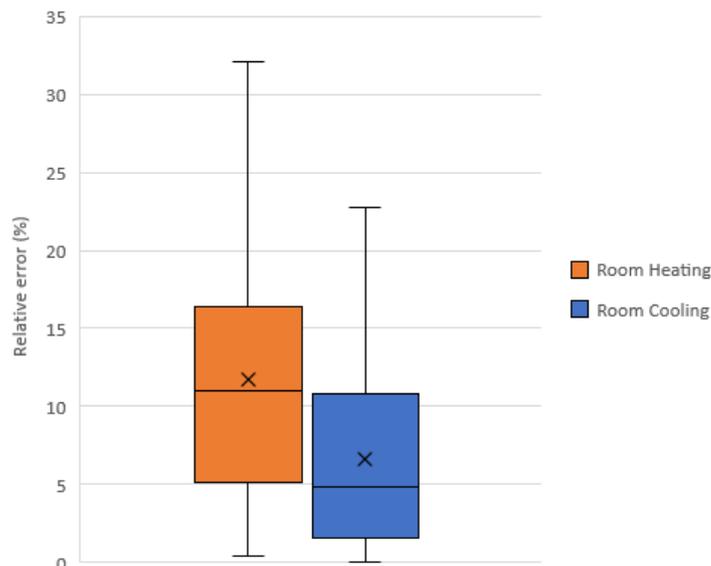


Figure 46: Results of sensitivity analysis of heat balance.

A total of 800 cases were tested against IDA. In Figure 46, the cases are represented by the relative error between BeDOT and IDA. Regarding room heating, 69% of the cases had an error below 15%, while in the case of room cooling this percentage increases to 88%. Some cases are still above the margin of 15%, mostly regarding room heating. However, this higher error may have a small impact. This occurs when the absolute different in kWh/m² is acceptable. Room heating cases are plotted in Figure 47.

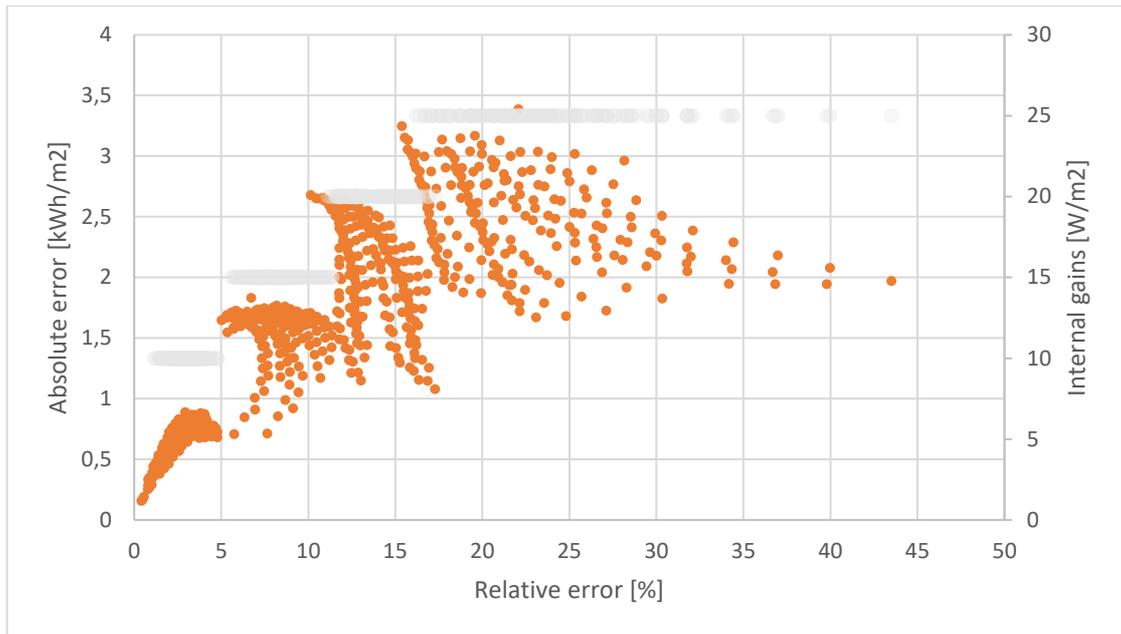


Figure 47: Sensitivity analysis of Room Heating.

The absolute error in these cases is not higher than 3.5 kWh/m^2 , so even though relative errors above 15% the absolute error value is acceptable. However, one can see that high internal gains cause an increment on the relative error. This is mainly because of the nodal distribution of loads that the standard proposes. Increasing the loads in one of the nodes increases the chance to uneven the local balance each node has with their most adjacent resistances (see Figure 7).

The same cases are plotted in Figure 48 for room cooling.

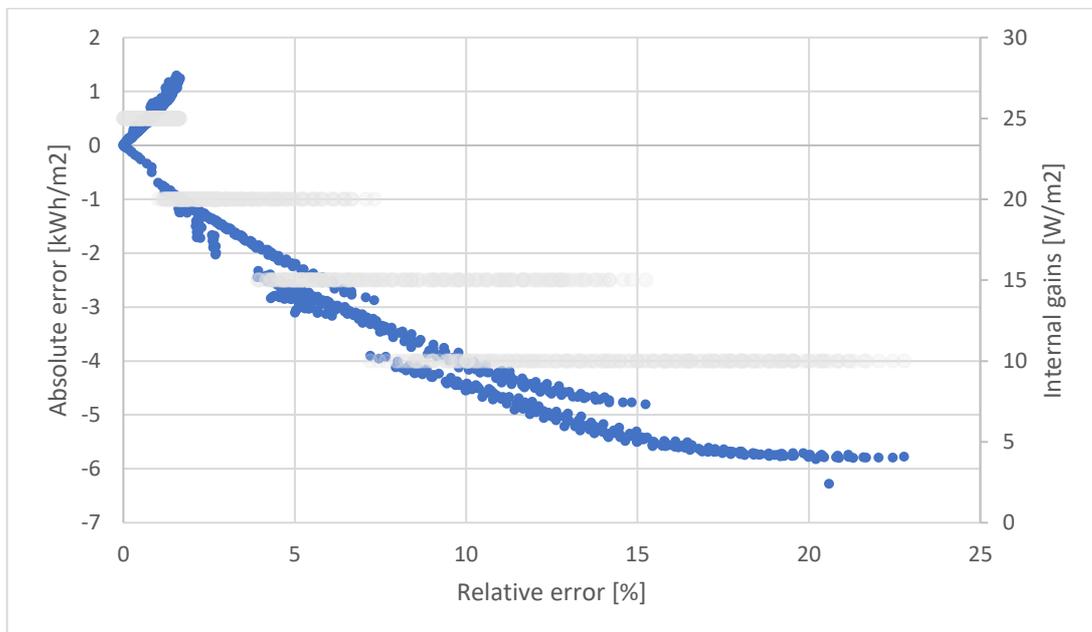


Figure 48: Sensitivity analysis of Room Cooling.

In these cases, the relative error increases while decreasing the internal gains. In the same way that increasing the loads uneven the local balance of each node in the heating demand, when it comes to cooling demand the same happens when decreasing the internal loads.

Notice that plots in Figure 47 and Figure 48 present also the internal load that was implemented in each case, so it is easier for the reader to localize the cases with low/high internal loads.

Despite this chapter being a sensitivity analysis of the heat balance, the cases were later simulated with an AHU. It should be pointed out that out of the 800 cases none of them showed an error higher than 5% regarding AHU heating and 0.5% for AHU cooling.

8.3.1 Zone comparison

With the purpose of investigating possible sources of inaccuracies, another sensitivity analysis was run, this time focusing on what happens at each zone. Similarly to the previous case, the model containing 5 zones (4 facing the cardinal directions and 1 in the center) was subjected to 49 combinations of internal gains and airflows. The results obtained in BeDOT are then compared to those from the correspondent case in IDA-ICE. Each zone's annual heating and cooling power, together with the average of all 5, are represented in Figure 49 and Figure 50 respectively.

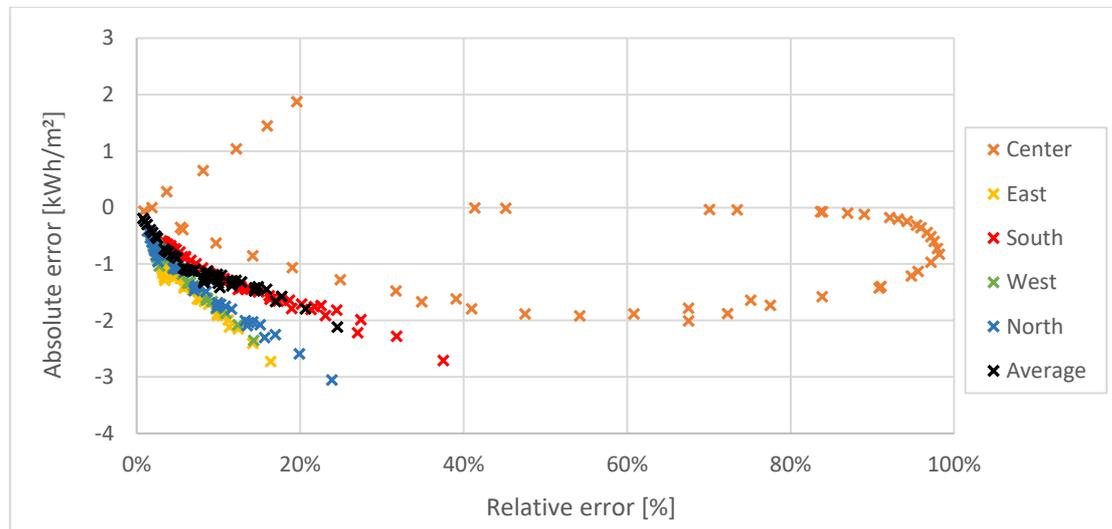


Figure 49: Heating power errors. Comparison among zones.

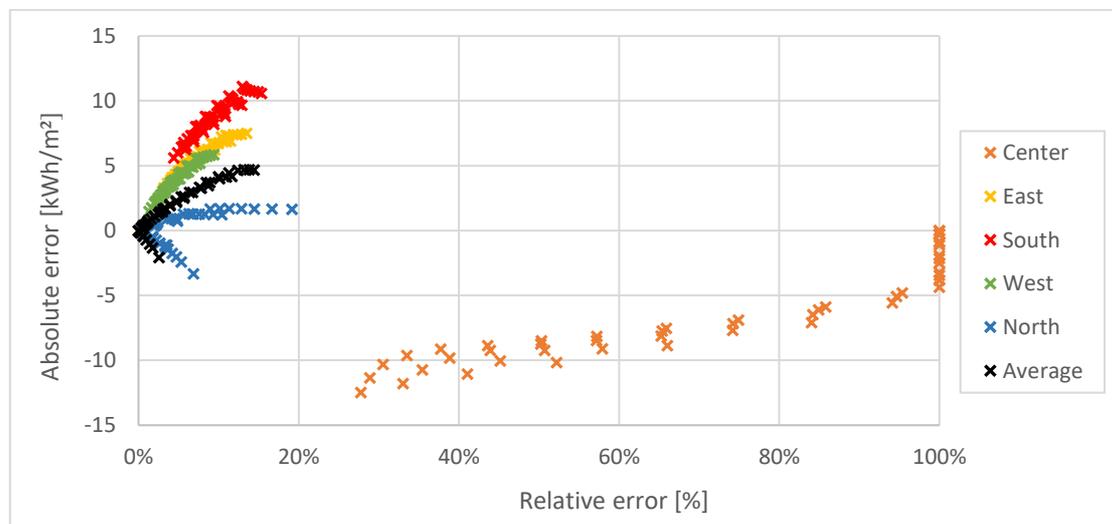


Figure 50: Cooling power errors. Comparison among zones.

It is possible to see that both in heating and cooling power all zones follow roughly the same pattern, except for the central one. Generally, the relative error increases as there is a raise in internal gains. When comparing to IDA-ICE's results, this corresponds to having a gradually larger underestimation of heating power and overestimation of cooling power.

From the figure it is also possible to identify differences due to solar radiation. When comparing zones North and South, the absolute error in heating power is larger on the North zone, while a higher error in absolute cooling power is seen in the South zone. In spite of that, both zones usually stay within a range of low relative errors, also displayed in Figure 51 and Figure 52 respectively.

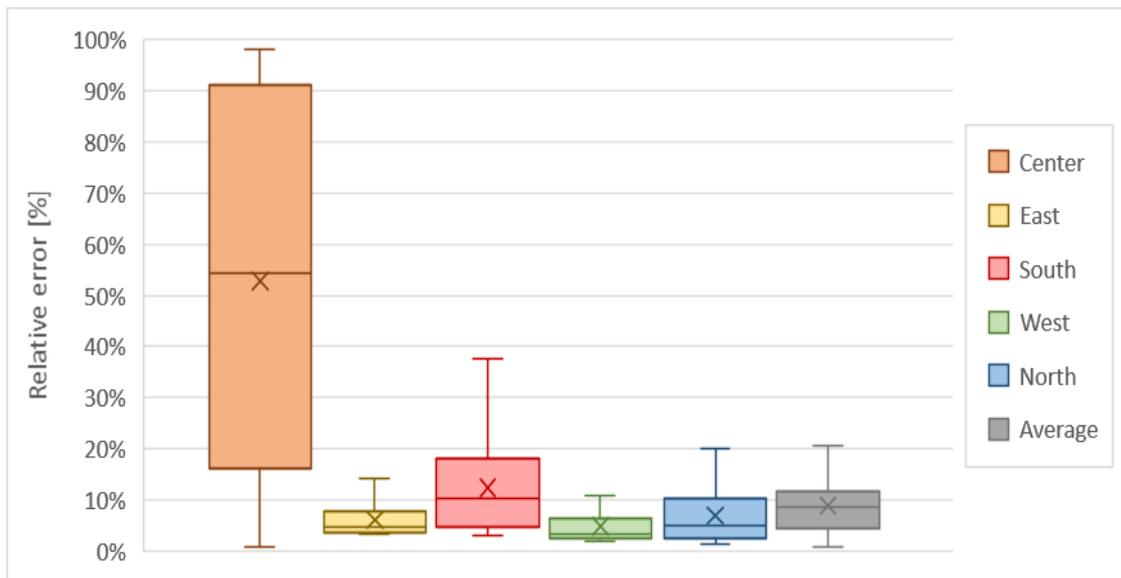


Figure 51: Spread analysis on heating power relative errors.

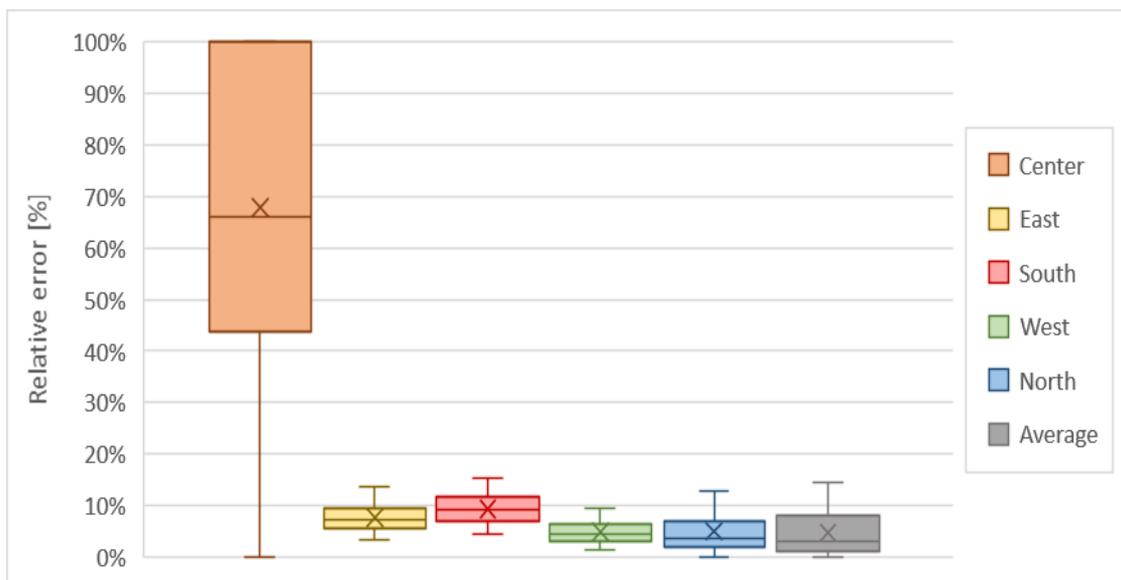


Figure 52: Spread analysis on cooling power relative errors.

Regarding the central zone, it greatly diverges from the pattern followed by the other zones, showing a tendency for underestimation of cooling power and reaching much higher relative errors. This zone is subjected to a much lower (in some cases inexistent) heat transmission and infiltration through the envelope, while solar radiation gains are also often absent. Consequently, in its thermal balance calculation the surface and mass nodes are subjected to much more internal gains than transmission losses. It has been noticed that this unevenness makes the model more prone to expressive errors.

Apart from the central zone, all others display an adequate level of accuracy. The South zone displays a relative error lower to 15% on approximately 65% of the studied cases, number that grows to 86% for the North zone. Zones East and West meet the same 15% criterion in all cases.

8.4 Case Studies

Case Study A

This case study is based on a real project design and provided by Bengt Dahlgren. The building consists of a ground floor with a commercial area, while the other floors are residences. However, the second and the third floors have different loads. The building energy demand was estimated in IDA ICE and the geometry is shown in Figure 53.

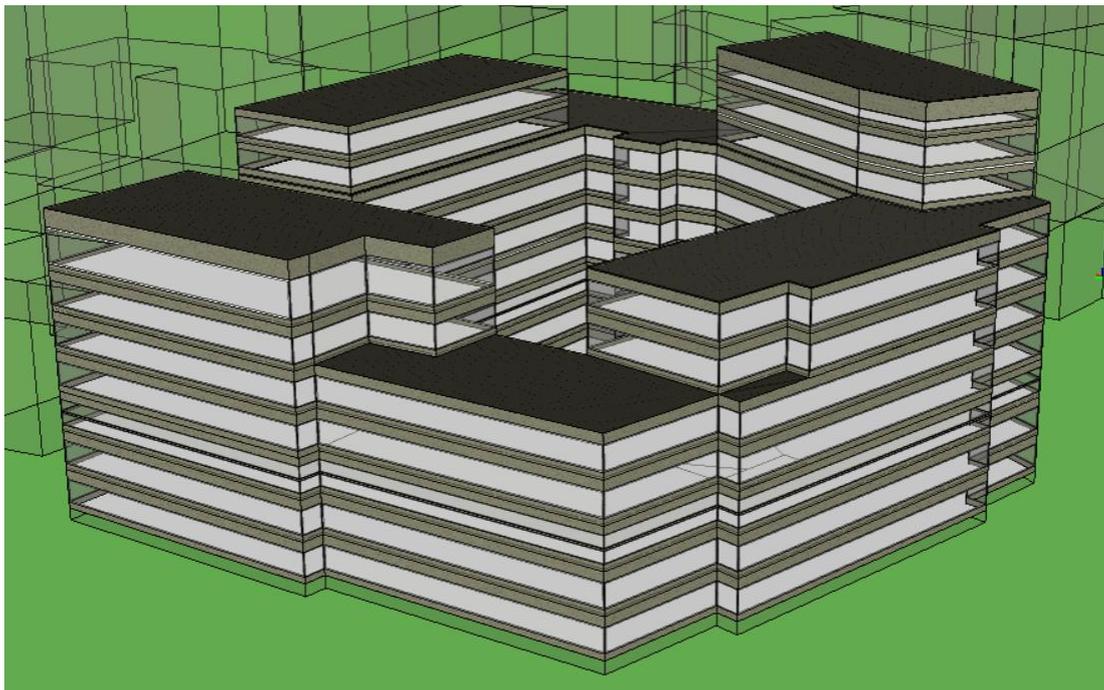


Figure 53: Case study A, geometry.

To simulate the building in BeDOT, a set of input data for each zone within the building needs to be assigned, as it is explained in section 4.6. The building contains two AHUs, one for the commercial area, and the other one for the residential floors. While the commercial floor has an AHU with heat recovery, air heater and air cooler, the residential areas do not need any air cooler.

Different transmissions through the building elements are compared in Figure 54.

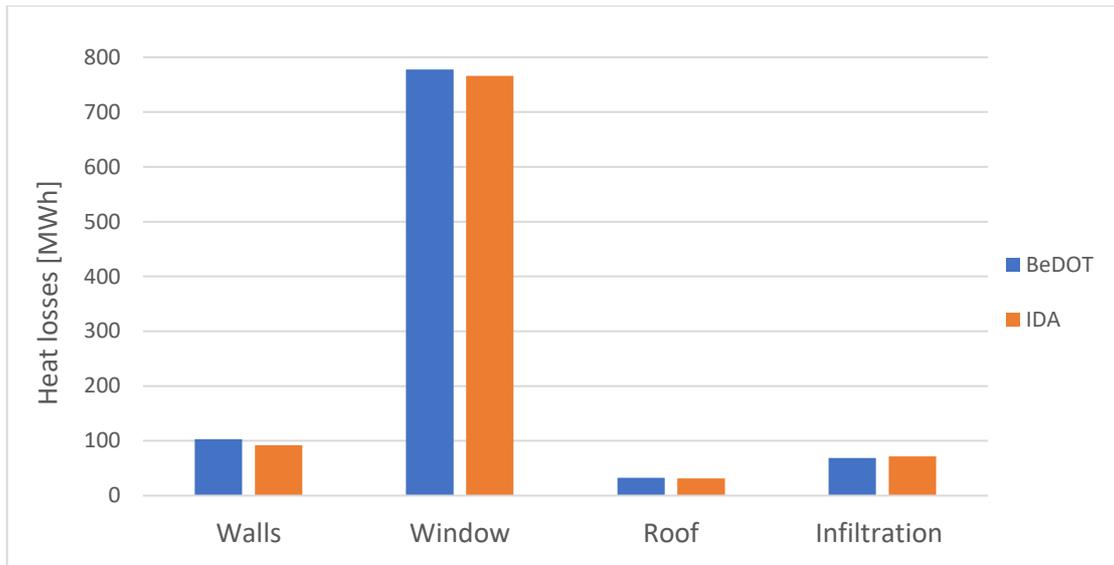


Figure 54: Case study A, heat losses through the building envelope.

Results from previous default cases were built to compare results in IDA vs BeDOT when these two had the same design and input data. However, IDA ICE includes parameters that BeDOT does not consider, such as reflectance and emissivity of walls. Now the intention is to evaluate how good is BeDOT when comparing real cases, so reflectance and emissivity are now considered in IDA. Even when considering this, the results from Figure 55 are within the margin of 15% error. Therefore, the conclusion is that disregarding these two parameters in BeDOT is acceptable in this project.

Another parameter that cannot be modelled so far in BeDOT are the windows' frames. In IDA, the frame of a window prevents part of the radiation from getting into the room. To compensate for this effect, the solar gains in BeDOT were reduced accordingly to the frames' ratio of the project. Since the frames' ratio were 10%, only 90% of the solar gains were considered in BeDOT.

According to Swedish Building Code (SBC) the total heat from equipment that reaches the room is 80 % of the total amount, so in BeDOT the load is multiplied by a factor of 0.8. Regarding human load, IDA considers both sensible and latent heat, while BeDOT only takes sensible heat. To have a good estimation, this was taken into account and the human load was extracted from IDA results.

Energy results from BeDOT and IDA ICE are presented in Figure 55. The largest absolute difference is room heating in the residential floors, where BeDOT is underestimating the demand by 1.58 kWh/m^2 .

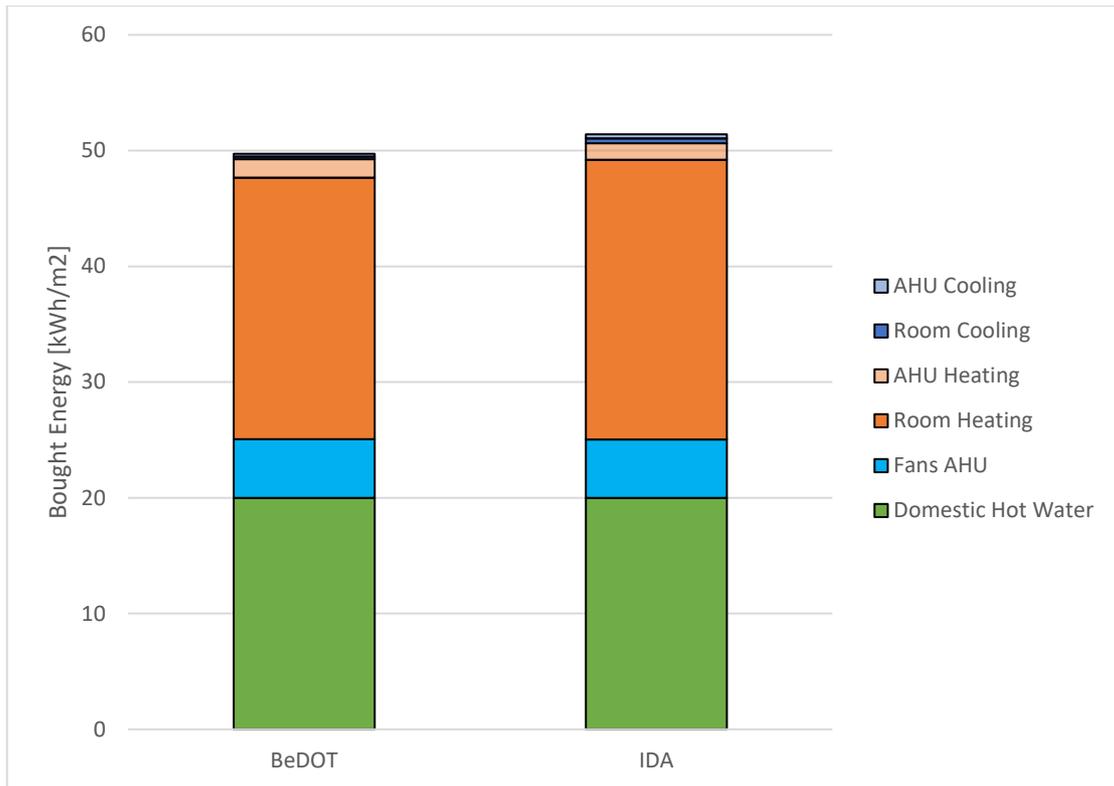


Figure 55: Case study A, energy results

In this project, the cooling demand is negligible if compared with the other demands. However, it needs to be pointed out that the absolute difference is 0.12 kWh/m^2 (32.6%) compared to IDA for the case of AHU Cooling. This difference may be due to a difference in the model of the AHU of IDA and BeDOT. From Figure 56 it is observed that the input data for the Air Cooler (AC) differs between the models. The temperature going in the AC is lower in BeDOT, which causes an underestimation of the cooling power.

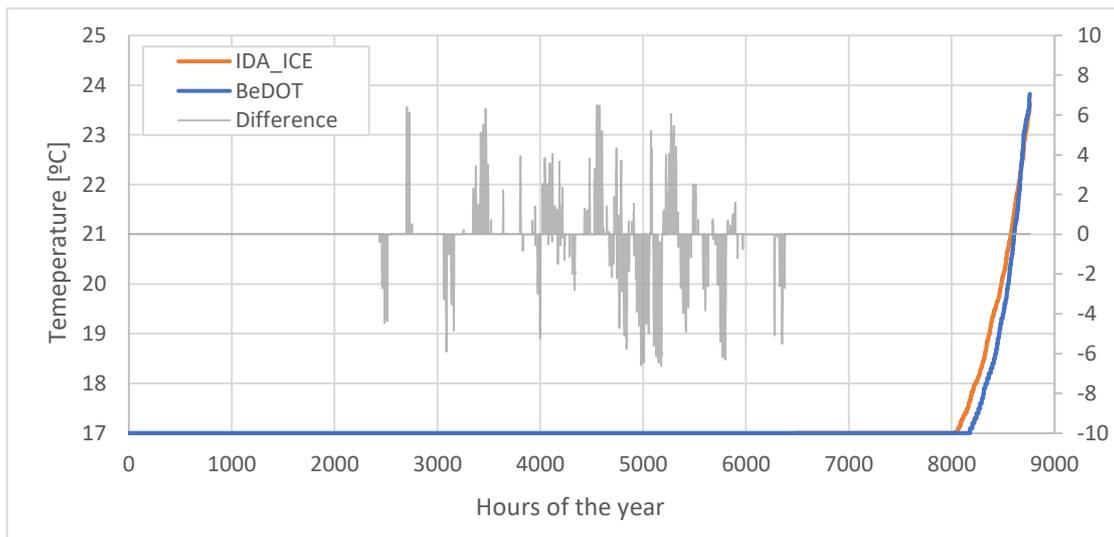


Figure 56: Case study A, inlet temperature to the air cooler.

A detailed list of input data is shown in APPENDIX A Input data Case Studies.

Case Study B

This case study is also based on a Bengt Dahlgren's project and it is an office building (see Figure 57). It only has an AHU composed of heat recovery and ideals air cooler and air heater. Also, a detailed list of input data is shown in APPENDIX A.

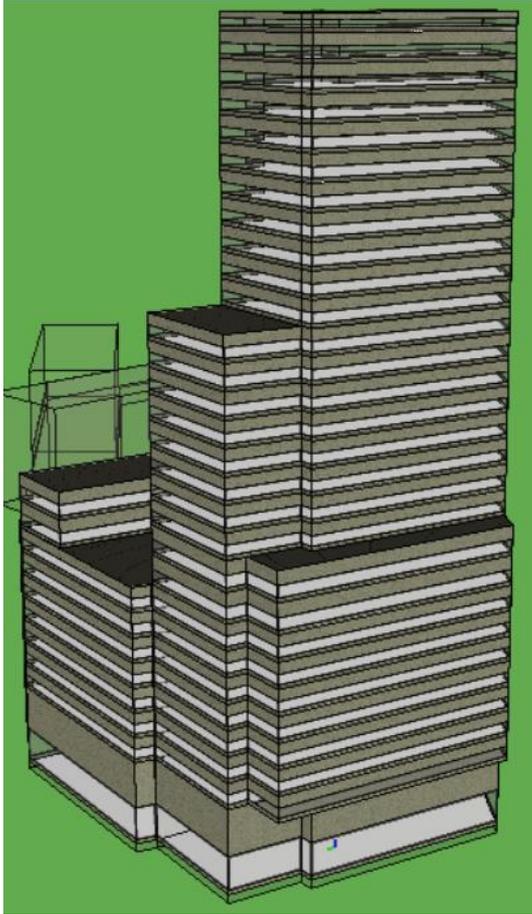


Figure 57: Case study B, geometry in IDA ICE.

For case study B, the building use is the same for all floors, so internal loads, occupancy and ventilation are the same throughout the building, as well as the envelope properties.

In the same way as case study A, all the input data were attached in the appropriate templates and then the simulation was run. Since the building does not require assigning data to different zones, the process of building the model was faster than case study A. However, from the simulation it was perceived that the room heating and cooling in BeDOT differed from the results in IDA ICE, while the AHU results were promising. Room heating was underestimated by 4 kWh/m^2 , while room cooling was overestimated by 4 kWh/m^2 .

The main input that affects cooling and heating at the same time is the solar radiation. From previous analysis, it is known that depending on the solar radiation model that is used, the input can vary. To check this, the solar gains of one of the rooms was exported from IDA and BeDOT. A more detailed approach would have been to get all the solar gains from both models, but in IDA ICE the user would need to take the gains one by one, which is time-consuming. The results for one zone are shown in Figure 58.

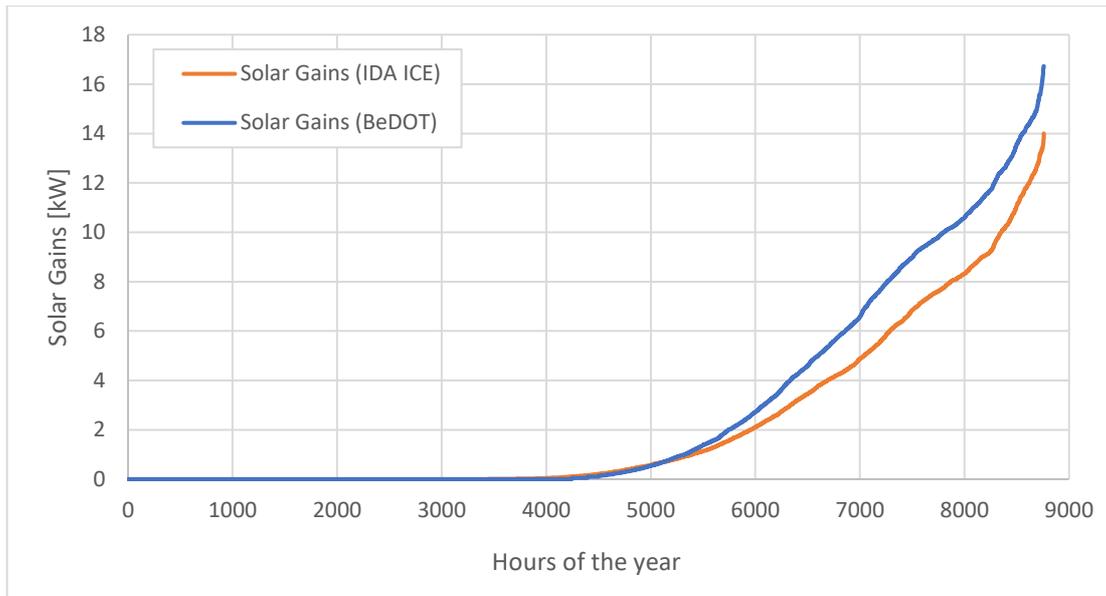


Figure 58: Case study B, solar gains analysis.

The solar gains in BeDOT were higher than IDA ICE, which explained why the cooling in BeDOT was higher. Implementing IDA ICE solar gains is out of the scope of this project since both methods to calculate solar gains are recognized and validated. However, to make the comparison, a reduction factor of 0.8, extracted from Figure 58, was applied in the solar gains of Daysim.

As in case study A, the human load is also extracted from IDA ICE and implemented in BeDOT. Transmissions through the building elements are plotted in Figure 59 and energy results are presented in Figure 60.

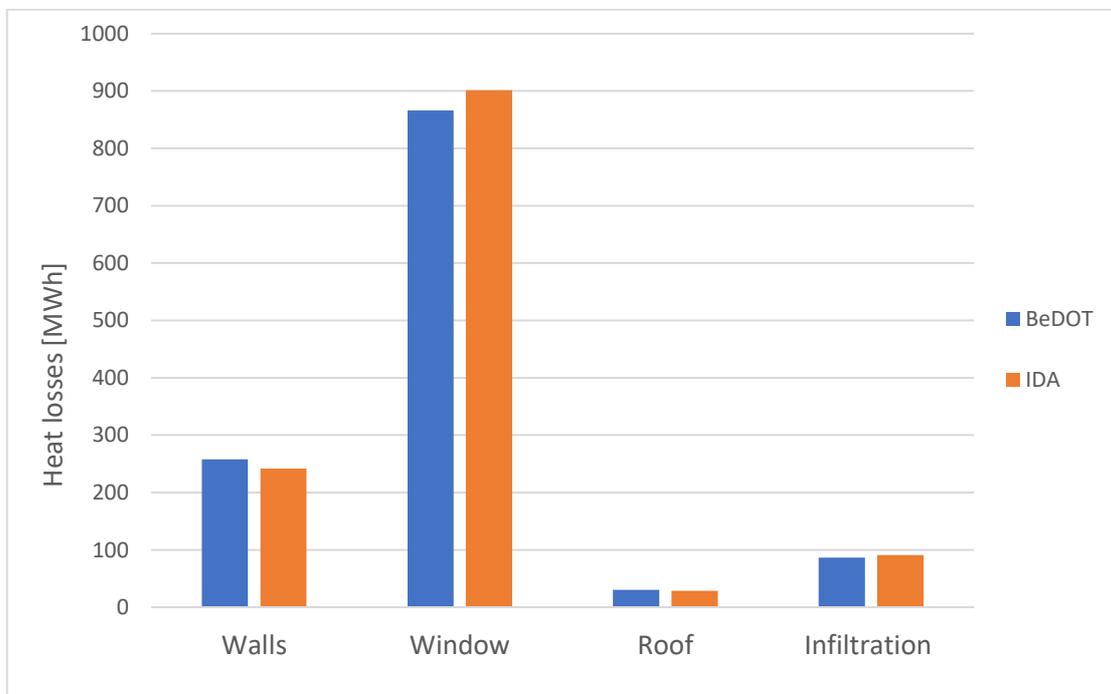


Figure 59: Case study B, transmission through building elements.

From the energy chart results are very promising. The bigger relative error is in the AHU Heating 0.29 kWh/m^2 (30.6%) but the demand is so little that is acceptable.

Then, the room heating has an error of 1.38 kWh/m^2 (6.8%) but the overall results are reasonable.

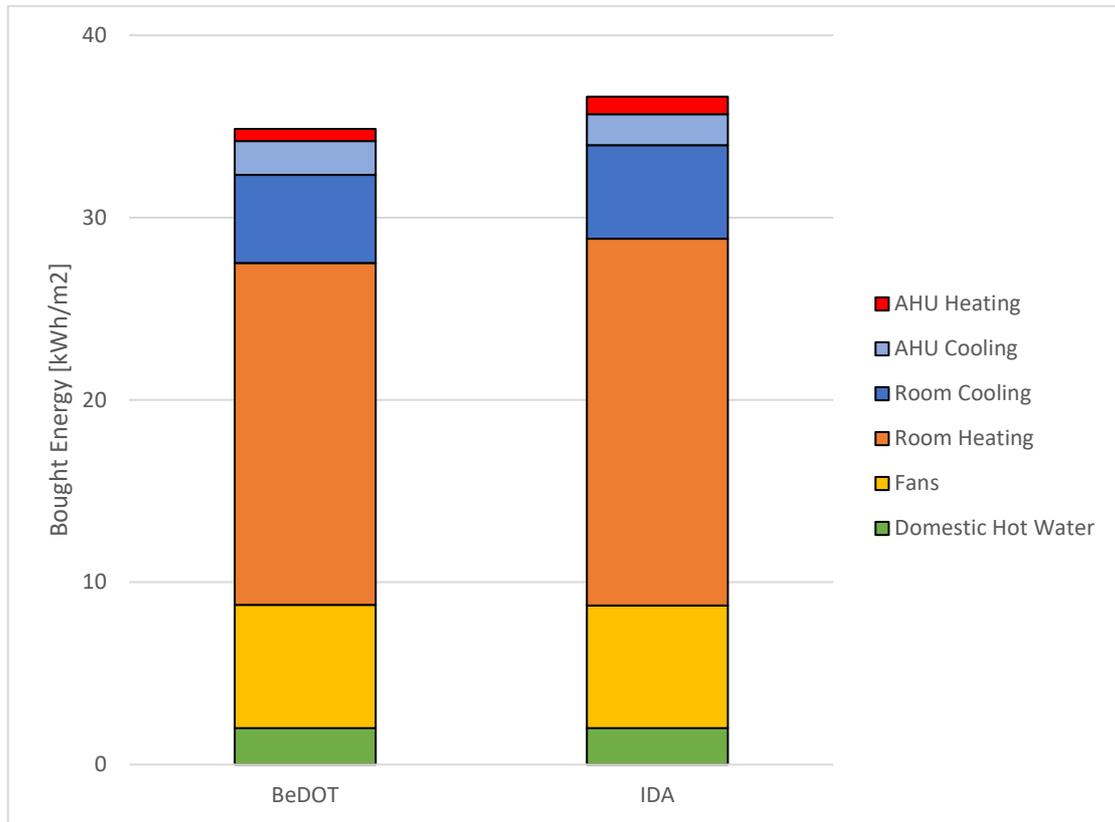


Figure 60: Case study B, energy results

9 Discussion

9.1 Strengths

The approach used in the current project differs from other programs in several ways. Firstly, the analysis is run over zones, which are independent from each other, and do not necessarily represent rooms or sectors. Although this was the main objective of energy estimation, zones might not even necessarily be related to buildings. It remains a user choice how to establish zones, which can be done either by subdividing a larger mass or stacking masses together.

The calculation components understand zones as a set of characteristics and parameters ultimately defined as numbers. This means that the program is not limited to having information provided from the lists of inputs. The user has total flexibility to add, replace, by-pass or modify data at any point of the process without causing a disruption in calculation, provided that the same format is kept along the way. Having the possibility to easily implement and edit a geometry by drawing in Rhino 3D canvas is surely a significant advantage. However, the flexibility in managing data in Grasshopper also allows changes to be made numerically, overwriting previous data and dismissing the need for altering geometry. By the same principle, the user can detach one or more components of BeDOT and replace them for others that he/she deem more fitting, or even add to the simulation data from other complementary models. To illustrate this, consider the section of solar radiation modelling. Right now, solar radiation is obtained using Daysim/Radiance components, but instead it could be brought from another software, or calculated using a different method. Another example of this would be using a separate model to calculate heat transfer to the ground and simply add it to BeDOT.

Among the advantages of the model, there is also a wide range of possibilities on post-processing and visualization. It has always been important to consider this and provide a natural design environment for designers. By having the calculation model integrated with geometry, results can be brought back into the 3D canvas, which makes it more tangible for the user to understand what they represent. This resource can also help identifying possible mistakes in design or in the implementation of the model.

In the post-processing stage, it is possible to group zones into a variety of categories, such as building unit, building sector, end use, realty, system boundaries, etc. Results can be analysed per category, therefore focusing not on the performance of individual zones, but on the project as a whole.

Additionally, BeDOT can be coupled to iterative methods for optimization, such as genetic algorithms, brute-force, neural networks, etc. This process is made easy in Grasshopper environment with the aid of plug-ins (i. e. Galapagos, Colibri, etc.).

Another important advantage of BeDOT is the time it consumes. For a familiarized user, the process of modelling a standard building from geometry to results is usually completed in a couple of hours. Regarding calculation time, the heat balance and energy performance calculation components run in a matter of minutes, varying depending on the number of zones. The calculation of solar radiation for each window is currently the most time-consuming part in the model, but as previously stated, this stage could be easily replaced by a more efficient method in the future.

9.2 Limitations

The program is subjected to limitations from two main sources. The first source of limitations is Grasshopper and its plug-ins, which supply the ready-made components. The second are limitations due to the numerical or physical models, applied during the implementation of energy calculations.

When talking about Grasshopper's limitations, it is important to notice that although they correspond to an inconvenience in BeDOT, it does not preclude the model's functioning. These limitations can be avoided if the user creates his/her own components, edits the ready-made ones or by-passes those components in any other way. Since these issues affect the starting part of the program, other solutions can (and should) be freely implemented, provided that their results are still kept in the right format. Some examples of this issues are:

- The impossibility of modelling curved building shapes, as it is not accepted by Grasshopper *zoning* component.
- Inclined surfaces are approximated into straight surfaces.

Another example of Grasshopper limitation would be the different glazing ratios on the façade. If the user wants to attach different ratios on different floors, it is necessary to multiply some honeybee components many times, one for each different glazing ratio. This is not a serious issue, but can compromise the cleanliness of the canvas when simulating multiple buildings, consequently being more confusing to the user.

Limitations regarding physical or numerical models are usually more complex to bypass. One important limitation of the model is that it follows a downstream flow of data, not being possible to return information upstream. This way, all parameters must be known before the calculation, or else have their values assumed beforehand by the user or the calculation itself. In other words, a parameter cannot be iterated between different modules. Iterations are possible only if fully implemented inside one single component, which has been done by using Python.

At times, the initial assumptions made in one step are far from the values later calculated in the following components, an inconsistency that also means a calculation error is established from the very beginning. It is still possible for the user to update the initially assumed values and run the simulation again. However, not only the procedure takes time, but also there is no guarantee of convergence.

BeDOT's resolution is limited to one hour, which is the smallest possible time-step. This resolution has been established based on the ISO's simplified hourly model, and is compatible with most weather files.

The amount of information handled by the program, mostly if many zones of different characteristics are implemented, can be very high. Because the program is based on zones, this means that the user has the possibility of modelling several buildings at once, or even detail one single building by dividing it into many zones. The combination of multiple zones, with characteristics varying hourly and per zone, result in a program that can be very demanding on computational power. Additionally, as it has been designed for supporting the import of pre-made input lists, the program has not been intensely developed for input lists creation. Although a method for list creation has been provided, it has limited user-friendliness and bounds the user to certain input formats, which might not be the ones preferred. Aiming to

reduce complexity and the amount of data, these input lists have been simplified in several points. One such example is the disregarding of the latent heat portion from human occupation loads, which is considered fully sensible heat.

Some heat transfer processes are still pending implementation in the program. One example is the modelling of ground heat transfer and thermal pillows. Accurately calculating transient ground temperature behaviour is usually a task of high complexity, as it requires time steps much larger from the ones used for air modelling. Moreover, the effect of solar radiation over opaque surfaces and heat transfer to the sky are both currently neglected. Additionally, thermal bridges and leakage are implemented, but lack a more detailed modelling. Right now, thermal bridges are considered as a percentage of the total U-value of the envelope, similarly as the infiltration. Besides, all U-values are considered constant and they are not affected by external conditions.

The thermal network method applied in heat balance calculations includes a few simplifications, which should be kept in mind. The model is based on nodal temperatures, resistances, applied loads and one thermal capacitance (see more in Chapter 3). Every node is connected, at least, to one heat source and to one heat loss to outdoor environment. The more these heat sources and losses cancel each other, the closer is the node to local balance and lower is the chance of error. The distribution of internal loads and solar radiation among air, surface and mass nodes is one of the main sources of error.

Regarding the air handling units, the program also has some limitations. Firstly, all components apart from the heat recovery unit are considered ideal coolers or heaters, with unlimited power, and the heat recovery system runs on any outdoor temperature, neglecting risks of condensation or frost. Secondly, heat gains from the operation of fans are so far not considered into the heat balance. Thirdly, the possibility of night-time natural ventilation has not yet been implemented, since it would result in inconsistencies when running alongside hygienic ventilation. This is another issue raised by the downstream programming flow. Also, this model does not contemplate air pressure difference so natural ventilation strategies cannot be simulated. The last limitation concerns the appearance of errors at specific and unusual cases, which have not been thoroughly explored.

It is important, as it is a mass model, that it is used in a consistent manner and that the user is aware of what the volume represents. Also, consider that the user does not know the inside conditions of the zones, or indoor details, such as rooms. Since zones do not necessarily correspond to rooms, indoor parameters (i.e. thermal comfort and daylight, etc) are not contemplated.

Finally, the model does not include daylight control of electrical lights or a more comprehensive shading control for windows, right now simplified as a reduction factor.

9.3 Further development

In the present moment, BeDOT lacks a detailed shading control. Although a shading factor can be applied, this one is constant for all the hours of the year. A more comprehensive control system, where the user can implement shadings according to irradiance, time or outside temperature needs to be designed.

It has been identified that the solar radiation mesh is one of the factors that affect computational time the most. Therefore, a better implementation on this method would be beneficial for the tool. For instance, right now if the user changes the size of the windows, the whole solar radiation analysis is run again. This should not be needed, since the mesh is described over the building's envelope regardless of façade elements.

The Air Handling Unit and the heat balance do not contemplate any power restriction. This flexibility may be needed at early stages designs since the user wants a quick estimation. However, a further implementation of power restriction may be required so BeDOT can be used in more detailed simulations. In addition, for the time being the air cooler and heater are considered ideal, but in the future, both could have efficiency values. Furthermore, the program could have a verification for frost risk in the heat recovery unit, which now operates the same way regardless of outside temperature. Along with these features, another possibility is to implement fan pressure curves. The possibility of having night ventilation cooling, should also be investigated.

At the moment, the way input data is handled in BeDOT may not be the best way regarding user-friendliness. Some parameters are still not straightforwardly visible to the user. Even though the proposed solution can be by-passed and replaced by imported databases, a better solution may be the user creating his/her own input data and setting it as default in BeDOT. By doing this, inputs can be provided in a wider variety of units, as preferred by the user, and pre-processed into the standard units and formats prior to calculations.

It was also identified in this project that softwares such as IDA ICE include smoothing factors with the purpose of avoiding abrupt changes. This could be seen in loads schedules, ventilation rates and control systems, which are then subjected to more progressive change. This progression may be included in future adjustments.

Regarding outputs, it is suggested that a correlation between energy and cost is established in the future, in order to estimate the project's real cost and energy savings. This also means identifying and considering, in the post-processing stage, the primary energy systems from which energy is provided to the project, and the energy cost related to each solution.

While programming the thermal balance of BeDOT, it was included an input to account for heat transfer through the ground. Right now, this input is a dummy null signal, so the ground is still considered adiabatic. However, a future compatibilization between BeDOT and other ground models can be developed and attached in the thermal balance, or post-processed after the heat balance calculation.

Finally, at the present moment BeDOT does not include a post-processing step to save and export input/output data. This point is of great interest, since the user may want to open project files later or store and share packages of inputs.

10 Conclusions

Decisions made during early design stages can greatly affect the final product's efficiency, as well as costs of production and maintenance, while the design effort is still low. Because of this, early-stage design demands analysis methods that provide fast and accurate feedback, while not requiring extensive input details.

The Building Early-stage Design Optimization Tool (BeDOT) can potentially help in this area, in great part because it is inserted in Rhinoceros, a platform that can be shared among engineers and architects, who can work in the same environment and directly share knowledge and information. Also, it is written in a common programming language (Python) that is kept open and simple to edit.

By using Rhinoceros, multiple options for visualization are available and both results and inputs can be visualized on a 3D model for better understanding. Additionally, it is not necessary to export geometries using e.g. BIM, the modelling of building volumes takes place in a CAD environment.

Buildings are described based on mass models instead of individual thermal zones. However, it preserves the possibility for increasing accuracy, if so needed further on in the construction process. BeDOT can model multiple buildings simultaneously and handle hundreds of thermal zones. These zones, together with technical systems, can be modelled fully independently of each other. Furthermore, zones and glazed surfaces are automatically generated, based on given parameters such as glass proportions and weather data. However, more detailed facades can still be described manually, since all information is editable in each analysis step. This makes it possible to avoid limitations we would otherwise have with pre-made components.

Regarding optimization, all analysis steps can be simulated independently of each other, which means that optimization can be done locally, for each component. Furthermore, BeDOT can be used along several optimization methods, including: manual optimization, brute force, Monte Carlos, Genetic Optimization and Artificial Intelligence.

When compared against IDA ICE, BeDOT's energy performance model displayed much faster modelling time. For experienced users, it takes from one to two hours, also depending on complexity of geometry, while it could take several hours or even days in IDA-ICE. The model's energy results stayed within a 15% accuracy range for most simulated cases. This is a valid expectation provided that both BeDOT and the software against which it is compared are using similar solar radiation models. Also, the solar models used in the project based on Daysim/Radiance provided an accurate daylight and sunlight analysis.

The energy model and BeDOT used, at this moment, an interface that handles input data by using templates. These templates can be easily replaced or reused in different simulations, depending on the user requirements.

BeDOT allows the implementation of a broad range of inputs and components, and allows the user to alter or by-pass parts of the simulation, if so desired. Thus, all analytical steps can be replaced if necessary, following the principle of modularity, and all system boundaries are open and easy to customize. This can apply to buildings, ownership, technical room systems, real estate systems and primary energy systems. Along with broad flexibility options come difficulties in handling and storing

data, as well as a reduction in user-friendliness. This balance should be pursued as BeDOT is developed further.

11 References

- Abel E., 2015. BUILDINGS AND ENERGY.
- Alduchov, O. A. & Eskridge, R. E., 1996. Improved Magnus Form Approximation of Saturation Vapor Pressure. *Journal of Applied Meteorology*.
- Anon., 2006. CEN/TC 89 Thermal performance of buildings and building components.
- ANSI/ASHRAE/IES, 2010. *Standard 90.1-2010 - Energy Standard for Buildings Except Low-Rise Residential Buildings*, s.l.: s.n.
- CEN, 2008. *SS-EN_ISO_13790_2008 - Energy performance of buildings - Calculation of energy use for space heating and cooling*. s.l.:s.n.
- Christoph Reinhart, n.d. *Advance Daylight Simulation Software*. [Online] Available at: <https://daysim.ning.com/>
- EU Commission & Council, 2014. A policy framework for climate and energy in the period from 2020 to 2030.
- EU Commission & Council, 2014. Energy Efficiency and its contribution to energy security and the 2030 Framework for climate and energy policy.
- EU Commission, 2015. *Energy efficiency in Buildings*. [Online] Available at: <https://ec.europa.eu/energy/en/topics/energy-efficiency/buildings>
- Jayathissa, P. et al., 2017. Optimising building net energy demand with dynamic BIPV shading. *Applied Energy*.
- Kircher, K. J. & Zhang, K. M., 2015. On the lumped capacitance approximation accuracy in RC network building models. *Energy and Buildings*.
- Li, B., 2017. *Use of Building Energy Simulation Software in Early-Stage of Design Process*, s.l.: s.n.
- Michalak, P., 2014. The simple hourly method of EN ISO 13790 standard in Matlab/Simulink: A comparative study for the climatic conditions of Poland. *Energy*.
- Nunes, A. I. F. & Oliveira Panão, M. J. N., 2013. Passive Cooling Load Ratio method. *Energy and Buildings*, Volume 64, pp. 209-217.
- Roulet, C.-A. & Anderson, B., 2006. CEN Standards for Implementing the European Directive on Energy Performance of Buildings. pp. 6-8.
- Sveby, 2012. *Brucarindata bostäder*, s.l.: s.n.
- Sveby, 2013. *Brucarindata kontor*, s.l.: s.n.
- U.S. Department of Energy's (DOE) Building Technologies Office (BTO), n.d. *Weather Data EnergyPlus*. [Online] Available at: <https://energyplus.net/weather>
- Vaisala Oyj, 2013. *Humidity conversion formulas*, Helsinki, Finland: s.n.
- Węglarz, A. & Narowski, P., 2011. The optimal thermal design of residential buildings using energy simulation and fuzzy sets theory.
- Yang, L., Yan, H. & Lam, J. C., 2014. *Thermal comfort and building energy consumption implications - A review*. s.l.:s.n.

APPENDIX A Input data Case Studies

A.1 Case Study A1

BUILDING USE		
1st Floor		
Loads	Internal gains	5.71 W/m ²
	Occupancy	20 m ² /p
	Schedule	8h to 17h Weekdays
HVAC System	CAV	1.2 l/sm ²
	Schedule	7h to 19h Weekdays
	Supply temperature	17 °C
	Heat Recovery	0.7
	Coil temperature	7.5 °C
	SFP	1.8 kWh/(m ³ /s)
	Set-point cooling	23 °C
	Set-point heating	21 °C
2nd Floor & 3rd Floor		
Loads	Internal gains	3.425 W/m ² (Always ON)
	Occupancy	34.48 m ² /p
	Schedule	18h to 8h all days
HVAC System	VAV	0.3 - 0.42 l/sm ²
	Schedule	18h to 19h all days
	Supply temperature	18 °C
	Heat Recovery	0.8
	Air cooler	OFF
	SFP	1.8 kWh/(m ³ /s)
	Set-point heating	25 °C
	Set-point cooling	21 °C
4th Floor onwards		
Loads	Internal gains	3.335 W/m ² (Always ON)
	Occupancy	35.4 m ² /p
	Schedule	18h to 8h all days
HVAC System	Same AHU as floors 2 & 3.	

GEOMETRY	
Infiltration	0.03 l/sm ² envelope
u-value walls	0.2 W/m ² K
u-value roof	0.11 W/m ² K
u-value windows	0.9 W/m ² K
g-value	0.3
g-shading	0.71 (Always drawn)
emissivity	0.9
reflectance	0.5
ground	adiabatic
Thermal bridges	0 W/m ² K

A.2 Case Study B1

BUILDING USE		
Loads	Internal gains	5.71 W/m ²
	Occupancy	20 m ² /p
	Schedule	8h to 17h Weekdays
HVAC System	CAV	1.2 l/sm ²
	Schedule	7h to 19h Weekdays
	Supply temperature	17 °C
	Heat Recovery	0.7
	Coil temperature	7.5 °C
	SFP	1.8 kWh/(m ³ /s)
	Set-point heating	21 °C
	Set-point cooling	23 °C

GEOMETRY	
Infiltration	0.03 l/sm ² envelope
u-value walls	0.2 W/m ² K
u-value roof	0.11 W/m ² K
u-value windows	0.9 W/m ² K
g-value	0.5
g-shading	0.71 (Always drawn)
emissivity	0.9
reflectance	0.5
ground	adiabatic
Thermal bridges	0 W/m ² K