

MASTER THESIS

Application of the AirflowNetwork model for selected test cases in EnergyPlus and evaluation of its input parameter sensitivity via parametric runs

Author (Student):	Mustafa Altattan
Date and Place of Birth:	14.07.1995, Syria
Institute:	Institut für Bauklimatik
Professor:	Prof.-Dr.-Ing. J. Grunewald
Supervisor:	Dr.-Ing. P. Freudenberg
Official Start Date:	17.01.2022
Official Submission Date:	17.06.2022

DECLARATION OF ORIGINALITY

I confirm that this assignment is my own work and that I have not sought or used inadmissible help of third parties to produce this work. I have fully referenced and used inverted commas for all text directly quoted from a source. Any indirect quotations have been duly marked as such.

This work has not yet been submitted to another examination institution – neither in Germany nor outside Germany – neither in the same nor in a similar way and has not yet been published.

Dresden, 17.06.2022

.....

CONTENTS

List of illustrations.....	IV
List of tables	V
List of abbreviations.....	VI
1 Introduction	1
1.1 Ventilation, natural ventilation, and air infiltration.....	1
1.1.1 Definition.....	1
1.1.2 Advantages and disadvantages of natural ventilation in buildings.....	2
1.1.3 Mechanism of natural ventilation	3
1.1.4 Simulation techniques for natural ventilation.....	4
1.2 The AirflowNetwork model in EnergyPlus	6
1.2.1 Overview	6
1.2.2 Application and restrictions.....	6
1.2.3 Input objects and calculation procedure	7
1.3 Simplified zonal model.....	16
2 Literature review	17
3 Methodology.....	21
3.1 Model description	21
3.2 Sensitivity analysis.....	22
3.3 Parametric run	24
3.4 Chosen parameters.....	25
4 Results and discussion.....	27
4.1 Parameters related to infiltration	27
4.2 Parameters related to airflow through openings.....	31

5	Conclusion and further work	33
5.1	Conclusion.....	33
5.2	Further work	34
A	Model settings.....	35
A.1	Simulation control.....	35
A.2	Node objects.....	35
A.3	Linkage objects	36
A.4	Component objects.....	38
A.4.1	Crack component	38
A.4.2	ELA component.....	39
A.4.3	Detailed opening component	39
A.4.4	Simple opening component	40
A.5	Wind pressure coefficient parameters.....	40
B	Program codes.....	43
B.1	Single-zone airflow network model.....	43
B.2	Parametric run program	44
B.3	Sensitivity analysis program	47
C	Workflow scheme.....	54
C.1	Workflow scheme of the sensitivity analysis	54
C.2	Workflow scheme of the parametric run	55
D	Sensitivity analysis results	56
6	Literature sources	58

LIST OF ILLUSTRATIONS

Figure 1: Airflow around a structure resulting in positive and negative pressures.....	3
Figure 2: A multi-zone airflow network model	7
Figure 3: AirflowNetwork model and EnergyPlus objects	8
Figure 4: Scheme for a three-way flow through a detailed opening (Engineering Reference)	12
Figure 5: Different flow patterns through a simple opening (Engineering Reference).....	14
Figure 6: Box model created using DesignBuilder	21
Figure 7: Plan view of the box model with AirflowNetwork components created in EnergyPlus	22
Figure 8: Example of a variable placeholder in the EnergyPlus IDF file.....	24
Figure 9: Workflow scheme of the sensitivity analysis.....	24
Figure 10: Workflow scheme of the parametric run.....	25
Figure 11: Tornado chart for the parameters related to infiltration ordered by their SI.....	27
Figure 12: Results of the parametric run on the air mass flow coefficient for the CC and SO using the distribution chosen in this study	28
Figure 13: Results of the parametric run on the air mass flow coefficient for the CC and SO using a wider range of distribution	29
Figure 14: Results of the parametric run on the air mass flow exponent for the CC and SO using the distribution chosen in this study	30
Figure 15: Results of the parametric run on the air mass flow exponent for the CC and SO using a wider range of distribution	30
Figure 16: Tornado chart for the parameters related to airflow through openings ordered by their SI.....	31
Figure 17: Modulation of the venting area according to the inside-outside temperature difference (Input / Output Reference)	32

LIST OF TABLES

Table 1: List of parameters used in the study of Belleri et al (Belleri, 2013).....	18
Table 2: List of parameters used in the study of Belleri et al. (Belleri, 2014).....	20
Table 3: Chosen parameters for the Sensitivity Analysis	26
Table 4: Sensitivity analysis results.....	57

LIST OF ABBREVIATIONS

CFD	Computational Fluid Dynamics
SHGC	Solar Heat Gain Coefficient
SRC	Standardized Regression Coefficient method
AIVC	Air Infiltration and Ventilation Center
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
ELA	Effective Leakage Area
BES	Building Energy Simulation
SA	Sensitivity Analysis
OAT	One-at-a-time method
SI	Sensitivity Index
CC	Crack component
SO	Simple opening component
DO	Detailed opening component
ACH	Air Change per Hour

1 INTRODUCTION

Natural ventilation plays an important role in buildings. It provides fresh air to occupants and is one of the largest factors influencing building energy costs. With that being said, proper simulation of natural ventilation is essential for an efficient design in complex buildings. Many tools are available for designers and one popular tool is the AirflowNetwork model in EnergyPlus. However, such tools are often complicated, over-parameterized and designers find it difficult to understand which parameters are important for the design process. This study aims to investigate the AirflowNetwork model and explore the sensitivity of its parameters using sensitivity analysis. This report is divided into 5 chapters. In Chapter 1, an introduction will be given on natural ventilation, its importance, the theory behind it and the state of art in estimating airflow in buildings. Moreover, the AirflowNetwork model in EnergyPlus will be explained including the calculation procedure and the main input objects. Chapter 2 will present a summary of similar studies and Chapter 3 will explain the model and the sensitivity analysis methodology used in this study. Finally, in Chapters 4 and 5 the results will be presented, and a conclusion will be drawn respectively.

1.1 VENTILATION, NATURAL VENTILATION, AND AIR INFILTRATION

1.1.1 Definition

Liddament defines ventilation as the process by which outdoor air is intentionally provided into a space which may be accomplished by either natural or mechanical means. Natural ventilation on the other hand, is the process of delivering air to and removing air from an indoor space without using mechanical systems and it is accomplished by the flow of external air to an indoor space as a consequence of pressure differences originating from natural forces. The mechanics of natural ventilation will be explained in more detail in Section 1.1.3. In addition to intentional ventilation, the uncontrolled flow of air into a space through adventitious or unintentional gaps and cracks in the building envelope is referred to as air infiltration (Liddament, 1996).

Many buildings around the world use natural ventilation as a source of cooling and supplying fresh air. In the past, the combination of uncontrolled air infiltration and window opening were the main methods of ventilation in buildings. But nowadays, buildings are more complex and ventilation requirements can be very demanding to save on cooling costs and prevent accidental heat losses. Therefore, modern systems have to provide much improved reliability and control and by careful design, natural ventilation can provide a good environment in fairly complex buildings (Liddament, 1996).

1.1.2 Advantages and disadvantages of natural ventilation in buildings

Adequate ventilation is a prominent benefactor to the health and comfort of building residents. Ventilation is required to supply fresh air and oxygen and to remove pollutants like carbon dioxide and odour. Ventilation is further used for cooling and to provide oxygen to combustion appliances (Liddament, 1996). As stated by Marschall, occupants of naturally ventilated buildings have more comfort ranges, highlighting the prospect of natural ventilation in increasing comfort while reducing energy demand (Marschall, 2020).

On the other hand, the main disadvantage of natural ventilation is the absence of control, in which unpredictable driving forces can result in durations of insufficient ventilation, followed by durations of over ventilation and extreme energy waste (Liddament, 1996). According to Prignon, air infiltration can be responsible for up to 30% of the heating demand in winter and it affects insulation thermal performances, hygrothermal performances, occupant comfort, ventilation system efficiency and acoustic insulation (Prignon, 2017). The quantity of energy required is dependent on the flow rate of ventilation and the amount of heating that is essential to attain thermal comfort (Liddament, 1996). In buildings constructed with very good thermal insulation, the portion of ventilation related energy loss can be much higher.

Good design can provide flow control but usually the occupant must modify ventilation openings to suit their comfort need. Even when it is difficult to control, natural ventilation is still depended upon to meet the requirement for fresh air in many types of building all over the world (Liddament, 1996).

1.1.3 Mechanism of natural ventilation

There are two types of natural ventilation in buildings. The first type is wind-driven ventilation, which originates from the different pressures formed by wind around a building and the openings on the perimeter which then enable airflow through the building. The second type is buoyancy-driven ventilation which occurs as an outcome of the buoyancy force that develops from temperature differences between the interior and exterior of the building. For any given layout of openings, the rate of natural ventilation differs to the degree of the driving forces of wind and indoor/outdoor temperature differences and how well-sealed the envelope of the building is (Liddament, 1996).

1.1.3.1 Wind-driven ventilation

Wind hitting a rectangular-shaped building generates a positive pressure on the windward facade and negative pressures on opposing and side facades. This forces air to enter openings and pass through the building from the high-pressure areas to the low-pressure areas (Liddament, 1996). A representation is shown in Figure 1. The pressure generated highly depends on the orientation of the building, the wind direction and speed and the obstacles surrounding the building. Therefore, a prediction is hard to make if the building is not rectangular in shape, and hence experimental methods like the wind tunnel test have to be used (Costola, 2009).

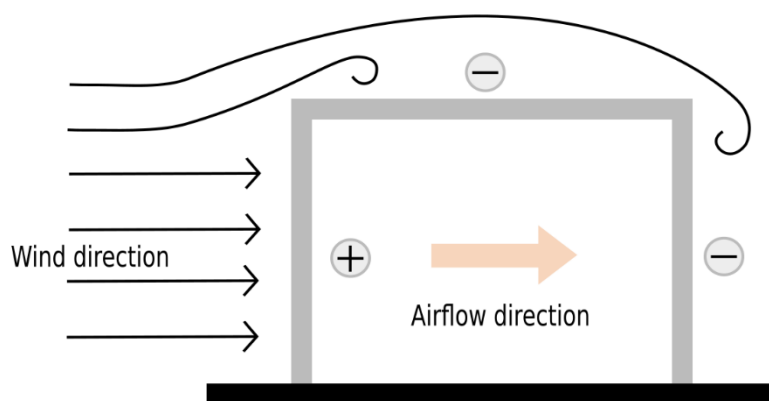


Figure 1: Airflow around a structure resulting in positive and negative pressures

1.1.3.2 Buoyancy-driven ventilation

Buoyancy-driven ventilation, also known as the stack effect, is produced as a result of differences in air temperature and air density, between the inside and outside of the building. This results in an imbalance in the pressure gradients of the internal and external air masses which results in a vertical pressure difference and since buildings are not completely closed the stack effect will cause air infiltration. When the inside air temperature is greater than the outside air temperature, air enters through openings in the lower part of the building and escapes through openings at a higher level. This is a common effect during the summer season. However, During the cooling season, the stack effect is reversed but is generally weaker due to lower temperature differences. In this case, the flow direction goes from the higher level to the lower level when the inside air temperature is higher than the outside air temperature (Liddament, 1996).

1.1.4 Simulation techniques for natural ventilation

Buildings are becoming more complex, and the performance tolerance for energy consumption is more demanding than ever. The simulation and analysis of natural ventilation are essential for such complex buildings. They help the designer examine the project before executing the final design. Several methods are available to support the designers in the decision-making process. The two most popular methods are Zonal Models (airflow network models) and Computational Fluid Dynamics (CFD) (Good, 2008).

1.1.4.1 Zonal Models

A zonal model (airflow network model) is used to describe a building with a series of zones interconnected by flow paths. Each zone usually designates a single room, while flow paths represent infiltration paths or openings like gaps, cracks, windows, or doors. The airflow in each zone is assumed to be well mixed and the bulk airflow through each opening is calculated using flow equations that relate pressure differences between zones and the exterior of the building. The designer can usually model the whole building as a single zone or divide the building into single connected zones (rooms). This way, the model can be described as a single-zone model or a multi-zone model (Liddament, 1996).

Zonal models calculate airflow through openings using outside weather conditions like wind and temperature in addition to the wind pressure coefficient data which will be discussed in Section 1.2.3.2. Moreover, operation schedules are also used to determine whether a window or a door is opened at a specific hour during the day. The airflow network model is also coupled with dynamic thermal simulations to resolve the thermal performance of the building over the whole year at an hourly time step (Good, 2008).

Some example software tools are IES Virtual Environment, ESP-r and the AirflowNetwork model in EnergyPlus. This study aims to investigate the AirflowNetwork model. Therefore, a detailed review will be given in Section 1.2.

1.1.4.2 CFD methods

Computational fluid dynamics (CFD) uses numerical methods with the help of computers to approximate the flow of fluids according to the conservation laws governing fluid motion. CFD could be used for thermal and ventilation analysis in buildings, and in the past years has been playing an increasingly important role in building design.

A CFD model, just like a Zonal model, requires the user to provide details at the boundaries so the airflow in and out of the zone can be estimated. However, CFD provides much more details compared to Zonal models about the airflow distribution within the naturally ventilated space inside the building. Airflow at different locations within the single zone can be calculated and temperature distribution in the zone can be estimated, giving information on areas that are too warm or too cold and enables detailed thermal comfort evaluations. This very high level of detail is very computationally demanding compared to a Zonal model. Therefore, only brief snapshots of time can be calculated (Good, 2008).

IES Virtual Environment and DesignBuilder are software examples that give the user the option to run a CFD analysis.

1.2 THE AIRFLOWNETWORK MODEL IN ENERGYPLUS

1.2.1 Overview

The AirflowNetwork model gives the ability to simulate multizone airflows driven by wind and buoyancy pressure with or without an HVAC system. The pressure and airflow model were established based on the AIRNET program to predict airflows through openings (Walton, 1989). This model is also used to model thermal conduction and air leakage losses for constant volume air distribution systems (Gu, 2007). However, the following description will only focus on the AirflowNetwork model in terms of natural ventilation. A detailed description of the input objects and the calculation procedure will be explained in Section 1.2.3. The majority of the information provided in the next sections, if otherwise referenced, is taken from the Engineering Reference, Chapter 13 - Alternative Modelling Processes, Section 13.1 - AirflowNetwork Model and the Input/Output Reference, Section 1.27 – Group AirflowNetwork of EnergyPlus.

1.2.2 Application and restrictions

Network methods are useful for the estimation of the airflow rate through separate openings. Therefore, this approach is considered one of the best approximations of real systems of natural ventilation and air infiltration. By estimating the rate and direction of flow through each flow path, it is achievable to assess almost every ventilation related parameter. According to (Liddament, 1996), usual applications of the network method include the computation of:

- air change rate as a function of climate and building air leakage,
- ventilation and air infiltration rate (mechanical and natural),
- the rate and direction of airflow through individual openings,
- the pattern of airflow between zones,
- internal room pressures,
- pollutant concentration,
- pollutant flow between zones and between the inside and outside of the building,
- back-draughting and cross-contamination risks.

However, the airflow network model comes also with limitations like:

- the need to quantify the driving forces and to account accurately for all openings in the structure of the building,
- the assumption that air in each zone is uniformly mixed,
- this approach, as with the preceding methods, does not provide information on airflow distribution within the individual zones.

1.2.3 Input objects and calculation procedure

The model is made up of a set of nodes, representing rooms, connected via linkages, representing airflow paths. Each linkage contains one or more component that describe the airflow given a pressure difference between nodes where the node variable is pressure, and the linkage variable is airflow rate. Figure 2 shows an illustration of how an airflow network model is set up. The solution is based on the balancing of the airflow going in and out of each zone. In other words, the sum of airflow going in and out of each zone must in theory equal to zero. However, an exact solution is not possible hence the model will compute the airflow balance iteratively until the flow balance at a certain tolerance is reached.

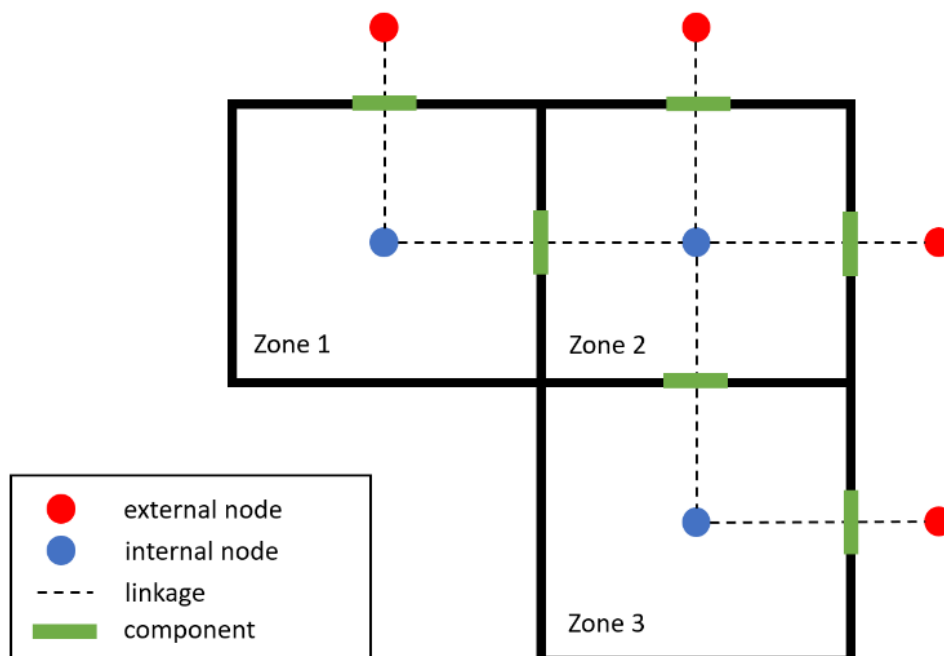


Figure 2: A multi-zone airflow network model

1.2.3.1 Input objects

The objects required to create an AirflowNetwork for natural ventilation can be divided into four main categories: 1) a simulation control object that specifies the main simulation settings like the convergences tolerance and the maximum number of iterations; 2) node objects that specify an internal or external node; 3) linkage objects that can be used to create a surface connecting two node objects; Finally, 4) component objects that can describe the airflow between two nodes and is connected via a linkage object (surface). A linkage object can connect only two nodes. However, it is possible to connect multiple component objects to a single linkage object to describe the airflow. For example, one component object can model a crack, and another component object can model an opening like a window, on a single wall. A list of the objects and their categories in addition to the objects created in EnergyPlus are shown in Figure 3.

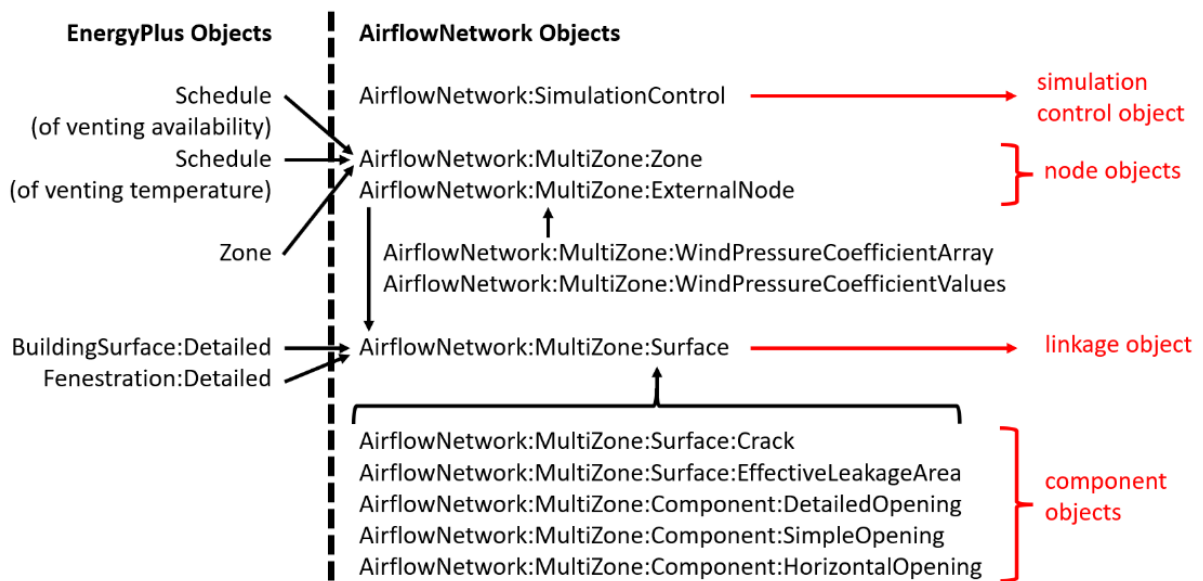


Figure 3: AirflowNetwork model and EnergyPlus objects

Additional AirflowNetwork objects that are required for setting up the model are the wind pressure coefficient array and values which will be described in next section. Objects from EnergyPlus are also required as an input for the AirflowNetwork model. An occupancy schedule to specify whether a zone is occupied at any given hour. In addition to a temperature set point schedule to set the conditions for a window or a door to be opened or closed. These EnergyPlus objects are used as input objects in the

AirflowNetwork:MultiZone:Zone object in the AirflowNetwork model. The **BuildingSurface:Detailed** and **Fenestration:Detailed** objects provide information on the dimensions and orientation of walls and openings and are used as input objects in the **AirflowNetwork:MultiZone:Surface** object. Finally, an external weather file with weather statistics is used as an input for temperature, wind speed and wind direction at an hourly interval.

1.2.3.2 Pressure calculation

For an airflow to occur, the pressure difference, ΔP , is first calculated and is given by Equation 1, where P_n and P_m are the total pressure difference between node n and m , P_s is pressure difference caused by the stack effect and P_w pressure difference caused by wind.

$$\Delta P = P_n - P_m + P_s + P_w \text{ [Pa]}$$

Equation 1

The nodes n and m represent two adjacent zones or a zone and an external node. In case of an internal zone, an initial pressure is first assumed for the initialization of the flow, and using the newton method, the true pressure can be found when the sum of airflow in and out of the zone is equal to zero. More on the solution method will be described in Section 1.2.3.4.

Calculation of the stack related pressure, P_s , is based on the temperature difference between the two air masses and the vertical spacing between openings. By using the Ideal Gas Law, this pressure difference is given by Equation 2, where ρ_0 is the air density at 273 K, g is the acceleration due to gravity, T_{ext} and T_{int} are the outdoor and inside air temperature respectively and h_1 and h_2 are the heights of opening 1 and 2.

$$P_s = \rho_0 g 273 (h_2 - h_1) \left[\frac{1}{T_{ext}} - \frac{1}{T_{int}} \right] \text{ [Pa]}$$

Equation 2

The pressure caused by wind, P_w , acting at any point on the façade of the building can be approximated by Equation 3, where C_p is the wind pressure coefficient, v is the local wind velocity at specified reference height and ρ is the air density.

$$P_w = \frac{\rho C_p v^2}{2} \text{ [Pa]}$$

Equation 3

The pressure coefficient, C_p , is a dimensionless empirical parameter, that can be derived from wind tunnel studies or using computational fluid dynamics techniques. It is not related to wind speed but changes with wind direction and the angle of the building surface. It is also affected by surrounding obstructions. According to Costola, 'the accurate evaluation of this parameter is one of the most difficult aspects of zonal modelling' (Costola, 2009). Several data sets are available for the assumption of wind pressure coefficients for a rectangular shaped building. The AirflowNetwork model gives the option to input the C_p data manually or the choice of average surface calculation. In the case of average surface calculation, for a low-rise building (the height is less than three times the width and length of the footprint), the C_p values are calculated from the equations of Swami and Chandra 1988. For a high-rise building (the height is more than three times the width and length of the footprint), the C_p data is provided from the ASHRAE Fundamentals Handbook. This option is provided in the **AirflowNetwork:SimulationControl** object.

1.2.3.3 Describing the airflow

After obtaining the initial pressure the airflow can then be calculated by balancing the airflow. The AirflowNetwork model allows the user to use different components to model the airflow through cracks or openings. **AirflowNetwork:MultiZone:Surface:Crack** and **AirflowNetwork:MultiZone:Surface:EffectiveLeakageArea** objects can be used to describe airflow through cracks and gaps and the **AirflowNetwork:MultiZone:Component:DetailedOpening** and **AirflowNetwork:MultiZone:Component:SimpleOpening** objects can be used to model a door or a window.

1.2.3.3.1 Airflow through cracks and gaps

Airflow through cracks is due to a pressure difference, ΔP , between both sides of the crack. The relation between the air mass flow rate and ΔP depends on the type of crack. The air mass flow rate is proportional to ΔP in cracks dominated by laminar flows and is proportional to $\sqrt{\Delta P}$ in cracks dominated by turbulent flows (Prignon, 2017).

The `AirflowNetwork:MultiZone:Surface:Crack` object specifies the properties of air flow through a crack and the associated measurement conditions. The power law form, shown in Equation 4, is used to describe airflow through the crack as a function of the pressure difference across the crack. Where \dot{m} is the air mass flow rate, C_Q is the air mass flow coefficient, C_T is the reference condition temperature correction factor and n is the air mass flow exponent.

$$\dot{m} = (\text{crack factor}) * C_T * C_Q (\Delta P)^n \text{ [kg/s]}$$

Equation 4

The air mass flow coefficient, C_Q , is the measurement of the efficiency of fluid flow through an opening and can be obtained via experimentation. The air mass flow exponent, n , defines the type of flow dominating the crack and can be a value between 0.5 and 1 for laminar and turbulent flow respectively. Finally, the *crack factor* is a user defined reduction factor that can take the values from 0 to 1.

Another way to describe the flow through a crack or a gap is to use the effective leakage area, *ELA*, which is defined by ASHRAE as “the area of an orifice [. . .] that would produce the same amount of leakage as the building envelope at the reference pressure” (ASHRAE, 2013). The `AirflowNetwork` object,

`AirflowNetwork:MultiZone:Surface:EffectiveLeakageArea` can be used, and Equation 5 applies for the relationship between pressure and airflow, where *ELA* is the effective leakage area, ρ is the air density, ΔP_r the reference pressure difference, C_d is the discharge coefficient and n is the air mass flow exponent.

$$\dot{m} = ELA * C_d \sqrt{2\rho} * (\Delta P_r)^{0.5-n} (\Delta P)^n \text{ [kg/s]}$$

Equation 5

1.2.3.3.2 Airflow through windows and doors

Windows and doors are modelled in the `AirflowNetwork` model as a detailed opening or a simple opening. The **`AirflowNetwork:MultiZone:Component:DetailedOpening`** object is used to calculate airflow through a door or window when they are closed or open. The input parameters are similar to those for **`AirflowNetwork:MultiZone:Surface:Crack`** object when the window or door is closed to model infiltration, but additional inputs are required to

$$\dot{m}_{z2,H} = C_d \theta \int_{z=0}^{z=H} \rho v(z) W dz$$

$$\dot{m}_{z1,z2} = C_d \theta \int_{z=z1}^{z=z2} \rho v(z) W dz$$

$$\dot{m}_{z2,H} = C_d \theta \int_{z=z2}^{z=H} \rho v(z) W dz$$

Equation 7

The detailed opening component allows the user to modulate how much a window, or a door is opened by linearly interpolating a user defined discharge coefficient, opening dimensions and the area reduction factor based on a set upper and lower limit for the inside/outside temperature difference with the precondition of the zone being occupied at a certain hour.

Another way to model windows or doors in the AirflowNetwork model is through the object **AirflowNetwork:MultiZone:Component:SimpleOpening**. Compared to the detailed opening object, which requires more inputs at different opening factors, this object needs comparatively less inputs. Therefore, it is called a simple opening. This opening also allows for the possibility of two-way flow due to temperature and resulting density differences. The air density however, for each node is assumed to be constant. The airflow is assumed to be a function of height and is given by Equation 8:

$$v(y) = C_d \sqrt{2 \frac{\Delta P(y)}{\rho}}$$

Equation 8

The neutral height Y , where the air velocity is zero, is calculated with Equation 9, where P_{0n} and P_{0m} are the pressure in zones n and m at the reference elevation and ρ_n and ρ_m are the air densities of zones n and m .

$$Y = \frac{P_{0n} - P_{0m}}{g(\rho_n - \rho_m)}$$

Equation 9

Figure 5 shows a simple opening with three associated flow patterns. When the neutral plane is within the opening, two-way flows occur. And when the neutral plane is above or below the opening, one-way flow occurs. Equation 10 describes the pressure and airflow relationship for two-way flow. Similar to the detailed opening component a reduction factor can be applied to modulate the amount of the opening width. However, the discharge coefficient and opening dimensions can not be modulated.

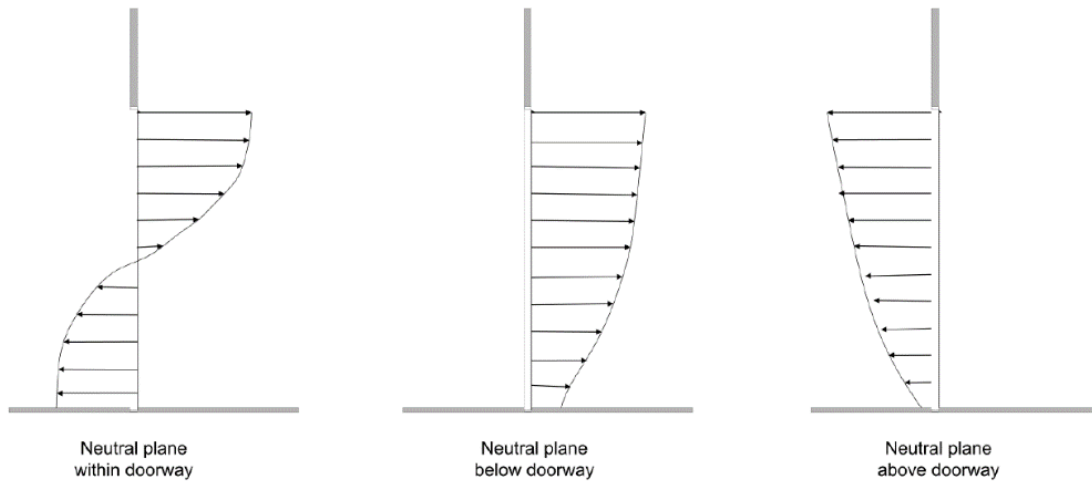


Figure 5: Different flow patterns through a simple opening (Engineering Reference)

$$\dot{m}_{0,Y} = C_d \theta \int_{y=0}^{y=Y} \rho v(y) W dy$$

$$\dot{m}_{Y,H} = C_d \theta \int_{y=Y}^{y=H} \rho v(y) W dy$$

Equation 10

1.2.3.4 Solving the mass flow balance

Newton's method is used to solve for node (zone) air pressures, and it requires an initial set of values. Conservation of air mass flow rate at each linkage provides the convergence criterion. When the sum of mass flow rates in all the linkages approaches zero within the convergence tolerance, the solution has converged. The **AirflowNetwork:SimulationControl** object is used to set the values for the convergence conditions.

Based on the relationship between airflow rate and pressure drop for each component, a system of equations for all components can be assembled together in an $n \times n$ square matrix,

where n is the number of nodes. Newton's method is used to iteratively solve for the air pressure at each node. A new estimated vector for all node pressures $\{P\}^*$, is computed from the current estimated vector of node pressures, $\{P\}$, by Equation 11:

$$\{P\}^* = \{P\} - \{C\}$$

Equation 11

where the correction vector, $\{C\}$, is computed by the matrix relationship as shown in Equation 12:

$$[J]\{C\} = \{B\}$$

Equation 12

$\{B\}$ is a column vector with each component given by Equation 13:

$$B_n = \sum_i \dot{m}_i$$

Equation 13

where n is the node number and i indicates all flow paths connecting node n to other nodes, and $[J]$ is the square Jacobian matrix whose elements are shown in Equation 14:

$$J_{n,m} = \sum_i \frac{\partial \dot{m}_i}{\partial P_m}$$

Equation 14

The convergence tolerance is used to check the sum of the mass flow rates by applying mass conservation. The convergence acceleration equation shown below is used to correct the node pressures to obtain a solution more rapidly. By assuming a constant ratio of correction values from one iteration to the next, Equation 15 applies, where r is the ratio of for the current iteration to its value for the previous iteration, C_n is correction value at the n -th node, P_n is the estimated pressure at the n -th node and P_n^* is the corrected pressure at the n -th node used in the next iteration.

$$P_n^* = P_n - \frac{C_n}{(1 - r)}$$

Equation 15

1.3 SIMPLIFIED ZONAL MODEL

To understand how a zonal model works, Liddament provides a good programming example in his book *A Guide to Energy Efficient Ventilation*, 1996. The program is made to represent a single-zone model with three flow components that describe the pressure and airflow relationship using the power law shown in Equation 4. The original algorithm in his book is written in BASIC but was rewritten in Python and is shown in Appendix B.1 (Liddament, 1996)

The program is provided as an example to help the reader understand how the internal pressure is first assumed for the initialization of the airflow and using any root finding method the true pressure can be found when the flow balance approaches zero. And hence, the airflow through each opening can be found. Furthermore, the reader can also see how the pressure difference due to wind and the stack effect is calculated.

The concept of the program is very similar to the `AirflowNetwork` model in `EnergyPlus`. However, the `AirflowNetwork` model is much more complex and can be used to model a multi-zone building. The pressure and airflow relationship can also be described in many ways which gives the user more options to model cracks and openings. Moreover, the program shown in Appendix B.1 will only solve the flow balance at a single time step while the `AirflowNetwork` model will resolve the airflow on an hourly time step over the whole year.

2 LITERATURE REVIEW

Sensitivity analysis plays an important role in building energy analysis. It can be used in determining the key variables influencing building thermal performance for energy simulation models and observational studies (Tian, 2013). According to Monari, sensitivity analysis has been increasingly applied to identify the most influential variable, quantify output uncertainty and to understand the relationships between inputs and outputs (Monari, 2016). In this chapter, recent attempts to study the most important factors influencing the ventilation rate in buildings using the AirflowNetwork model are reviewed.

In their study on the sensitivity analysis of natural ventilation design parameters for non-residential buildings (Belleri, 2013), Belleri et al used the Elementary Effects method on a naturally ventilated office building, under a range of climate conditions and ventilation strategies to identify the design parameters that have largest impact on natural ventilation performance.

The authors modelled a four-storey office building representing a typical European medium sized office using EnergyPlus with two different ventilation strategies: 1) night stack driven cross ventilation to reduce cooling needs; and 2) wind driven cross ventilation during the day to improve thermal comfort. Both strategies were modelled using the AirflowNetwork model and for both scenarios wind pressure coefficient values were used from the Air Infiltration and Ventilation Center (AIVC) dataset for a building surrounded by obstructions equivalent to half the height. Moreover, the ELA values were derived from the KlimaHaus building regulations.

The Elementary Effect method determines which parameters have negligible, linear, or nonlinear effect on the output using a relatively small number of simulations. A uniform distribution for all the parameters was defined and the full list of variables is shown in Table 1. In addition, the parametric analysis was done using three different weather files: Bolzano, Palermo, and San Francisco, to show the climatic dependency.

No.	Description	Min	Max
1	Exterior wall insulation thickness [m]	0.1	0.25
2	Exterior roof insulation thickness [m]	0.16	0.28
3	Exterior window U-value [W/m ² K]	0.5	1.7
4	Exterior window Solar Heat Gain Coefficient (SHGC)	0.3	0.9
5	Exterior wall density [kg/m ²]	230	430
6	Slab density [kg/m ²]	150	415
7	Overhang depth [m]	0.3	1.5
8	Inside reveal depth [m]	0	0.24
9	People fraction radiant	0.3	0.6
10	Lights fraction radiant	0.18	0.72
11	Effective Leakage Area	0.5	2
12	Window opening factor	0.4	1
13	Window discharge coefficient	0.3	0.9
14	Vent discharge coefficient	0.3	0.9
15	Number of people per Zone	7	16
16	Lighting Watts per Zone Floor Area [W/m ²]	5	20
17	Electric equipment Watts per Zone Floor Area [W/m ²]	5	20
18	Wind velocity profile: Exponent α , Boundary layer thickness δ [m]	$\alpha = 0.10$ $\delta = 210$	$\alpha = 0.33$ $\delta = 460$

Table 1: List of parameters used in the study of Belleri et al (Belleri, 2013)

The sensitivity analysis results showed that the parameters affecting solar and internal gains have a dominant effect on the natural ventilation performance in all climates; these include the solar heat gain coefficient (SHGC), number of people per zone, lighting, and equipment. In case of the weather in Palermo, the window opening factors and the discharge coefficient showed high impact on thermal comfort which was likely due to the lower wind speed and indoor/outdoor temperature difference. However, in case of high wind speeds, wind pressure coefficients have more influence compared to window opening factors and discharge coefficient.

Another study worth noting was conducted by Belleri (Belleri, 2014), on natural ventilation design. The author performed uncertainty and sensitivity analyses to determine which design parameters have most impact on the uncertainty associated with ventilation performance predictions. The model was created in EnergyPlus with an airflow network model for a small office building.

The standardized regression coefficient (SRC) method was used to point out the sensitivity in the parameters. The method assumes a linear regression between input and output and works best if inputs are pairwise statistically independent. SRC allows to quantify the effect of moving each variable away from its mean value by a fixed fraction of its standard deviation while all other variables are kept at their means. The parameters used in the sensitivity analysis are shown in Table 2.

No.	Description	Distribution
1	Wind-speed profile (exponent e boundary layer thickness [m])	Discrete [0.14–270, 0.20–333, 0.27–397, 0.33–460]
2	U of wall [W/m ² K]	N[0.43, 0.0307]
3	U of roof [W/m ² K]	N[0.237, 0.029]
4	U of floor [W/m ² K]	N[0.315, 0.0405]
5	U of window [W/m ² K]	N[3.646, 0.3785]

6	Solar heat gain coefficient (SHGC)	N[0.4, 0.05]
7	Maximum window opening factor	U[0.1–0.5]
8	Effective leakage area [cm ² /m ²]	Triang.dist[0.4, 5.1, 7.8]
9	Discharge coefficient	N[0.65, 0.05]
10	Wind-pressure coefficient array	Discrete [AIVC, TPU, ASHRAE, CpGen]
11	Lighting power [W/m ²]	Triang.dist[8, 12, 13]
12	Electric equipment power [W/m ²]	Triang.dist[8, 12, 18]

Table 2: List of parameters used in the study of Belleri et al. (Belleri, 2014)

The results showed that the wind-speed-profile parameter and the wind pressure coefficients had the most significant impact on air change rate. The next three parameter are all related to internal gains, and these are SHGC, lightning and electric equipment power density. Moreover, the discharge coefficients showed more influence than the individual window opening factor, but this is not the case for the combined effect of all windows. The author concluded that the proper prediction of windows operation plays a big role in the predicting of ventilation performance.

3 METHODOLOGY

In this chapter, the methodology used in this study will be explained. In Section 3.1 the model and main model settings will be explained. Section 3.2 and Section 3.3 will explain the methodology behind the sensitivity analysis and the parametric run. Finally, the parameters chosen for the variation are presented in Section 3.4.

3.1 MODEL DESCRIPTION

The model is based on a theoretical box model with four openings and four cracks on each side representing a single zone building as shown in Figure 6. The model was created in DesignBuilder 6 and the AirflowNetwork was further edited in EnergyPlus 8.9. To create the AirflowNetwork model in DesignBuilder, a calculated ventilation option is available. This option will automatically create a crack component on each surface and a detailed opening component for every window or door. DesignBuilder however, does not give the user access to the AirflowNetwork model. Therefore, the IDF Editor of EnergyPlus is used to further modify the model and add additional components.

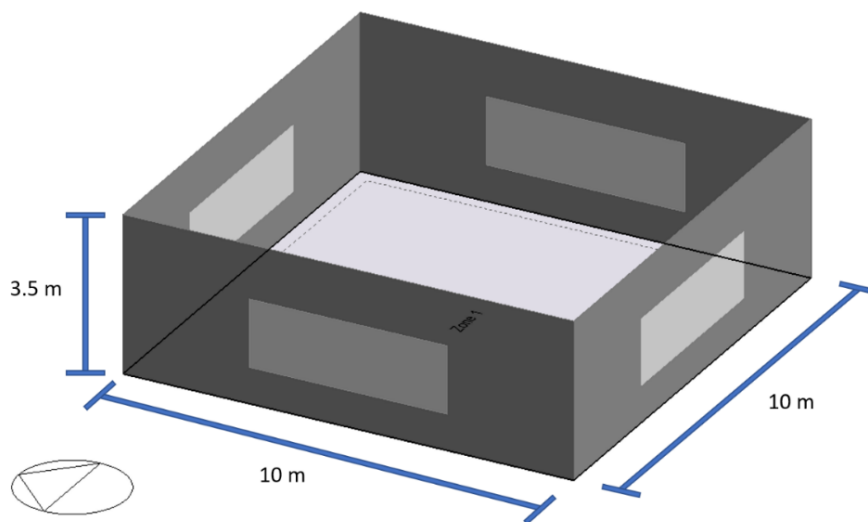


Figure 6: Box model created using DesignBuilder

The building is occupied from 07:00 to 19:00, Monday to Friday. The zone is also heated with a radiator with a heating setpoint temperature of 22 °C and a setback temperature of 12 °C.

The openings are set in DesignBuilder as a window with an opening-to-wall ratio of 20% and

a height of 1.5 m. The natural ventilation setpoint temperature is 24 °C. The occupancy and the ventilation setpoint temperature are important because they set the conditions for the detailed opening and simple opening components to be open or closed.

In the IDF Editor, the walls on the north and south sides are changed from a crack component and a detailed opening component to an ELA component and a simple opening component. The result is two opposing surfaces with the same type of components. An illustration is shown in Figure 7. A complete list of objects created in the AirflowNetwork model can be found in Appendix A.

The the air mass coefficient, discharge coefficient and air mass exponent data are all set based on default values in DesignBuilder or EnergyPlus. DesignBuilder will also automatically set the wind pressure coefficient type in the AirflowNetwork simulation control to 'input' and will input the C_p data from the AIVC dataset for a low-rise building surrounded by obstructions equivalent to half the height of the building. The C_p data is listed in Appendix A.5.

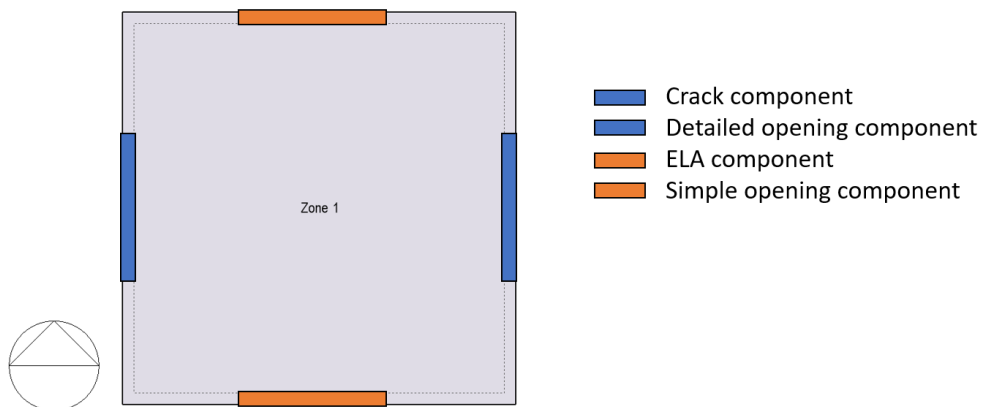


Figure 7: Plan view of the box model with AirflowNetwork components created in EnergyPlus

The weather file used in the simulation is an IWEC file for Berlin, entered as an EPW file in EnergyPlus.

3.2 SENSITIVITY ANALYSIS

Sensitivity Analysis (SA) is the study of how uncertainty in the input of a model is related to uncertainty in the output. It has become recently a very important tool for building energy

simulation (BES) programs to characterize the model uncertainty and help the designer with decision-making, determining the most important influential parameters and understanding the relationship between model inputs and outputs.

Sensitivity Analysis can be divided into two main groups: 1) One-factor-at-a-time methods (OAT); and 2) Global SA techniques. OAT methods belong to the class of Local-SA and work by fixing a base case and varying the target inputs one at a time to compare the change in the output to the average response of the base case. On the other hand, Global SA techniques explore the space of all input parameters and the interaction between them but at the cost of large computation (Tian, 2013).

In this study, the OAT method will be used to analyse the model. This method was chosen because it is very straightforward, easy to interpret and does not require computational effort. The number of simulations will be equal to $2N + 1$, where for each parameter variation there will be an upper and a lower boundary in addition to the average response of the base case. The disadvantages of this method are that the interaction between the parameters can not be studied and the sensitivity of the parameters could be different for a different spatial model.

There are different ways to quantify the sensitivity of the input parameters. The method used in this study is called the Sensitivity Index developed by Hoffman and Gardner (1983). The Sensitivity Index, SI , is calculated using Equation 16, where D_{min} and D_{max} are the output values for the lower and upper input parameters (Hamby, 1994).

$$SI = \frac{D_{max} - D_{min}}{D_{max}}$$

Equation 16

Figure 9 shows the workflow scheme of the sensitivity analysis. The following steps are taken: 1) The main model is prepared in DesignBuilder and exported as an IDF file; 2) Input files are prepared where, a number of IDF files, equal to the number of target variables, are copied from the main file and a variable placeholder is inserted in the place of the target variable using a text editor as shown in Figure 8; 3) Using a Python script, the placeholder in each input file is varied by a lower and an upper value; 4) The edited input files and the

weather file are passed to EnergyPlus for running the simulation; 5) The average Air Change per Hour (ACH) from the output files, together with the values from the input files are used to calculate the Sensitivity Index and the results are saved in a CSV file; Finally, 6) The CSV file is imported into Excel to create a tornado chart. The full sensitivity analysis program is listed in Appendix B.3 and the workflow scheme is also shown in Appendix C.1

```

placeholder ← AirflowNetwork:MultiZone:Surface:Crack,
                Block1:Zone1_Wall_2_0_0_Crack, !- Name
                var, !- Air Mass Flow Coefficient at Reference Conditions {kg/s}
                0.7, !- Air Mass Flow Exponent {dimensionless}
                Reference Crack Conditions; !- Reference Crack Conditions
  
```

Figure 8: Example of a variable placeholder in the EnergyPlus IDF file

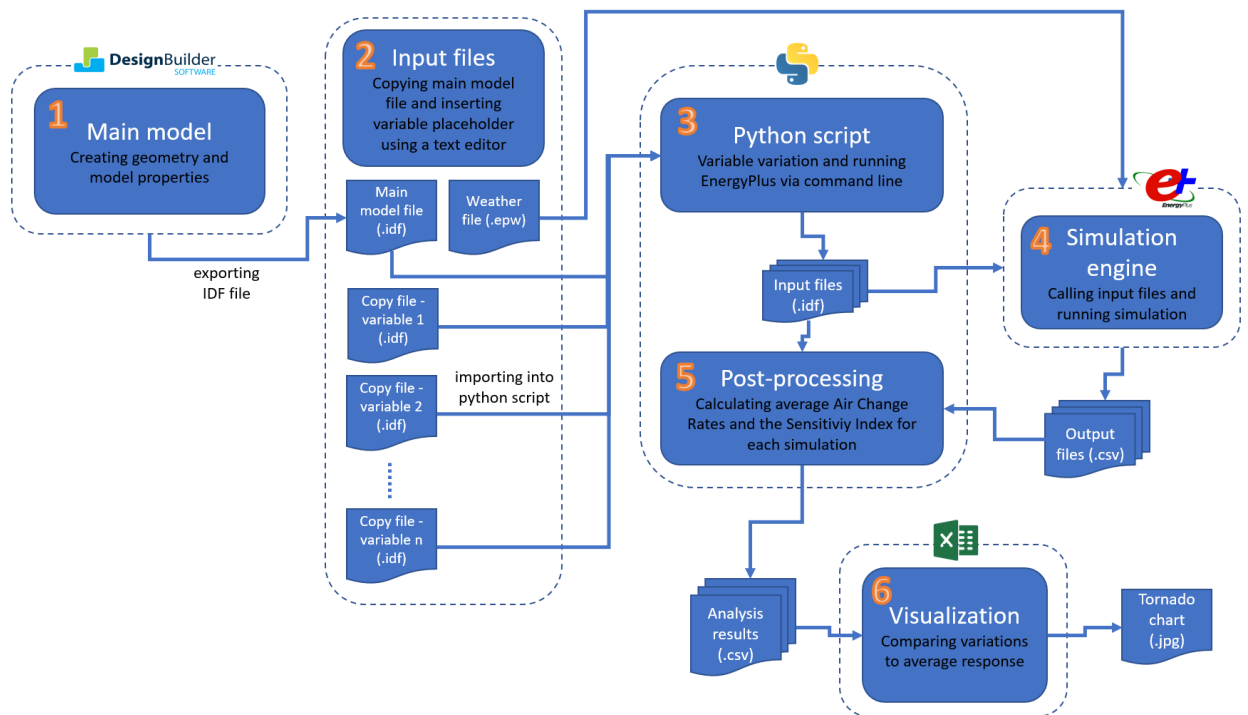


Figure 9: Workflow scheme of the sensitivity analysis

3.3 PARAMETRIC RUN

For further interpretation of the results, a parametric run was done to show the response of certain variables at a wider range of variations. Figure 10 shows the workflow scheme of the parametric run where the following steps are taken: 1) The model is created in DesignBuilder and exported as an IDF file; 2) A variable placeholder for the target variable is inserted in the IDF file using a text editor; 3) The model file is passed to a Python script for the variation of

the target variable; 4) The edited input files and the weather file are passed to EnergyPlus for running the simulation; 5) The average ACH from the output files together with the input parameters are used to create plots for further interpretation. The full parametric run program is listed in Appendix B.2 and the workflow scheme is also shown in Appendix C.2.

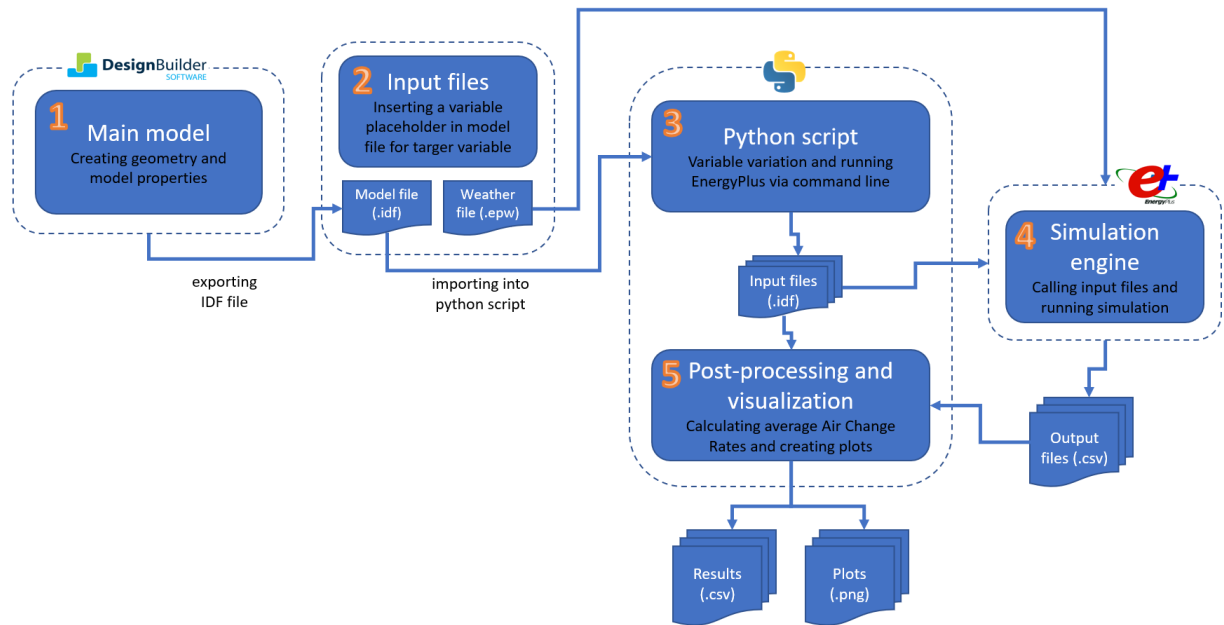


Figure 10: Workflow scheme of the parametric run

3.4 CHOSEN PARAMETERS

One of the most important steps in sensitivity analysis is determining the parameters and their distribution for the variation process. However, due to the time constraint and lack of data, default values or values recommended by the input/output reference of EnergyPlus were chosen for the base model.

For a proper sensitivity analysis, a probability distribution, representing real possibilities is usually used for the parametric variation. However, in this study, a simple variation of 20% was done on the parameters of the base model. This approach comes with a few disadvantages as it can be misleading and not represent real scenarios. Another disadvantage is that some parameters can exceed a value which is physically possible. For example, the maximum value for a crack factor is equal to 1. When the crack factor value is varied by 20%, we get an upper and a lower boundary of 1.2 and 0.8 respectively. In the case of the upper boundary, the value is no longer physically possible. Therefore, attention is paid

making sure that no values exceed the maximum or minimum limit for any variable in the Sensitivity Analysis.

A total of 16 parameters were chosen, belonging to the crack component (CC), Effective Leakage Area component (ELA), simple opening component (SO), and finally, detailed opening component (DO). The parameters are shown in Table 3.

Variable	Base value
CC: Air Mass Flow Coefficient at Reference Conditions [kg/s]	0.0028875
CC: Air Mass Flow Exponent [-]	0.7
CC: Crack Factor [-]	0.5
ELA: Effective Leakage Area [m ²]	0.01
ELA: Discharge Coefficient [-]	0.65
ELA: Air Mass Flow Exponent [-]	0.65
SO: Air Mass Flow Coefficient When Opening is Closed [kg/s-m]	0.00014
SO: Air Mass Flow Exponent When Opening is Closed [-]	0.65
SO: Discharge Coefficient [-]	0.65
SO: Window Opening Factor [-]	0.5
SO: Minimum Venting Open Factor [-]	0.5
DO: Air Mass Flow Coefficient When Opening is Closed [kg/s-m]	0.00014
DO: Air Mass Flow Exponent When Opening is Closed [-]	0.65
DO: Discharge Coefficient for Opening Factor 1 and 2 [-]	0.65
DO: Window Opening Factor [-]	0.5
DO: Minimum Venting Open Factor [-]	0.5

Table 3: Chosen parameters for the Sensitivity Analysis

According to the manual of DesignBuilder, the IEA Annexe 20 Air flow Through Large Openings in Buildings indicates that discharge coefficients reported in the literature vary between 0.3 and 0.8. using a discharge coefficient between 0.60 and 0.65 should provide sufficient accuracy. The air mass flow exponent of 0.65 is used as a default value by EnergyPlus. Finally, the air mass flow coefficient and effective leakage are the hardest values to estimate without available data. Therefore, the default values given by DesignBuilder were used for moderate airtightness in the building.

4 RESULTS AND DISCUSSION

A total of 33 simulations were done for the sensitivity analysis and the results are presented in a tornado chart. The parameters are ordered by their Sensitivity Index (SI) and the results are interpreted. A direct comparison between all parameters was not possible due to the big difference in ACH values between parameters related to infiltration and parameters related to airflow through windows and doors. Therefore, this section was divided into two subsections. One related to infiltration and the other related to airflow through openings. The full list of the results can be found in Appendix D.

4.1 PARAMETERS RELATED TO INFILTRATION

Infiltration in the AirflowNetwork model can be modelled using a crack component, ELA component, or a simple and detailed opening component when windows and doors are closed. The parameters related to infiltration are presented in Figure 11. The parameters are ordered according to their SI value from largest to smallest.

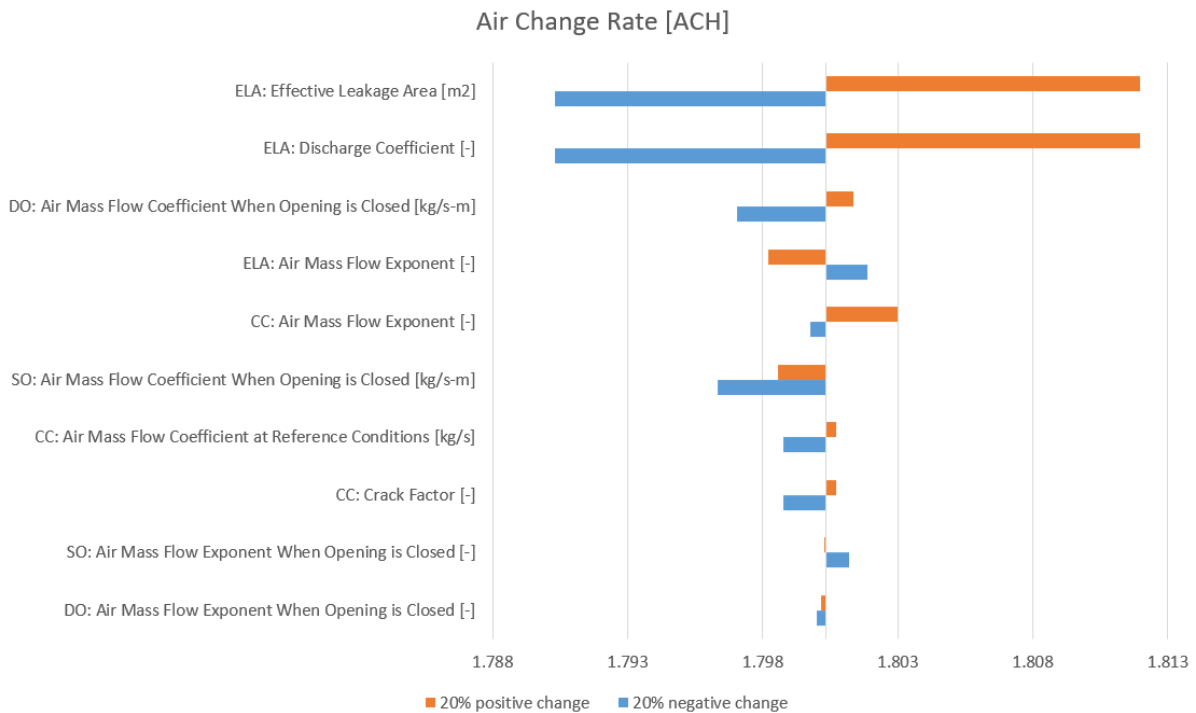


Figure 11: Tornado chart for the parameters related to infiltration ordered by their SI

From the chart, it can be seen that the parameters with the greatest influence on infiltration are the ELA: Effective Leakage Area and the ELA: Discharge Coefficient. Both parameters

have the same influence because when compared to Equation 5, both parameters are increased and reduced by the same percentage, which goes along with theory. The same effect can be seen with CC: Crack Factor and CC: Air Mass flow Coefficient when compared to Equation 4. However, the magnitude of the positive change compared to the negative change in the air mass flow coefficient and exponent of all components is hard to explain as there can be many factors that could cause this influence. A parametric run was done on some of these parameters and the results are shown in Figure 12. The result is a fluctuation in the ACH that makes the comparison between these parameters difficult. One reason could be that the rate by which the parameters are varied is too small for a notable difference. For this reason, the parametric run was done again but on a larger range to show whether this could be the issue. The range chosen for the air mass flow coefficient is 0.0001 to 0.01. Figure 13 shows the same parameters but with larger boundaries. The ACH makes more sense with such a distribution. Therefore, it is concluded that the chosen distribution in this study was too small to show effect.

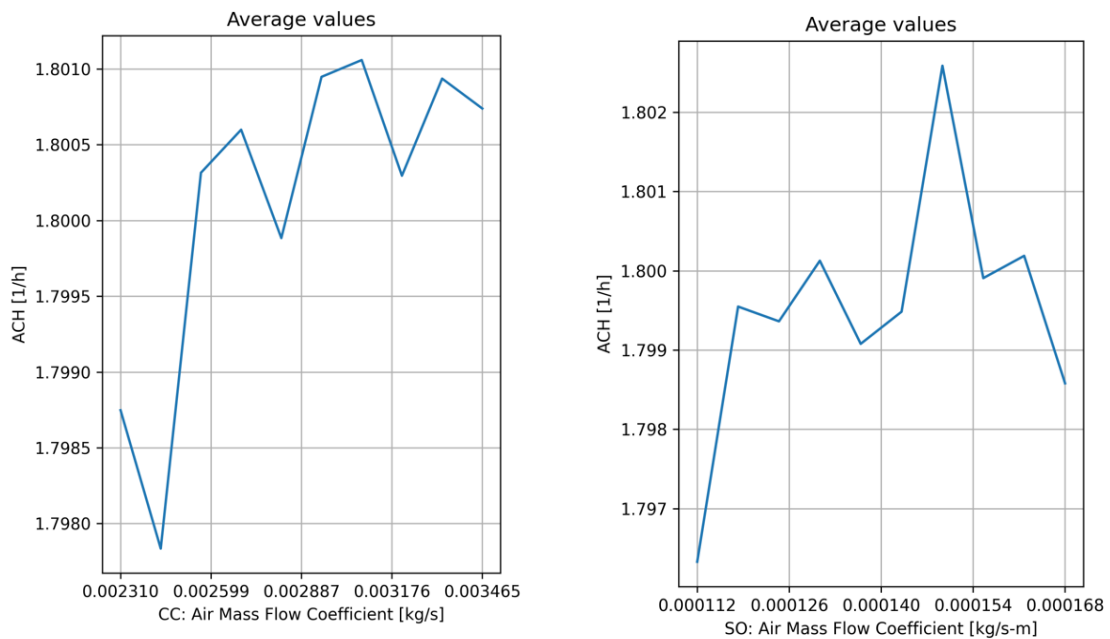


Figure 12: Results of the parametric run on the air mass flow coefficient for the CC and SO using the distribution chosen in this study

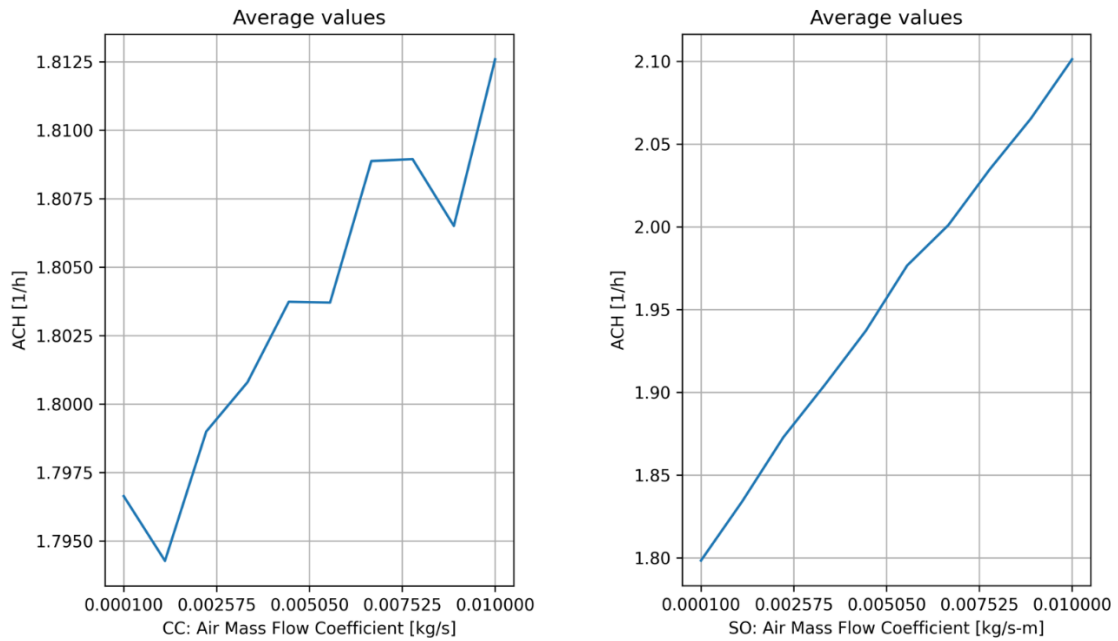


Figure 13: Results of the parametric run on the air mass flow coefficient for the CC and SO using a wider range of distribution

The parametric run was also done on the air mass flow exponent to show its influence on a higher range. However, this variable is limited to a range of 0.5 to 1.0 for a laminar and turbulent flow respectively. Figure 14 shows the results for the range chosen in this study and Figure 15 shows the results for higher ranges. It is not clear what causes the fluctuation in the results. One reason could be that there is some form of interaction between parameters that causes this kind of phenomenon. One approach to overcome this is to use a global sensitivity analysis technique instead of a local one.

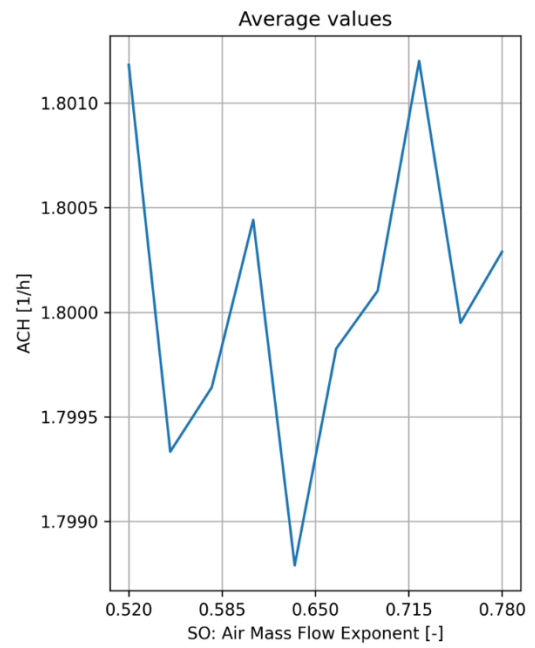
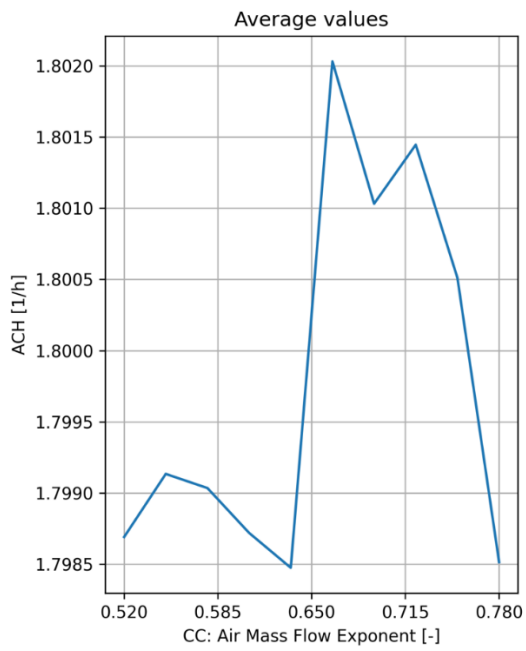


Figure 14: Results of the parametric run on the air mass flow exponent for the CC and SO using the distribution chosen in this study

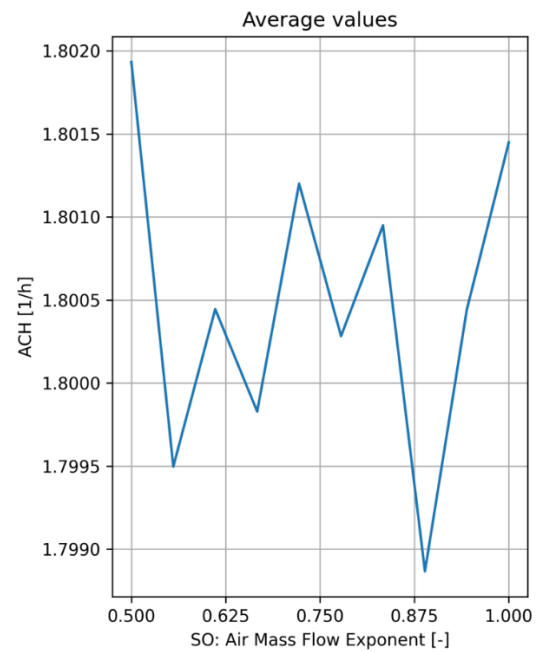
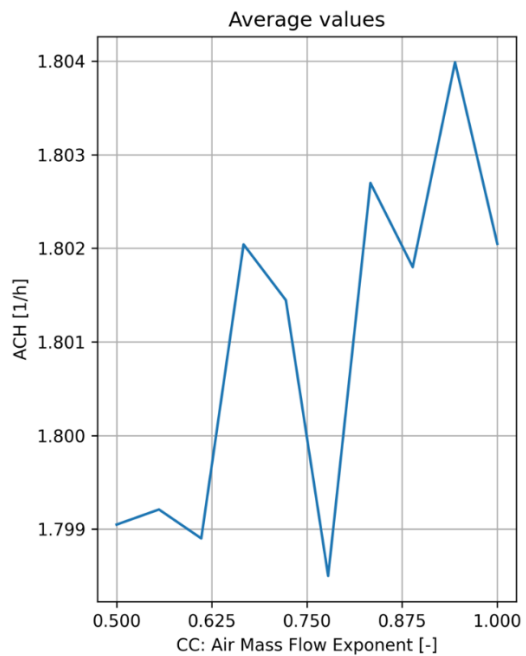


Figure 15: Results of the parametric run on the air mass flow exponent for the CC and SO using a wider range of distribution

4.2 PARAMETERS RELATED TO AIRFLOW THROUGH OPENINGS

Two opening components were tested for their sensitivities. The detailed opening and the simple opening component. The detailed opening component gives the designer more freedom with the modulation of the opening. Multiple discharge coefficients, opening widths and opening heights can be set to simulate the degree of an opening based on the temperature difference. The simple opening, on the other hand, allows the user to modulate the degree of opening, but only using a reduction factor. The parameters related to airflow through openings are presented in Figure 16. The parameters here are also sorted according to their SI value from highest to lowest.

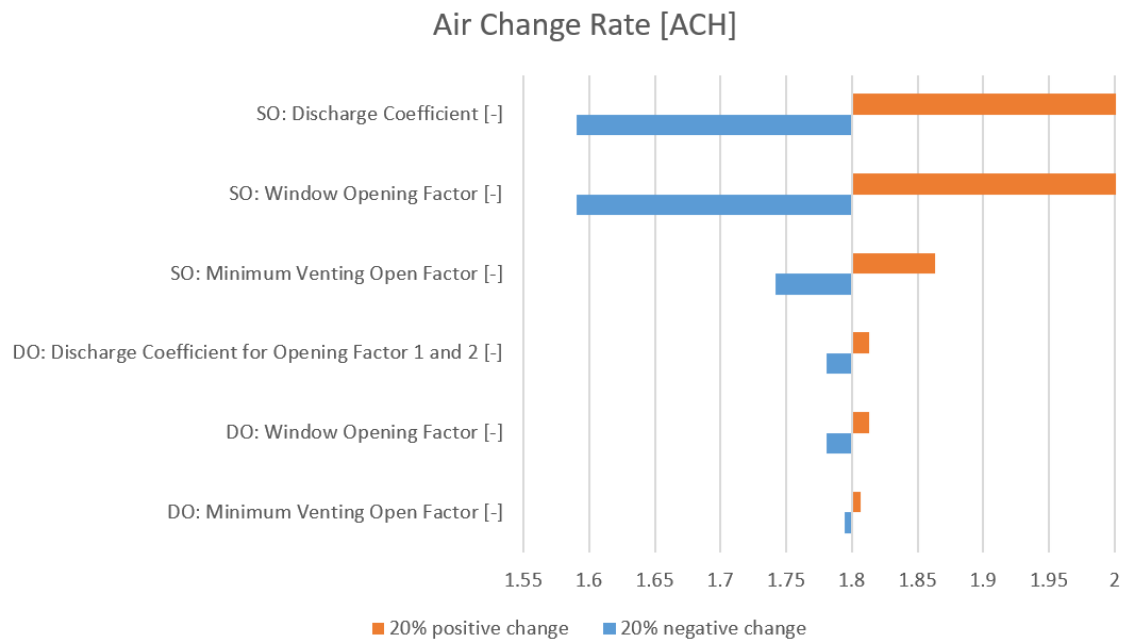


Figure 16: Tornado chart for the parameters related to airflow through openings ordered by their SI

It is found that the parameters of the simple opening component have a greater influence than the detailed opening component parameters. And this can be explained by the Modulation Venting Open Factor (modulation factor) for both simple opening and detailed opening components. For the simple opening component, the modulation factor is simply interpolated based on the temperature difference between the outside and the internal zone. However, for the detailed opening component, reducing or increasing the modulation factor also results in changing the discharge coefficient and the window opening area. Thus, the degree of opening is almost always smaller than that of the simple opening reducing the

average ACH in the zone. Figure 17 shows a graph from the Input/Output Reference of EnergyPlus illustrating how the modulation factor is interpolated based on the temperature difference.

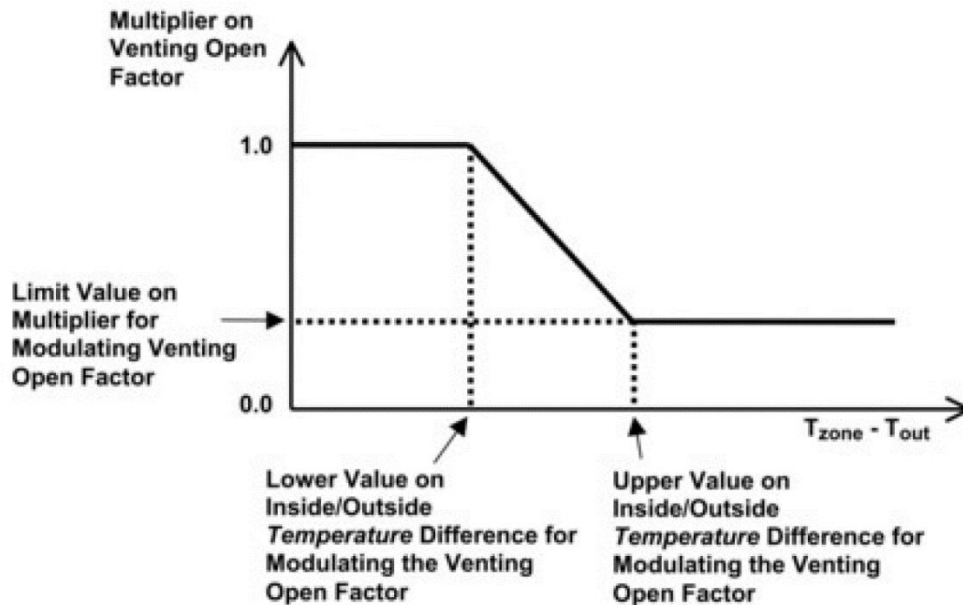


Figure 17: Modulation of the venting area according to the inside-outside temperature difference (Input / Output Reference)

The simple opening component could be over-estimating the ACH due to the lack of opening control compared to the detailed opening. On the other hand, the detailed opening component could be also underestimating the airflow. A study on an existing building including airtightness data could show the correct answer.

What can also be noted, is that a similar phenomenon previously discussed in Section 4.2 can also be seen here. The discharge coefficients and the window opening factors for both components have an equal magnitude. The simple opening and detailed opening components share similar equations (Equation 7 and Equation 10) to describe the airflow. The discharge coefficient, C_d , is directly multiplied by the reduction factor, θ (equal to the sum of the Window Opening Factor and the Modulation Venting Open Factor). Therefore, increasing or reducing both parameters with the same percentage results in similar outputs. This shows the disadvantage of varying all the parameters with the same percentage instead of assuming a probability distribution.

5 CONCLUSION AND FURTHER WORK

5.1 CONCLUSION

Natural ventilation plays an important role in buildings. It provides fresh air to occupants and is one of the biggest factors influencing building energy costs. Proper simulation of natural ventilation is essential for an efficient design in complex buildings. Many tools are available for designers and one popular tool is the AirflowNetwork model in EnergyPlus. However, such tools are often complicated, over-parameterized and designers find it difficult to understand which parameters are important for the design process. In this study, the AirflowNetwork model in EnergyPlus was reviewed. The model is made up of a set of nodes, representing rooms, connected via linkages, representing airflow paths. Each linkage contains one or more components that describe the airflow given a pressure difference between nodes.

To investigate the AirflowNetwork model, a box model was created, representing a single zone building with one crack and one opening on each surface. A One-at-a-time sensitivity analysis was done to investigate the sensitivity of the parameters and a parametric run was also done on single parameters to further investigate their effect. Finally, a direct comparison between the results and the theory was done and the main findings were:

- the ELA component showed the most influence compared to all the infiltration-related parameters,
- the simple opening component showed more influence than the detailed opening component and the reason could be the lower degree of simple opening modulation,
- the variation of the air mass flow coefficient of all components was too small to show a comparative result,
- the variation of the air mass flow exponent for all components did not match with theory and the reason is unclear,

5.2 FURTHER WORK

While this study gives a review of the theory of the AirflowNetwork model and its application. It still requires a lot of improvement for a more extensive review. The following is recommended for future work:

- the input parameters follow a probability distribution for a more realistic approach as this was one of the biggest disadvantages of this study,
- a realistic multi-zone model of an exciting building including airtightness data is also recommended for the verification and validation of results.
- a global sensitivity analysis technique could be applied to explore the interaction between parameters and provide quantitative results,
- the horizontal opening component (which was not reviewed in this study) could also be included in future studies,
- for the validation of the results, the analysis can be done using different weather conditions to explore the effect of average temperatures and wind speeds.

A MODEL SETTINGS

A.1 SIMULATION CONTROL

```
!- ===== ALL OBJECTS IN CLASS: AIRFLOWNETWORK:SIMULATIONCONTROL
=====
```

```
AirflowNetwork:SimulationControl,
  VENT,                                     !- Name
  MultizoneWithoutDistribution,             !- AirflowNetwork Control
  Input,                                    !- Wind Pressure Coefficient Type
  OpeningHeight, !- Height Selection for Local Wind Pressure Calculation
  LowRise,                                   !- Building Type
  1000,                                     !- Maximum Number of Iterations {dimensionless}
  ZeroNodePressures,                       !- Initialization Type
  0.0001, !- Relative Airflow Convergence Tolerance {dimensionless}
  0.000001, !- Absolute Airflow Convergence Tolerance {kg/s}
  -0.5, !- Convergence Acceleration Limit {dimensionless}
  0, !- Azimuth Angle of Long Axis of Building {deg}
  1; !- Ratio of Building Width Along Short Axis to Width Along Long Axis
```

A.2 NODE OBJECTS

```
!- ===== ALL OBJECTS IN CLASS: AIRFLOWNETWORK:MULTIZONE:ZONE
=====
```

```
AirflowNetwork:MultiZone:Zone,
  Block1:Zone1,                             !- Zone Name
  Temperature,                               !- Ventilation Control Mode
  20001, !- Ventilation Control Zone Temperature Setpoint Schedule Name
  , !- Minimum Venting Open Factor {dimensionless}
  , !- Indoor and Outdoor Temperature Difference
  Lower Limit For Maximum Venting Open Factor {deltaC}
  , !- Indoor and Outdoor Temperature Difference
  Upper Limit for Minimum Venting Open Factor {deltaC}
  , !- Indoor and Outdoor Enthalpy Difference
  Lower Limit For Maximum Venting Open Factor {deltaJ/kg}
  , !- Indoor and Outdoor Enthalpy Difference
  Upper Limit for Minimum Venting Open Factor {deltaJ/kg}
  Office_OpenOff_Occ; !- Venting Availability Schedule Name
```

```
!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:EXTERNALNODE =====
```

```
AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Roof_1_0_0, !- Name
  4.67, !- External Node Height {m}
  Block1:Zone1_Roof_1_0_0; !- Wind Pressure Coefficient Curve Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_2_0_0, !- Name
  2.33, !- External Node Height {m}
  Block1:Zone1_Wall_2_0_0; !- Wind Pressure Coefficient Curve Name
```

```

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_2_0_0_0_0_0_Win,  !- Name
  2.07,                                !- External Node Height {m}
  Block1:Zone1_Wall_2_0_0_0_0_0_Win;  !- Wind Pressure Coefficient Curve
Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_3_0_0,           !- Name
  2.33,                               !- External Node Height {m}
  Block1:Zone1_Wall_3_0_0;           !- Wind Pressure Coefficient Curve Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_3_0_0_0_0_0_Win, !- Name
  2.07,                                !- External Node Height {m}
  Block1:Zone1_Wall_3_0_0_0_0_0_Win;  !- Wind Pressure Coefficient Curve
Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_4_0_0,           !- Name
  2.33,                               !- External Node Height {m}
  Block1:Zone1_Wall_4_0_0;           !- Wind Pressure Coefficient Curve Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_4_0_0_0_0_0_Win, !- Name
  2.07,                                !- External Node Height {m}
  Block1:Zone1_Wall_4_0_0_0_0_0_Win;  !- Wind Pressure Coefficient Curve
Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_5_0_0,           !- Name
  2.33,                               !- External Node Height {m}
  Block1:Zone1_Wall_5_0_0;           !- Wind Pressure Coefficient Curve Name

AirflowNetwork:MultiZone:ExternalNode,
  Block1:Zone1_Wall_5_0_0_0_0_0_Win, !- Name
  2.07,                                !- External Node Height {m}
  Block1:Zone1_Wall_5_0_0_0_0_0_Win;  !- Wind Pressure Coefficient Curve
Name

```

A.3 LINKAGE OBJECTS

```

!- ===== ALL OBJECTS IN CLASS: AIRFLOWNETWORK:MULTIZONE:SURFACE
=====

```

```

AirflowNetwork:MultiZone:Surface,
  Block1:Zone1_Roof_1_0_0, !- Surface Name
  Block1:Zone1_Roof_1_0_0, !- Leakage Component Name
  Block1:Zone1_Roof_1_0_0, !- External Node Name
  1;      !- Window/Door Opening Factor, or Crack Factor {dimensionless}

AirflowNetwork:MultiZone:Surface,
  Block1:Zone1_Wall_2_0_0,           !- Surface Name
  Block1:Zone1_Wall_2_0_0_Crack,    !- Leakage Component Name
  Block1:Zone1_Wall_2_0_0,           !- External Node Name
  0.5;      !- Window/Door Opening Factor, or Crack Factor {dimensionless}

AirflowNetwork:MultiZone:Surface,
  Block1:Zone1_Wall_2_0_0_0_0_0_Win, !- Surface Name
  Block1:Zone1_Wall_2_0_0_0_0_0_Win_Detailed, !- Leakage Component Name

```

```

Block1:Zone1_Wall_2_0_0_0_0_0_Win,           !- External Node Name
0.5,      !- Window/Door Opening Factor, or Crack Factor {dimensionless}
Temperature,           !- Ventilation Control Mode
20001,           !- Ventilation Control Zone Temperature
Setpoint Schedule Name
0.5,           !- Minimum Venting Open Factor {dimensionless}
5,           !- Indoor and Outdoor Temperature Difference
Lower Limit For Maximum Venting Open Factor {deltaC}
10,           !- Indoor and Outdoor Temperature Difference
Upper Limit for Minimum Venting Open Factor {deltaC}
,           !- Indoor and Outdoor Enthalpy Difference
Lower Limit For Maximum Venting Open Factor {deltaJ/kg}
,           !- Indoor and Outdoor Enthalpy Difference
Upper Limit for Minimum Venting Open Factor {deltaJ/kg}
Office_OpenOff_Occ;           !- Venting Availability Schedule Name

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_3_0_0,           !- Surface Name
Block1:Zone1_Wall_3_0_0_ELA,       !- Leakage Component Name
Block1:Zone1_Wall_3_0_0,           !- External Node Name
1;           !- Window/Door Opening Factor, or Crack Factor {dimensionless}

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_3_0_0_0_0_0_Win,           !- Surface Name
Block1:Zone1_Wall_3_0_0_0_0_0_Win_Simple,    !- Leakage Component Name
Block1:Zone1_Wall_3_0_0_0_0_0_Win,           !- External Node Name
0.5,      !- Window/Door Opening Factor, or Crack Factor {dimensionless}
Temperature,           !- Ventilation Control Mode
20001,           !- Ventilation Control Zone Temperature
Setpoint Schedule Name
0.5,           !- Minimum Venting Open Factor {dimensionless}
5,           !- Indoor and Outdoor Temperature Difference
Lower Limit For Maximum Venting Open Factor {deltaC}
10,           !- Indoor and Outdoor Temperature Difference
Upper Limit for Minimum Venting Open Factor {deltaC}
,           !- Indoor and Outdoor Enthalpy Difference
Lower Limit For Maximum Venting Open Factor {deltaJ/kg}
,           !- Indoor and Outdoor Enthalpy Difference
Upper Limit for Minimum Venting Open Factor {deltaJ/kg}
Office_OpenOff_Occ;           !- Venting Availability Schedule Name

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_4_0_0,           !- Surface Name
Block1:Zone1_Wall_4_0_0_Crack,       !- Leakage Component Name
Block1:Zone1_Wall_4_0_0,           !- External Node Name
0.5;      !- Window/Door Opening Factor, or Crack Factor {dimensionless}

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_4_0_0_0_0_0_Win,           !- Surface Name
Block1:Zone1_Wall_4_0_0_0_0_0_Win_Detailed, !- Leakage Component Name
Block1:Zone1_Wall_4_0_0_0_0_0_Win,           !- External Node Name
0.5,      !- Window/Door Opening Factor, or Crack Factor {dimensionless}
Temperature,           !- Ventilation Control Mode
20001,           !- Ventilation Control Zone Temperature
Setpoint Schedule Name
0.5,           !- Minimum Venting Open Factor {dimensionless}
5,           !- Indoor and Outdoor Temperature Difference
Lower Limit For Maximum Venting Open Factor {deltaC}
10,           !- Indoor and Outdoor Temperature Difference
Upper Limit for Minimum Venting Open Factor {deltaC}
,           !- Indoor and Outdoor Enthalpy Difference

```

```

Lower Limit For Maximum Venting Open Factor {deltaJ/kg}
,                                     !- Indoor and Outdoor Enthalpy Difference
Upper Limit for Minimum Venting Open Factor {deltaJ/kg}
Office_OpenOff_Occ;                 !- Venting Availability Schedule Name

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_5_0_0,            !- Surface Name
Block1:Zone1_Wall_5_0_0_ELA,        !- Leakage Component Name
Block1:Zone1_Wall_5_0_0,            !- External Node Name
1;                                   !- Window/Door Opening Factor, or Crack Factor {dimensionless}

AirflowNetwork:MultiZone:Surface,
Block1:Zone1_Wall_5_0_0_0_0_0 Win,    !- Surface Name
Block1:Zone1_Wall_5_0_0_0_0_0 Win_Simple, !- Leakage Component Name
Block1:Zone1_Wall_5_0_0_0_0_0 Win,    !- External Node Name
0.5,                                  !- Window/Door Opening Factor, or Crack Factor {dimensionless}
Temperature,                          !- Ventilation Control Mode
20001,                                 !- Ventilation Control Zone Temperature
Setpoint Schedule Name
0.5,                                  !- Minimum Venting Open Factor {dimensionless}
5,                                     !- Indoor and Outdoor Temperature Difference
Lower Limit For Maximum Venting Open Factor {deltaC}
10,                                   !- Indoor and Outdoor Temperature Difference
Upper Limit for Minimum Venting Open Factor {deltaC}
,                                     !- Indoor and Outdoor Enthalpy Difference
Lower Limit For Maximum Venting Open Factor {deltaJ/kg}
,                                     !- Indoor and Outdoor Enthalpy Difference
Upper Limit for Minimum Venting Open Factor {deltaJ/kg}
Office_OpenOff_Occ;                 !- Venting Availability Schedule Name

```

A.4 COMPONENT OBJECTS

A.4.1 Crack component

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:SURFACE:CRACK =====

AirflowNetwork:MultiZone:Surface:Crack,
DefaultCrack,                        !- Name
0.000000000001,                      !- Air Mass Flow Coefficient at Reference Conditions
{kg/s}
0.66,                                 !- Air Mass Flow Exponent {dimensionless}
Reference Crack Conditions;            !- Reference Crack Conditions

AirflowNetwork:MultiZone:Surface:Crack,
Block1:Zone1_Roof_1_0_0,              !- Name
0.01,                                  !- Air Mass Flow Coefficient at Reference Conditions
{kg/s}
0.7,                                   !- Air Mass Flow Exponent {dimensionless}
Reference Crack Conditions;            !- Reference Crack Conditions

AirflowNetwork:MultiZone:Surface:Crack,
Block1:Zone1_Wall_2_0_0_Crack,        !- Name
0.0028875,                            !- Air Mass Flow Coefficient at Reference Conditions
{kg/s}
0.7,                                   !- Air Mass Flow Exponent {dimensionless}
Reference Crack Conditions;            !- Reference Crack Conditions

```

```

AirflowNetwork:MultiZone:Surface:Crack,
  Block1:Zone1_Wall_4_0_0_Crack,                               !- Name
  0.0028875,                                                  !- Air Mass Flow Coefficient at Reference Conditions
{kg/s}
  0.7,                                                         !- Air Mass Flow Exponent {dimensionless}
  Reference Crack Conditions;                                  !- Reference Crack Conditions

```

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:REFERENCECRACKCONDITIONS =====

```

```

AirflowNetwork:MultiZone:ReferenceCrackConditions,
  Reference Crack Conditions, !- Name
  20,                         !- Reference Temperature {C}
  101320,                     !- Reference Barometric Pressure {Pa}
  0.005;                     !- Reference Humidity Ratio {kgWater/kgDryAir}

```

A.4.2 ELA component

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:SURFACE:EFFECTIVELEAKAGEAREA =====

```

```

AirflowNetwork:MultiZone:Surface:EffectiveLeakageArea,
  Block1:Zone1_Wall_3_0_0_ELA,                               !- Name
  0.01,                                                       !- Effective Leakage Area {m2}
  1,                                                           !- Discharge Coefficient {dimensionless}
  4,                                                           !- Reference Pressure Difference {Pa}
  0.65;                                                       !- Air Mass Flow Exponent {dimensionless}

```

```

AirflowNetwork:MultiZone:Surface:EffectiveLeakageArea,
  Block1:Zone1_Wall_5_0_0_ELA,                               !- Name
  0.01,                                                       !- Effective Leakage Area {m2}
  1,                                                           !- Discharge Coefficient {dimensionless}
  4,                                                           !- Reference Pressure Difference {Pa}
  0.65;                                                       !- Air Mass Flow Exponent {dimensionless}

```

A.4.3 Detailed opening component

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:COMPONENT:DETAILEDOPENING =====

```

```

AirflowNetwork:MultiZone:Component:DetailedOpening,
  Block1:Zone1_Wall_2_0_0_0_0_0_Win_Detailed,               !- Name
  0.00014,           !- Air Mass Flow Coefficient When Opening is Closed {kg/s-m}
  0.65,             !- Air Mass Flow Exponent When Opening is Closed {dimensionless}
  NonPivoted,      !- Type of Rectangular Large Vertical Opening (LVO)
  0,                !- Extra Crack Length or Height of Pivoting Axis {m}
  2,                !- Number of Sets of Opening Factor Data
  0,                !- Opening Factor 1 {dimensionless}
  0.65,            !- Discharge Coefficient for Opening Factor 1 {dimensionless}
  0,                !- Width Factor for Opening Factor 1 {dimensionless}
  0.05,            !- Height Factor for Opening Factor 1 {dimensionless}
  0.95,            !- Start Height Factor for Opening Factor 1 {dimensionless}

```

```

1,                                     !- Opening Factor 2 {dimensionless}
0.65,    !- Discharge Coefficient for Opening Factor 2 {dimensionless}
1,                                     !- Width Factor for Opening Factor 2 {dimensionless}
0.05,    !- Height Factor for Opening Factor 2 {dimensionless}
0.95,    !- Start Height Factor for Opening Factor 2 {dimensionless}

```

```

AirflowNetwork:MultiZone:Component:DetailedOpening,
Block1:Zone1_Wall_4_0_0_0_0_0_Win_Detailed, !- Name
0.00014, !- Air Mass Flow Coefficient When Opening is Closed {kg/s-m}
0.65, !- Air Mass Flow Exponent When Opening is Closed {dimensionless}
NonPivoted, !- Type of Rectangular Large Vertical Opening (LVO)
0, !- Extra Crack Length or Height of Pivoting Axis {m}
2, !- Number of Sets of Opening Factor Data
0, !- Opening Factor 1 {dimensionless}
0.65, !- Discharge Coefficient for Opening Factor 1 {dimensionless}
0, !- Width Factor for Opening Factor 1 {dimensionless}
0.05, !- Height Factor for Opening Factor 1 {dimensionless}
0.95, !- Start Height Factor for Opening Factor 1 {dimensionless}
1, !- Opening Factor 2 {dimensionless}
0.65, !- Discharge Coefficient for Opening Factor 2 {dimensionless}
1, !- Width Factor for Opening Factor 2 {dimensionless}
0.05, !- Height Factor for Opening Factor 2 {dimensionless}
0.95, !- Start Height Factor for Opening Factor 2 {dimensionless}

```

A.4.4 Simple opening component

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:COMPONENT:SIMPLEOPENING =====

```

```

AirflowNetwork:MultiZone:Component:SimpleOpening,
Block1:Zone1_Wall_3_0_0_0_0_0_Win_Simple, !- Name
0.00014, !- Air Mass Flow Coefficient When Opening is Closed {kg/s-m}
0.65, !- Air Mass Flow Exponent When Opening is Closed {dimensionless}
0.001, !- Minimum Density Difference for Two-Way Flow {kg/m3}
0.65; !- Discharge Coefficient {dimensionless}

```

```

AirflowNetwork:MultiZone:Component:SimpleOpening,
Block1:Zone1_Wall_5_0_0_0_0_0_Win_Simple, !- Name
0.00014, !- Air Mass Flow Coefficient When Opening is Closed {kg/s-m}
0.65, !- Air Mass Flow Exponent When Opening is Closed {dimensionless}
0.001, !- Minimum Density Difference for Two-Way Flow {kg/m3}
0.65; !- Discharge Coefficient {dimensionless}

```

A.5 WIND PRESSURE COEFFICIENT PARAMETERS

```

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:WINDPRESSURECOEFFICIENTARRAY =====

```

```

AirflowNetwork:MultiZone:WindPressureCoefficientArray,
Cp Data, !- Name
0, !- Wind Direction 1 {deg}
45, !- Wind Direction 2 {deg}
90, !- Wind Direction 3 {deg}
135, !- Wind Direction 4 {deg}
180, !- Wind Direction 5 {deg}
225, !- Wind Direction 6 {deg}

```

```

270,                !- Wind Direction 7 {deg}
315;                !- Wind Direction 8 {deg}

!- ===== ALL OBJECTS IN CLASS:
AIRFLOWNETWORK:MULTIZONE:WINDPRESSURECOEFFICIENTVALUES =====

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
  Block1:Zone1_Roof_1_0_0, !- Name
  Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
  -.6,      !- Wind Pressure Coefficient Value 1 {dimensionless}
  -.5,      !- Wind Pressure Coefficient Value 2 {dimensionless}
  -.4,      !- Wind Pressure Coefficient Value 3 {dimensionless}
  -.5,      !- Wind Pressure Coefficient Value 4 {dimensionless}
  -.6,      !- Wind Pressure Coefficient Value 5 {dimensionless}
  -.5,      !- Wind Pressure Coefficient Value 6 {dimensionless}
  -.4,      !- Wind Pressure Coefficient Value 7 {dimensionless}
  -.5;      !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
  Block1:Zone1_Wall_2_0_0, !- Name
  Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
  -.3,      !- Wind Pressure Coefficient Value 1 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 2 {dimensionless}
  0.4,      !- Wind Pressure Coefficient Value 3 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 4 {dimensionless}
  -.3,      !- Wind Pressure Coefficient Value 5 {dimensionless}
  -.35,     !- Wind Pressure Coefficient Value 6 {dimensionless}
  -.2,      !- Wind Pressure Coefficient Value 7 {dimensionless}
  -.35;     !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
  Block1:Zone1_Wall_2_0_0_0_0_0_Win, !- Name
  Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
  -.3,      !- Wind Pressure Coefficient Value 1 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 2 {dimensionless}
  0.4,      !- Wind Pressure Coefficient Value 3 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 4 {dimensionless}
  -.3,      !- Wind Pressure Coefficient Value 5 {dimensionless}
  -.35,     !- Wind Pressure Coefficient Value 6 {dimensionless}
  -.2,      !- Wind Pressure Coefficient Value 7 {dimensionless}
  -.35;     !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
  Block1:Zone1_Wall_3_0_0, !- Name
  Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
  0.4,      !- Wind Pressure Coefficient Value 1 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 2 {dimensionless}
  -.3,      !- Wind Pressure Coefficient Value 3 {dimensionless}
  -.35,     !- Wind Pressure Coefficient Value 4 {dimensionless}
  -.2,      !- Wind Pressure Coefficient Value 5 {dimensionless}
  -.35,     !- Wind Pressure Coefficient Value 6 {dimensionless}
  -.3,      !- Wind Pressure Coefficient Value 7 {dimensionless}
  0.1;      !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
  Block1:Zone1_Wall_3_0_0_0_0_0_Win, !- Name
  Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
  0.4,      !- Wind Pressure Coefficient Value 1 {dimensionless}
  0.1,      !- Wind Pressure Coefficient Value 2 {dimensionless}
  -.3,      !- Wind Pressure Coefficient Value 3 {dimensionless}

```



```

-.35,          !- Wind Pressure Coefficient Value 4 {dimensionless}
-.2,          !- Wind Pressure Coefficient Value 5 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 6 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 7 {dimensionless}
0.1;        !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
Block1:Zone1_Wall_4_0_0, !- Name
Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
-.3,          !- Wind Pressure Coefficient Value 1 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 2 {dimensionless}
-.2,         !- Wind Pressure Coefficient Value 3 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 4 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 5 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 6 {dimensionless}
0.4,         !- Wind Pressure Coefficient Value 7 {dimensionless}
0.1;        !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
Block1:Zone1_Wall_4_0_0_0_0_0_Win, !- Name
Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
-.3,          !- Wind Pressure Coefficient Value 1 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 2 {dimensionless}
-.2,         !- Wind Pressure Coefficient Value 3 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 4 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 5 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 6 {dimensionless}
0.4,         !- Wind Pressure Coefficient Value 7 {dimensionless}
0.1;        !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
Block1:Zone1_Wall_5_0_0, !- Name
Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
-.2,          !- Wind Pressure Coefficient Value 1 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 2 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 3 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 4 {dimensionless}
0.4,         !- Wind Pressure Coefficient Value 5 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 6 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 7 {dimensionless}
-.35;       !- Wind Pressure Coefficient Value 8 {dimensionless}

AirflowNetwork:MultiZone:WindPressureCoefficientValues,
Block1:Zone1_Wall_5_0_0_0_0_0_Win, !- Name
Cp Data, !- AirflowNetwork:MultiZone:WindPressureCoefficientArray Name
-.2,          !- Wind Pressure Coefficient Value 1 {dimensionless}
-.35,        !- Wind Pressure Coefficient Value 2 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 3 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 4 {dimensionless}
0.4,         !- Wind Pressure Coefficient Value 5 {dimensionless}
0.1,         !- Wind Pressure Coefficient Value 6 {dimensionless}
-.3,         !- Wind Pressure Coefficient Value 7 {dimensionless}
-.35;       !- Wind Pressure Coefficient Value 8 {dimensionless}

```

B PROGRAM CODES

B.1 SINGLE-ZONE AIRFLOW NETWORK MODEL

```
import numpy as np
import matplotlib.pyplot as plt

# Input ***** #
V = 250 # building volume [m3]
L = 3 # number of flow paths

H = np.array([2, 4, 7]) # height of flow path [m]
C = np.array([0.03, 0.06, 0.02]) # flow coefficient [m3/s at 1 Pa]
N = np.array([0.7, 0.5, 0.6]) # flow exponent [-]
P = np.array([0.3, -0.25, -0.4]) # wind pressure coefficient [-]

D = 1.29 # air density at 0 degC
E = 18 # external temperature [degC]
I = 20 # internal temperature [degC]
U = 3 # wind speed at building height [m/s]

# Pressure Calculation ***** #
W = np.empty(L)
S = np.empty(L)
T = np.empty(L)

for i in range(L):
    W[i] = 0.5 * D * P[i] * U ** 2
    S[i] = -3455 * H[i] * (1 / (E + 273) - 1 / (I + 273))
    T[i] = W[i] + S[i]

print('wind pressure =', W)
print('stack pressure =', S)
print('wind + stack pressure =', T)

# Calculate Infiltration ***** #
R = -50 # assumed internal pressure for initialization [Pa]
step = 0.01 # solution steps
tol = 0.001 # convergence tolerance

Q_i = np.empty(L) # volume flow rate through the i'th flow path [m3/s]
count = 0 # counter to measure number of iterations
B = 0 # flow balance

for i in range(L):
    count += 1
    delta_P = T[i] - R
    if delta_P == 0:
        Q_i[i] = 0
    else:
        Q_i[i] = C[i] * (abs(delta_P) ** N[i]) * delta_P / abs(delta_P)
    B += Q_i[i]

while B > tol:
    R += step
    B = 0
    for i in range(L):
        count += 1
```

```

    delta_P = T[i] - R
    if delta_P == 0:
        Q_i[i] = 0
    else:
        Q_i[i] = C[i] * (abs(delta_P) ** N[i]) * delta_P / abs(delta_P)
    B += Q_i[i]

Q = 0 # air flow rate [m3/s]
for i in range(L):
    if Q_i[i] > 0:
        Q += Q_i[i]

ACH = Q * 3600 / V # air change rate [1/h]

print('internal pressure at convergence =', round(R, 4), '[Pa]')
print('\nnumber of iterations =', count)
print('flow rate =', Q_i)
print('flow balance =', round(B, 6))
print('\nair flow rate =', round(Q, 4), '[m3/s]')
print('air change rate =', round(ACH, 2), '[1/h]')

```

B.2 PARAMETRIC RUN PROGRAM

```

import numpy as np
import os
import pandas as pd
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import ttk
import time

root = tk.Tk()
root.geometry('350x200')
root.title('simulation progress')

# reading original IDF input file
***** #
input_file = open('Model/DB004BOX.idf', 'r')
edit = input_file.read()
input_file.close()

# editing input file
***** #
edit2 = edit.replace('{', '(')
edit3 = edit2.replace('}', ')')
edit4 = edit3.replace('var', '{var}')

# specifying run parameters
***** #
steps = 15
base_value = 0.0028875
deviation = 0.2
a = base_value - base_value * deviation
b = base_value + base_value * deviation
var = np.linspace(a, b, steps)

text_variable1 = tk.StringVar()
text_variable1.set('progress:\nsimulation: 0/' + str(len(var)) + '\n\ntime
elapsed: 0 [mins]')

```

```

def start():
    # creating input files and running simulation
    ***** #
    text_variable1.set('running...\nsimulation: 0/' + str(len(var)) +
'\ntime elapsed: 0 [mins]')
    root.update_idletasks()
    start_time = time.perf_counter()
    for i in range(len(var)):
        input_file1 = open('InputFiles/InputFile' + str(i) + '.idf', 'w')
        edit5 = edit4.format(var=var[i])
        input_file1.write(edit5)
        input_file1.close()
        os.system(
            'C:/EnergyPlusV8-9-0/energyplus'
            ' -w
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_Fina
lModel/Weather/DEU_BERLIN_IWEC.epw'
            ' -d
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_Fina
lModel/ParametricRun/OutputFiles/OutputFile' + str(i) +
            ' -r
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_Fina
lModel/ParametricRun/InputFiles/InputFile' + str(i) + '.idf"')
        pb1['value'] += (1 / len(var)) * 100
        finish_time = time.perf_counter()
        text_variable1.set('running...\nsimulation: ' + str(i+1) + '/' +
str(len(var)) +
            '\ntime elapsed: ' + str(round((finish_time -
start_time) / 60, 2))
            + ' [mins]')
        root.update_idletasks()

    # post-processing
    ***** #
    AirChangeRate = np.empty(len(var), dtype=float)
    pb1['value'] = 0
    text_variable1.set('post-processing...\nfigure: ' + '0/' +
str(len(var)) +
            '\ntime elapsed: ' + str(round((finish_time -
start_time) / 60, 2)) +
            ' [mins]')
    root.update_idletasks()
    for i in range(len(var)):
        df1 = pd.read_csv(
'C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_Fin
alModel/ParametricRun/OutputFiles/OutputFile' + str(i) + '/eplusout.csv')
        df1.set_index('Date/Time', inplace=True)

        average = df1['BLOCK1:ZONE1:AFN Zone Infiltration Air Change Rate
[ach] (Hourly)'].mean()
        AirChangeRate[i] = average

        fig = plt.figure(figsize=(15, 6), dpi=300)
        df1['BLOCK1:ZONE1:AFN Zone Infiltration Air Change Rate
[ach] (Hourly)'].plot()
        plt.ylabel('BLOCK1:ZONE1:AFN Zone Infiltration Air Change Rate
[ach] (Hourly)')
        plt.axhline(average, color='red')
        plt.text(0.01, average + 1, 'Average: ' + str(round(average, 3)) +
' [ach]', color='red')

```

```

plt.grid()
plt.title('var = ' + str(round(var[i], 3)) + ' [-]\n')
plt.savefig('Figures/fig' + str(i) + '.png')
plt.close(fig)
finish_time = time.perf_counter()
text_variable1.set('post-processing...\nfigure: ' + str(i + 1) +
'/' + str(len(var)) +
'\ntime elapsed: ' + str(round((finish_time -
start_time) / 60, 2)) +
' [mins]')
pb1['value'] += (1 / len(var)) * 100
root.update_idletasks()

print('\n', 'Variable:\n', var, '\n')
print('Air Change Rate:\n', AirChangeRate)

fig = plt.figure(figsize=(15, 6), dpi=300)
plt.subplot(132)
plt.grid()
plt.title('Average values')
plt.xlabel('var')
plt.ylabel('Air Change Rate [ach]')
listOf_Xticks = np.linspace(a, b, 5)
plt.xticks(listOf_Xticks)
listOf_Yticks = np.linspace(AirChangeRate[0], AirChangeRate[len(var) -
1], 10)
plt.yticks(listOf_Yticks)
plt.plot(var, AirChangeRate)
plt.savefig('Figures/fig' + str(len(var)) + '.png')
plt.close(fig)
text_variable1.set('completed successfully\nfigure: ' + str(len(var)) +
'/' + str(len(var)) +
'\ntime elapsed: ' + str(round((finish_time -
start_time) / 60, 2)) +
' [mins]')
button2 = tk.Button(root, text='exit', command=root.destroy, padx=60)
button2.grid(column=1, row=4, padx=30)

labell1 = tk.Label(root, textvariable=text_variable1)
labell1.grid(column=1, row=1, pady=10, padx=30)

pb1 = ttk.Progressbar(root, orient='horizontal', mode='determinate',
length=280)
pb1.grid(column=1, row=2, pady=10, padx=30)

button1 = tk.Button(root, text='start', command=start, padx=60)
button1.grid(column=1, row=3, pady=10, padx=30)

root.mainloop()

```

B.3 SENSITIVITY ANALYSIS PROGRAM

```
import module
import pandas as pd
import os

# average response
***** #
os.system('"C:/EnergyPlusV8-9-0/energyplus'
          ' -w
          C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/Weather/DEU_BERLIN_IWEC.epw'
          ' -d
          C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/OutputFiles/AverageResponse' +
          ' -r
          C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/Model/DB004BOX.idf"')

df1 =
pd.read_csv('C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/OutputFiles/AverageResponse/eplust.csv')
df1.set_index('Date/Time', inplace=True)
flow_rate = df1['BLOCK1:ZONE1:AFN Zone Infiltration Air Change Rate [ach] (Hourly)'].mean()

# Crack component
***** #
# variable 1
***** #
name = 'CC: Air Mass Flow Coefficient at Reference Conditions [kg/s]'
base_value = 0.0028875
deviation = 0.2
variable_num = 1

variable = [base_value - base_value * deviation, base_value + base_value * deviation]
simulation1 = module.Simulation(variable, variable_num)
simulation1.start_simulation()

list1 = [name, variable[0], variable[1], base_value, simulation1.flow_rate[0], simulation1.flow_rate[1], simulation1.SI]

# variable 2
***** #
name = 'CC: Air Mass Flow Exponent [-]'
base_value = 0.7
deviation = 0.2
variable_num = 2

variable = [base_value - base_value * deviation, base_value + base_value * deviation]
simulation2 = module.Simulation(variable, variable_num)
simulation2.start_simulation()

list2 = [name, variable[0], variable[1], base_value, simulation2.flow_rate[0], simulation2.flow_rate[1], simulation2.SI]

# variable 3
***** #
```

```

name = 'CC: Crack Factor [-]'
base_value = 0.5
deviation = 0.2
variable_num = 3

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation3 = module.Simulation(variable, variable_num)
simulation3.start_simulation()

list3 = [name, variable[0], variable[1], base_value,
simulation3.flow_rate[0], simulation3.flow_rate[1], simulation3.SI]

# ELA component *****
#
# variable 4
***** #
name = 'ELA: Effective Leakage Area [m2]'
base_value = 0.01
deviation = 0.2
variable_num = 4

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation4 = module.Simulation(variable, variable_num)
simulation4.start_simulation()

list4 = [name, variable[0], variable[1], base_value,
simulation4.flow_rate[0], simulation4.flow_rate[1], simulation4.SI]

# variable 5
***** #
name = 'ELA: Discharge Coefficient [-]'
base_value = 1
deviation = 0.2
variable_num = 5

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation5 = module.Simulation(variable, variable_num)
simulation5.start_simulation()

list5 = [name, variable[0], variable[1], base_value,
simulation5.flow_rate[0], simulation5.flow_rate[1], simulation5.SI]

# variable 6
***** #
name = 'ELA: Air Mass Flow Exponent [-]'
base_value = 0.65
deviation = 0.2
variable_num = 6

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation6 = module.Simulation(variable, variable_num)
simulation6.start_simulation()

list6 = [name, variable[0], variable[1], base_value,
simulation6.flow_rate[0], simulation6.flow_rate[1], simulation6.SI]

# simple opening component

```

```

***** #
# variable 7
***** #
name = 'SO: Air Mass Flow Coefficient When Opening is Closed [kg/s-m]'
base_value = 0.00014
deviation = 0.2
variable_num = 7

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation7 = module.Simulation(variable, variable_num)
simulation7.start_simulation()

list7 = [name, variable[0], variable[1], base_value,
simulation7.flow_rate[0], simulation7.flow_rate[1], simulation7.SI]

# variable 8
***** #
name = 'SO: Air Mass Flow Exponent When Opening is Closed [-]'
base_value = 0.65
deviation = 0.2
variable_num = 8

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation8 = module.Simulation(variable, variable_num)
simulation8.start_simulation()

list8 = [name, variable[0], variable[1], base_value,
simulation8.flow_rate[0], simulation8.flow_rate[1], simulation8.SI]

# variable 9
***** #
name = 'SO: Discharge Coefficient [-]'
base_value = 0.65
deviation = 0.2
variable_num = 9

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation9 = module.Simulation(variable, variable_num)
simulation9.start_simulation()

list9 = [name, variable[0], variable[1], base_value,
simulation9.flow_rate[0], simulation9.flow_rate[1], simulation9.SI]

# variable 10
***** #
name = 'SO: Window Opening Factor [-]'
base_value = 0.5
deviation = 0.2
variable_num = 10

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation10 = module.Simulation(variable, variable_num)
simulation10.start_simulation()

list10 = [name, variable[0], variable[1], base_value,
simulation10.flow_rate[0], simulation10.flow_rate[1], simulation10.SI]

```

```

# variable 11
***** #
name = 'SO: Minimum Venting Open Factor [-]'
base_value = 0.5
deviation = 0.2
variable_num = 11

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation11 = module.Simulation(variable, variable_num)
simulation11.start_simulation()

list11 = [name, variable[0], variable[1], base_value,
simulation11.flow_rate[0], simulation11.flow_rate[1], simulation11.SI]

# Detailed opening component
***** #
# variable 12
***** #
name = 'DO: Air Mass Flow Coefficient When Opening is Closed [kg/s-m]'
base_value = 0.00014
deviation = 0.2
variable_num = 12

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation12 = module.Simulation(variable, variable_num)
simulation12.start_simulation()

list12 = [name, variable[0], variable[1], base_value,
simulation12.flow_rate[0], simulation12.flow_rate[1], simulation12.SI]

# variable 13
***** #
name = 'DO: Air Mass Flow Exponent When Opening is Closed [-]'
base_value = 0.65
deviation = 0.2
variable_num = 13

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation13 = module.Simulation(variable, variable_num)
simulation13.start_simulation()

list13 = [name, variable[0], variable[1], base_value,
simulation13.flow_rate[0], simulation13.flow_rate[1], simulation13.SI]

# variable 14
***** #
name = 'DO: Discharge Coefficient for Opening Factor 1 and 2 [-]'
base_value = 0.65
deviation = 0.2
variable_num = 14

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation14 = module.Simulation(variable, variable_num)
simulation14.start_simulation()

list14 = [name, variable[0], variable[1], base_value,
simulation14.flow_rate[0], simulation14.flow_rate[1], simulation14.SI]

```

```

# variable 15
***** #
name = 'DO: Window Opening Factor [-]'
base_value = 0.5
deviation = 0.2
variable_num = 15

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation15 = module.Simulation(variable, variable_num)
simulation15.start_simulation()

list15 = [name, variable[0], variable[1], base_value,
simulation15.flow_rate[0], simulation15.flow_rate[1], simulation15.SI]

# variable 16
***** #
name = 'DO: Minimum Venting Open Factor [-]'
base_value = 0.5
deviation = 0.2
variable_num = 16

variable = [base_value - base_value * deviation, base_value + base_value *
deviation]
simulation16 = module.Simulation(variable, variable_num)
simulation16.start_simulation()

list16 = [name, variable[0], variable[1], base_value,
simulation16.flow_rate[0], simulation16.flow_rate[1], simulation16.SI]

# printing and exporting data to csv file
***** #
print('\naverage response:\n', flow_rate)
print('\nsimulation 1:\n', simulation1.flow_rate)
print('simulation 2:\n', simulation2.flow_rate)
print('simulation 3:\n', simulation3.flow_rate)

print('\nsimulation 4:\n', simulation4.flow_rate)
print('simulation 5:\n', simulation5.flow_rate)
print('simulation 6:\n', simulation6.flow_rate)

print('\nsimulation 7:\n', simulation7.flow_rate)
print('simulation 8:\n', simulation8.flow_rate)
print('simulation 9:\n', simulation9.flow_rate)
print('simulation 10:\n', simulation10.flow_rate)
print('simulation 11:\n', simulation11.flow_rate)

print('\nsimulation 12:\n', simulation12.flow_rate)
print('simulation 13:\n', simulation13.flow_rate)
print('simulation 14:\n', simulation14.flow_rate)
print('simulation 15:\n', simulation15.flow_rate)
print('simulation 16:\n', simulation16.flow_rate)

dict1 = {
    'Variable': [list1[0], list2[0], list3[0], list4[0], list5[0],
list6[0], list7[0], list8[0],
                list9[0], list10[0], list11[0], list12[0], list13[0],
list14[0], list15[0], list16[0]],
    'Negative change': [list1[4], list2[4], list3[4], list4[4], list5[4],
list6[4], list7[4], list8[4],

```

```

        list9[4], list10[4], list11[4], list12[4],
list13[4], list14[4], list15[4], list16[4]],
    'Positive change': [list1[5], list2[5], list3[5], list4[5], list5[5],
list6[5], list7[5], list8[5],
        list9[5], list10[5], list11[5], list12[5],
list13[5], list14[5], list15[5], list16[5]],
    'Lower limit': [list1[1], list2[1], list3[1], list4[1], list5[1],
list6[1], list7[1], list8[1],
        list9[1], list10[1], list11[1], list12[1], list13[1],
list14[1], list15[1], list16[1]],
    'Upper limit': [list1[2], list2[2], list3[2], list4[2], list5[2],
list6[2], list7[2], list8[2],
        list9[2], list10[2], list11[2], list12[2], list13[2],
list14[2], list15[2], list16[2]],
    'Base value': [list1[3], list2[3], list3[3], list4[3], list5[3],
list6[3], list7[3], list8[3],
        list9[3], list10[3], list11[3], list12[3], list13[3],
list14[3], list15[3], list16[3]],
    'Average response': [flow_rate, flow_rate, flow_rate, flow_rate,
flow_rate, flow_rate, flow_rate, flow_rate,
        flow_rate, flow_rate, flow_rate, flow_rate,
flow_rate, flow_rate, flow_rate, flow_rate],
    'Sensitivity Index': [list1[6], list2[6], list3[6], list4[6], list5[6],
list6[6], list7[6], list8[6],
        list9[6], list10[6], list11[6], list12[6],
list13[6], list14[6], list15[6], list16[6]]
}

```

```

df1 = pd.DataFrame(dict1)
df1.set_index('Variable', inplace=True)
df1.to_csv('OutputFiles/output.csv')

```

```

import numpy as np
import pandas as pd
import os

```

```

class Simulation:
    def __init__(self, variable, variable_num):
        self.variable = variable
        self.variable_num = variable_num
        self.flow_rate = np.empty(len(self.variable), dtype=float)
        self.SI = 0

    def start_simulation(self):
        input_file = open('InputFiles/Variable' + str(self.variable_num) +
'/DB004BOX.idf', 'r')
        edit = input_file.read()
        input_file.close()
        edit2 = edit.replace('{', '(')
        edit3 = edit2.replace('}', ')')
        edit4 = edit3.replace('var', '{var}')

        for i in range(len(self.variable)):
            input_file = open('InputFiles/Variable' +
str(self.variable_num) + '/InputFile' + str(i) + '.idf', 'w')
            edit5 = edit4.format(var=self.variable[i])
            input_file.write(edit5)
            input_file.close()
            os.system(

```

```

        'C:/EnergyPlusV8-9-0/energyplus'
        '-w
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/Weather/DEU_BERLIN_IWEC.epw'
        '-d
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/OutputFiles/Variable' + str(self.variable_num) +
'/OutputFile' + str(i) +
        '-r
C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/InputFiles/Variable' + str(self.variable_num) +
'/InputFile' + str(i) + '.idf"')

    for i in range(len(self.variable)):
        dfl = pd.read_csv(

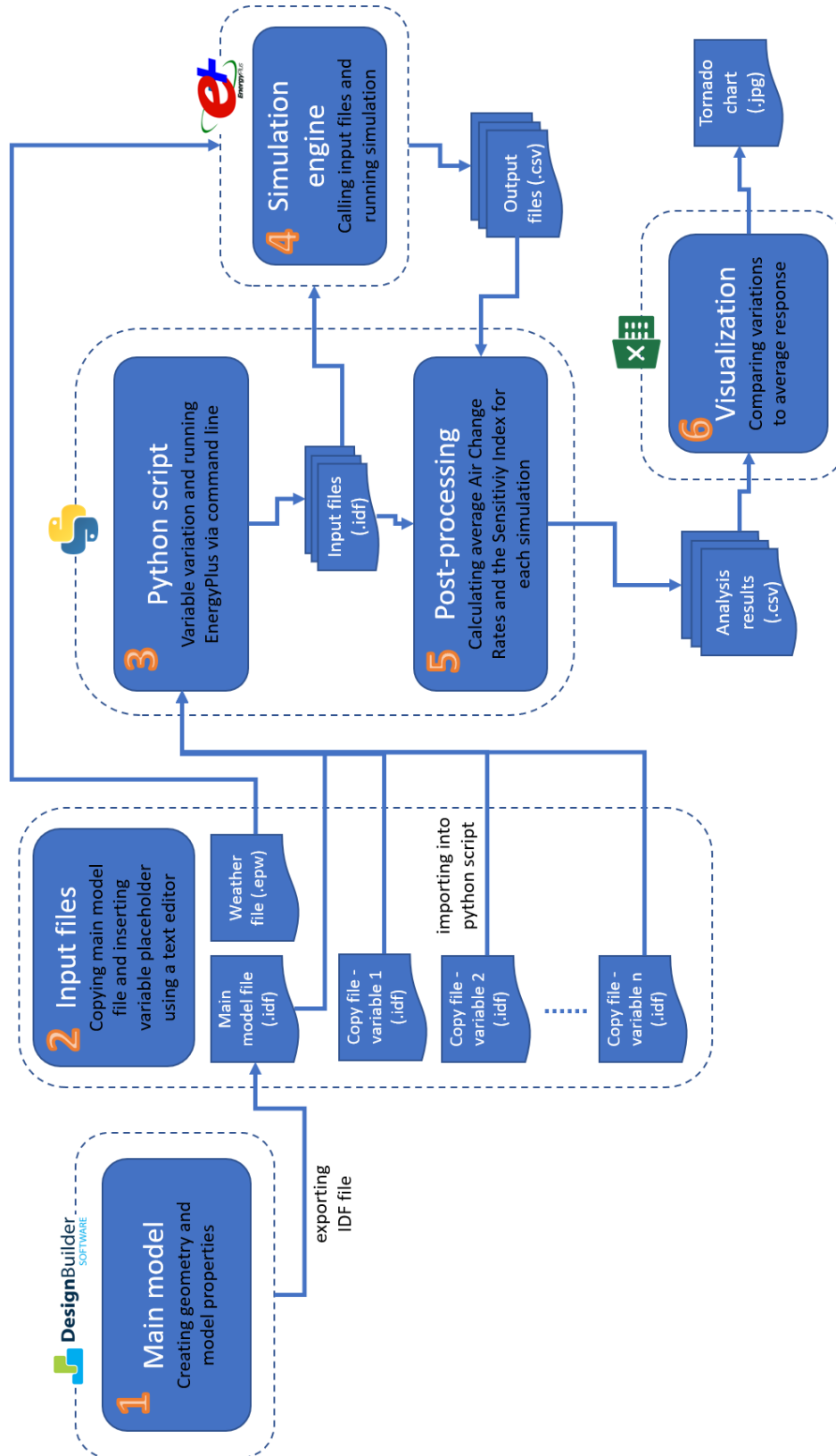
'C:/Users/musta/Desktop/Masterarbeit/SoftwareFiles/09_AFN_ParametricRun_FinalModel/SensitivityAnalysis/OutputFiles/Variable' + str(self.variable_num)
+ '/OutputFile' + str(i) + '/eplusout.csv')
        dfl.set_index('Date/Time', inplace=True)
        average = dfl['BLOCK1:ZONE1:AFN Zone Infiltration Air Change
Rate [ach] (Hourly)'].mean()
        self.flow_rate[i] = average

        if self.flow_rate[0] < self.flow_rate[1]:
            self.SI = (self.flow_rate[1] - self.flow_rate[0]) /
self.flow_rate[1]
        else:
            self.SI = (self.flow_rate[0] - self.flow_rate[1]) /
self.flow_rate[0]

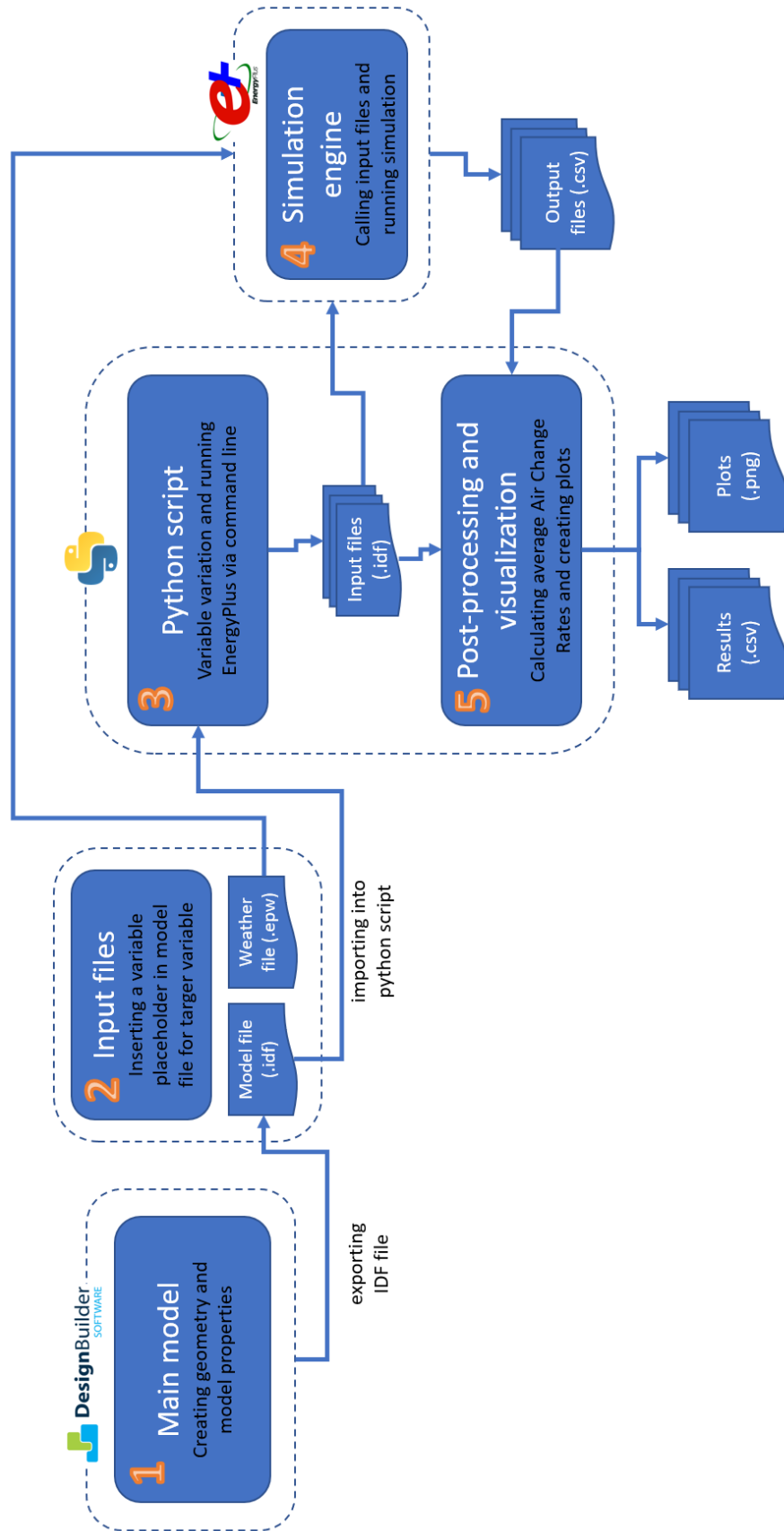
```

C WORKFLOW SCHEME

C.1 WORKFLOW SCHEME OF THE SENSITIVITY ANALYSIS



C.2 WORKFLOW SCHEME OF THE PARAMETRIC RUN



D SENSITIVITY ANALYSIS RESULTS

Variable	Negative change	Positive change	Lower limit	Upper limit	Base value	Average response	Sensitivity Index
DO: Air Mass Flow Exponent When Opening is Closed [-]	1.800018 837	1.800174 507	0.52	0.78	0.65	1.800351 774	8.65E-05
SO: Air Mass Flow Exponent When Opening is Closed [-]	1.801181 726	1.800288 711	0.52	0.78	0.65	1.800351 774	0.000495 794
CC: Crack Factor [-]	1.798747 729	1.800739 087	0.4	0.6	0.5	1.800351 774	0.001105 856
CC: Air Mass Flow Coefficient at Reference Conditions [kg/s]	1.798747 729	1.800739 087	0.0023 1	0.0034 65	0.0028 875	1.800351 774	0.001105 856
SO: Air Mass Flow Coefficient When Opening is Closed [kg/s-m]	1.796328 32	1.798579 947	0.0001 12	0.0001 68	0.0001 4	1.800351 774	0.001251 891
CC: Air Mass Flow Exponent [-]	1.799758 307	1.802996 568	0.56	0.84	0.7	1.800351 774	0.001796 044
ELA: Air Mass Flow Exponent [-]	1.801862 603	1.798187 046	0.52	0.78	0.65	1.800351 774	0.002039 865
DO: Air Mass Flow Coefficient	1.797059 332	1.801352 157	0.0001 12	0.0001 68	0.0001 4	1.800351 774	0.002383 113

When Opening is Closed [kg/s-m]							
DO: Minimum Venting Open Factor [-]	1.794356 202	1.806201 525	0.4	0.6	0.5	1.800351 774	0.006558 14
ELA: Discharge Coefficient [-]	1.790291 195	1.811957 78	0.8	1.2	1	1.800351 774	0.011957 555
ELA: Effective Leakage Area [m2]	1.790291 195	1.811957 78	0.008	0.012	0.01	1.800351 774	0.011957 555
DO: Window Opening Factor [-]	1.781072 677	1.813150 948	0.4	0.6	0.5	1.800351 774	0.017692 002
DO: Discharge Coefficient for Opening Factor 1 and 2 [-]	1.781072 677	1.813150 948	0.52	0.78	0.65	1.800351 774	0.017692 002
SO: Minimum Venting Open Factor [-]	1.741486 138	1.863024 487	0.4	0.6	0.5	1.800351 774	0.065237 119
SO: Window Opening Factor [-]	1.590026 706	2.000524 885	0.4	0.6	0.5	1.800351 774	0.205195 237
SO: Discharge Coefficient [-]	1.590026 706	2.000524 885	0.52	0.78	0.65	1.800351 774	0.205195 237

Table 4: Sensitivity analysis results

6 LITERATURE SOURCES

ASHRAE. (2013). *ASHRAE handbook – fundamentals*.

Belleri, A. (2013). *A Sensitivity Analysis of Natural Ventilation Design Parameters For Non Residential Buildings*. Bolzano: Institute for Renewable Energy, EURAC research.

Belleri, A. (2014). Natural ventilation design: An analysis of predicted and measured performance. *Building and Environment*, 123-138.

Costola, D. (2009). Overview of pressure coefficient data in building energy simulation and airflow network programs. *Building and Environment*, 2027–2036.

Good, J. (2008). The Role of Wind in Natural Ventilation Simulation Using Airflow Network Models. *Third National Conference of IBPSA*, (S. 140-147). California.

Gu, L. (2007). Airflow Network Modeling in EnergyPlus. *Proceedings: Building Simulation 2007*.

Hamby, D. M. (1994). A Review of Techniques for Parameter Sensitivity Analysis of Environmental Models. *Environmental Monitoring and Assessment*, (32) 135-154.

Liddament, M. W. (1996). *A Guide to Energy Efficient Ventilation*. Air Infiltration and Ventilation Centre (AIVC).

Marschall, M. (2020). Simulating Natural Ventilation in Early Stage Design: Combining an Occupant Behavior Model with an Airflow Network Approach. *Impact: Design With All Senses*, 119-131.

Monari, F. (2016). Characterization of an airflow network model by sensitivity analysis: parameter screening, fixing prioritizing and mapping. *Journal of Building Performance Simulation*.

Prignon, M. (2017). Factors influencing airtightness and airtightness predictive models: A literature review. *Energy and Building*, 87-97.

Tian, W. (2013). A review of sensitivity analysis methods in building energy analysis.
Renewable and Sustainable Energy Reviews, 411-419.

Walton, G. N. (1989). *AIRNET - A Computer Program for Building Airflow Network Modeling*.
Washington: National Institute for Standards and Technology.