Technische Universität München

TUM School of Engineering and Design

Chair of Computational Modeling and Simulation

Parametric Building Energy Performance Simulation with Sensitivity Analysis Using BIM Models in Early Design Stages

Master Thesis

for the Master of Science Degree in Civil Engineering

Author: Viet Cuong Dao

Matriculation Number:

Supervised by: Prof. Dr.-Ing. André Borrmann

M. Sc. Kasimir Forth

M. Sc. Jiabin Wu

Chair of Computational Modeling and Simulation

Date of Submission: 15. December 2023

Acknowledgement

This thesis acknowledgement is a tribute to all the individuals who made my academic journey worthwhile at TUM during the last two years.

I express my sincerest gratitude to the Chair of Computational Modeling and Simulation at TUM, led by Prof. André Borrmann, for all the support and guidance throughout my research.

I am indebted to my two supervisors, Kasimir Forth and Jiabin Wu, for the dedicated support and insightful discussions from the beginning of my research. Without your quick response and amazing suggestions, I would not have accomplished this thesis.

I would like to thank Hai Van Le for his willingness to provide valuable technical assistance despite the time difference.

My special appreciation to Herr Duc Chien Le and Linhvan Team for the valuable support I received during my relocation from Finland.

Special mention for all members of the AED2 Team, with Duc Hoang Nguyen, for all the mental support and valuable experiences during my academic journey.

And lastly, my heartfelt appreciation extended to my parents, my family, and friends, for the unwavering support throughout my journey. Thank you for always believing and staying with me.

Abstract

The current effort to prioritize energy efficiency and energy performance improvement in buildings is crucial to reduce the high CO_2 emissions and energy consumption of the building and construction industry. In this context, integrating building energy performance simulation (BEPS) into the early stages provides opportunities to influence the actual building performance at an early stage. However, the major challenge for BEPS in the early design stages is the uncertainties of design parameters required for detailed energy simulation. Hence, this study represents an attempt to shift BEPS into the early design stages by developing a framework for parametric BEPS with sensitivity analysis (SA) originating from early design Building Information Modeling (BIM). A novel closed-BIM approach integrating Autodesk Revit, Rhino.Inside.Revit, and the SALib package is introduced. The major findings emerged from the study were as follows: (i) A framework of BIM-based BEPS was developed. with quick, robust, and automated geometry transformation and semi-automated semantic data transformation was developed. (ii) Early design BIM models at Building Development Level (BDL) 3 provided sufficient information for detailed energy simulations. (iii) The interpretation of SA results, including differentiation of influential parameters and parameter rankings, indicated SA should be performed on the simulation output of BIM model at each BDL. This thesis validates the advantages of utilizing BIM models as the core data models for BEPS in the early design stages and contributes to the general knowledge of BIM-BEPS interoperability and SA for BEPS. For future enhancements of the study, the author recommends increasing the simulation spaces, conducting more sophisticated SA methods, and integrating Machine Learning into the current workflow.

Keywords: Building Energy Performance Simulation (BEPS), Building Information Modeling (BIM), Sensitivity Analysis, Morris Method, Early Design Stages, closed-BIM

Table of Content

List of Figures

List of Tables

List of Abbreviations

AEC	Architectural, Engineering, and Construction
API	Application Programming Interface
BDL	Building Development Level
BEM	Building Energy Modeling
BEPS	Building Energy Performance Simulation
BIM	Building Information Modeling
BPA	Building Performance Analysis
EPW	EnergyPlus Weather Format
EUI	End Use Intensity
gbXML	Green Building Extensible Markup Language
GSA	Global Sensitivity Analysis
GUIs	Graphical User Interfaces
HB	Honeybee
HBJSON	Honeybee JavaScript Object Notation
HVAC	Heating, Ventilation, and Air Conditioning
IFC	Industry Foundation Classes
LCEA	Life Cycle Energy Assessment
LOD	Level of Development
LSA	Local Sensitivity Analysis

MC	Monte Carlo
OAT	One-At-a-Time
PDFs	Probability Distribution Functions
R.I.R	Rhino.Inside.Revit
SA	Sensitivity Analysis
SDK	Software Development Kit
UA	Uncertainty Analysis
VPL	Visual Programming Language

1 Introduction

1.1 Overview

The building and construction industry in the European Union accounts for 40% of energy consumption and 36% of annual carbon dioxide emissions (European Commission 2020). To address this challenge, European Commission (2019) aims to prioritize energy efficiency and energy performance improvement in buildings as a key principle to reduce CO_2 emissions. In the context of energy efficient buildings design, architects and energy specialists have strived to develop methodologies that shift the building energy performance simulation (BEPS) into early design stages over the last two decades (Harter et al. 2020). One of the powerful approach to achieve energy efficiency in buildings with BEPS is the Building Energy Modeling (BEM) technology, which emphasizes the evaluation of alternative designs to offer optimized building designs (Gao et al. 2019a).

The emerge of Building Information Modeling (BIM) in the planning process facilitates the integration of building performance simulation and analysis into the early design phases (Borrmann et al. 2018). Indeed, BIM offers quick and efficient access to geometric and semantic information of building components. Moreover, BIM provides a parametric digital representation of a built facility, in which the modification of an element automatically adjusts an adjacent element or assembly to preserve the established relationship (Gao et al. 2019a). Thus, design uncertainties in the early design stages can be considered parametrically varying geometric or semantic information in BIM models. The parametric definition of the design model also provides designers the ability to explore and choose from a wider range of design alternatives based on building energy performance simulation and then further adjust the chosen solutions.

Existing building energy performance simulation (BEPS) tools function as performance validation of a project with defined geometry in the later design stages, rather than as dynamic assistance to the decision-making process in the early design stages (Touloupaki and Theodosiou 2017). However, the use of BEPS in the early design stage substantially affects actual building performance and construction/operational cost via design parameters such as shape, orientation, and envelope configuration. By utilizing BIM as the central data model for BEPS, continuous verification of building energy performance is available for the whole building life cycle (Laine and Karola 2007). Moreover, BIM-based approach significantly reduces the amount of time preparing input data for BEM tools (Gao et al. 2019a).

While most of the information required for energy analysis is stored in the BIM model, insufficient interoperability between BIM and BEPS tools is the major challenge to achieve reliable energy simulation and analysis (Jin et al. 2019). In the current practice, parametric relations in BIM model is lost during the translation to BEM, which requires manual re-enter of data and additional definition of parameters for BEPS tools besides the input data obtained from BIM prior to the simulation (Reisinger and Kovacic 2019). Another major challenge for the integration of BEPS tools in the early design stages is the design uncertainties in geometric and semantic information of building components in the BIM models. Currently, energy simulation experts have to make estimations of missing information considering materials or geometrical configuration, which require expert prior knowledge in energy-efficient buildings design and lack transparency (Harter et al. 2020).

In BEM related research, sensitivity analysis (SA) methods have recently been used to explore and determine influential input design parameters and their corresponding output variation. The choice of an appropriate SA for BEM investigations, whether local method or global variance-based Sobol' or screening-based Morris's method, depends on the purpose of the analysis (Kristensen and Petersen 2016). With influential input parameters determined from the sensitivity analysis, uncertainty analysis is performed to quantify uncertainty in the model output (Pang et al. 2020). The utilization of SA methods offers instruction to reduce uncertainties and consequently enhance the precision of overall simulation results (Schneider-Marin et al. 2020).

1.2 Objective

The goal of this thesis is to develop a framework for BIM-based parametric BEPS with sensitivity analysis to support the decision-making process in the early design stages. The thesis aims to answer the following questions:

- How can a framework of BEPS originating from early design BIM models with sensitivity analysis improve the decision-making process in the early design stages?
- What are the minimum information requirements for early design BIM models to secure detailed BEPS?

- What are the most and least sensitive parameters for building energy performance in the early design stages?
- How does the proposed framework contribute to the improvement of the existing BIM-BEM interoperability?

1.3 Structure

The rest of the thesis is structured as follows:

- Chapter 2, Literature Review, engages an in-depth literature review, inquiring into the existing body of knowledge concerning BIM, BIM-based BEPS, and SA methods for BEPS. The chapter aims to establish a solid theoretical background for the study.
- Chapter 3, Methodology, presents the research methodology used in this study. The chapter advances the study by proposing a general framework for integrating parametric BIM-based BEPS with SA methods.
- Chapter 4, Prototypical Implementation, involves the implementation of the general framework using selected software tools. Furthermore, a case study is designed to evaluate the prototypical software implementation.
- Chapter 5, Results & Discussion, presents the results from the case study. Moreover, the chapter provides a detailed overview of the research findings and limitations.
- Chapter 6, Conclusion and Outlook, concludes the thesis by answering the research questions and outlining suggestions for future enhancement and extension of this study.

2 Literature Review

The following chapter reviews the concept of BIM and discusses the development of BIM in different design stages. Furthermore, the workflow for BIM-based building energy performance simulation and the existing BIM-BEM exchange formats are introduced. Finally, a review of sensitivity analysis methods and their applications in previous building performance analysis (BPA) research is presented.

2.1 Building Information Modeling

A Building Information Model is defined as the digital representation of physical and functional attributes of a construction facility with great information insight (Borrmann et al. 2018). A typical Building Information Model contains the 3D geometry and semantic information of the building components. The concept of Building Information Modeling (BIM) then illustrate the process of generating and managing the Building Information Model throughout the life-cycle of the built facility using authoring tools (Borrmann et al. 2018).

In conventional workflows, printed or digital technical drawings are the main means of information exchange between stakeholders from different disciplines. This paper-based workflow results in the disruptions in the information flow due to the error-prone and laborious manual re-entering of information to downstream applications for further analysis of the built facility (Borrmann et al. 2018). The implementation of BIM-related technologies in the Architectural, Engineering, and Construction (AEC) industry aims to replace the conventional workflows by continuously reusing data from the centralized BIM-model (Kolbeck 2020).

The adoption of BIM involves several software products from different vendors and varies significantly between companies and countries. The data exchange between individual software solutions is required due to the involvement of various disciplines and the distribution of tasks across different companies (Borrmann et al. 2018). Consequently, the format used for data exchange is classified into “closed” or “open” BIM. Closed BIM refers to the workflows employing proprietary data formats from only one vendor such as Autodesk or Nemetschek. On the other hand, open BIM approaches utilizes vendor-neutral and open-source formats such as Industry Foundation Class (IFC) and Green Building XML (gbXML) to exchange comprehensive digital building models. However, the usage of open BIM includes an export and import interface which potentially cause data loss and complication of the workflow (Kolbeck 2020).

Through different stages of a construction project, the building model is continuously elaborated from conceptual design to as-built model. To formalize the progressive nature of the building design process, the Level of Development (LOD) concept was introduced and widely used in the AEC industry. The LOD concept defines the required geometric detail and alphanumeric information of individual building components within a building model in a specific design phase (Borrmann et al. 2018). However, the existing definitions of LOD lack the relevant parameters considering energy calculation perspective in different design stages (Harter et al. 2020). To address this lack, Abualdenien and Borrmann (2019) introduced the Building Development Level (BDL) concept, which enables the integration of project-specific parameters in the early design stages. In general, the BDL describes the required maturity of an entire digital building model at a certain stage by composing specifications for components of diverse LODs (see Figure 1). The early design stages of buildings are represented by BDL 1 to 4 (Harter et al. 2020).

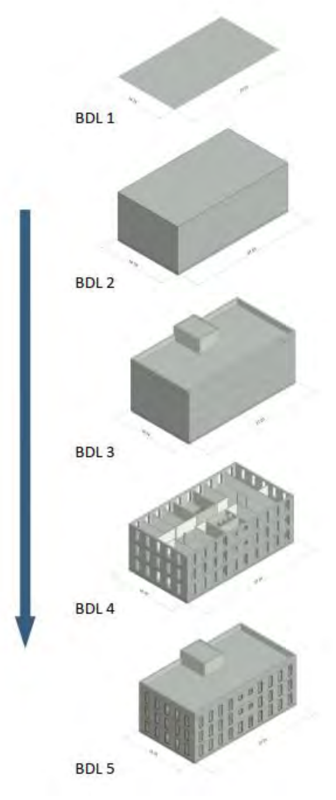


Figure 1: Refinement of building model at early design stages using the BDL scale (Abualdenien and Borrmann 2019)

2.2 BIM-based Building Energy Performance Simulation

The term building performance simulation, also known as building energy performance simulation, building energy modeling or energy simulation, refers to a physic-based software simulation used to predict and analyze building energy use. BEPS tools provide versatile and multipurpose approaches to assist various use cases in architectural design, HVAC design and operation, building performance rating, and building stock analysis (National Renewable Energy Laboratory 2023a). The majority of BEPS programs incorporate an engine to enable detailed simulations based on simple text-based input files (Maile T. et al. 2007). The inputs for BEPS programs include description of building geometry, construction materials, system configurations, component efficiencies, control strategies, operational schedules, and weather data.

Gao et al. (2019a) cited the tedious preparation process for energy model in conventional BEPS workflow as the major challenge for BEPS implementation in early design stages. The increasing use of BIM throughout the life-cycle of buildings and growing software support for open BIM contribute to emerge of the BIM-based BEPS approach (Andriamamonjy et al. 2018). In fact, BIM models store more than 70% of the information required for BEPS (Casini 2022). Therefore, retrieving input for BEPS tools from pre-designed BIM model significantly reduces the time and effort spent in preparing energy model and thus increase the consistency and accuracy of the process (Gao et al. 2019a).

Different BIM-based workflows for BEPS were investigated in previous studies, such as for automated building energy performance model generation (Giannakis et al. 2020), for automated BEPS with IFC (Andriamamonjy et al. 2018), for energy analysis and building sustainability assessment (Carvalho et al. 2021), or for energy audits (Congiu et al. 2022). Review on studies of BIM integration with BEPS (Jin et al. 2019) highlighted the interoperability as a major research problem. Despite the accurate representation of material quantities and building components provided by BIM models, insufficient interoperability between BIM and BEPS impedes the development of reliable BIM-based BEPS (Jin et al. 2019).

2.3 BIM-BEM interoperability and exchange formats

The term interoperability describes the ability to exchange data without loss between different software products by different vendors (Borrmann et al. 2018). In addition, interoperability should also enable bidirectional updates and data exchange for building information. Interoperability between BIM and BEM forms the essential base for the BIM-based BEM technology, which enables energy model generation from the direct access of building design information (Gao et al. 2019a). A higher level of BIM-BEM interoperability would directly improve BEPS usability within the design stages. In fact, progress in BIM-BEM interoperability eliminate potential human errors and data repetition, and enable BEPS to less specialized professionals (Casini 2022).

The first commonly seen approach for BIM-BEM transformation process is the integration of the transformation process into the BIM software (Andriamamonjy et al. 2018). In particular, the Application Programming Interface (API) of the BIM software is utilized to develop interfaces that handle BIM to BEM transformation within the BIM authoring tools. The main drawback of this approach is the dependence on a specific version of a proprietary software, which requires high maintenance efforts (Andriamamonjy et al. 2018).

Alternatively, the data from BIM authoring tool is exported to a file using open BIM schemas and subsequently importing that file into the simulation software (Giannakis et al. 2020). The typical workflow of this approach consists of three essential steps. First, the BIM model is simplified within the BIM authoring tool by refining the building geometry, internal loads, and equipment systems and eliminating insignificant data. Next, the optimized BIM model is exported into open-source formats. Lastly, the BIM files are imported into the modeler graphical user interfaces (GUIs) of BEPS tools (Casini 2022). Spielhaupter (2021) listed five available data file schemas for conversing data from BIM to BEM, including the Industry Foundation Classes (IFC), Green Building XML (gbXML), Honeybee Schema, SEMERGY Building Model, Simulation Domain Model, and OpenStudio Model. Among the listed schemas, IFC and gbXML are the most widely acknowledged building information exchange schemas for energy simulations (O'Donnell et al. 2020).

IFC is an open, vendor neutral data exchange format developed and maintained by buildingSMART International (buildingSMART International 2023). The IFC schema utilizes object-oriented approach to represent both geometry and semantic structure of a building model (Borrmann et al. 2018). IFC remains the only standardized and ISO certified non-proprietary BIM-based format that supports a wide variety of disciplines and use cases in the whole building life cycle (O'Donnell et al. 2020). In contrast, the gbXML schema by Autodesk Green Building Studio focuses on the data exchange exclusively for energy performance simulation purposes. Hence, the previous BIM-BEM integration studies showed more preference towards gbXML than IFC (Casini 2022). The advantages of gbXML in BEM compared to IFC include the straightforward bottom-up structure, the ability to transfer project location data, and wider range of support from current energy modeling programs (Casini 2022). Nevertheless, the bidirectional updates and data exchange between design software applications and BEPS software are unavailable despite the exchange format being used (Casini 2022).

Lack of required data is a commonly reported problem in BIM-BEM interoperability, which causes by the improper data transfer of the BIM authoring tools or the inability to retrieve the required data of the BEM tools (Kamel and Memari 2019). A solution to this problem is to utilize corrective middleware tools (Kamel and Memari 2019) or ecosystem that links simulation and performance analysis services to design applications proves to be a potential solution. A notable example of such ecosystem is Ladybug Tool’s Pollination, which utilizes the Honeybee JSON (HBJSON) schema and provide cloud-based simulation (Pollination 2023). The HBJSON schema is designed to provide a simple and robust geometry model with semantic information to support several analyses by abstracting the individual requirements of analytical engines (Office of Energy Efficiency & Renewable Energy 2022). In addition, the geometry model in HBJSON format is cleanly separated from metadata, which allows easy update of geometry changes to the analytical model. Hence, the information loss during direct data exchange of a BIM model to gbXML is significantly reduced by using Pollination and HBJSON schema (Roudsari 2021).

2.4 Parametric Building Energy Performance Analysis

The performance-based design concept, which prioritizes the energy performance in design, is crucial for the shift of performance assessment into the conceptual design stages (Aksamija and Brown 2018). One objective of the performance-based design is the option to explore various design alternatives and select the optimal alternative for the project (Asl et al. 2013). Parametric modeling proves to be the solution for this requirement. Indeed, parametric modeling facilitates generative form-making based on performance metrics of building and allows automatic update of objects (Asl et al. 2014).

The integration of parametric modeling and BEPS provides more cohesive and effective design process. Touloupaki and Theodosiou (2017) provided a workflow to integrate parametric modeling and building energy performance analysis in architectural design process (see Figure 2). One of the methods to integrate parametric modeling and building energy performance simulation is the integrated dynamic model (Aksamija and Brown 2018). The integrated dynamic model utilizes a middleware component, which is a visual programming language (VPL), to exchange the data between the design and BPS tools. Visual programming provides a more friendly approach for non-programmers or novice programmers to manipulate complex parametric models than conventional textual programming. Notable example VPL for parametric modeling in architectural design are Grasshopper for Rhinoceros and Dynamo for Revit (Asl et al. 2014).

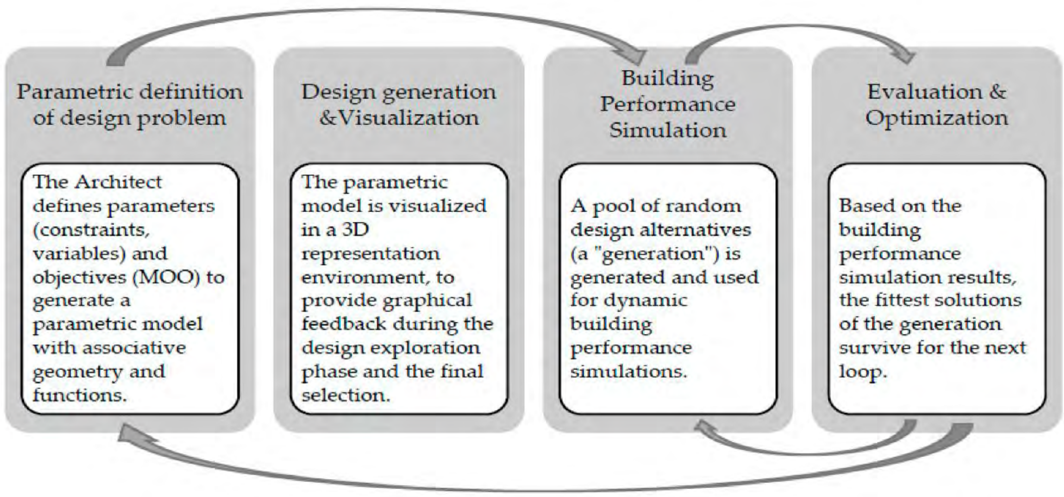


Figure 2: Computational performance-driven design workflow by Touloupaki and Theodosiou (2017)

A variety of BPS tools are available to address different aspects of energy performance analysis for BIM design platforms, such as EnergyPlus, eQuest, DesignBuilder, and Autodesk Green Building Studio. When integrating BIM and parametric design to assess building energy-efficiency applications, previous studies show a predominant use of Autodesk Revit and Dynamo (Zardo et al. 2019). For instance, by Gao et al. (2019b) proposed a Revit-based real time building energy simulation and optimization using Revit API and Dynamo. Regarding Grasshopper, an interoperability package is required for the Grasshopper-BIM connection (Zardo et al. 2019).

Grasshopper for Rhinoceros is currently the most favored software for parametric 3D modeling and energy performance analysis among architects (Touloupaki and Theodosiou 2017). The workflow for the parametric energy simulation in Rhinoceros with

Grasshopper involves the open-source plugins Ladybug and Honeybee. Designers can utilize Ladybug and Honeybee to visualize and analyze weather data in Grasshopper, and to connect Grasshopper environment to validated simulation engines like EnergyPlus or OpenStudio respectively. Previous studies involving Grasshopper, Ladybug, and Honeybee for parametric energy analysis include for example the optimization of building shape for minimum energy use density by Konis et al. (2016), optimization of office building for minimum energy consumption by Qingsong and Fukuda (2016), and the parametric energy analysis by Aksamija and Brown (2018).

2.5 Sensitivity Analysis

Sensitivity analysis (SA) is the study which investigates different techniques to apportion the uncertainty in the output of a model to different sources of uncertainty in model input parameters. The SA methods are usually categorized into local sensitivity analysis (LSA) and global sensitivity analysis (GSA) (Kristensen and Petersen 2016).

LSA methods are based on One-At-a-Time (OAT) technique, which evaluates the variation of a single input parameter at discrete points of the input space while other input parameters are fixed at their reference value (Kristensen and Petersen 2016). In LSA methods, the characteristics of the input parameters are neglected, which results in the inability to provide insight about the correlations between parameters. Also, the LSA methods only explore a reduced set of uncertain parameters. The LSA methods are capable of handling nonlinear and monotonic models. Kristensen and Petersen (2016) recommended the use of LSA only when the object of the analysis is identifying a group of the most influential input parameters instead of actual ranking them.

In contrast to the LSA methods, the GSA methods investigate the influence of an input parameter on the output by varying all other input parameters withing the input parameter space (Kristensen and Petersen 2016). GSA methods adopt a probabilistic framework to incorporate the effect of range and shape of the input probability distribution functions (PDFs). The probabilistic framework of GSA requires a large number of Monte Carlo-based analysis of the model to investigate the model output repeatedly on randomly selected input samples from the entire input space (Kristensen and Petersen 2016). The accuracy of the MC evaluations in this case has a strong dependency on the choice of sampling techniques for the input space (Mara and Tarantola 2008). Among various available sampling techniques, Kristensen and Petersen (2016) presented three most prevailing sampling techniques, namely Monte Carlo (MC) sampling, Sobol’ sequences, and Latin hypercube sampling. In practice, the GSA methods are divided into different groups such as screening-based approach, variance-based methods, linear analysis methods, or regional sensitivity analysis.

A closely related practice with SA is the uncertainty analysis (UA). The major difference between UA and SA lies in the UA’s objective of quantifying the variability of the output due to the variability of input (Pang et al. 2020). In practice, it is recommended to utilize UA when performing SA to verify whether if the output variability resulted from sensitivity indices lies within a feasible range of model behavior (Pianosi et al. 2016).

2.6 Sensitivity Analysis in Building Performance Analysis

In the context of BPA studies, SA has been used extensively to analyze the model behavior and determine influential input parameters in the building energy model for computer simulations and observational studies (Pang et al. 2020). Particularly, SA can be utilized to assist a wide range of model-based applications, namely the model simplification, model-based optimization, model calibration, and input-output understanding (Pang et al. 2020). The benefits of SA application in BPA include model simplification via the parameter screening, improving model robustness, identifying the unexpected uncertainties which may lead to errors, and providing decision support by varying the input parameters (Pang et al. 2020).

A general three-steps workflow for SA in building performance simulation is described in Figure 3. The first step focuses on generating a baseline model and identifying the uncertainties of the interested parameters. Designers can utilize different PDFs and sampling strategies to determine the uncertainties of the parameters. Subsequently, the previously determined uncertainties are incorporated to the baseline model, which forms multiple samples for MC simulation. Lastly, uncertainty and sensitivity analysis will be conducted on the inputs and outputs from the MC simulation. The parameter ranking resulted from the UA/SA provides reference for model optimization and calibration. For example, a workflow for building energy optimization with SA involving Rhino, Grasshopper, Honeybee, Ladybug and EnergyPlus is showed in Figure 4 (Shahsavari et al. 2019).

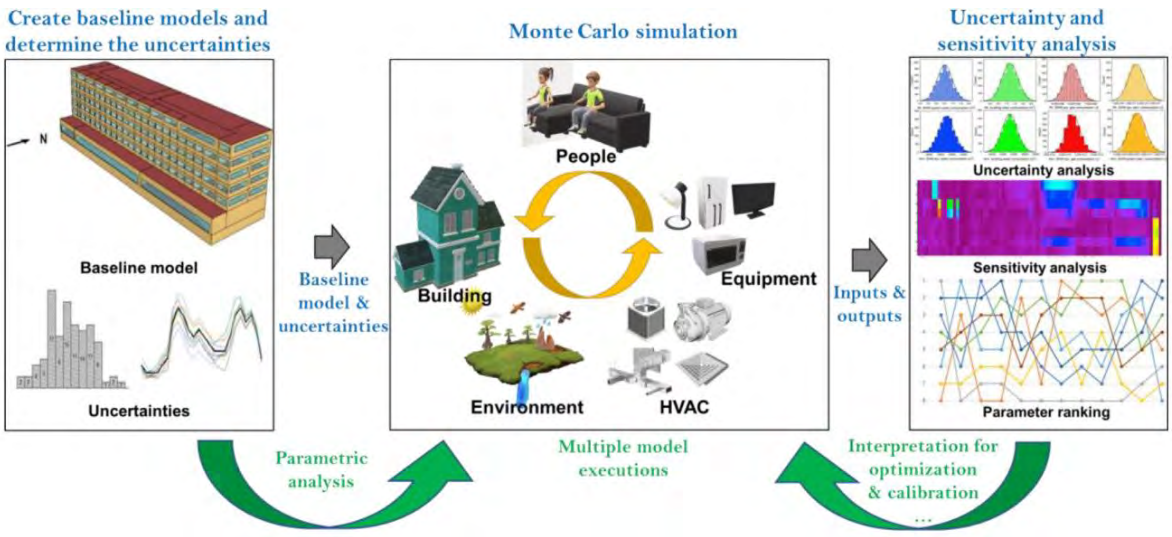


Figure 3: Flowchart for the UA/SA in building performance simulation by Pang et al. (2020)

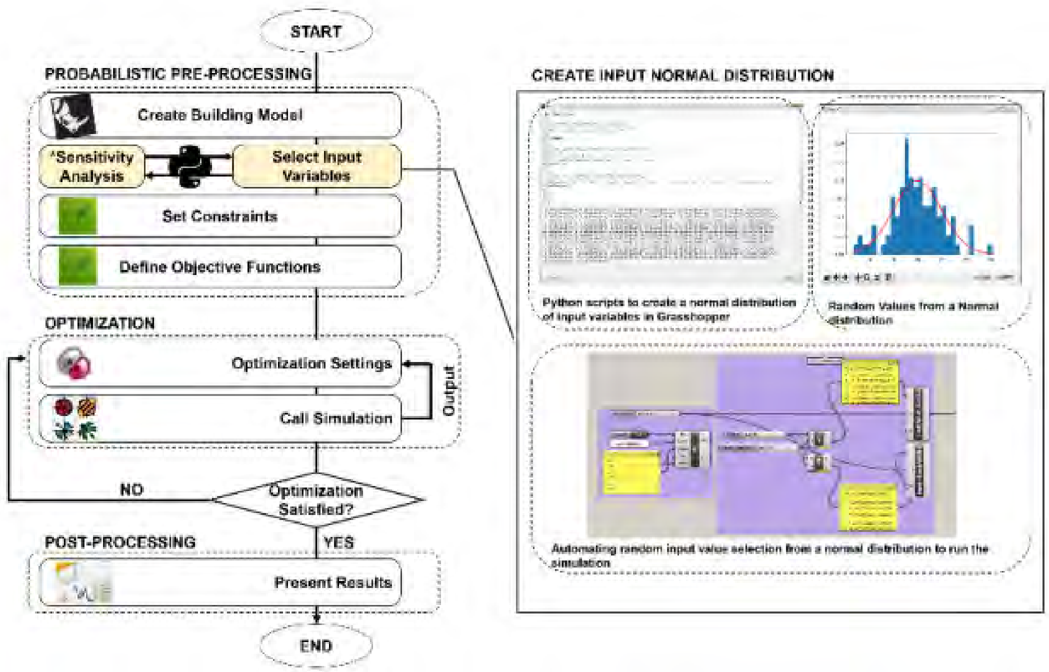


Figure 4: Sensitivity-based building design optimization by Shahsavari et al. (2019)

The choice of SA methods for BEPS is highly dependent on the objective of the research and the availability of the computational resources (Pang et al. 2020). In general, the accuracy of the results grows proportionally with the complexity and number of model evaluations of the SA methods. The LSA method is generally not recommended for BPA due to the high nonlinear nature of a building system (Pang et al. 2020). Alternatively, LSA method can be utilized to reduce the size of parameter set prior to more sophisticated GSA methods (Kristensen and Petersen 2016). On the other hand, GSA methods, in particular the screening-based approach by Morris and the variance-based approach by Sobol, are commonly used in BEM-based analysis.

The global screening-based approach by Morris is regarded as the extension of the local OAT approach (Pang et al. 2020). Instead of varying only one parameter at a time as in the LSA method, the global OAT method varies all the input parameters consecutively in one iteration. The influence of a given parameter is quantified by applying the absolute mean of a population of elementary effects. However, due to the OAT nature of the method, the screening method potentially disregards the individual interaction between two input parameters.

The global variance-based Sobol' method provide a comprehensive description of the model behavior by decomposing the variance of the outputs into a sum of contributions by the inputs (Kristensen and Petersen 2016). The contribution of the input parameter determined by the Sobol' method includes the interactions between input parameters and variance caused by such interactions. In addition, the global variance-based approach is model-independent, which neglects the assumption between the input and output (Pang et al. 2020). However, the relative computational time of Sobol' method is significantly higher than that of LSA and the Morris method (Kristensen and Petersen 2016).

Figure 5 illustrates a decision diagram for the selection of SA methods for BPA. The Sobol' method is recommended in case of established continuous or discrete range and shape of the parameter distributions (Kristensen and Petersen 2016). However, when the variation of input parameters is uniformly distributed between chosen boundaries, both the Sobol' and Morris method showed the same result. Thus, the less computational challenging Morris method is recommended for such case (Kristensen and Petersen 2016). In general, the Sobol's method showed higher robustness than the Morris method when ranking parameters in terms of importance to the model output. (Pang et al. 2020).

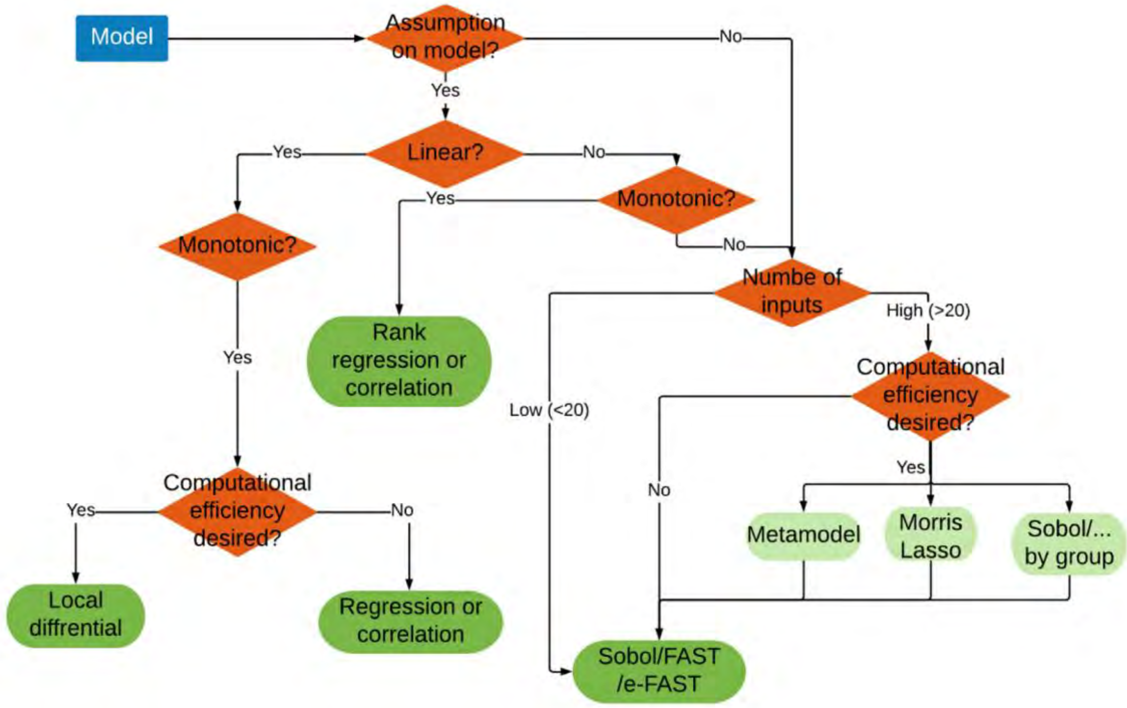


Figure 5: Decision diagram for selection of SA methods by Pang et al. (2020)

In addition, Pang et al. (2020) discussed strategies to increase the computational efficiency for variance-based SA in case of large input parameter space. These strategies include replacing the whole-building energy simulation with a metamodel, preliminarily identifying and eliminating noninfluential parameters with screening-based method, and dividing the input space into sub-spaces to perform SA by group.

The previous investigations of Morris method for BEM-based analysis include identification of influential parameters for energy efficiency building design in different climates (Maučec et al. 2021), reducing the number of uncertain parameters in the early design stage (Østergård et al. 2016), and evaluation of influence of building geometry on building energy use (Hemsath and Bandhosseini 2015). The use of Sobol' method for BEM-based analysis is less common than Morris method. In previous studies, the Sobol' method is utilized alongside the Morris method to compare the results of the two GSA techniques for better validation, as described in Kristensen and Petersen (2016), Neale et al. (2022), and Nouri (2023). A workflow for identifying influential and non-influential parameters for a building involving both Morris and Sobol' method is described in Figure 6 (Neale et al. 2022).

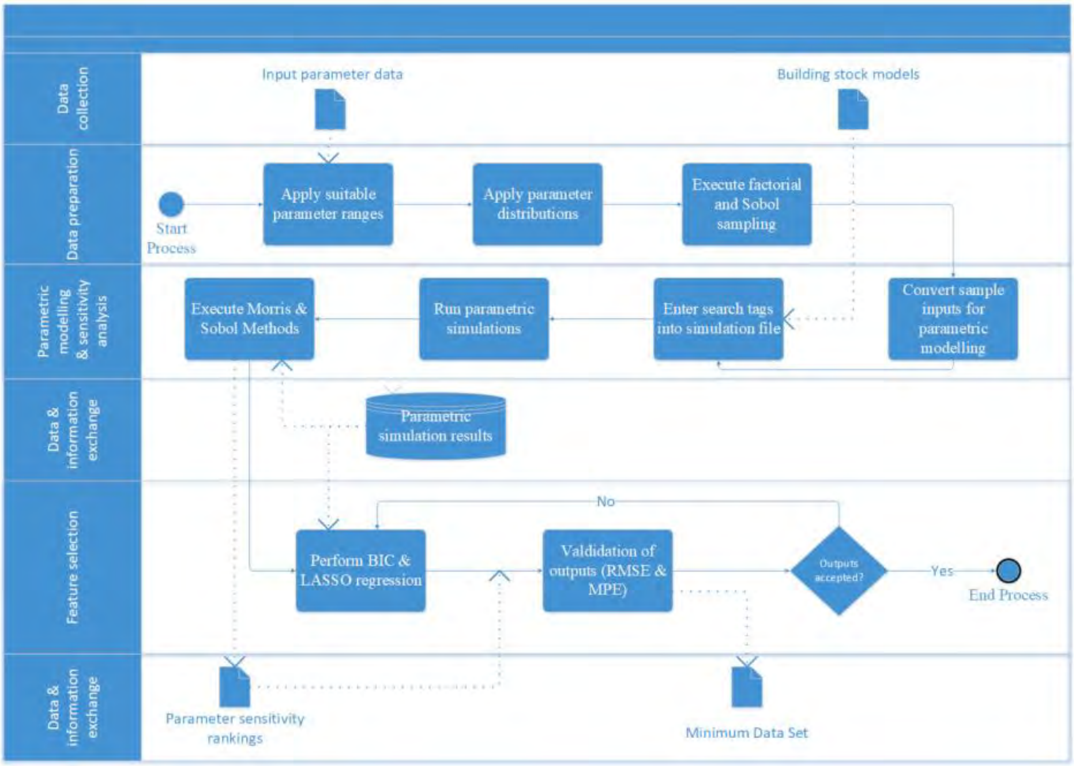


Figure 6: Identification of influential and non-influential parameters for a building process by Neale et al. (2022)

2.7 Uncertainty of BEPS Parameters

The improvement in the accuracy and detail of building performance simulation consequentially increases the number of inputs involved in the analysis. At the early design stage, the uncertainty in the input parameters remains the major cause for the prediction gap in the result of BPS (Singh et al. 2020). In previous studies of SA/UA in the early design stages, researchers tried different approaches to define and categorize the input parameters. The choice of input parameters for building performance analysis varies based on the objectives of the research.

For SA/UA investigation, a Building Information Model adopting the LOD approach provides related input parameters with uncertainties for calculations. Harter et al. (2020) suggested defining additional project-specific parameters and integrate such parameters into the BDL 2, 3, and 4 for analysis of early design stages. A value indicating the range for generating the sample set is assigned to each parameter group to represent the uncertainty at each BDL. For example, the uncertainty of design parameters at BDL 2 to 4 are defined as showed in Table 1.

Table 1: Specific Definition of Uncertainty of LCEA-Parameters at each BDL by (Harter et al. 2020)

Group	Uncertainty		
	BDL 2	BDL 3	BDL 4
Geometrical	±10%	±2%	±1%
Technical Specifications	±25%	±25%	±2%
Window Constructions	±25%	±25%	±25%
Building Operation	±5%	±5%	±5%
System Efficiency	±5%	±5%	±5%

Singh and Geyer (2019) utilized a parameter space for an investigation of multi-LOD BIM using variance-based sensitivity analysis for energy performance. The multi-LOD approach defines design parameters in the beginning and focuses on a few parameters at each LOD. The design parameters are categorized into five groups, namely Geometrical, Technical Specifications, Window Construction, Operational Design, and System Efficiency. A similar categorization of design parameters with specific definition of Life Cycle Energy Assessment (LCEA) parameters was demonstrated in the uncertainty analysis of LCEA in early design stages by Harter et al. (2020). For the UA of embedded energy and greenhouse gas, Schneider-Marin et al. (2020) utilized a reduced set of parameters with four categories excluding the Operational Design and System Efficiency. A mean value of a defined minimum and maximum is assigned to each design parameter.

In previous BPA studies, parameter groups considering the behavior of occupants and the thermal properties of the building components showed the strongest correlation with each other (Pang et al. 2020). The design parameters referring to boundaries conditions and usage scenarios which remain uncertain in post-design phase, such as heat gain from light and equipment, operating hours, occupant load, etc., are called scenario parameters. Besides, such parameters represent the lack of information in design features that have yet to be settled, for example U-values of different building components, G-values of windows, air tightness, and internal mass, are called undecided design parameters (Singh et al. 2020). The common parameters group in previously mentioned studies is summarized in Appendix A.

The parameters in the Geometrical group describe the uncertainty in the building shape at early design stages of design. To represent this uncertainty, researchers in previous study implement from five (Singh et al. 2020) up to seven (Harter et al. 2020) design options for building shape. For small and medium-size office building, the most common building shapes include rectangular, plus-shape, L-shape, U-shape, H-shape, and T-shape (Singh and Geyer 2019). To provide comparable results, the geometrical parameters are constrained to keep the floor area of all shapes constant throughout the BDLs (Singh and Geyer 2019).

3 Methodology

3.1 Research Gap

An efficient BIM-based energy simulation in early design stages requires a solution to quickly generate BEM models from the geometrical and semantic data retrieved from BIM models. Thus, the first challenge is how to establish a robust and fast BIM-BEM data exchange framework. In addition, the minimal requirement for the early design BIM models to secure a detailed energy simulation must be addressed. As discussed in Section 2.6, the results of the SA methods in building performance simulation serve as the reference for model optimization and calibration. Hence, another challenge is to interpret the SA results to support model optimization.

3.2 Research Method

To answer the challenges stated in the problem formulation, a 3-step workflow is proposed. An overview of the research method is presented in Figure 10. Step 1 discusses existing workflows for BIM-based BEM and SA for BEPS to formulate a general framework for parametric BIM-based building energy performance simulation with sensitivity analysis for office buildings. Based on the proposed general framework in Step 1, Step 2 performs a prototypical software implementation with discussions on coupling tools selection. The prototypical implementation is evaluated through a case study in step 3. Eventually, the result of the case study is validated to give feedback and outlook on the prototypical software implementation for future improvements.

The two sub-steps of Step 1, Discussions of Existing Workflows and Proposed Framework, are described in Section 3.3 and 3.4 respectively. Details of the proposed framework is illustrated in Section 3.5, 3.6, and 3.7. Subsequently, the Step 2, Prototypical Implementation, is depicted in Chapter 4. Step 3 of the method, Case Study & Results, is illustrated in Chapter 5.

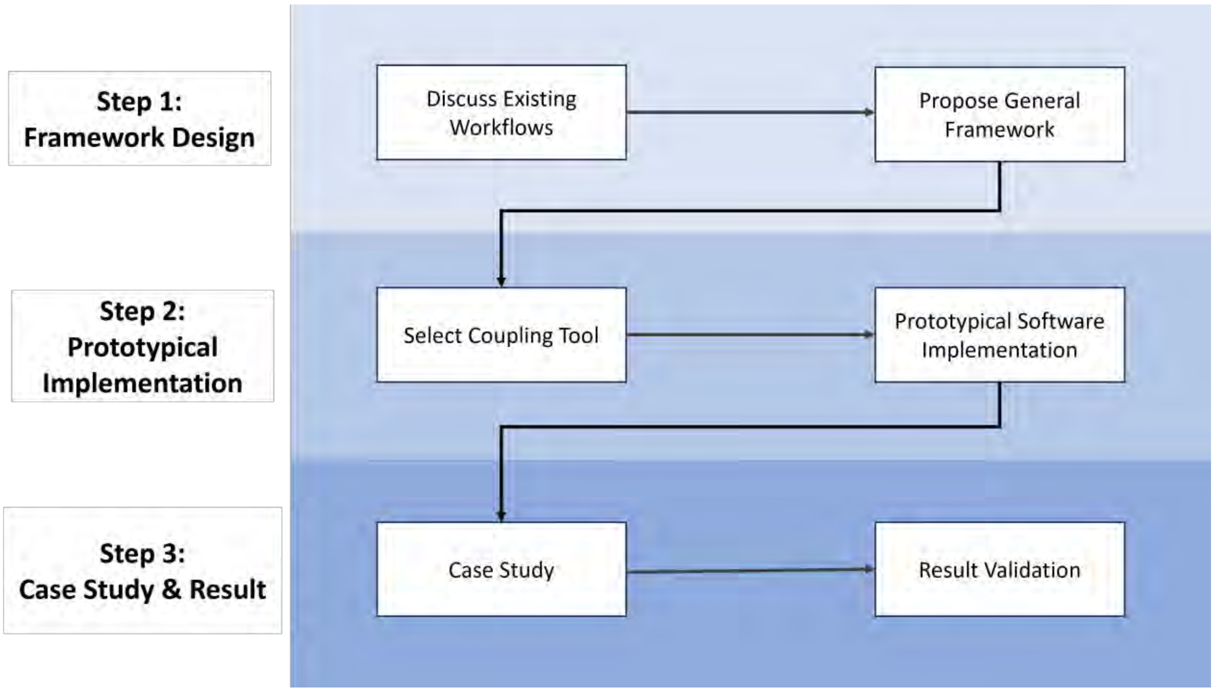


Figure 7: Research method outline (own illustration)

3.3 Discussion on Existing Workflows

3.3.1 BIM-based BEM

One major advantage of BIM-based BEM workflow is the automated geometry transformation, instead of manual iterative geometry fix in the conventional workflow (Kamel and Memari 2019). Ideally, the fully automated BIM-based BEM workflow (see Figure 8) should obtain almost all the required data from the BIM model and quickly perform simulation on the obtained data. However, the existing challenges on BIM-BEM interoperability significantly hindered the development of the fully automated BIM-based BEM workflow. Indeed, a semi-automated BIM-BEM workflow is rather realistic. In fact, the workflows with the semi-automated provides more control over the generation of BEM models with significantly less effort compared to the manual BEM model generation.

As described in Section 2.3, the two most used BIM-BEM transformation approaches are explicit data exchange using vendor neutral, open-source exchange formats and the implicit data exchange using the API of the BIM authoring tool. The explicit data exchange approach took prevalence in previous research. The typical BIM-BEM exchange procedure with open exchange formats involves three major components, namely the BIM authoring tool, the model schema exchange format, and the BEM software. IFC and gbXML are the most frequently used building information exchanged formats for energy simulations in the BEM software. In addition, previous workflows with open data exchange formats often implemented the corrective middleware tools to support BIM-BEM transformation (Kamel and Memari 2019). For instance, the previously developed and implemented middleware tools in existing literature include Space Boundary Tool (Lawrence Berkeley National Laboratory 2013), OBES (Choi et al. 2016), CBIP via SimModel (Giannakis et al. 2020), OsmSerializer (Ramaji et al. 2020), and IfcOpenShell-python (IfcOpenShell Contributors 2023).

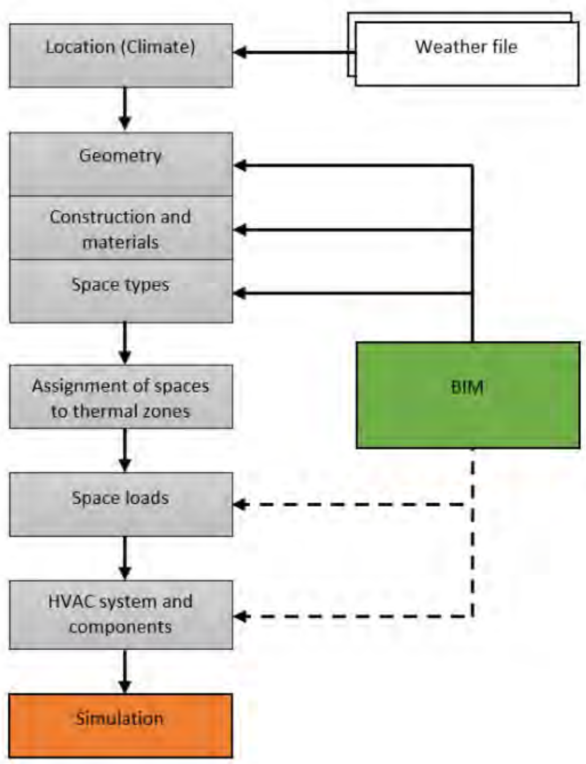


Figure 8: Ideal BIM-based BEM workflow (Maile T. et al. 2007)

Apart from the explicit file exchange via vendor neutral, open-source data exchange formats, implicit access to BIM data with the API of the respective BIM authoring tool is possible (Andriamamonjy et al. 2018). The implicit data exchange approach utilizes plugins within the BIM authoring tools to handle the BIM-BEM data exchange. In contrast to the availability of literature on explicit file exchange via open-source data schema, the implementation of implicit exchange approach in existing literature is significantly less. Considering the data exchange from Autodesk Revit, three plugins have the potential to perform the data exchange to subsequent energy simulation tool, namely Rhino.Inside.Revit (Robert McNeel & Associates 2023), Pollination’s Revit plugins (Pollination 2023), and the Data Exchange Connector apps (Videau

2023). Since the Data Exchange Connector is currently in the beta testing, Rhino.Inside.Revit and Pollination are more suitable for implementation.

3.3.2 SA for BEPS

A variety of software packages written in different programming languages are available to perform SA (see Table 2). Considering open-source tools for SA for BPS, Pang et al. (2020) mentioned the “sensitivity” package in R platform, the SIMLAB, the SALib Python package, the OpenTURNS C++ software, and the SAFE toolbox. As of 2023, the SIMLAB is unavailable for public downloading (EU Science Hub 2023). Among the other available software, the “sensitivity” package in R platform by Iooss et al. (2023) is considered the most comprehensive package which support a wide variety of SA methods investigations. For Python users, the SALib package and OpenTURNS software could be utilized. Considering MATLAB, the SAFE toolbox by Pianosi et al. (2015) provides a non-specialist friendly approach to SA.

Table 2: Categorization of SA package based on language (adapted from Pang et al. (2020))

Language	Package name	SA methods	Reference
R	sensitivity multisensi sensobol	LSA/GSA GSA GSA	Iooss et al. (2023) Bidot et al. (2022) Puy et al. (2021)
Python	SALib sensitivity OpenTURNS	LSA/GSA LSA LSA/GSA	Iwanaga et al. (2022) DeRobertis (2022) Baudin et al. (2017)
MATLAB	GSA Toolbox GSUA Toolbox UQlab SAFE Toolbox	GSA LSA/GSA LSA/GSA Regional/GSA	flax (2023) Carlos and Velez (2023) Marelli and Sudret (2014) Pianosi et al. (2015)
C, C++, C#	Dakota SobolGSA PSUADE VARS-Tool	GSA GSA GSA GSA	Adams et al. (2020) Kucherenko and Zaccheus (2020) Tong (2015) Razavi et al. (2019)

As discussed in Section 2.6, a typical workflow for SA in BEPS illustrated by Pang et al. (2020) utilizes the SA tools in the last step to perform SA on the generated inputs and outputs from the BPA software. In other words, the sampling input parameters to generate model inputs is executed within the BEPS tool. However, it is possible to perform the sampling of input parameters with the SA tool. In fact, most of the listed SA packages provides built-in functions to perform different sampling methods, such as factorial sampling, LHS techniques, Sobol' sequence, etc. Subsequently, the generated model inputs by the SA tool are imported into the BEPS tool to serve as inputs for parametric simulation. The advantages of generating model inputs using built-in sampling functions of SA tool are the wider choices and customization of the sampling techniques. In contrast, the sampling of input parameters within the BEPS tool requires manual work from the users. The two workflows for SA in BEPS with different input parameters sampling strategies are illustrated in Figure 9.

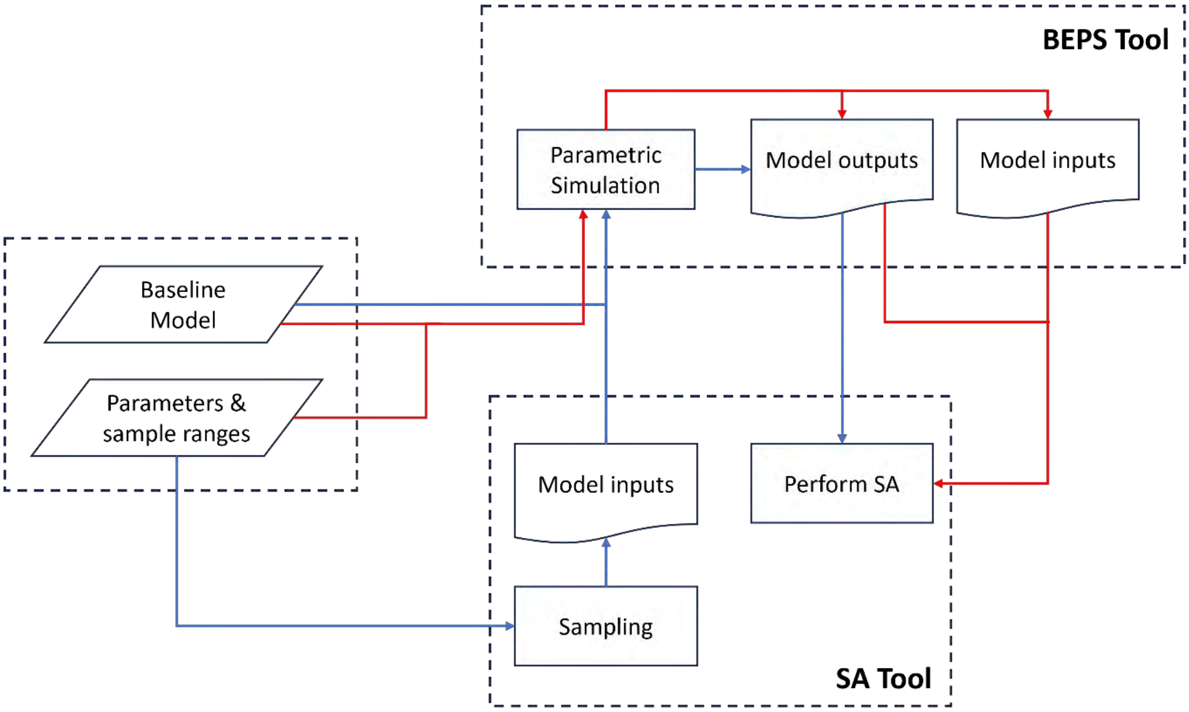


Figure 9: SA in BEPS workflow with different sampling strategies: sampling using the SA tool (blue) and BEPS tool (red) (own illustration)

3.3.3 Energy Simulation Tools

As discussed in Section 2.6, the general SA workflow involves adding uncertainties into a baseline model to generate multiple sample variations for subsequent energy simulation with a simulation engine. Considering the highly repetitive and computationally intensive demands of the process, the simulation engine used in SA ideally possesses the ability to create BEM models in batch and efficiently run simulation in parallel. In addition, the use of programming language or added-in parametric analysis function would be beneficial to support the parametrization of the BEM model.

Considering the previously mentioned prerequisite for simulation engines, EnergyPlus and OpenStudio show significant potential among other simulation tools in the market such as TRNSYS, eQUEST, ESP-r, etc. EnergyPlus is an open-source whole-building energy simulation engine that supports energy analysis and thermal load simulation of both residential and commercial buildings (National Renewable Energy Laboratory). Batch simulations are supported in EnergyPlus using the built-in EP-Launch component. One notable feature of EnergyPlus is the availability of programmable external interface for modeling control sequences and interfacing with other analyses. On the other hand, OpenStudio is an open-source software development kit (SDK) which provides an API to access the EnergyPlus modelling engine (National Renewable Energy Laboratory 2023b). The main objective of OpenStudio is to support the growth of end-user BEM tools ecosystem by reducing the required effort to maintain BEM tools. In addition, the Parametric Analysis Tool of OpenStudio supports parametric SA and parallel batch running either locally or on cloud server.

EnergyPlus and Openstudio served as the base simulation engine for various public and private-sector end-user applications, notably the Honeybee energy modeling plug-in. Particularly, the Honeybee plugin links the Grasshopper VPL to EnergyPlus/OpenStudio simulation engine to create, run, and visualize energy analysis. As discussed in Subsection 3.3.1, the implicit BIM-BEM data exchange approach involves the API of the respective BIM software. In this context, the use of a VPL to handle the BIM-BEM data exchange is more friendly towards novice and non-programmers than the use of conventional textual programming language. Hence, a cohesive workflow for BIM-based BEM would be achieved by utilizing one common VPL for both data exchange and perform energy simulation.

3.4 Proposed Framework

The proposed general framework for parametric BIM-based BEPS with SA for office buildings is illustrated in Figure 10. The three main components within the framework are Pre-processing, Parametric Modelling and Simulation, and Sensitivity Analysis. The initial step of the framework, Pre-processing, aims to define relevant data for subsequent steps. The second part of the framework, Parametric Modelling and Simulation, handles the BIM-BEM data exchange and energy simulation. In the final step, Sensitivity Analysis, a SA software performs sensitivity analysis on relevant inputs and interpret the results.

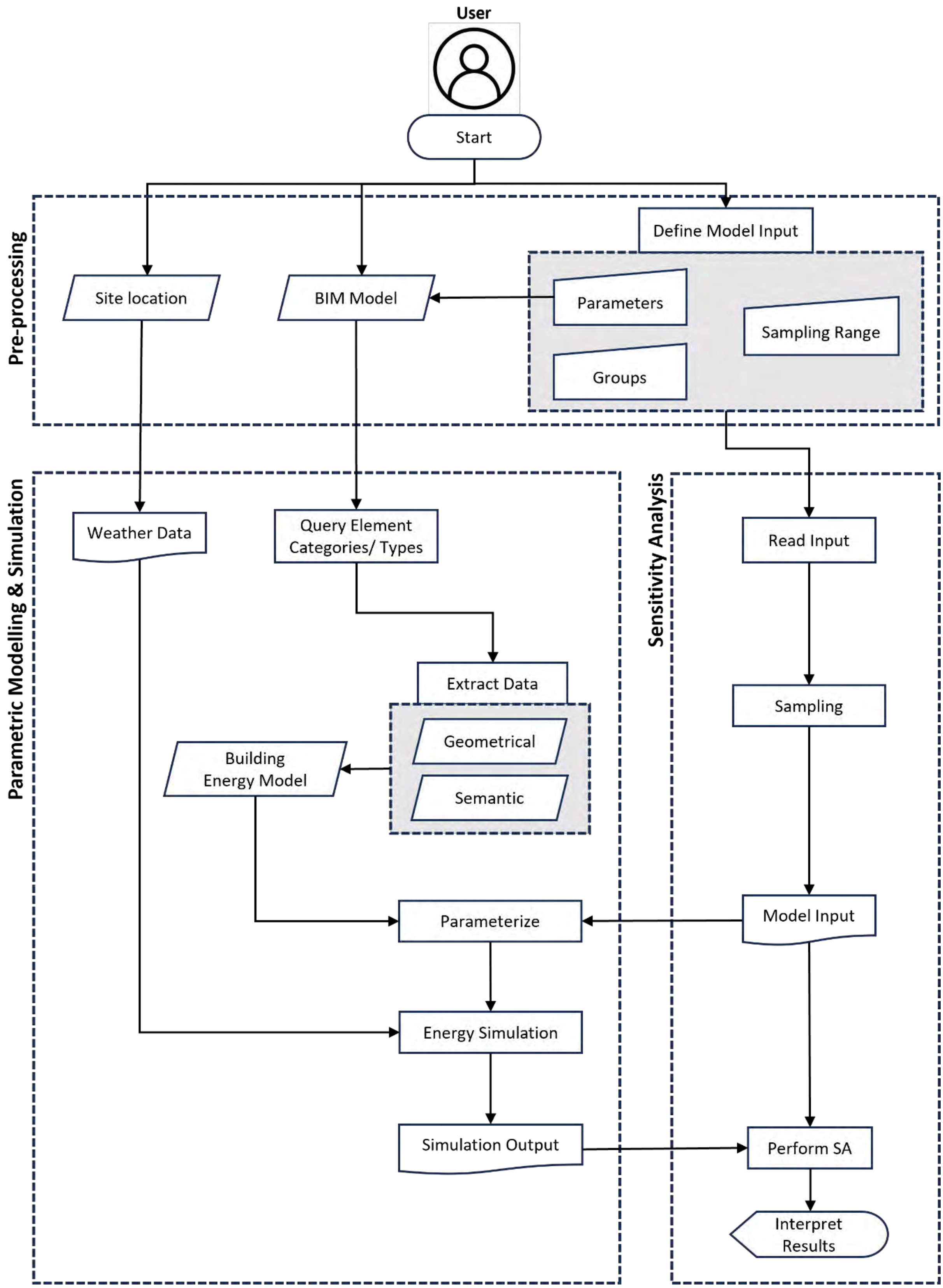


Figure 10: General framework for parametric BIM-based building energy performance simulation with sensitivity analysis for office buildings

3.5 Pre-processing

Pre-processing is the initial process in the proposed method. The major outputs of this process include BIM models to serve as the core data models for the building energy performance simulation, a set of user-defined input parameters to represent the uncertainties of the early design stages, and the site location to determine representative weather data.

3.5.1 Generate BIM Models

In this study, BIM models are modeled to serve as the core data models for the building energy performance simulation. The main purpose of utilizing BIM model is to achieve quick and consistent transfer of geometric data with minimal energy-related information. In addition, by applying the BDL concept, BIM models at different BDLs provide related input parameters with uncertainties for generating model inputs for the subsequent SA. As mentioned in Section 2.1, BIM models at BDL 1 to 4 represent the early design stages.

Considering the proposed general framework, the generation of BIM models must consider certain aspects. First, since the implicit BIM-BEM data exchange involves the API of the respective BIM authoring tool, the utilization of a BIM authoring tool with a comprehensive and well-documented SDK to support working with the API is beneficial. Second, the BIM-BEM transformation is the main consideration when generating the BIM models. Consequently, a set of prerequisites considering potential errors must be set to secure a smooth BIM-BEM conversion. Finally, the components within the BIM model are modeled with LODs requirement derived from the respective BDL of the BIM model.

3.5.2 Define Input Parameters

The set of relevant input parameters for the calculation of annual energy demand is defined and categorized into four groups, namely Geometrical, Technical, Window, and System. The parameter groups are defined with respect to the design process to represent the decisions consecutively made by designers in the design stages. First, the group Geometrical represents the uncertainties in building dimensions at an early stage of designs. Parameters within the Technical group describe the structural requirements of the building elements. The group Windows includes parameters concerning the openings on exterior walls. Finally, the System group represents the efficiency of the building energy system. The interested input parameters in their respective groups in this study are presented in Table 3.

Table 3: Input parameter groups for annual energy demand calculations (own illustration)

Group	Parameters	Symbol [unit]	Deviation at BDL 3	Deviation at BDL 4
Geometrical	Building Length Building Width Building Height	L [m] W [m] H [m]	±2%	±1%
Technical	External Wall Thickness External Roof Thickness	T_ExWa [mm] T_ExRo [mm]	±25%	±2%
Windows	Window Area Window-to-Wall Ratio Infiltration	Wi_A [m^2] WWR [%] Infil [-]	±25%	±20%
System	Cooling COP Heating COP	C_COP [-] H_COP [-]	±5%	±5%

A mean value and percentage of possible deviation are assigned to each parameter within a parameter group. The mean values of input parameters within the Geometrical, Technical, and partly from Windows groups are extracted directly from the BIM models. On the other hand, mean values of parameters within the System group are consulted from the IECC Scorecard for the Prototype Building Models provided by the U.S. Building Energy Codes Program (2023). The possible deviation value represents a range of uncertainty with a uniform distribution. Parameters within the same group share a common possible deviation value. In addition, the concept of BDL (Abualdenien and Borrmann 2019) is applied to specify the uncertainty of each parameter group at each design stages. As shown in Table 3, the uncertainty of each parameter group represents the required maturity of the BIM model at the respective BDL. Subsequently, the input parameters and their deviation are sampled by SA software to generate model inputs.

3.6 Parametric Modeling & Simulation

The Parametric Modeling and Simulation process of the framework primarily handles the BIM-BEM data exchange, the parametrization of BEM models, and building energy simulation within the BEM software environment.

3.6.1 Weather Data

The appropriate choice of weather data has significant impact on the energy performance of buildings (Moradi et al. 2023). For EnergyPlus-based energy simulation with the HoneyBee plugin, the EnergyPlus Weather Format (EPW) is adopted as the standard format for weather data. The EPW files consist of 8,760 values for each meteorological parameter included in a Typical Meteorological Year dataset, which represents the number of hours in a year (Ladybug Tools LLC 2023). Meteorological parameters stored in the EPW file include hourly values of weather variables and solar radiation related variables for the Typical Meteorological Year (Maučec et al. 2021). Weather files from different sources is combined and represent explicitly in the Ladybug Tool EPW Map (Ladybug Tools LLC 2023). In practice, the Ladybug Tool EPW Map provides designers with different weather dataset of the interested location to choose based on the type of study.

3.6.2 BIM-BEM Export

In this study, the implicit data exchange approach is applied to handle the BIM-BEM transformation. Particularly, the geometrical and semantic data of the BIM models is exported into a BEM software environment using available addon or plugin. Subsequently, the extracted data forms the required geometry and semantic inputs to create detailed BEM model. A VPL is utilized to support the data exchange and BEM model generation process.

As described in Section 2.3, it is beneficial to simplify the BIM model by refining the building components and eliminating insignificant data prior to the data exchange. Hence, an optimized BIM model containing only necessary data for the subsequent simulation with the BEM tool is obtained. Following the same concept, this study seeks to export only relevant data for quickly creating a detailed BEM model. Particularly, geometrical data considering the volumes created in the BIM model forms the basis for the geometry of BEM model. On the other hand, semantic data regarding element thermal properties and building materials in the BIM model is utilized to generate construction set in the BEM model.

3.6.3 Parametrization of BEM Model

As discussed in Subsection 3.3.3, the same VPL used for BIM-BEM data exchange is utilized to manipulate the parametrization process. Particularly, the model inputs generated from the SA sampling method are imported into the BEM software environment in a text file format. Next, the imported model inputs serve as the values set to parametrize the BEM model. Eventually, the parametrization process aims to generate a batch of BEM models for consequent energy simulation.

3.6.4 Energy Simulation

In this study, an energy simulation to estimate the building end use intensity (EUI) is of interest. The EUI indicates annual total energy use of a building as a function of its gross floor area. Generally, a high EUI signifies low energy performance. In addition, the EUI could be divided into specific end uses, such as cooling, heating, etc. The batch of BEM models and the EPW weather data are the main input to run the energy simulation. The parametrized BEM models are expected to provide sufficient inputs to secure a detailed energy simulation. Otherwise, necessary parameters must be added manually. Eventually, the simulation result is recorded as the model output to perform SA.

3.7 Sensitivity Analysis

The final part of the framework involves the implementation of SA methods to analyze the results of the previous energy simulation.

3.7.1 Selection of SA Approaches

As discussed in Section 2.6, the choice of SA methods for BEPS highly depends on the objective of the research and the available computational resources. With the assumption of a nonlinear and nonmonotonic model representing the typical building system, the choice of SA methods relies on the number of inputs. For an analysis of less than 20 inputs, the Sobol', FAST, and e-FAST method are recommended (Pang et al. 2020). In contrast, the use of the Sobol', FAST, and e-FAST method for analysis of more than 20 inputs only if the computational resource is significant. In the case of analysis of more than 20 inputs with limited computational resource, the method of Morris is generally recommended.

Considering the number of inputs previously defined in Subsection 3.5.2, analysis using the Sobol' method could be implemented for the proposed framework. However, due to desired computational efficiency, the Morris method would be more beneficial.

3.7.2 Morris Method

The global OAT screen-based method by Morris (1991) utilizes an elementary effect to evaluate the influence of individual input parameters. An input space Ω consists of k input parameters x_i is defined in the Morris method. Subsequently, the range of each parameter is divided in p levels with a distance Δ between them. The definition of the elementary effect is described in Equation 1, where a k-dimensional input space established by k independent input parameters x_i is discretized into a p-level grid Ω.

$$EE_i = \frac{y(x_1, \dots, x_{1-1}, x_i + \Delta, \dots, x_k) - y(x_1, \dots, x_k)}{\Delta} \quad (1)$$

The two sensitivity measures introduced in the Morris method are the mean value μ_i and the standard deviation σ_i, as described in Equation 2 and Equation 3 respectively. The mean value μ_i assesses the overall influence of the input parameter x_i on the output. Hence, the mean value is utilized to rank the input parameters in order of importance. On the other hand, the standard deviation σ_i estimates the collective effects of an input due to nonlinearity and/or interactions with other inputs.

$$\mu_i = \frac{1}{r}\sum_{t=1}^{r} EE_{i,t} \quad (2)$$

$$\sigma_i = \sqrt{\frac{1}{r-1}\sum_{t=1}^{r} |EE_{i,t} - \mu_i|^2} \quad (3)$$

In addition, Campolongo et al. (2007) proposed an enhanced method to compute the mean value by using absolute average of elementary effects, denoting as μ_i^*. The use of μ_i^* resolved the Type II error - fail to identify influential input, when using μ_i. The definition of μ_i^* is depicted in Equation 4.

$$\mu_i^* = \frac{1}{r}\sum_{t=1}^{r} |EE_{i,t}| \quad (4)$$

3.7.3 Interpretation of Morris Method's Results

The evaluation of the arithmetic mean μ_i, the standard deviation σ_i, and the absolute mean μ_i^* over a considerable number of trajectories r produces the global sensitivity measures of the importance of the i-th parameter. Morris (1991) suggested three possible categories of parameter importance, namely non-influential, influential with linear and/or additive effects, and influential with non-linear and/or interaction effects. The visualization of conventional parameter classification using the σ_i - μ_i^* plane is illustrated in Figure 11.

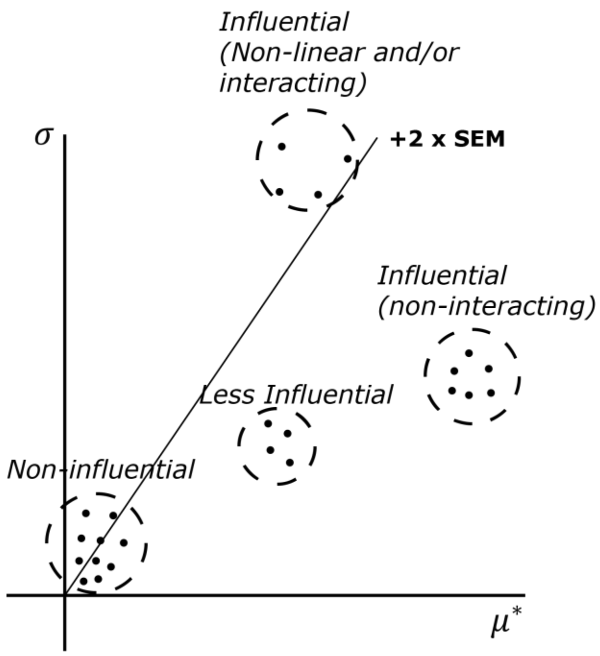


Figure 11: Visual inspection of the elementary effect statistics on the σ_i - μ_i^* plane (Wicaksono 2016)

The relative locations of the parameters' statistics on the σ_i - μ_i^* plane signify the assumption of influential and non-influential parameters. Generally, the statistics of influential parameters are grouped with pronounced boundary further from the origin of the σ_i - μ_i^* plane than those of non-influential. In the case of uniformly spreading statistics across the plane, the differentiation between influential and non-influential becomes ambiguous. Moreover, the non-linearity effects from parameter interactions on the output is indistinguishable for a parameter with large μ_i and σ_i.

4 Prototypical Implementation

4.1 Requirements for Generating the BIM Model

The foreseen BIM to BEM conversion serves as the main consideration to generate BIM models from the beginning of the design process. Error-prone details that affect the later parametric energy simulations must be attended to ensure a smooth workflow. In this study, the requirements for the generation of BIM model in Autodesk Revit are set.

First, the Room Separators are used to form the room boundaries instead of other room bounding elements, such as walls and floors. Subsequently, the “Room Bounding” parameter of room bounding elements within the model is unchecked. Moreover, all wall elements are created with location lines lying in the respective Finish Face: Interior planes. For example, Figure 12 shows the properties of an exterior wall with the required “Location Line” and “Room Bounding” parameter. The reason for the requirement of room separators and wall properties is to ensure the coplanarity between the room faces and the interior faces of the wall elements. Alternatively, designers must choose to compute the room area at wall finish if wall elements were implemented as the room bounding elements instead of separator lines (see Figure 12).

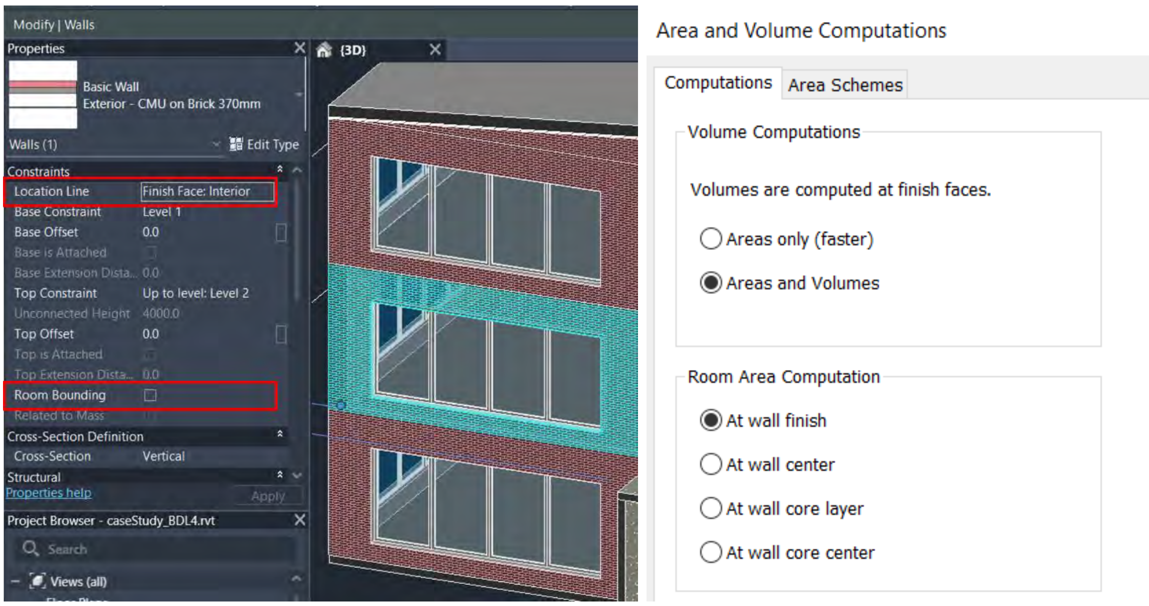


Figure 12: Requirement when creating a Wall element (left) and area & volume computations (right) (own illustration)

Second, the name of the rooms within the BIM model must be derived from a building program provided in Honeybee energy standard library. A list of room program identifiers based on the selected building program and building vintage would be obtained by using the “HB Search Programs” component. For instance, the list of room program identifiers that meet the requirement for the “Medium Office” building program as described in the Commercial Prototype Building Models provided by the U.S. Building Energy Codes Program (2023) is showed in Figure 13. This specific definition of room names is the key factor to achieve the automated semantic enrichment of the whole BEM model using the BIM model in early stages as described by Forth (2023).

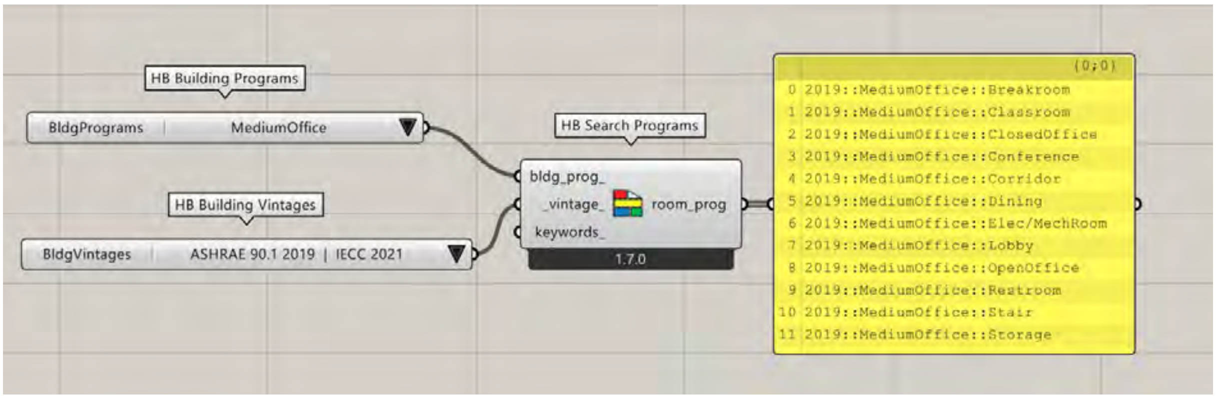


Figure 13: List of room programs based on building program and building vintage with the "HB Search Programs" component (own illustration)

In addition, all wall openings including doors, windows, or straight rectangular openings must have a base offset of at least 100 mm. The lack of a base offset for wall openings would result in errors when creating Honeybee rooms due to unmatched subfaces and parent faces.

4.2 Coupling Tools Selection

Two possible workflows for accessing information from Revit BIM models and exporting geometry model to Rhinoceros using Revit API are demonstrated in Figure 14. The first workflow considers the Pollination Revit Plugin, which exports clean analytical models from Revit models for cloud simulation (Pollination 2023). The second workflow implements the Revit addon Rhino.Inside.Revit (R.I.R), which introduces the parametric modeling power of Rhino and Grasshopper into Revit environment (Robert McNeel & Associates 2023).

The Pollination Revit plugin provides automated generation of analytical model from Revit model with options to control geometrical properties, built-in repairing tools, and linked models support (Pollination 2023). The Pollination Revit plugin is capable of exporting analytical model in six data exchange formats, including HBJSON, Dragonfly Model (DFJSON), gbXML, Input Data File (IDF), OpenStudio Model (OSM), IES-VE Geometry (GEM), and eQuest Geometry (INP) (Pollination 2023). For simulations with HoneyBee in Rhino environment, the option of exporting HBJSON model would be most suitable in terms of interoperability. Pollination also provides cloud computing service to facilitate prolonged execution time of parametric environmental simulation (Roudsari 2021). In addition, most of the common simulation workflows are available in Pollination's cloud computing as validated recipes, which are a collection of reusable and customizable simulation workflows (Pollination 2023).

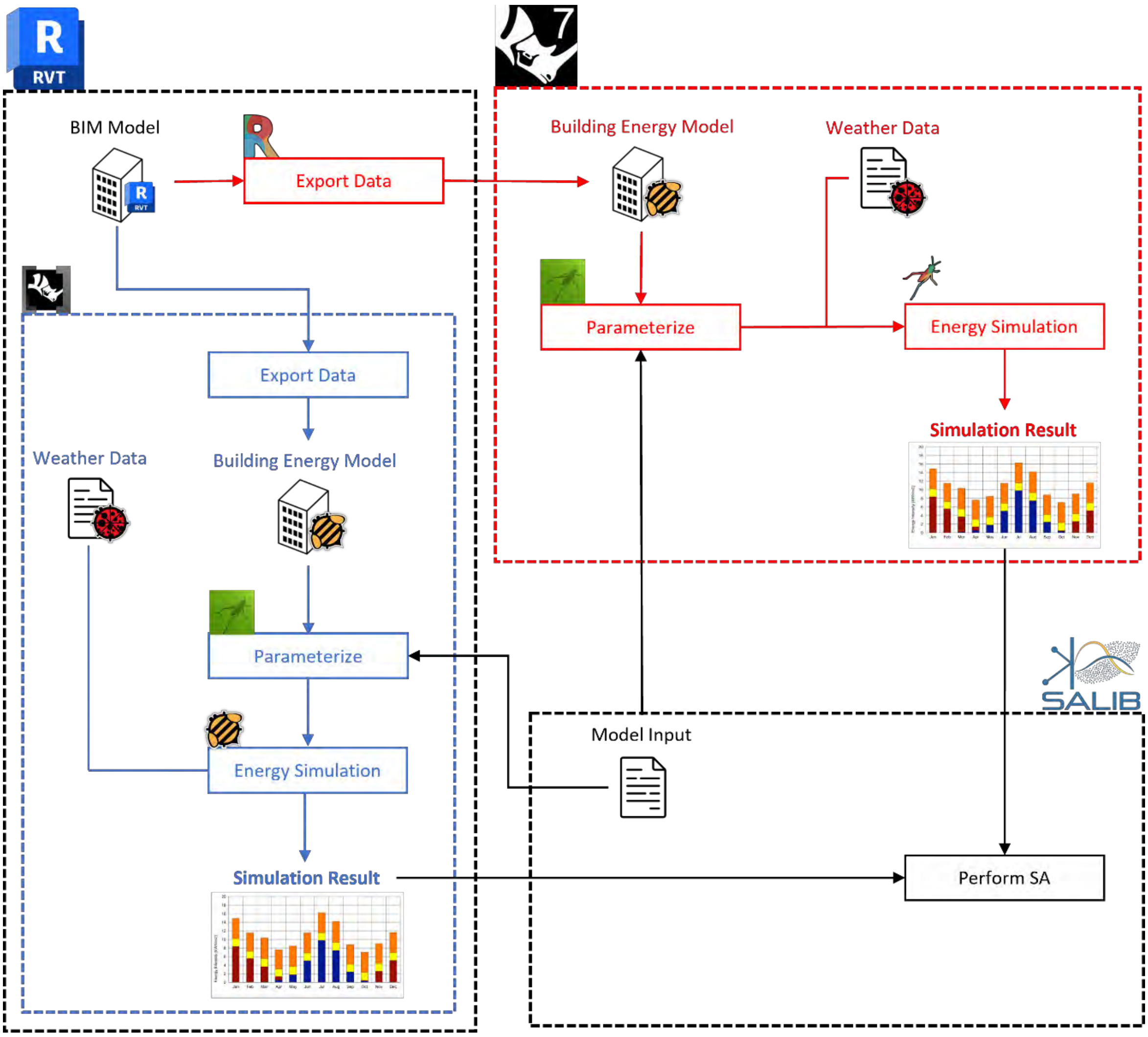


Figure 14: Two workflows for parametric energy simulation: with R.I.R in blue and with Pollination in red (own illustration)

The Rhino.Inside is an open-source project developed by Robert McNeel & Associates to embed Rhino and Grasshopper into other 64-bit Windows applications such as Revit or AutoCAD (Robert McNeel & Associates 2023). The R.I.R runs as a Revit addon that loads Rhino into Revit's memory space. One of the most important aspects of the R.I.R addon is the ability to extract geometry of Revit elements and create Revit-specific Grasshopper components for further processing (Robert McNeel & Associates 2023). The simulation results with R.I.R can be stored either locally or on cloud services using the Pollination Grasshopper plugin.

In this case study, the workflow with R.I.R is implemented. The key factor for the choice of R.I.R is the free-to-use and open-source nature of the addon. Consequently, the workflow would be more friendly towards further enhancements in future research.

4.3 BIM-BEM Exportation

In the scope of this thesis, R.I.R and Grasshopper are chosen to export data from the BIM model in Autodesk Revit and create a Honeybee model for energy simulation. In Autodesk Revit, individual elements, such as Wall or Floor, placed in a model are referred as Instances. Particularly, the Revit Instances are parametric objects which inherit a set of Parameters from their respective Category, Family and Type. In addition, an Instance optionally carries instance parameters belonging to only the specific instance. For example, the data structure of a wall element in Autodesk Revit is depicted in Figure 15.

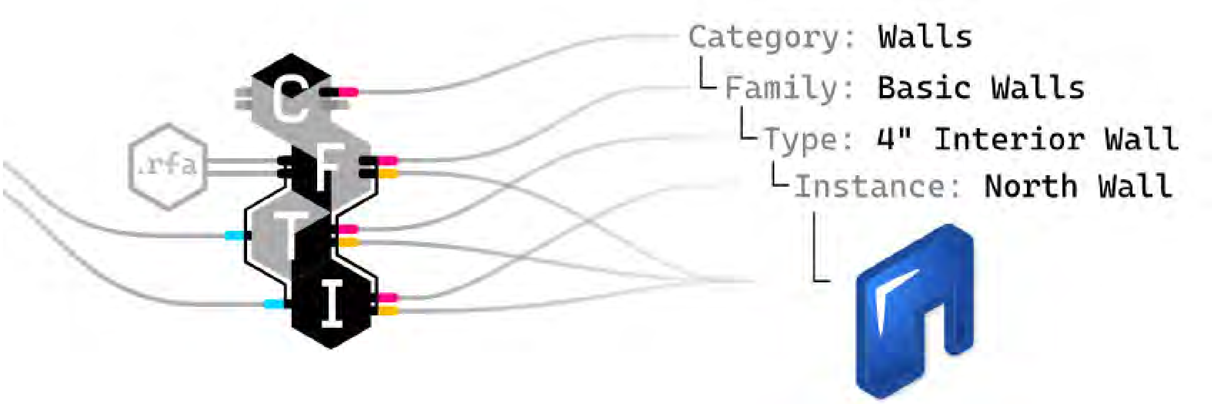


Figure 15: Data structure of a Wall element in Autodesk Revit (Robert McNeel & Associates 2023)

R.I.R provides various Grasshopper components to query and analyze Revit elements based on built-in categories and types. The combination of Grasshopper components used in this thesis to collect all Revit Instances of a specific Type in a Category is shown in Figure 16. For Wall, Ceiling, Floor, and Roof instances, it is required to select a specific Type from the respective Built-in Category before filtering and querying the desired instances. On the other hand, the Room can be queried directly without a specific filter. For Window and Door instances, the respective types are queried since the essential thermal properties of Revit windows and doors are stored as Type parameters instead of Instance parameters. Upon all the desired instances are queried, geometrical and semantic data are extracted to create the Honeybee BEM model.

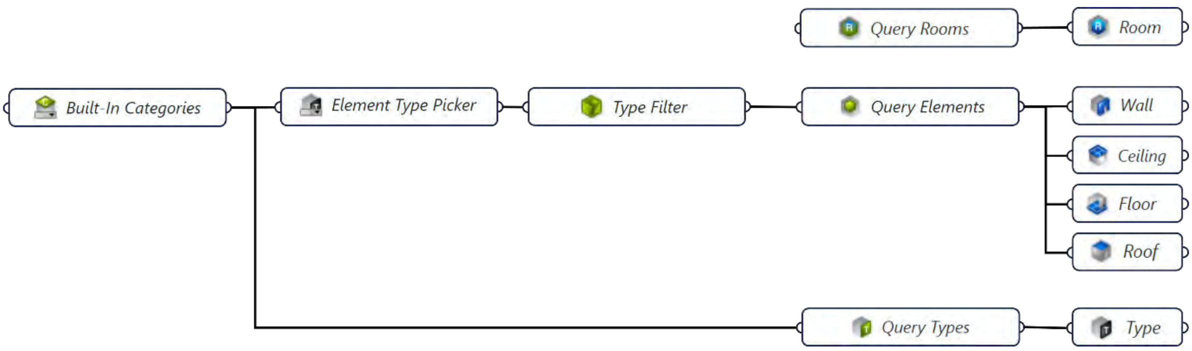


Figure 16: Workflow to retrieve Revit element based on Category and Type with R.I.R (own illustration)

4.3.1 Geometrical Data Extraction

The geometry of the queried Revit rooms formed the basis to create rooms in the Honeybee BEM model. R.I.R provides the “Spatial Element Geometry” component to extract the geometry of Revit spatial elements such as Rooms, Spaces, or Areas based on the boundary location of the spatial elements (Robert McNeel & Associates 2023). The geometry of each room defined in the Revit model is represented by a Closed Brep. Details of the “Spatial Element Geometry” component are shown in Figure 17.

Figure 17: Details of the "Analyze Spatial Element" component by R.I.R (own illustration)

The doors and apertures in the Honeybee model are created based on the geometry of the openings within the queried Revit wall instances. The process of extracting the geometry of Revit wall openings to create Honeybee windows and doors are illustrated in Figure 18. First, the queried Wall instances are decomposed into faces using the “Host Faces” component. As described in Section 4.1, the boundary of the Revit rooms and the interior faces of wall are co-planar due to the predefined requirement. Hence, only the set of Revit faces representing the interior wall faces is extracted. Second, the “Solid Difference” component was utilized for separating the geometry of wall openings from their respective host wall surfaces. The two input sets of Brep for the “Solid Difference” component are the set of interior wall faces extracted in the previous step and a set of plane surface constructed based on the bounding boxes of the interior wall faces. The output of the “Solid Difference” component returns a set of Trimmed Surface which could be used directly to represent the geometry of doors and apertures in the Honeybee model.

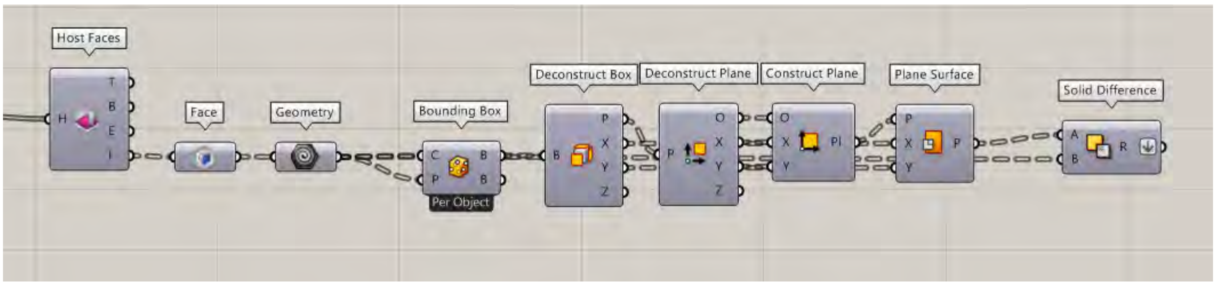


Figure 18: Wall openings geometry extraction process (own illustration)

4.3.2 Semantic Data Extraction

Semantic data considering the materials of the queried Revit instances and types are extracted to create different Honeybee materials for customized construction set for simulation. The method to extract thermal properties of queried instances using R.I.R components is described in Figure 19. For queried Room instances, the name of each room is extracted using the “Spatial Element Identity” component to serve as input to construct Honeybee rooms and assign building programs. R.I.R provides the “Analyze Basic Wall Type” and “Host Type Compound Structure” which return the definition of the compound structure of the given Wall and Ceiling, Floor, Roof type respectively. Subsequently, the compound structure definitions are deconstructed into layers, materials, and material’s assets. For Revit materials, relevant thermal properties for energy analysis are categorized into material’s thermal assets. Hence, the material’s thermal asset provides inputs to create Honeybee materials. Regarding the queried Window and Door types, the thermal properties of the respective types are extracted with the “Inspect Element” component. The handling of required material parameters to create opaque and window materials in Honeybee from Revit materials is shown in Table 4.

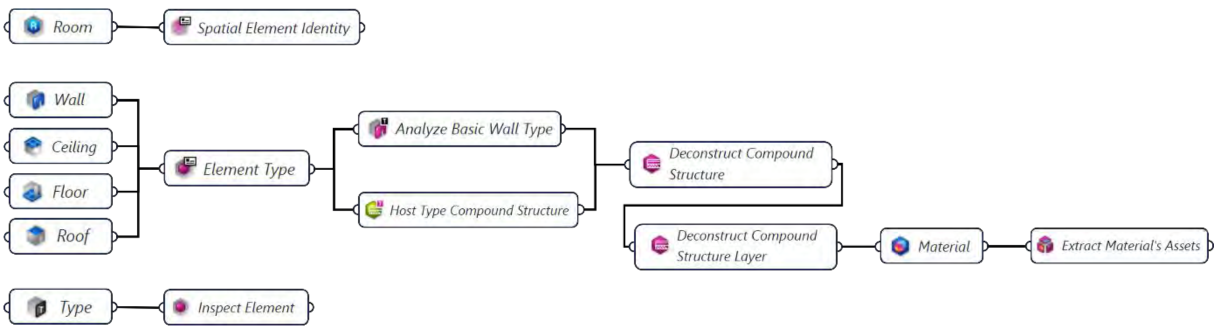


Figure 19: Semantic data extraction of queried Revit instances and types (own illustration)

The BIM model at early design stages lacks precise information to construct a fully detailed Honeybee construction set. Hence, the concept of Model Healing introduced by Wu et al. (2022) is applied to overcome the challenge of insufficient information in early stages BIM model. Particularly, Model Healing concerns the automatic adaptation of building models towards similar designs by providing a range of design options to cover the uncertainties caused by the non-compliance. In this study, default construction sets in the standards library of Honeybee are utilized to provide options to fulfill the requirements of detailed construction set. In other words, a base construction set is defined using the Honeybee standards library. Subsequently, any missing materials to construct a detailed construction set is derived from the pre-determined base construction set.

Table 4: Availability of required material parameters in Revit and Honeybee (own illustration)

	Parameter [unit]	Revit	HB	Required
Opaque material	Name [string]	+	+	
	Thickness [mm]	+	*/	x
	Density [kg/m^3]	+	*/	x
	Thermal conductivity [W/m·K]	+	*/	x
	Specific heat [J/kg·K]	+	*/	x
	Roughness [string]	-	-	
	Thermal absorptance [-]	-	-	
	Solar absorptance [-]	-	-	
	Visible absorptance [-]	-	-	
Window material	Name [string]	+	+	x
	Thermal transmittance (U-value) [W/(m^2·K)]	+	*/	x
	Solar heat gain coefficient [-]	+	*/	x
	Visible transmittance [-]	+	*/	
Legend: (+): available, (-): unavailable, (*/): required unit conversion				

Most required parameters to create Honeybee opaque materials could be directly extracted from the queried Revit instances or types, however, additional unit conversion is required. For example, the differences between the display values of thermal properties of the “Concrete Masonry Units” material in Revit and the extracted values with R.I.R are showed in Figure 20. The cause of the incompatibility between display and extracted values is the Revit internal unit system. Particularly, Autodesk Revit implements seven internal units for base quantities. Among the seven internal units, only the unit for length is in the Imperial system while the others are in Metric system (Autodesk Inc. 2023). Consequently, derived units involving length are returned in non-standard units based on both Metric and Imperial systems. To facilitate the unit conversion between the Revit non-standard derived units to conventional units, the

Revit API provides the ConvertFromInternalUnits method from the Autodesk.Revit.DB.UnitUtils class.

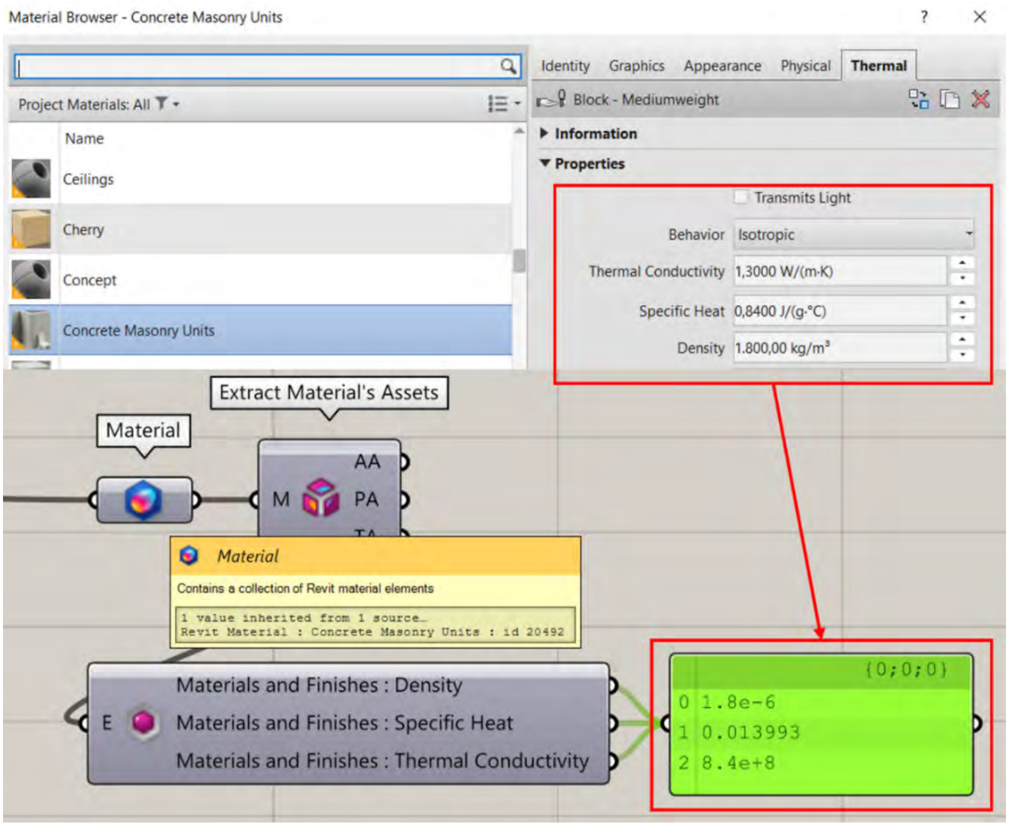


Figure 20: Incompatibility between displayed and extracted values of thermal properties (own illustration)

In addition, it is important to note that Autodesk Revit implemented the Autodesk.Revit.DB.ForgeTypeId class to identify units of measurements, symbols, and unit types instead of the old enumerations since the release of Autodesk Revit 2022 (Autodesk Inc. 2021). The ForgeTypeId class utilizes Forge schemas with the aim to support data interchange between applications. The properties of the ForgeTypeId type can be accessed through the class Autodesk.Revit.DB.UnitTypeId, which contains constants identifying measurement units. The Python code implemented in this thesis to extract thermal properties and converse internal units from the thermal asset of a Revit material is illustrated Figure 21.

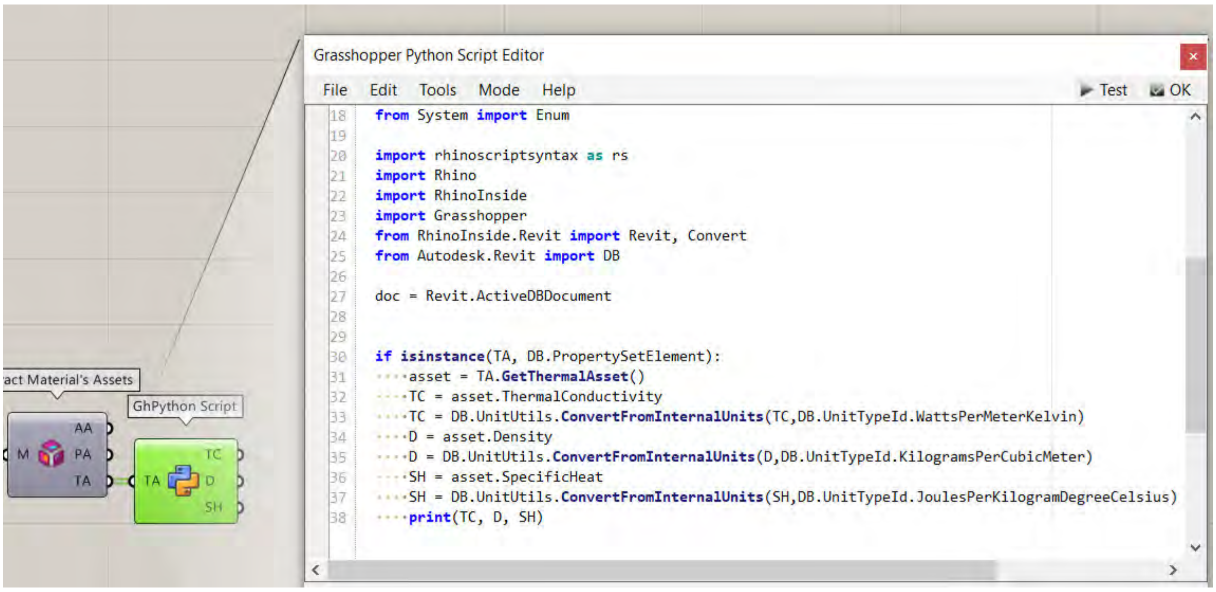


Figure 21: Python script for unit conversion of thermal properties (own illustration)

4.3.3 Creation of the BEM model

In this thesis, the creation of the Honeybee energy model follows the “Solids-to-Rooms Workflow” as described in the Ladybug Tools samples (see Figure 22). The set of extracted Brep representing the Revit rooms serves as the required input to create Honeybee rooms with the “HB Room from Solid” component. In addition, the “HB Intersect Solids” component is implemented prior to the “HB Room from Solid” component to ensure matching coplanar faces in multi-room energy model. The “HB Skylights by Ratio” component is utilized to assign skylight apertures to the roof of the Honeybee rooms. The two sets of Trimmed Surface representing openings on exterior and interior walls serve as the input to create Honeybee apertures and doors respectively. Subsequently, the doors and apertures were attached to the Honeybee rooms using the “HB Add Subface” component. Since the Honeybee model consists of a series of rooms with matching faces, it is required to analyze the Honeybee rooms using the “HB Solve Adjacency” to determine adjacencies between matching room faces. Without the determination of matching room faces, exterior and interior elements of the Honeybee model are indistinguishable. Eventually, the “HB Extruded Border Shades” component is utilized to add shades to all outdoor apertures.

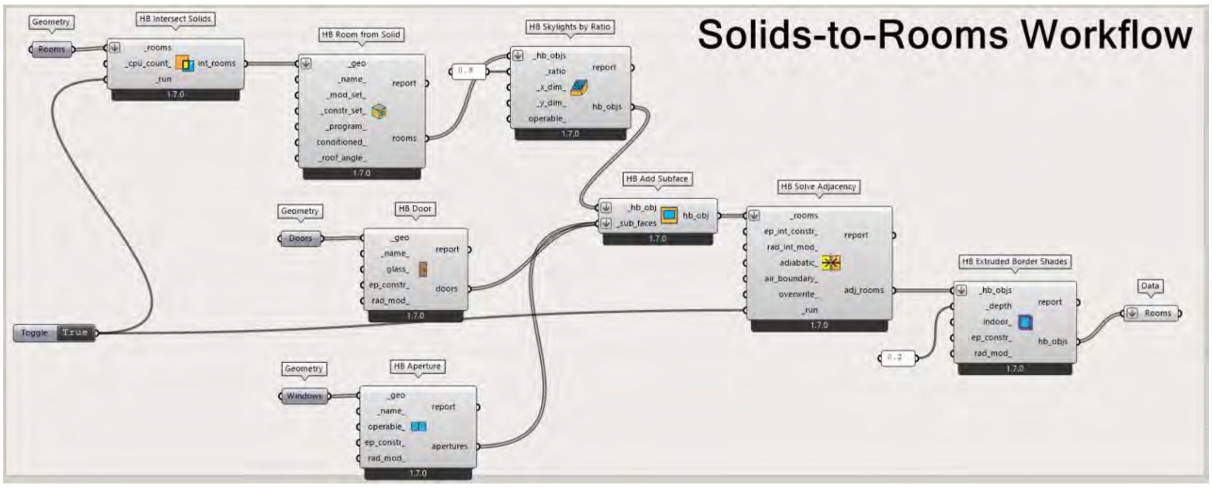


Figure 22: The "Solids-to-Rooms" workflow to create Honeybee energy model (own illustration)

4.4 Parametrization of the BEM model

Table 5 summarizes input parameters in their respective group with the mean values and bounds at BDL 3 and BDL 4. For each set of input values, the initial BEM model is parametrized to generate respective design variants. The values of parameters within the Technical and System group could be used directly to parametrize the BEM model. In contrast, input values for parameters from the Geometrical and Windows groups requires additional manipulation using Grasshopper components.

Table 5: Input parameters for annual energy demands calculations (own illustration)

Group	Parameter	Mean Value	Bounds at BDL 3		Bounds at BDL 4	
			low	high	low	high
Geometrical	L [m]	46.37	45.4426	47.2974	45.9063	46.8337
	W [m]	22.37	21.9226	22.8174	22.1463	22.5937
	H [m]	12.0	11.76	12.24	11.88	12.12
Technical	T_ExWa [mm]	370	277.5	462.5	362.6	377.4
	T_ExRo [mm]	315	236.25	393.75	308.7	321.3
Windows	Wi_A [m^2]	4.16	-	-	3.328	4.992
	WWR [-]	0.33	0.2475	0.4125	-	-
	Infil [-]	0.002	0.0015	0.0025	0.0016	0.0024
System	C_COP [-]	4.0	3.8	4.2	3.8	4.2
	H_COP [-]	0.85	0.8075	0.8925	0.8075	0.8925

The additional steps taken to parametrize the building dimension and windows are described in Figure 23. First, the room and window/door geometry are collected and combined into a list using the “Entwine” component. The reason for implementing the “Entwine” component is to ensure that the room and window/door geometry remain throughout the parametrization process. Subsequently, the room and window geometry are parametrized separately using the values of model input concerning the building dimensions (Geometrical group) and window area (Windows group) respectively. In the case of room geometry, the “Scale NU” component is utilized to scale the building length, width, height in global x-, y-, z-direction with the non-uniform scaling factors derived from the model input values of the Geometrical group. On the other hand, each window geometry is scaled uniformly using scaling factor derived from the Wi_A parameter.

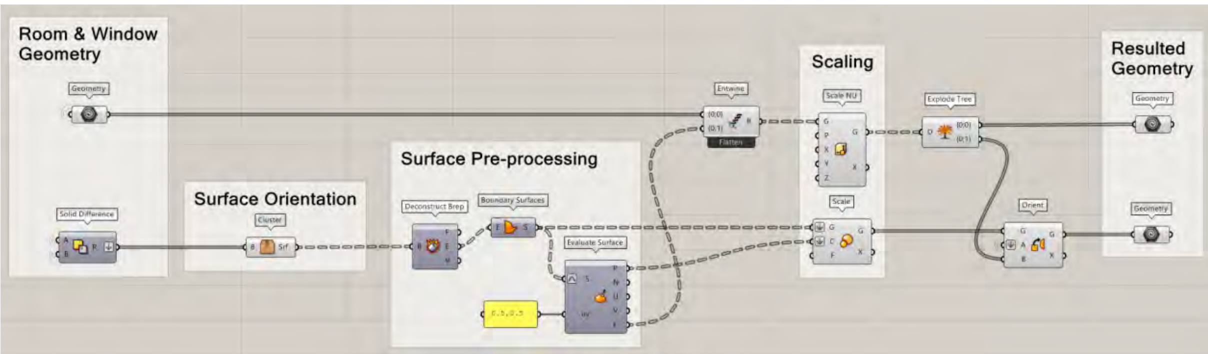


Figure 23: Global dimensions of building and Windows/Doors parametrization (own illustration)

The cluster “Surface Orientation” is implemented to manipulate the surface directions to generate correct centers of scaling for window geometry (see Figure 24). A comparison between the resulting surface centers when using the original Trimmed Surfaces and the new Surfaces is showed in Figure 25. When directly using the Trimmed Surfaces representing the window geometry, the centers locate in various corners of rectangular surfaces instead of the midpoints of the lower edges as expected. The cause of this error is the construction of surface directions of the Trimmed Surfaces. The varying surface centers are justified by creating a set of new Surfaces from the original Trimmed Surfaces.

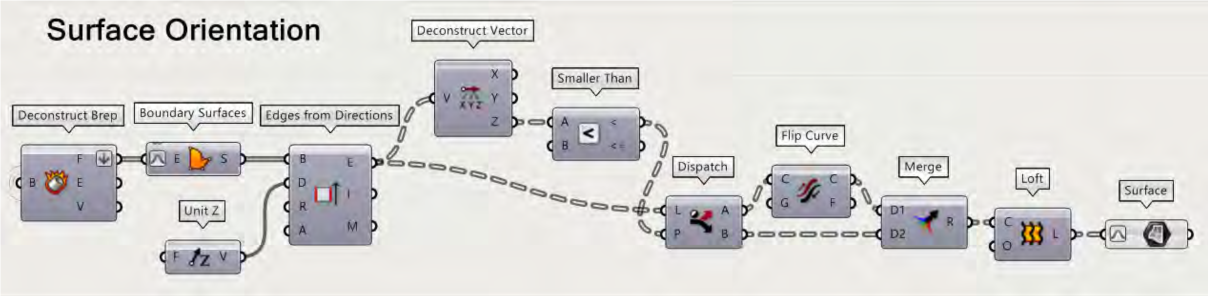


Figure 24: Manipulation of surface orientations (own illustration)

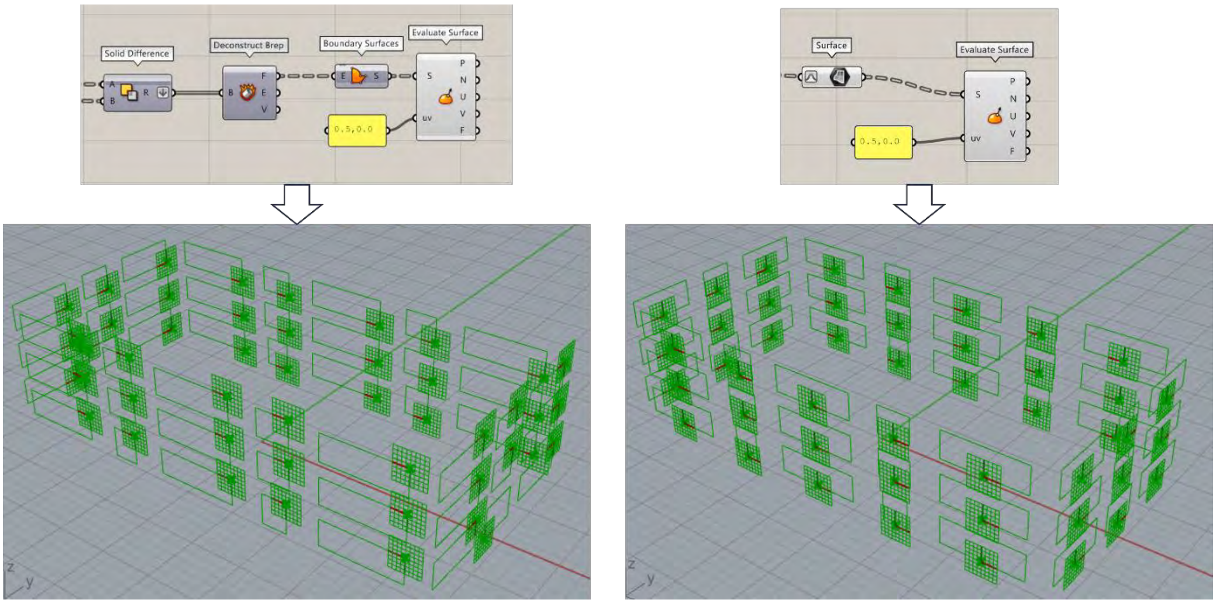


Figure 25: Retrieving centers of scaling with the “Evaluate Surface” component using Trimmed Surfaces (left) and new Surfaces (right) (own illustration)

4.5 Energy Simulation

The Honeybee plugin provides the Honeybee-energy package to facilitate energy simulation of using the EnergyPlus engine. In addition, the OpenStudio SDK is utilized to provide customizable simulation properties and capabilities. The main components used for energy simulation within the Honeybee-energy package include the “Model to OSM”, “Annual Loads”, and “Peak Loads” components.

A general workflow to perform simulation using the “HB Model to OSM” component and compute the EUI result is illustrated in Figure 26. In this workflow, a Honeybee model must be generated as the main input for simulation. Next, the “HB Model to OSM” component writes the Honeybee model into different formats including Honeybee JSON, OpenStudio Workflow JSON, OpenStudio Model, EnergyPlus Input Data File, SQL, and HTML. The properties of the simulation output could be customized using the “HB Simulation Parameter” component. Subsequently, the EUI could be obtained from the SQL file. In addition, the “HB End Use Intensity” component divides the EUI into end uses to estimate the annual heating, cooling, lighting, equipment, and service hot water loads of the building. Alternatively, the EUI could be computed from other file formats, for example IDF as shown in Figure 26.

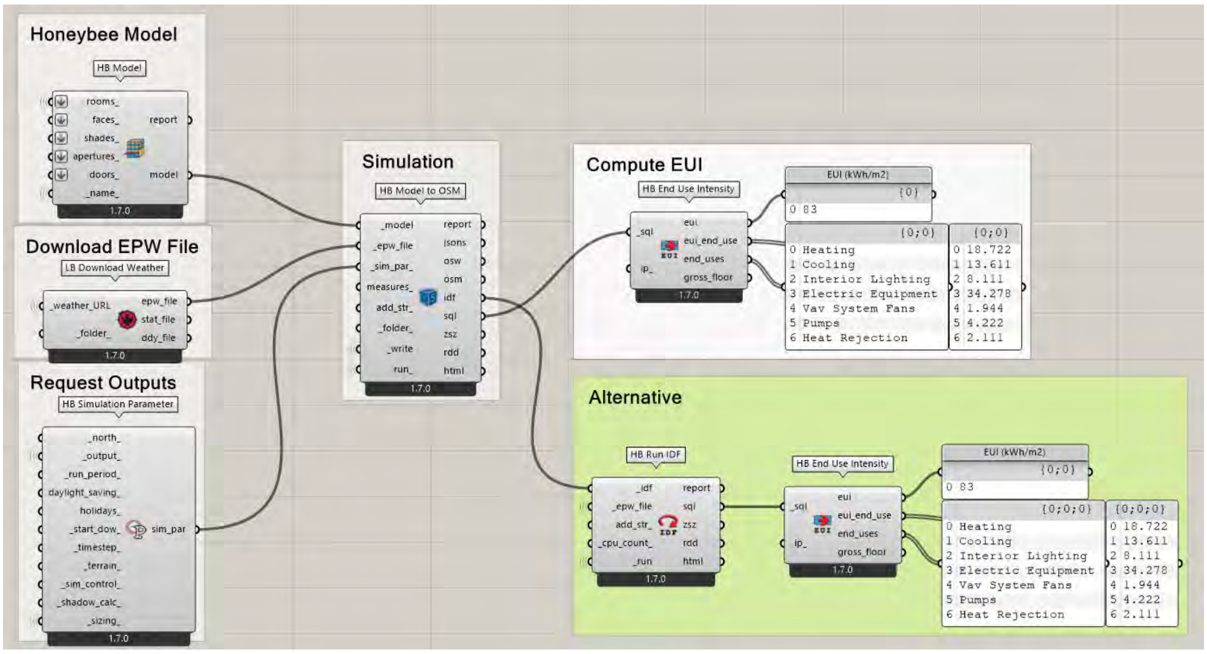


Figure 26: General EUI workflow with the "HB to OSM" component (own illustration)

The workflow for simulation with the “HB Annual Loads” and “HB Peak Loads” is illustrated in Figure 27. Generally, the two components utilize the EnergyPlus engine to perform quick energy simulation over the input Honeybee rooms. The “HB Annual Loads” component returns estimate of annual heating, cooling, lighting, equipment, and service hot water loads normalized by the floor area of the input Honeybee rooms. Honeybee developers recommend the “HB Annual Loads” component in case of annual loads evaluation with up to 5% error tolerance (Ladybug Tools LLC 2023). Options to define the Coefficients of Performance of the cooling and heating systems are available. In the same mechanism, the “HB Peak Loads” returns an estimate of room-level peak cooling and heating on summer and winter design days. For both annual loads and peak loads simulation, the “_timestep_” parameter affects the speed and accuracy of the simulation. Particularly, a higher value for timesteps secures more accurate results but requires longer simulation time.

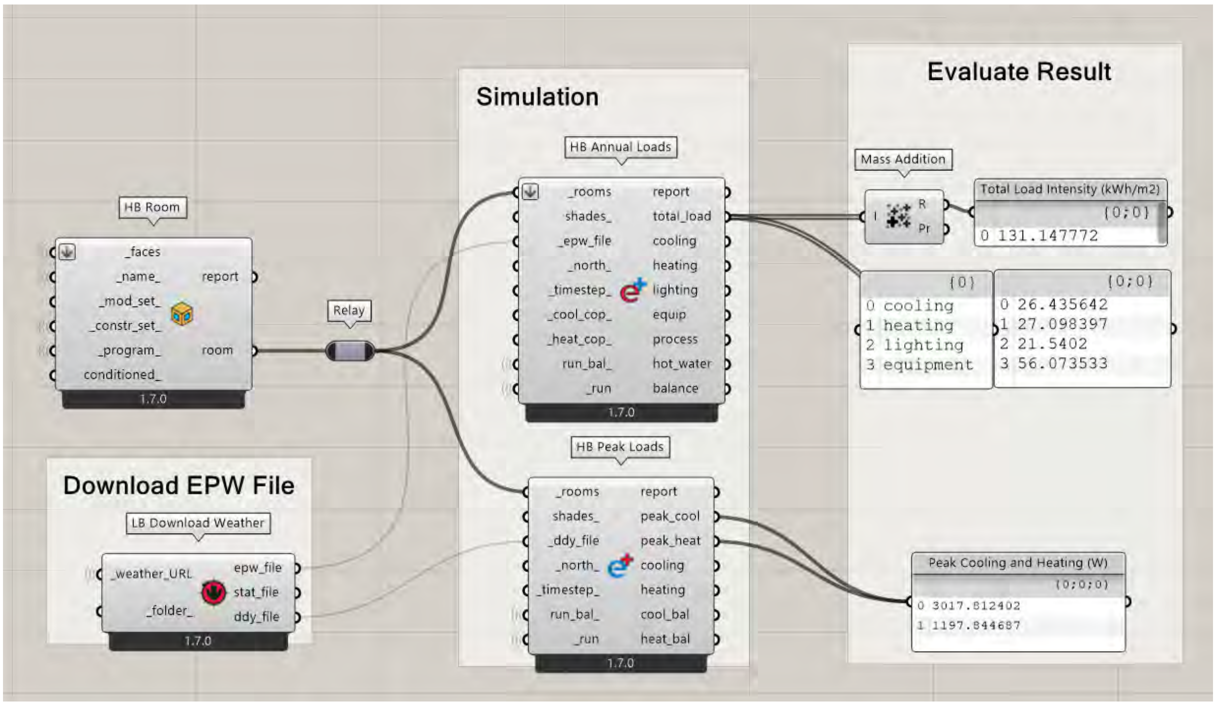


Figure 27: General workflow with the "HB Annual Loads" and "HB Peak Loads" component (own illustration)

In this implementation, the workflow with the “HB Annual Loads” is selected due to the comparably shorter time of simulation compared to the workflow with the “HB Model to OSM”.

4.5.1 Data Preparation for SA

The “Expression” component is utilized to define the problem dictionary containing the names, sampling ranges, and groups of the input parameters. Similarly, the results of the annual loads simulation are recorded and categorized as the model output by the “Expression” component. Subsequently, the problem dictionary and model outputs are written into text files using the “WriteTXT” component by the Wombat for Grasshopper package. The definition of the expressions for both the problem dictionary and the model output should strictly follow the instructions provided by the SA software (see Figure 28).

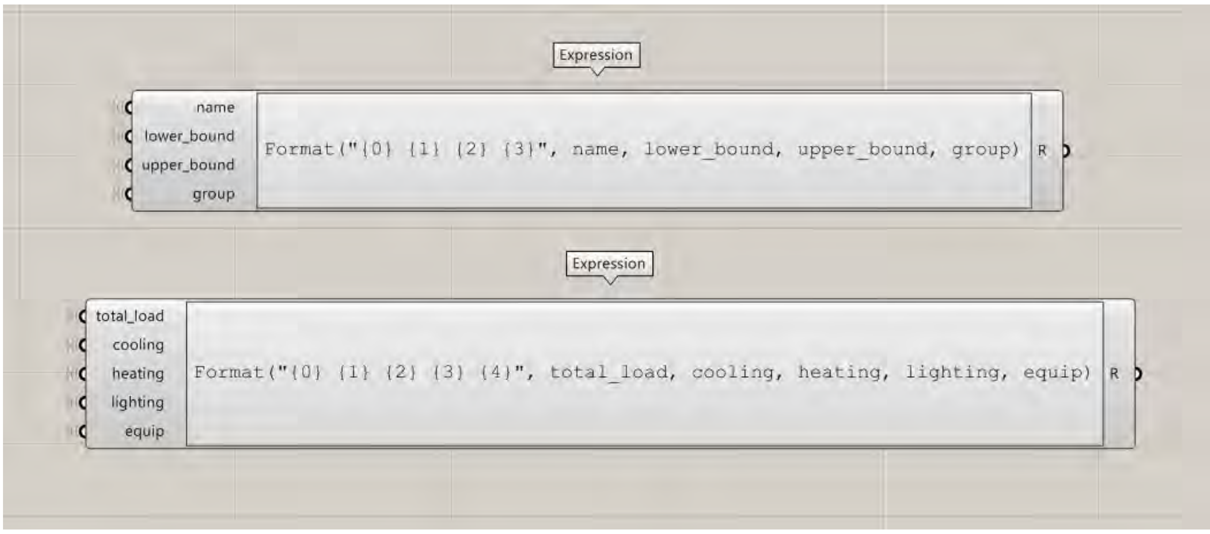


Figure 28: Problem dictionary and model output definition (own illustration)

4.6 Sensitivity analysis

4.6.1 SALib Package

As mentioned in Subsection 3.3.2, the SALib package provides an open-source Python library to perform several SA methods, namely Sobol's, Morris, and FAST. The main principle of SALib is to decouple the SA with the computational model. The typical four-step workflow of SA using SALib is demonstrated in Figure 29. First, the parameter names, distributions, and their respective groups are defined. Subsequently, the defined parameters information is compiled into a “Problem dictionary” using provided interfaces and functions by SALib. In the second step, a sample function is utilized to generate model inputs on the problem dictionary. SALib supports several sampling methods including Morris, Sobol's, FAST, Fractional Factorial, etc. The third step involves the evaluation of the computational model using the generated inputs. This step is fully separated from the SALib script. Eventually, the analyze function is run on the model outputs to generate sensitivity indices. In addition, default plotting functions are available to assist the visualization of the SA results.

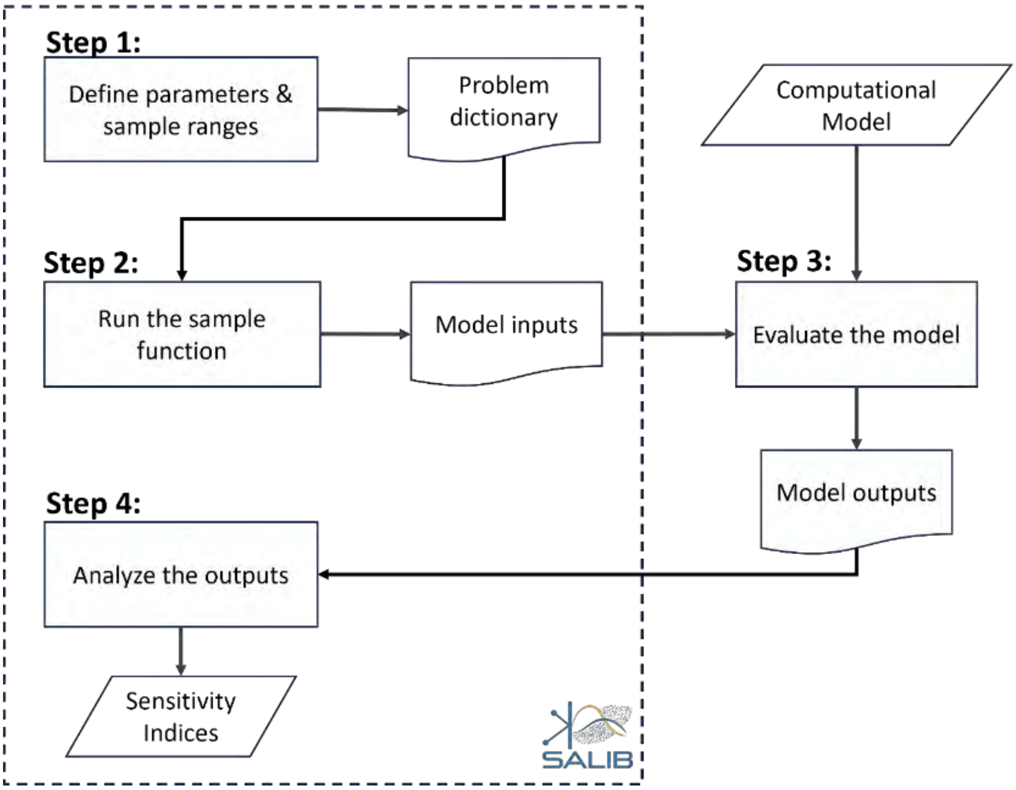


Figure 29: General SALib workflow (own illustration)

4.6.2 Morris Method with SALib Package

SALib provides packages to support both sampling of input parameters and analyze model outputs using the Morris method. Particularly, SALib supports four sampling techniques designed for the Morris method, including the brute force optimization strategy, local optimization algorithm, Morris sampling method, or generating samples with a defined family of algorithms (Herman et al. 2023). Furthermore, SAlib provides the analyze function to perform Morris analysis on model outputs. In addition, basic functions for plotting charts for the method of Morris results are available. In this software implementation, a combination of Morris sampling method and sensitivity analysis was chosen.

The SALib sample function for Morris method returns an array containing model inputs of (G+1) * N rows and D columns, where G, N, and D represent the number of groups, the number of trajectories and the number of parameters respectively. The sample function supports three variations of the Morris's sampling for elementary which could be selected based on the values of the function parameters. In case of False values for both the "optimal_trajectories" and "local_optimization" parameters, the sampling process is based on the factorial sampling technique as described by

Morris (1991). When the “optimal_trajectories” is set to True with an integer value, the optimized trajectories technique is used according to the enhancements by Campolongo et al. (2007). Specifically, the optimal trajectories approach by Campolongo pursues maximizing the parameter space of N trajectories by randomly generating a higher number of possible trajectories and choosing a subset of trajectories with highest spread in the parameter space. Optionally, the enhancement by Ruano et al. (2012) is used when the “local_optimization” parameter is set to True. The Ruano’s local optimization approach, which utilizes an iterative process to maximize the distance between subgroups of generated trajectories, would significantly reduce the processing time for higher values of trajectories and grid levels. When the problem definition contains groups of parameters, the Morris sampling technique with groups is utilized to reduce the number of required model runs.

The SALib analyze function for Morris method takes the same problem dictionary and grid levels used for the sampling function. In addition, the model inputs and the model outputs are passed into the analyze function as Numpy arrays. Furthermore, customization of the analysis is available via the values of the number of resamples used to compute the confidence intervals and confidence interval level. Eventually, the analyze function returns a dictionary containing sensitivity indices including the mean elementary effect “mu”, the absolute of the mean elementary effect “mu_star”, the standard deviation of the elementary effect “sigma”, and the bootstrapped confidence interval “mu_star_conf”.

4.6.3 SALib-R.I.R Integration

Different approaches are available to import the SALib package inside R.I.R since Grasshopper provides the Python script component GHPython. However, the GHPython component implements Python programming language using IronPython, which was unable to import the SALib and other required packages written in CPython. Hence, a switch to a Grasshopper component with the ability to import CPython libraries and packages is necessary. Several methods to import CPython packages into Grasshopper environment are available, including GH Python Remote, COMPAS, GHCPython, and the Grasshopper Hops server.

The workflow with GH Python Remote, COMPAS, GHCPython limits to Python version 2.7. Nevertheless, SALib is incompatible with Python 2 from version 1.2 onwards. As a result, the collaboration of the current SALib version 1.4.7 and GH Python Remote, COMPAS, and GHCPython is not supported. The Grasshopper Hops server remains the only available method for importing CPython packages into Grasshopper with compatible Python version. Particularly, the implementation of Hops requires a Rhinoceros 3D version 3.4 or above, CPython version 3.8 or above, the Hops component in Grasshopper, and the code editor Visual Studio Code. The main idea behind the Hops server is to create CPython functions that can be used inside the Grasshopper scripts. The workflow with Hops utilizes the built-in default HTTP server to either distribute the CPython functions as Grasshopper components, or act as a middleware to a Flask app (McNeel 2021). The existing approaches to incorporate CPython packages into the Grasshopper environment is summarized in Table 6.

Table 6: Approaches to incorporate CPython packages into Grasshopper environment (own illustration)

Component	Version	Release Date	Python Version
GH Python Remote	v1.4.6	21/11/2020	2.7
GH_CPython	v0.1-alpha	28/11/2017	2.7
COMPAS	1.0	18/01/2021	2.7
Hops	0.16.2	23/08/2023	3.8 or above

Alternatively, the SALib library is accessed outside of the Grasshopper environment. For example, Tokarzewski (2020) imported the result from energy simulation in with Honeybee to run Python SA scripts written in Google Collab. This workflow involved recording all the necessary inputs for the SA into text files, reading the input text files in the Python scripts, and utilizing the SALib package to perform SA methods on the imported inputs. The result of the SA included a text file and several plots.

Since the significance of importing the SALib package into Grasshopper environment does not outweigh the effort spent on technical setup of the Hops server, this thesis implemented the SALib package outside of the Grasshopper environment. Particularly, Visual Studio Code is utilized to write a Python script for sensitivity analysis with the SALib package.

4.7 Case Study

4.7.1 Overview

The proposed framework is validated by a hypothetical case study of a three-story office building. The building has a gross floor area of 3000 m^2 with a rectangular floor layout. The structural system of the building consists of a structural steel frame, precast concrete floors, precast concrete elevator shaft, and exterior walls in concrete masonry units. The roof has a sloped glazing system to allow more sunlight in the interior. The interior walls are made of masonry. The 3D view of the building is shown in Figure 30.

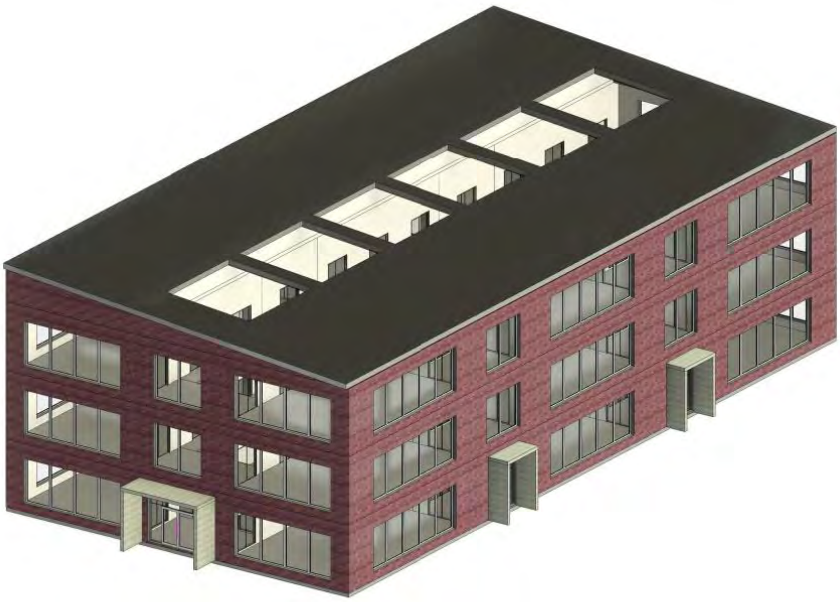
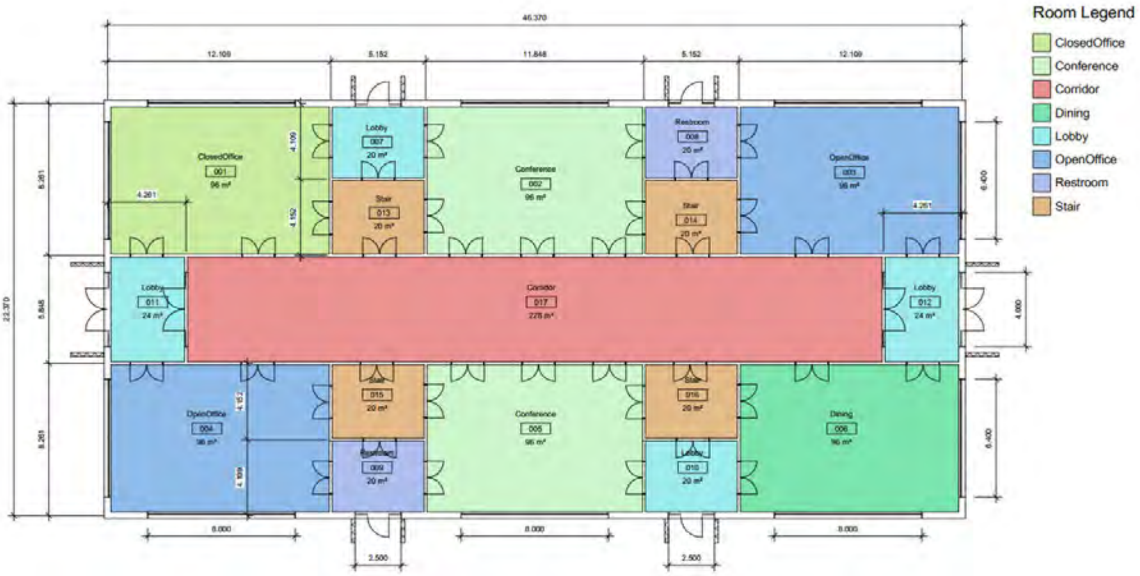
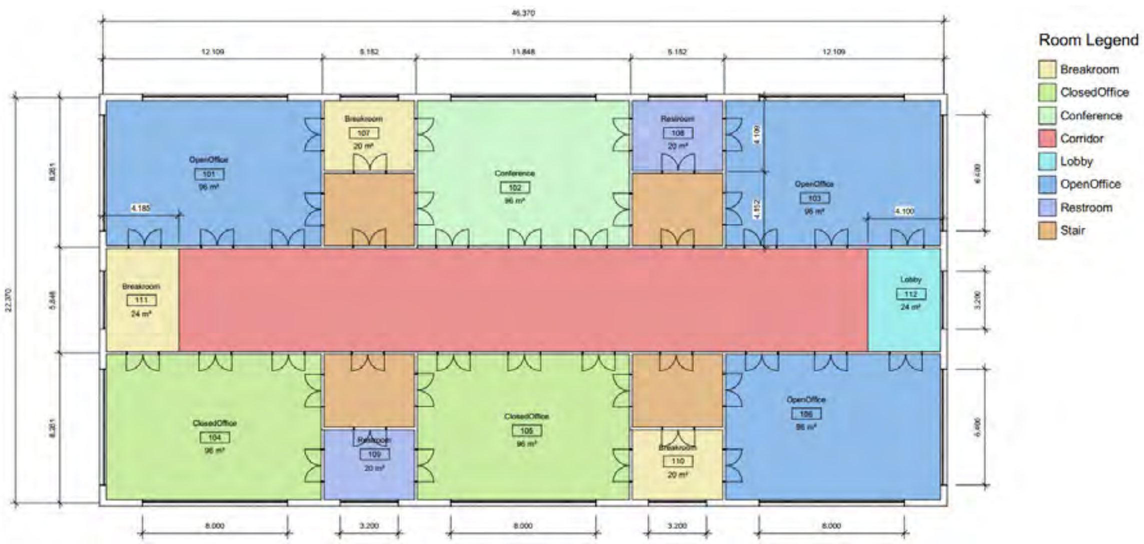


Figure 30: 3D view of the case study building in Autodesk Revit (own illustration)

The building in the case study is located in Munich Airport near Munich, Germany. The climate data of Munich Airport is obtained via the Ladybug Tool EPW Map with the weather dataset from Climate.OneBuilding repository (Ladybug Tools LLC 2023). The climate is classified as Cfb in Köppen-Geiger climate classification, which represents a warm temperate climate with full humid and warm summer.

The specific room usages of ground floor, first floor, and second floor of the building is depicted in Figure 31. The entire ground floor is designed as a communicative space to hold conferences and meetings. At the core of the building interior space, a three-story atrium spans from the ground floor. The glazing systems in the sloped roof are directly connected to the atrium space. The first and second floor are divided into interconnected workspaces with a combination of open and closed offices. A variety of break rooms are placed on the first and second floor to serve leisure activities of the employees. Four structural cores contain the necessary escape stairs and elevators.





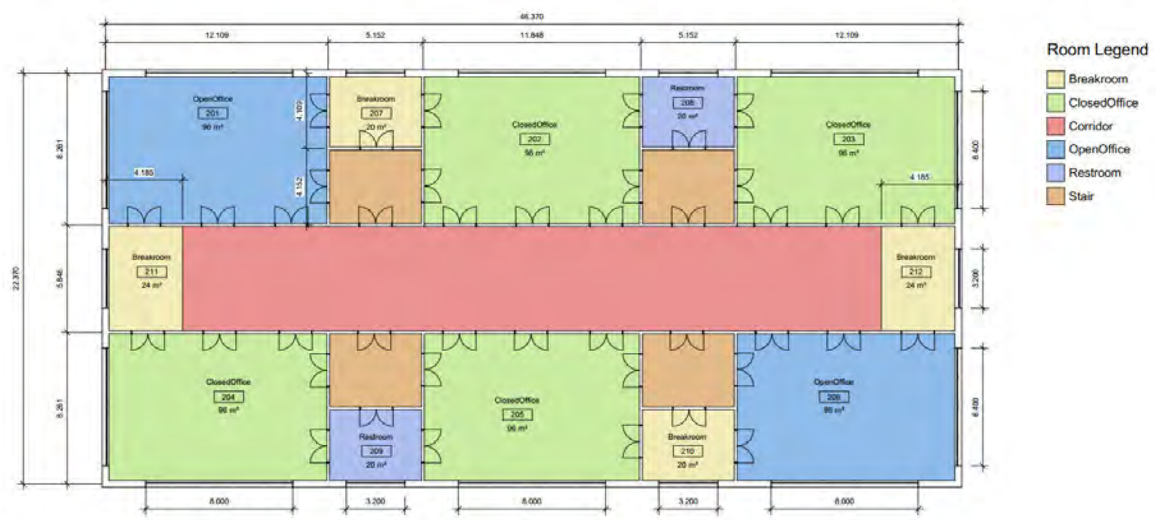


Figure 31: Ground floor, first floor, and second floor plan in Autodesk Revit (own illustration)

The case study building is equivalent in total floor area with a Medium Office as described in the Commercial Prototype Building Models provided by the U.S. Building Energy Codes Program (2023). The Commercial Prototype Building Models are derived from the commercial reference building models for new construction with extensive input according to the ASHRAE Standard 90.1-2019. As described in Section 4.1, the names of the queried Revit rooms were derived from the list of room program identifiers for the “Medium Office” building program and the “ASHRAE 90.1 2019 / IECC 2021” building vintage. Alternatively, the building program could be created using weighed average of the program ratios with the “HB Blend ProgramTypes” component.

4.7.2 BIM Models

Two BIM models were built from scratch in Autodesk Revit 2024 based on the BDL 3 and 4 definition described by Abualdenien and Borrmann (2019). Both models shared identical floor plan layout and room usages. The general requirement for creating BIM model for later simulation purpose as described in Section 4.1 was applied to both cases. The walls, floors, ceiling, and roofs type in both models were modelled to possess detailed information of construction type and materials.

The BDL 3 model aims to define the space in each story and the load-bearing components of the building. Hence, the exterior walls, floors, and sloped roof composed the main elements of the BIM model. To ease the process of manually selecting element types in the BIM-BEM exportation, all wall, floor, and roof types which are irrelevant for the energy simulation were deleted. Moreover, a Window-to-Wall ratio for exterior walls was implemented instead of wall openings with fixed location. However, Autodesk Revit lacks an option to place window instance on walls based on the Window-to-Wall ratio. Hence, no openings were modeled in the BDL 3 model. The 3D views and 3D section of the BDL 3 model are depicted in Figure 32.

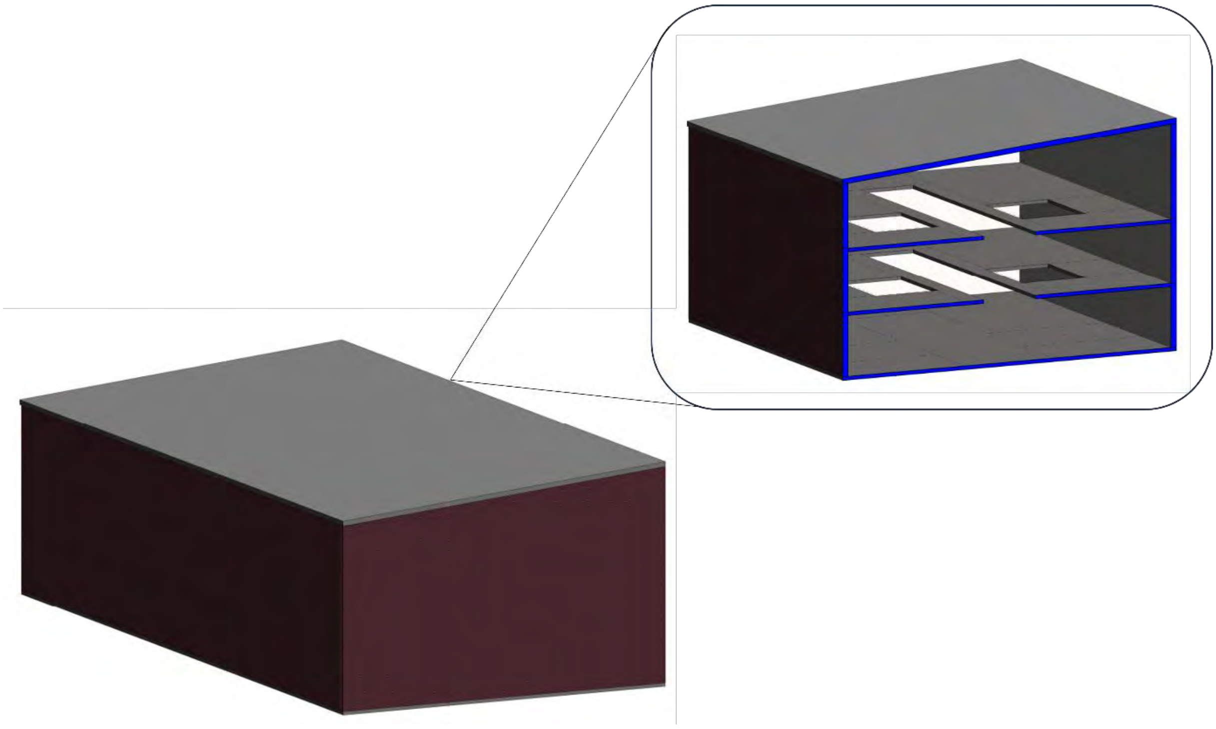


Figure 32: 3D views and 3D section of the BDL 3 model (own illustration)

Generally, the BDL 4 model is the refinement of the BDL 3 model. The main difference between the two models is the refinement of interior structure and openings in the BDL 4 model. Particularly, a more accurate definition of the interior structure with interior partition walls and interior doors was provided. Only one type of door was selected for all interior doors with the aim to ease the manual element type selection in later step.

The BDL 4 model implemented the “Exterior Glazing” type of the “Curtain Wall” family to model openings on exterior walls. All the types of doors, windows and curtain wall panels used in the BIM model were loaded from the Autodesk Family library. The exterior windows and doors were placed as panels on the exterior glazing curtain walls. A fixed sill height and size was defined for all the exterior windows. Comparable to the interior door type, only one type of window was selected for the exterior windows. Moreover, the roof openings were modeled using a sloped glazing system. An 3D illustration of the BDL 4 model with the interior structure and roof configuration is showed in Figure 33.

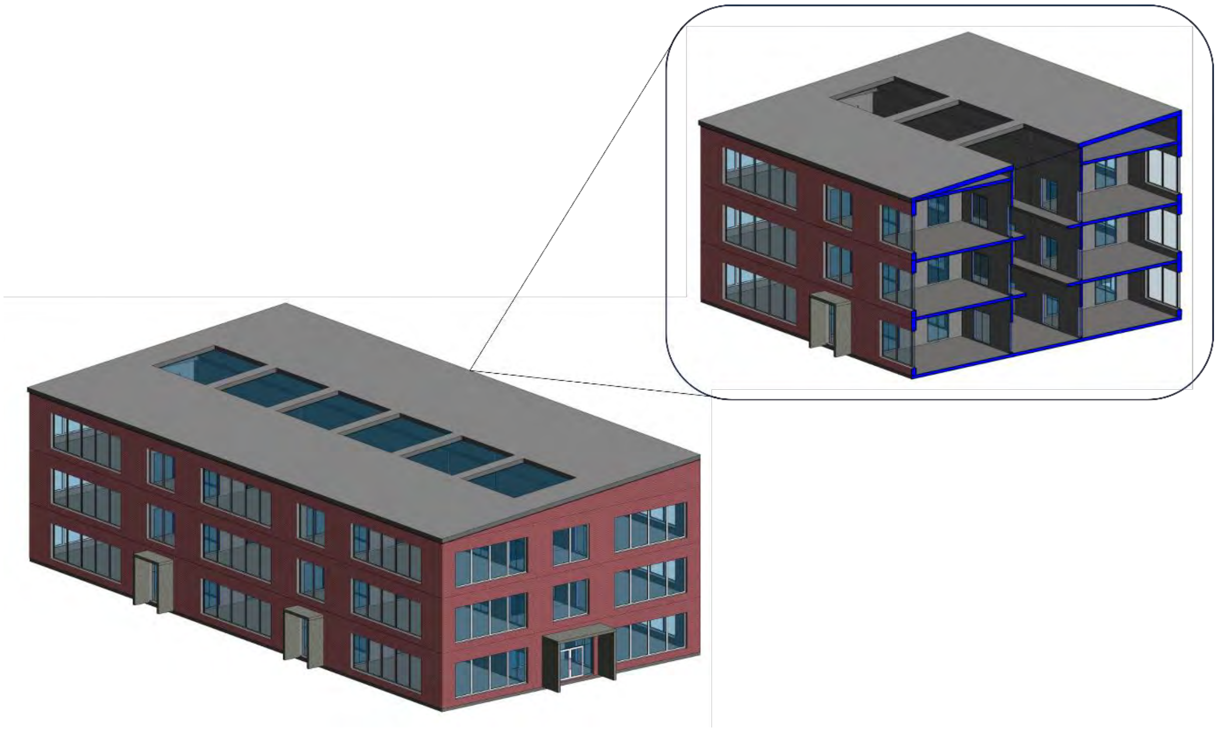


Figure 33: 3D views of the BDL 4 model with and without the roof (own illustration)

4.7.3 Simulation Input Space

A total number of $r(k+1)$ samples is generally suggested for a SA using method of Morris (1991), where k and r are the number of input parameters and number of trajectories respectively. Consequently, each of the k input parameters receive r elementary effects. In this case study, an input space of 9 parameters and 12 trajectories was generated. Hence, 1200 samples were obtained for each BEM model.

In this case study, the sampling process utilized the factorial sampling technique as described by Morris (1991) without the optimized trajectories technique by Campolongo et al. (2007) and the local optimization approach by Ruano et al. (2012). Uniform PDFs were assumed for all parameters.

5 Results & Discussions

5.1 BIM-BEM Exportation

The results of the BIM-BEM exportation with respect to the BIM models at BDL 3 and 4 are summarized in Table 7. The details of the exportation results for each model are reviewed in the following subsections. In general, the exportation of geometrical data of the BIM models only considered the relevant inputs for the “Solids-to-Rooms” workflow as described in Subsection 4.3.3. For simplification purpose, the extraction of roof geometry was neglected. Particularly, the top surfaces of the room geometry represented the roof elements of the BEM model instead. In the case of semantic data, most of the extracted data required additional unit conversion as described in Subsection 4.3.2.

Table 7: Summary of BIM-BEM exportation results (own illustration)

Processing task	Data	BDL 3	BDL 4
	Geometry		
Create HB room	Room geometry	+	+
Define HB shade	Roof geometry	-	-
Define HB aperture/door	Roof openings	-	-
	Wall face with openings	-	*/
	Semantic data		
Define construction sets	Material name	+	+
	Wall, roof material’s thermal asset	-	*/
	Floor, ceiling material’s thermal asset	-	*/
Define model input	Window, door material’s thermal asset	-	*/
	Wall length	+	+
	Wall unconnected height	+	+
	Wall thickness	*/	*/
Create HB room	Roof thickness	*/	*/
	Window area	-	*/
	Roof slope	*/	*/
Legend: available (+), not available (-), working with unit conversion (*/)			

A comparison between the dimensions of the original BIM models and the transformed BEM models is showed in Table 8. Overall, the transformed gross floor areas and volumes of the buildings matched the original values. In the case of exterior wall areas, variations of 1-2% between the original and transformed values occurred.

Table 8: Comparison between the dimensions of the original BIM model and transformed BEM models at BDL 3 and 4 (own illustration)

	BDL 3		BDL 4	
	BIM	BEM	BIM	BEM
Building area [m^2]	2420	2420	2420	2420
Building volume [m^3]	13150.68	13150.68	13150.68	13150.68
Exterior wall area [m^2]	1787	1767.28	1059	1039.36

5.1.1 BDL 3 model

The BEM model resulted from the BIM-BEM transformation for the BIM model at BDL 3 is illustrated in Figure 34. Since the data concerning the windows was unavailable in the BDL 3 BIM model, the apertures in the BEM model were added with defined sill height using the "HB Apertures by Ratio" component. Similarly, the roof openings were represented by using the ""HB Skylights by Ratio" instead of actual roof opening geometry. In addition, the interior walls of the BEM model were automatically generated based on the exported room geometry. Considering the thermal construction set, only relevant properties from the thermal assets of the exterior wall and roof material were extracted to create the HB room construction set. Other construction sets were derived from the HB standard library using predefined climate zone, building vintages, and construction types.

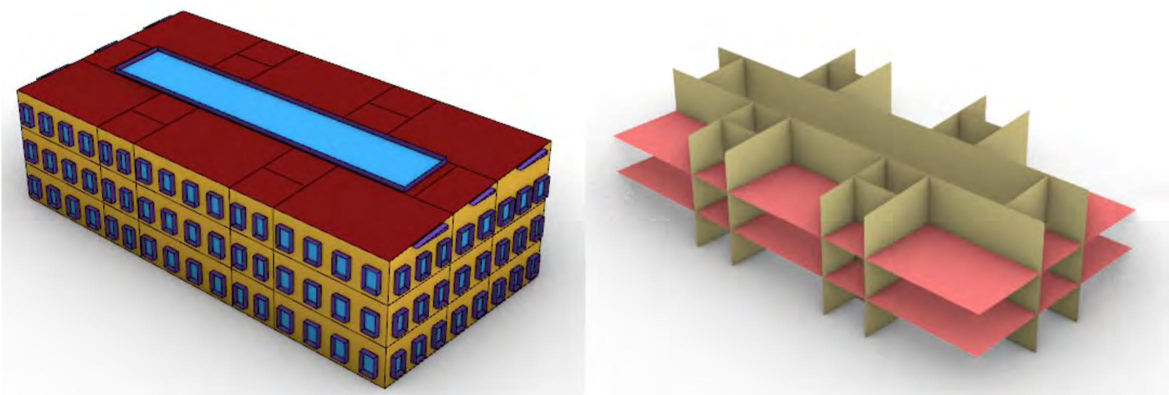


Figure 34: Visualization of the BDL 3 BEM model and its interior structure (own illustration)

5.1.2 BDL 4 model

The BEM model generated from the extracted data from the BDL 4 BIM model is showed in Figure 35. Due to the refinement in the interior structure of the BDL 4 BIM model, the interior walls and interior doors were defined based on the exported room and wall opening geometry from the BIM model. In addition, exterior apertures of the BEM model were generated using the exported exterior wall face geometries. Regarding the thermal construction set, a more detailed construction set was established with the thermal properties from the material thermal assets of the exterior, interior, and window/door elements.

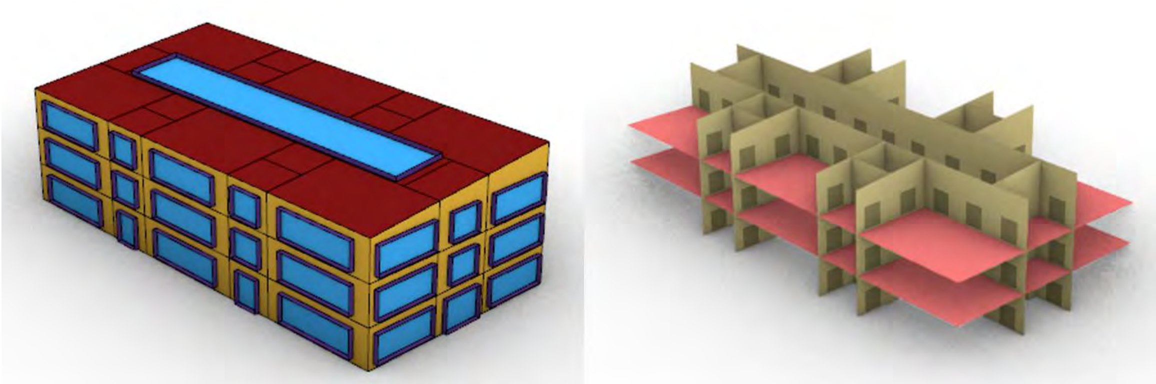


Figure 35: Visualization of the BDL 4 BEM model and its interior structure (own illustration)

5.2 Parametrization of the BEM models

The main inputs for the parametrization process of the BEM models at BDL 3 and 4 was the set of samples generated by SALib as described in Section 4.4. A summary of the parametrization process is described in Table 9. For each BEM model, a total number of 1200 variants were generated. When using the “HB Model to OSM” component, batch of the design variants could be stored locally in the OSM and IDF formats. Computational cost of the parametrization process is efficient, which amounts to 1 and 2.3 hours to generate 1200 design variants for the BEM model at BDL 3 and BDL 4 respectively. The visualization of the two batches of design variants generated from the parametrization process of the BEM models at BDL 3 and 4 is illustrated in Figure 36.

Table 9: Summary of parametrization process (own illustration)

	Usages	BDL 3	BDL 4
Geometrical			
L [m]	Scaling model dimensions	+	+
W [m]	Scaling model dimensions	+	+
H [m]	Scaling model dimensions	-	-
Wi_A [m^2]	Parametrize HB aperture/door		+
Semantic			
T_ExWa [mm]	Parametrize HB construction sets	+	+
T_ExRo [mm]	Parametrize HB construction sets	+	+
WWR [-]	Parametrize HB aperture/door	+	
Infil [-]	Parametrize HB building program	+	+
C_COP [-]	Parametrize HB HVAC system	+	+
H_COP [-]	Parametrize HB HVAC system	+	+
Number of variants		1200	1200
Error percentage		0.9%	1.1%
Legend: working (+), causing conflicts (-)			

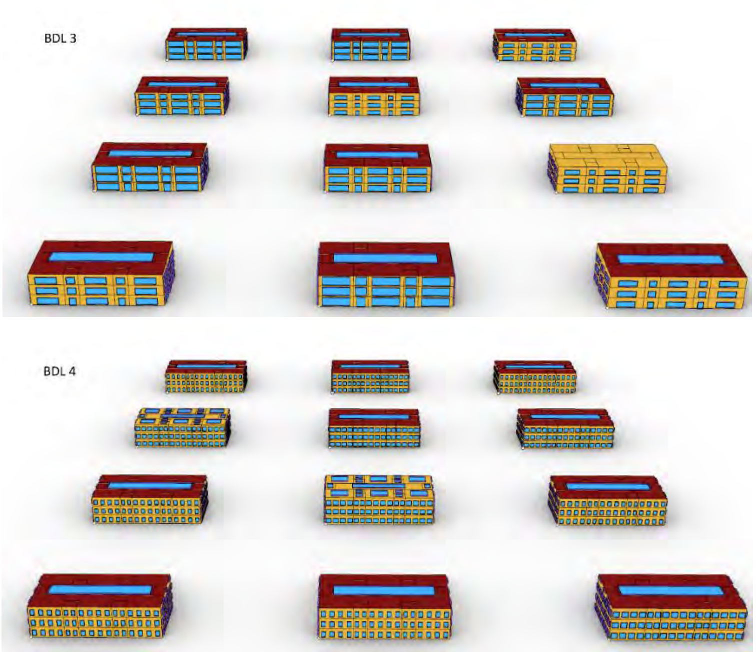


Figure 36: Design variants batch of parametrized BEM models at BDL 3 (upper) and BDL 4 (lower) (own illustration)

In general, the parametrization of the BEM models using model inputs from SALib showed precise results, except for the incorrect definition of roof elements appeared in approximately 1% of the total design variants. As shown in Figure 37, the error in the roof elements subsequently resulted in incorrect skylight geometry. The direct cause of this error was the conflicts between the parametrized building height H and the extracted roof slope parameter. Hence, the use of the roof slope when creating Honeybee room was neglected. In terms of parametrizing semantic data, namely parameters in the Technical, System, and parts of the Windows groups, no error was detected.

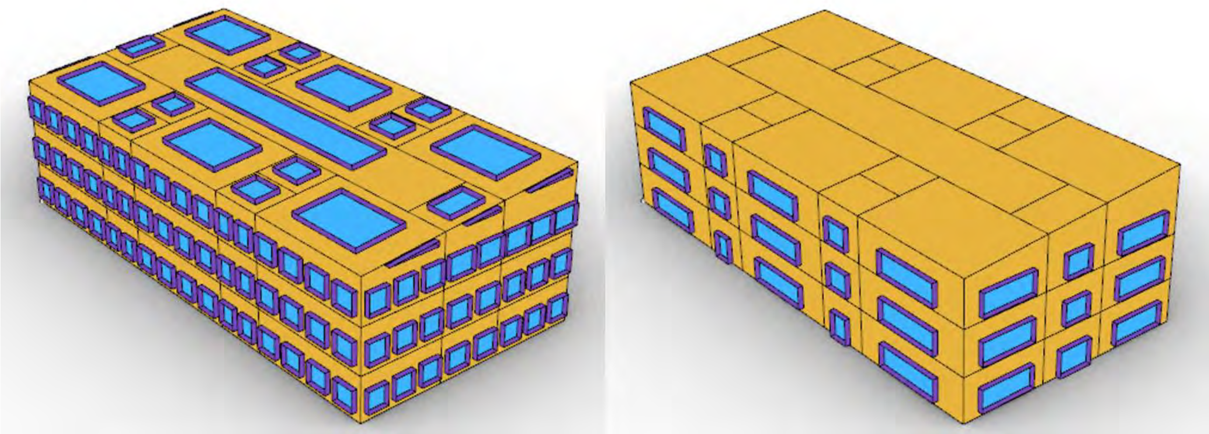


Figure 37: Error in definition of roof elements for parametrized BEM model at BDL 3 (left) and BDL 4 (right) (own illustration)

5.3 Simulation Results

5.3.1 BDL 3

As described in Section 4.5, the annual loads simulation returns estimated values of annual heating, cooling, lighting, equipment load intensity of the studied building. The sum of the four previously mentioned annual loads is stored as the total load intensity. A summary of the annual loads simulation results for the batch of 1200 parametrized BEM models at BDL 3 is described in Table 10. Overall, the heating loads contributed the highest share in the total load intensity, with a median value of 74.65%. Moreover, the intensities of lightning and electric equipment load remained constant throughout 1200 simulations. This reflected the default values of lighting and equipment load provided in the base building program utilized in this case study. Eventually, the cooling load had the lowest contribution in the total load intensity.

Table 10: Summary of annual loads simulation of BEM models at BDL 3 (own illustration)

Value	Minimum	Maximum	Mean
Total Load Intensity [kWh/m²]	216.224	326.822	269.895
Cooling Load Intensity [kWh/m²]	2.319	3.369	2.801
Heating Load Intensity [kWh/m²]	148.069	258.115	201.478
Lighting Load Intensity [kWh/m²]	18.412	18.412	18.412
Equipment Load Intensity [kWh/m²]	47.204	47.204	47.204

The result of the annual loads simulation was visualized by the monthly load and energy balance bar charts. With the energy balance bar chart, a graphical display of all the energy inputs supplied into a building and their uses was available. For instance, visualization of the simulation result for the design variant with highest total load intensity is illustrated Figure 38.

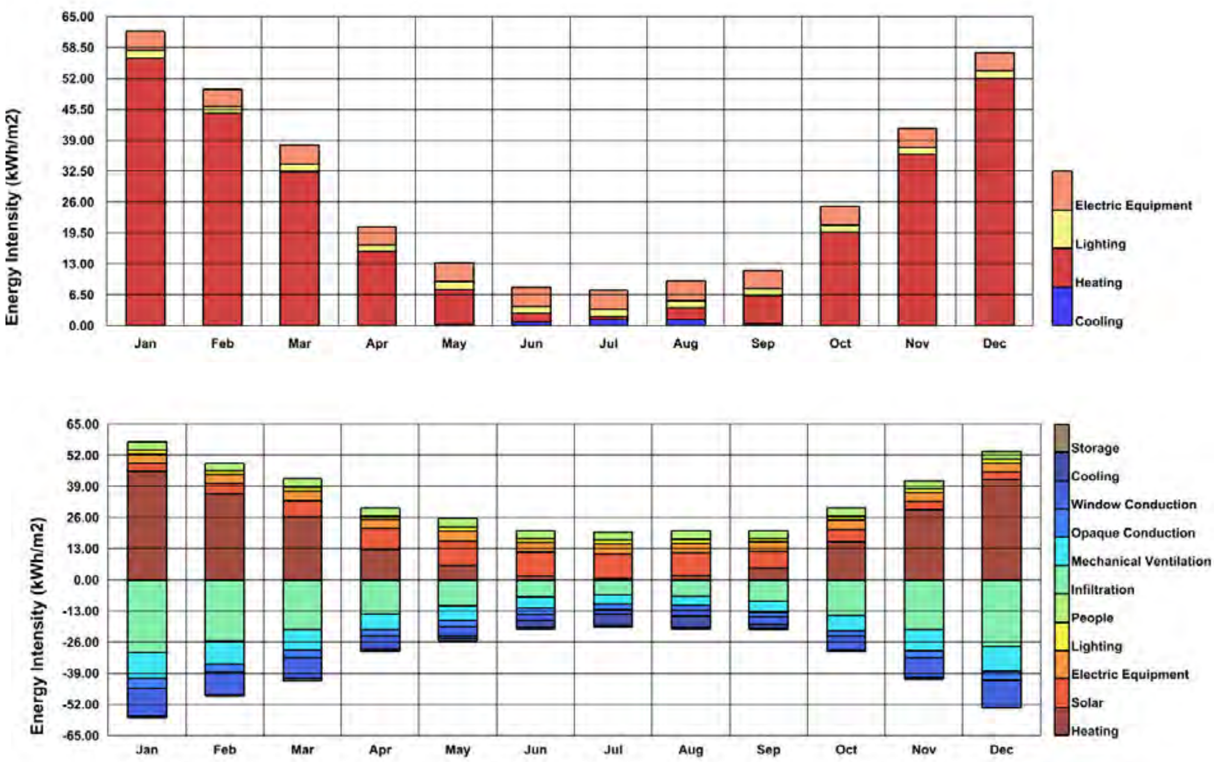


Figure 38: Monthly load and energy balance bar chart for design variant with highest total load intensity of the BEM models at BDL 3 batch (own illustration)

5.3.2 BDL 4

The results of annual loads simulation for the batch of 1200 parametrized BEM models at BDL 4 is summarized in Table 11. In general, the contribution of the cooling, heating, lightning, and electric equipment load to the total load intensity in the simulation with BEM models at BDL 4 resembled the results from the BDL 3 simulation. The mean result for total load intensity of the BDL 4 simulation varied by 5% that of the BDL 3 simulation. The cause of the result variation could be the higher timestep value used for the BDL 4 simulation, which resulted in longer simulation time but more accurate outcomes. Another possible cause of the result variation could be the more detailed construction set of the BDL 4 BEM model. The visualization of the simulation result for the design variant with the highest total load intensity is illustrated in Figure 39.

Table 11: Summary of annual loads simulation of BEM models at BDL 4 (own illustration)

Value	Minimum	Maximum	Mean
Total Load Intensity [kWh/m^2]	211.287	304.285	255.534
Cooling Load Intensity [kWh/m^2]	4.519	7.885	6.138
Heating Load Intensity [kWh/m^2]	140.772	231.791	183.78
Lighting Load Intensity [kWh/m^2]	18.412	18.412	18.412
Equipment Load Intensity [kWh/m^2]	47.204	47.204	47.204

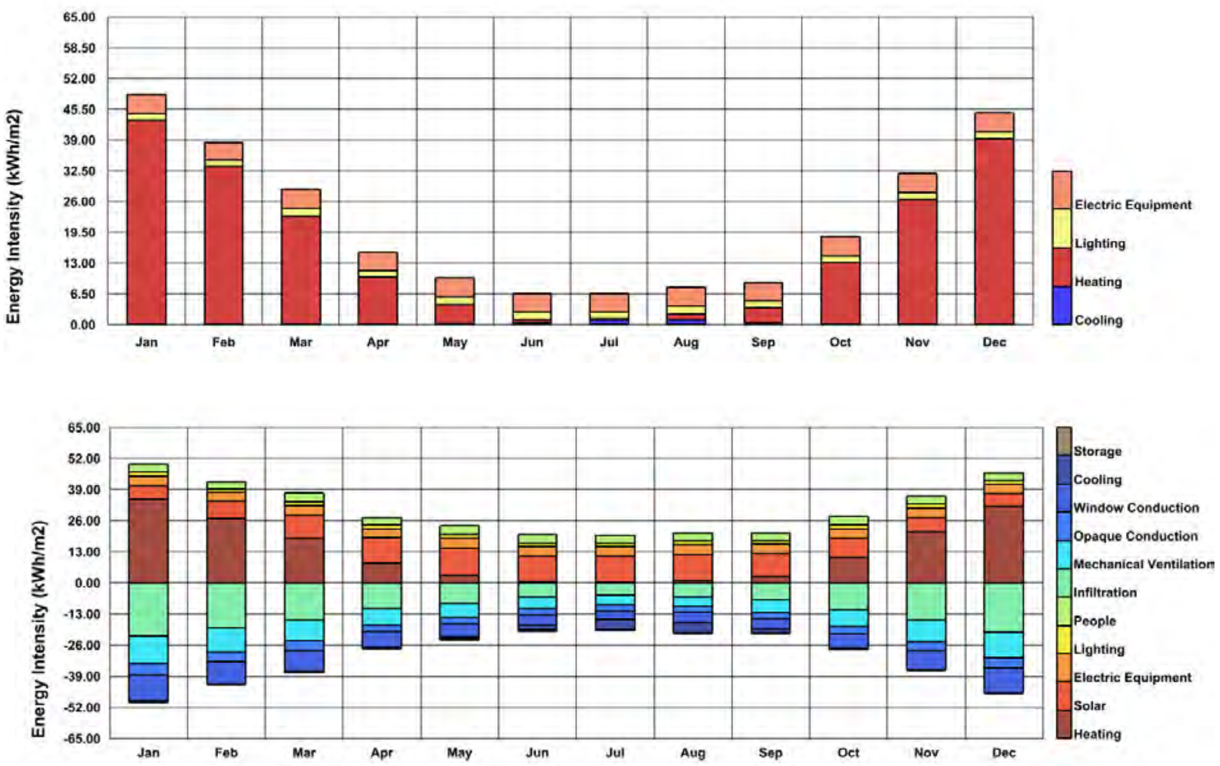


Figure 39: Monthly load and energy balance bar chart for design variant with highest total load intensity of the BEM models at BDL 4 batch (own illustration)

5.4 SA Results

In general, each of the 9 input parameters defined for the BIM models at BDL 3 and 4 were analyzed using Morris method over 12 trajectories. The analyzed output in this SA was the total load intensity recorded from the annual loads simulation. As mentioned in Subsection 3.7.3, the standard deviation σ and the absolute mean μ^* resulted from the Morris method over a considerable number of trajectories were utilized to evaluate the global sensitivity measures of the importance of the parameters. First, the covariance plot using σ and μ^* was generated to visualize the classification of parameters in terms of influential effects. Next, the ranking of parameters in terms of the absolute mean μ^* was plotted in the horizontal bar plots, with the error bar representing the confidence interval μ^*_{conf}. Furthermore, the absolute mean μ^* and relative parameter ranking were plotted against incrementally increasing the number of samples to assess the performance of μ^* and the resulting ranking of parameters.

5.4.1 BDL 3

The results of the Morris analysis for total load intensity of the BEM model at BDL 3 is showed in Figure 40. According to the covariance plot, all the parameters in the analysis were influential with non-linear and/or interaction effects. The values of μ^* for parameters within the Windows group (WWR and Infil) are the lowest among other groups. Parameters of the Technical group considering the thickness of exterior wall and roof ranked second to last above the Window group parameters. The higher μ^* values belong to the parameters within the Geometrical (L, W, and H) and System (C_COP and H_COP) groups. Particularly, the parameter W representing the building height had the largest impact on the total load intensity. Furthermore, it is important to note that the confidence interval μ^*_{conf} was high over all parameters, ranging from 24.7 to 31.8%.

The μ^* values obtained from Morris method and the resulting ranking of parameter influences over incrementally increasing the number of samples from 200 to 1200 is showed in Figure 41. In the two plots illustrated in Figure 41, crossing lines signifed changes in parameter ranking between individual evaluations. As the number of samples increased, the distribution of resulting μ^* became more continuous and presumptively more robust. For evaluation with number of samples from 400 to 1000, changes in parameter rankings occurred frequently for all parameters, except the most and least influential parameters. It is important to note that variations in parameter rankings also happened for evaluation with high number of samples (1100).

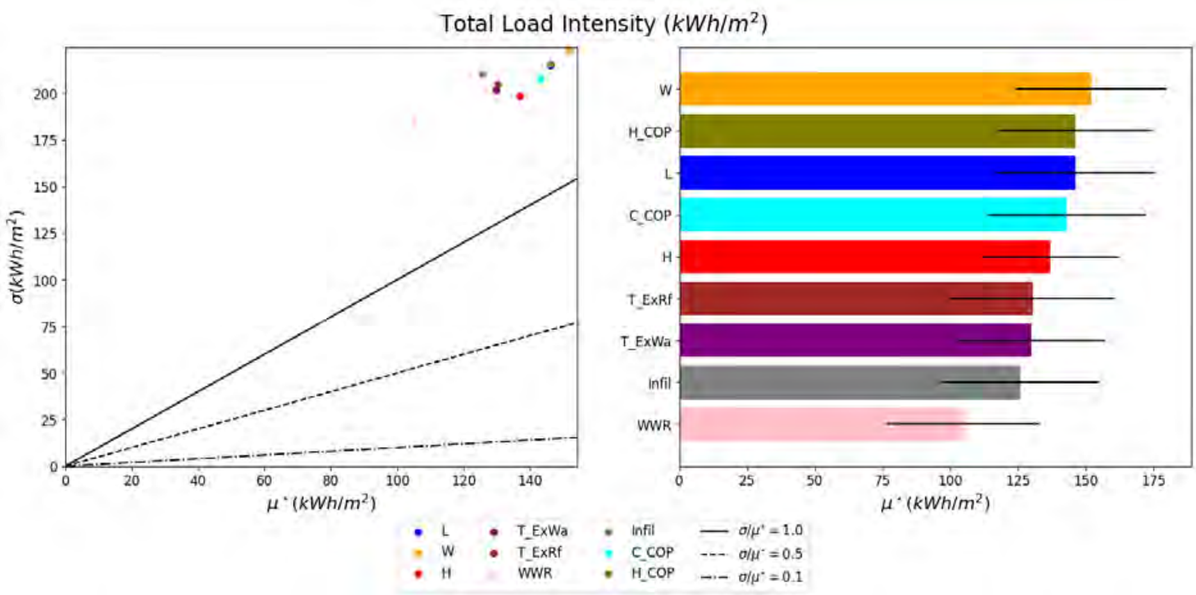


Figure 40: Results of the SA with Morris method for the total load intensity of the BEM model at BDL 3, showing the covariance plot (left) and horizontal bar plot with confident interval (right) (own illustration)

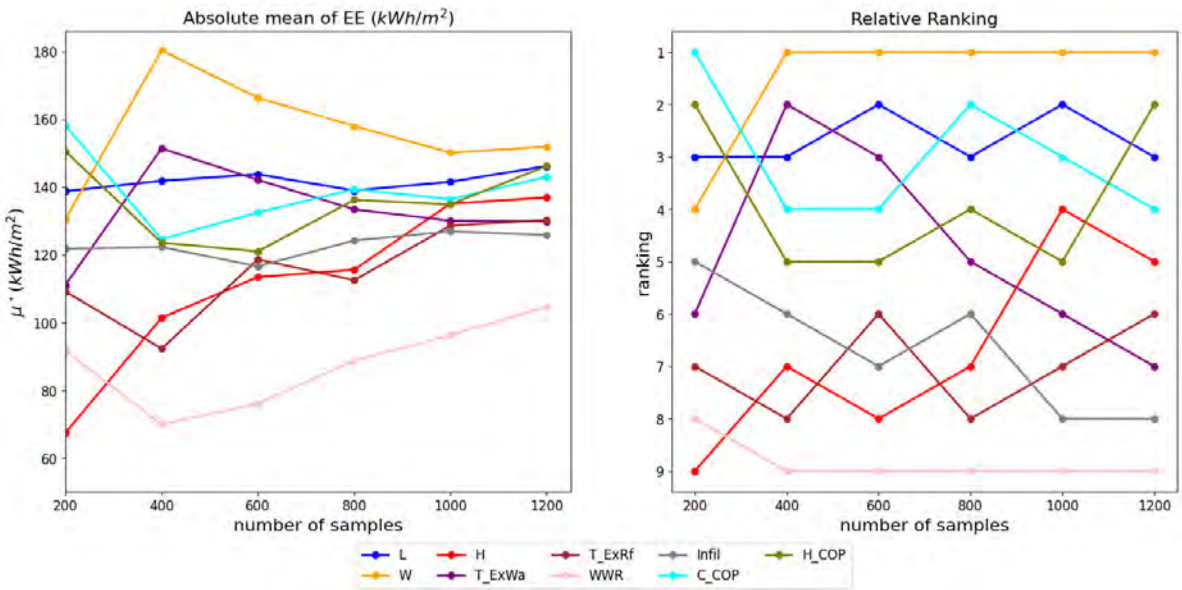


Figure 41: Evolution of Morris SA results for the total load intensity of the BEM model at BDL 3 for an increasing number of samples, showing the absolute mean μ^* values (left) and the resulting ranking of parameters using μ^* (right) (own illustration)

5.4.2 BDL 4

The results of the Morris analysis for total load intensity of the BEM model at BDL 4 is showed in Figure 42. Comparable to the result for the BEM model at BDL 3, all the parameters were classified as influential with non-linear and/or interaction effects. In addition, parameters within the Windows group (WWR and Infil) received the lowest values of μ^* among other groups. A clear shift occurred in the ranking of parameters of the Technical and System groups when compared to the BDL 3 result. Particularly, the parameters of the Technical group became the most dominant parameters while only ranked second to last in the analysis for BDL 3. In contrast, the rankings of parameters in the System group are significantly lower, which represented a reduction in influence. However, the confidence interval μ^*_{conf} over all parameters was significantly higher. For instance, the most influential parameters, T_ExWa, had a confidence interval of 40.9%.

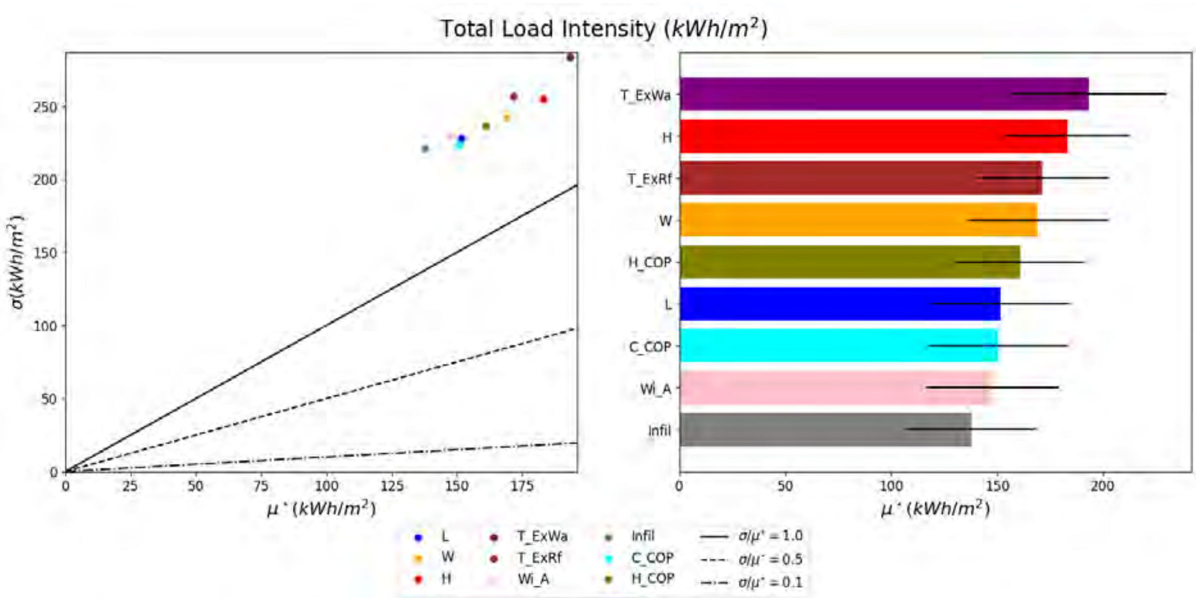


Figure 42: Results of the SA with Morris method for the total load intensity of the BEM model at BDL 4, showing the covariance plot (left) and horizontal bar plot with confident interval (right) (own illustration)

Compared to the result for BDL 3, a continuous and robust distribution μ^* required a lower number of samples (see Figure 43). Particularly, a stable parameter ranking without variations can be obtained after evaluation of 100 samples. In addition, the changes in parameter rankings became significantly less frequent after evaluation of 800 samples.

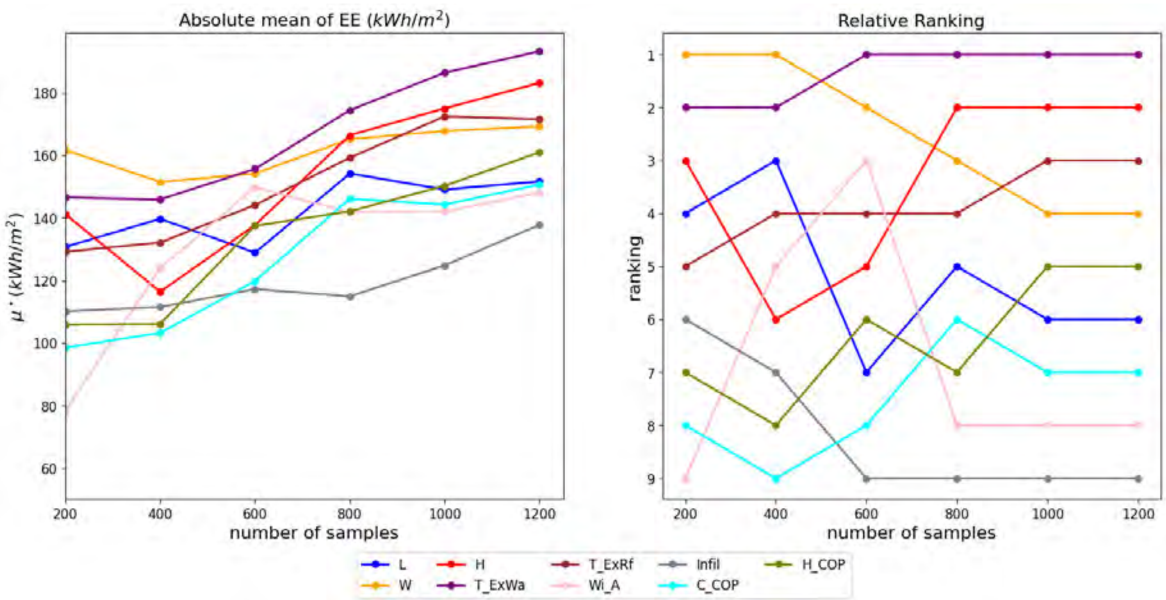


Figure 43: Evolution of Morris SA results for the total load intensity of the BEM model at BDL 4 for an increasing number of samples, showing the absolute mean μ^* values (left) and the resulting ranking of parameters using μ^* (right) (own illustration)

5.5 Overall Efforts

The overall efforts to validate the prototypical software implementation with the case study is summarized in Table 12. Overall, the time required for the BIM-BEM exportation is negligible despite the increasing refinement of the BIM model at ascending BDLs. Similarly, the results of the SA with SALib package for 1200 samples could be obtained almost instantly.

Table 12: Summary of effort to pursue the case study (own illustration)

Tasks	BDL 3	BDL 4
BIM-BEM exportation [minutes]	instant	instant
Parametrize and write 1200 BEM models [minutes]	64.3	136.2
Run Annual Loads simulation [minutes]	498.5	1176.5
Perform SA with SALib [minutes]	instant	instant
Overall duration [minutes]	562.8	1312.7

Considering the parametrization of the BEM model and subsequent generation of a batch of 1200 design variants in the OSM and IDF formats, an average time of 3.1 and 6.8 seconds is required for each iteration for the BEM model at BDL 3 and 4 respectively. The most computational expensive task of the case study was the simulation of annual loads, which lasted 8.3 and 19.6 hours for the BEM models at BDL 3 and 4 respectively. In both the parametrization and annual loads simulation tasks, the time needed for the BEM model at BDL 4 was more than doubled that for the BEM model at BDL 3.

5.6 Discussion

5.6.1 Key Findings

A semi-automated framework of BIM-based BEPS with SA for office buildings was validated by this case study. The advantages of utilizing BIM models as the core data model was proven as all geometrical and most of the semantic data relevant to BEPS was extracted from the original BIM models. In addition, the workflow integrating Autodesk Revit and R.I.R provided more control over the whole parametric modeling and simulation process, which was crucial when errors occurred. Furthermore, the integration of the SALib in the framework was smooth with minimal technical setup.

A quick and robust method for BIM-BEM transformation using the implicit data exchange approach was developed in this case study. It was found that the efficiency of the data exportation was unaffected by the increasing refinement of the original BIM models at ascending BDLs. In addition, an automated geometry transformation was successfully developed without any manual iterative geometry fix. However, additional manipulations were required for semantic data transformation due to unit incompatibility between the BIM authoring tool and simulation tool internal unit systems. The parametrization and generation of BEM models batch showed precise results. A strong impact of the level of refinement of the transformed BEM models on the duration required for generating batch of design variants was detected. A similar impact was found on the annual loads simulation on the batch of BEM models.

The lack of precise data relevant for a detailed energy simulation in the BIM model at early design stages was overcome by applying the Model Healing concept. Indeed, options from pre-determined base construction set and building program were utilized to fulfill the requirement of detailed energy simulation. Comparing the annual loads simulation results of the batch of BEM models at BDL 3 and 4, a variation of 5% in the mean value of total load intensity and corresponding contribution of member loads was recorded. This indicates that a detailed energy simulation with reliable results is possible in the early design stages with BIM models at BDL 3.

The performance of the SA with Morris method could be assessed based on two aspects, namely a clear separation between influential and non-influential parameters, and an apparent and robust ranking of parameters. Considering the results of SA with Morris method in this case study, the differentiation between influential and non-influential was indistinct. All parameters involved in the analysis were categorized as influential with non-linear and/or interaction effects. In addition, the parameter ranking using the absolute mean value μ^* of the elementary effects was potentially unstable, as the confident intervals were high for both BDL 3 and 4. Furthermore, the varying parameter rankings observed from the two analyses for different BDLs indicate the shift of parameter influences over BEM model at ascending BDLs. Consequently, it is beneficial to perform SA for BEM models at each BDL instead of assuming the parameter influences based solely on the results from BEM models at lower BDLs.

5.6.2 Limitations

In the scope of this thesis, the topic of BIM-based BEPS was partly evaluated. The BIM-BEM transformation process investigated only the implicit data exchange approach while neglecting the explicit approach using vendor neutral, open-source exchange formats.

An automated model checking procedure is absent in this study. Instead, only one simple verification of the exported geometrical and semantic data after the BIM-BEM data exchange was executed using available Honeybee components. Considering parametrization and subsequent generation of BEM models batch, the framework lacks a detailed model checking procedure to detect potential errors.

In this study, only one SA method, namely the Morris method, was utilized to investigate the parameter influences. Hence, a comparison between results of different SA methods is unavailable.

Furthermore, the efforts for the technical setup of the BIM authoring tool, the parametric modeling and simulation tool, and the SA package were not evaluated since these efforts are highly dependent on the familiarity of the users with the software environment.

6 Conclusion & Outlook

This chapter concludes the study by answering research questions and proposes opportunities for future research. In addition, a review on the value and contribution of the study to the general body of knowledge is presented.

6.1 Research Questions

6.1.1 Question 1

- How can a framework of BEPS originating from uncertain BIM models with SA improve the decision-making process in the early design stages?

Answer:

The proposed framework in this thesis helps shift the BEPS into the early design stages. The use of early design BIM model as the core data model is sufficient to perform detailed BEPS. In addition, the SA on the output of the BEPS provides parameter rankings and classification of inputs in terms of influence. The result of SA enables the opportunity for designers to determine influential input parameters and subsequently aim for model-based optimization at an early design stage. Consequently, the decisions made in the early design stage become more influential towards the actual building performance.

6.1.2 Question 2

- What are the minimum information requirements for early design BIM models to secure detailed BEPS?

Answer:

Detailed BEPS in early design stages could be performed using BIM models at BDL 3. The basic information requirements for a BIM model at BDL 3 include information regarding the building dimensions, structural system, construction type, material, and definition of building space with defined stories and room usage. In addition, the lack of precise data relevant for a detailed energy simulation in the BIM models at early design stages could be overcome by providing alternatives from pre-determined base construction set and building program.

6.1.3 Question 3

- What are the most and least sensitive parameters for building energy performance in the early design stages?

Answer:

Based on results from the SA using the Morris method on the total load intensity of BEM models at ascending BDLs, the ranking of parameter influences varies over different BDLs. For BEM models at BDL 3, the most sensitive parameters are building width, heating COP, and building length. On the other hand, the external wall thickness, building height, and external rood thickness are most sensitive in the case of BEM models at BDL 4. The parameters within Windows group, namely window area, infiltration, and window to wall ratio, have the least influence across the BDLs.

6.1.4 Question 4

- How does the proposed framework contribute to the improvement of the existing BIM-BEM interoperability?

Answer:

The proposed framework in this thesis provides a simple but quick and robust method to perform BIM-BEM transformation. The computational efficiency of the BIM-BEM transformation is fast and independent on the refinement of the original BIM model. Automated geometry transformation was successfully obtained. However, the overall BIM-BEM transformation process is still semi-automated as the exportation of semantic data required additional manipulation due to incompatible internal unit system. Moreover, the framework is a closed BIM approach limiting to Autodesk Revit.

6.2 Outlooks

Based on the elaborated topics, some recommendations are available for future enhancements of this study. First, it is recommended that other combinations of tools should be implemented to evaluate the interoperability between Autodesk Revit and McNeel Rhinoceros. For instance, the Pollination plugin and the Data Exchange for Dynamo/Grasshopper are notable nominations to carry the BIM-BEM transformation task. With the utilization of different software combinations, the comparison of BIM-BEM transformation results using different approaches is available to give more in-depth validation of the BIM-BEM interoperability.

Second, parallel energy simulation of the BEM models batch could be implemented when computational resources are available. For simulation of larger scale, the use of cloud computing could be investigated. For instance, the Pollination Cloud Computing and the OpenStudio Parametric Analysis Tool are available cloud services specified for energy simulation purposes.

In addition, an automated or semi-automated model checking procedure should be developed to inspect the results of the BIM-BEM transformation and BEM model parametrization process.

Considering the SA on the BEPS results, it is beneficial to perform a more sophisticated analysis, such as the variance-based Sobol’ method when computational resources are sufficient. The enlargement of the simulation space in terms of number of samples and number of inputs is possible. In case of higher number of inputs, a hybrid SA approach which utilizes the Morris method to identify the non-influential parameters prior to the computationally expensive Sobol’ method is recommended. In addition, the use of optimized techniques and other PDFs when sampling the input parameters should be investigated.

Eventually, the integration of machine learning into the framework to predict the building energy use and capture the effects of input parameters in early design stages on building energy use should be investigated. The use of machine learning is beneficial in the case of higher number samples. For instance, such approach to integrate machine learning and convolutional neural network into early stage energy design is described by Singh and Smith (2023).

6.3 Conclusion

This study aimed to develop a framework for BIM-based parametric BEPS with SA to support the decision-making process in the early design stages. Based on the results of the case study in this thesis, the following statements are applicable to the proposed BIM-based parametric BEPS with SA. First, a semi-automated BIM-BEM workflow with quick and robust BIM-BEM transformation and detailed energy simulation originating from early design BIM models are feasible. Next, the computational efficiency of the energy simulation is directly proportional to the refinement of the original BIM model and the available computational resources. Finally, the results from SA on the energy simulation output provide dynamic assistance for the decision-making process in the early design stages. The main limitation of the study is the implemented closed BIM approach confined to only Autodesk Revit.

Bibliography

Abualdenien, J.; Borrmann, A. (2019): A meta-model approach for formal specification and consistent management of multi-LOD building models. In *Advanced Engineering Informatics*, pp. 20–42.

Adams, B. M.; Bohnhoff, W. J.; Dalbey, K.; Ebeida, M. S.; Eddy, J. P.; Eldred, M. S. et al. (2020): Dakota A Multilevel Parallel Object-Oriented Framework for Design Optimization Parameter Estimation Uncertainty Quantification and Sensitivity Analysis: Version 6.12 User's Manual. Livermore, CA (United States): Sandia National Lab.

Aksamija, A.; Brown, D. N. (2018): Integration of Parametric Design Methods and Building Performance Simulations for High-Performance Buildings: Methods and Tools. In *Research Journal* 10.01, Article 02. Available online at https://www.semanticscholar.org/paper/Integration-of-Parametric-Design-Methods-and-for-Aksamija-Brown/9200f3768e744a5cf2e3010003aa1f3c3551b3f9.

Andriamamonjy, A.; Saelens, D.; Klein, R. (2018): An automated IFC-based workflow for building energy performance simulation with Modelica. In *Automation in Construction* 91, pp. 166–181. DOI: 10.1016/j.autcon.2018.03.019.

Asl, M.; Zarrinmehr, S.; Yan, W. (2013): Towards BIM-based Parametric Building Energy Performance Optimization. In Philip Beesley, Omar Khan, Michael Stacey (Eds.): Proceedings of the 33rd Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA). ACADIA 2013: Adaptive Architecture. Cambridge (Ontario), Canada, 10/24/2013 - 10/26/2013: ACADIA (ACADIA proceedings), pp. 101–108.

Asl, M. R.; Bergin, M.; Menter, A.; Yan, W. (2014): BIM-based Parametric Building Energy Performance Multi-Objective Optimization. In Emine Mine Thompson (Ed.): Proceedings of the 32nd International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe) [Volume 2]. eCAADe 2014: Fusion. Newcastle upon Tyne, UK, 9/10/2014 - 9/12/2014: eCAADe (eCAADe proceedings), pp. 455–464.

Autodesk Inc. (2021): Revit 2022 Release Notes: Enhancements. Available online at https://help.autodesk.com/view/RVT/2022/ENU/?guid=RevitReleaseNotes_2022release_Enhancements_html.

Autodesk Inc. (2023): Revit Developer's Guide. Available online at https://help.autodesk.com/view/RVT/2024/ENU/?guid=Revit_API_Revit_API_Developers_Guide_Introduction_Application_and_Document_Units_html.

Baudin, M.; Dutfoy, A.; Iooss, B.; Popelin, AL. (2017): OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation. In *Handbook of Uncertainty Quantification*. DOI: 10.1007/978-3-319-12385-1_64.

Bidot, C.; Lamboni, M.; Monod, H. (2022): Package ‘multisensi’. Multivariate Sensitivity Analysis. Version 2.1-1. Available online at https://cran.r-project.org/web/packages/multisensi/multisensi.pdf.

Borrmann, A.; König, M.; Koch, C.; Beetz, J. (2018): Building Information Modeling: Why? What? How? In A. Borrmann, M. König, C. Koch, J. Beetz (Eds.): Building Information Modeling. Cham: Springer International Publishing, pp. 1–24.

Building Energy Codes Program (2023): Prototype Building Models. Available online at https://www.energycodes.gov/prototype-building-models#Commercial.

buildingSMART International (2023): Industry Foundation Classes (IFC) – An Introduction. buildingSMART International, Ltd. Available online at https://technical.buildingsmart.org/standards/ifc/.

Campolongo, F.; Cariboni, J.; Saltelli, A. (2007): An effective screening design for sensitivity analysis of large models. In *Environmental Modelling & Software* (22(10)), pp. 1509–1518. DOI: 10.1016/j.envsoft.2006.10.004.

Carlos, M.; Velez, S. (2023): Global sensitivity and uncertainty analysis (GSUA). Version 4.5: MATLAB Central File Exchange. Available online at https://www.mathworks.com/matlabcentral/fileexchange/47758-global-sensitivity-and-uncertainty-analysis-gsua?s_tid=srchtitle.

Carvalho, J.; Almeida, M.; Bragança, L.; Mateus, R. (2021): BIM-Based Energy Analysis and Sustainability Assessment—Application to Portuguese Buildings. In *Buildings* 11 (6), p. 246. DOI: 10.3390/buildings11060246.

Casini, M. (2022): Building performance simulation tools. In Woodhead Publishing Series in Civil and Structural Engineering. In *Construction 4.0*, pp. 221–262. DOI: 10.1016/B978-0-12-821797-9.00004-0.

Choi, J.; Shin, J.; Kim, M.; Kim, I. (2016): Development of openBIM-based energy analysis software to improve the interoperability of energy performance assessment. In *Automation in Construction* 72 (0926-5805), pp. 52–64. Available online at https://doi.org/10.1016/j.autcon.2016.07.004.

Congiu, E.; Desogus, G.; Quaquero, E.; Argiolas, R.; Di Pilla, L.; Frau, V. (2022): BIM-based workflow for energy audits. In : Proceedings of the 2022 European Conference on Computing in Construction. 2022 European Conference on Computing in Construction, Jul. 24, 2022: University of Turin (Computing in Construction).

DeRobertis, N. (2022): sensitivity 0.2.8. Version 0.2.8. Available online at https://pypi.org/project/sensitivity/#description.

EU Science Hub (2023): SIMLAB and other software. Available online at https://joint-research-centre.ec.europa.eu/sensitivity-analysis-samo/simlab-and-other-software_en.

European Commission (2019): The European Green Deal. Available online at https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1576150542719&uri=COM%3A2019%3A640%3AFIN.

European Commission (2020): Energy efficiency in buildings. Available online at https://commission.europa.eu/news/focus-energy-efficiency-buildings-2020-02-17_en.

flax (2023): Global Sensitivity Analysis Toolbox. Version 1.57.0.0: MATLAB Central File Exchange. Available online at https://www.mathworks.com/matlabcentral/fileexchange/40759-global-sensitivity-analysis-toolbox.

Forth, K. (2023): Multilingual semantic enrichment of room-specific load profiles using BIM models for whole building energy simulation. In *Forum Bauinformatik* 34. DOI: 10.13154/294-10093.

Gao, H.; Koch, C.; Wu, Y. (2019a): Building information modelling based building energy modelling: A review. In *Applied Energy* 238, pp. 320–343. DOI: 10.1016/j.apenergy.2019.01.032.

Gao, H.; Zhang, L.; Koch, C.; Wu, Y. (2019b): BIM-based real time building energy simulation and optimization in early design stage. In *IOP Conf. Ser.: Mater. Sci. Eng.* 556 (1), p. 12064. DOI: 10.1088/1757-899X/556/1/012064.

Giannakis, Georgios I.; Katsigarakis, Kyriakos I.; Lilis, Georgios Nektarios; Rovas, Dimitrios V. (2020): A Workflow for Automated Building Energy Performance Model Generation Using BIM Data. In Vincenzo Corrado, Enrico Fabrizio, Andrea Gasparella, Francesco Patuzzi (Eds.): Proceedings of Building Simulation 2019: 16th Conference of IBPSA. Building Simulation 2019. Rome, Italy, 9/2/2019 - 9/4/2019: IBPSA (Building Simulation Conference proceedings), pp. 167–174.

Harter, H.; Singh, Manav M.; Schneider-Marin, P.; Lang, W.; Geyer, P. (2020): Uncertainty Analysis of Life Cycle Energy Assessment in Early Stages of Design. In *Energy and Buildings* 208, p. 109635. DOI: 10.1016/j.enbuild.2019.109635.

Hemsath, L.; Bandhosseini, K. (2015): Sensitivity analysis evaluating basic building geometry's effect on energy use. In *Renewable Energy* 76, pp. 526–538. DOI: 10.1016/j.renene.2014.11.044.

Herman, J.; Usher, W.; Iwanaga, T. (2023): SALib. Version 1.4.7. Available online at https://salib.readthedocs.io/en/latest/api/SALib.html.

IfcOpenShell Contributors (2023): IfcOpenShell. Version 0.7.0. Available online at https://ifcopenshell.org/.

Iooss, B.; Da Veiga, S.; Janon, A.; Pujol, G. (2023): Package ‘sensitivity’. Global Sensitivity Analysis of Model Outputs. Version 1.29.0: CRAN. Available online at https://cran.r-project.org/web/packages/sensitivity/sensitivity.pdf.

Iwanaga, T.; Herman, J.; Usher, W. (2022): Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. In *Socio-Environmental Systems Modelling* 4 (18155). Available online at doi:10.18174/sesmo.18155.

Jin, R.; Zhong, B.; Ma, L.; Hashemi, A.; Ding, L. (2019): Integrating BIM with building performance analysis in project life-cycle. In *Automation in Construction* 106 (102861), Article ISSN 0926-5805. DOI: 10.1016/j.autcon.2019.102861.

Kamel, E.; Memari, A. M. (2019): Review of BIM's application in energy simulation: Tools, issues, and solutions. In *Automation in Construction* 97, pp. 164–180. DOI: 10.1016/j.autcon.2018.11.008.

Kolbeck, L. (2020): Interoperability of BIM-based Life-Cycle Energy Analysis in Early Design Stages. Master. Technical University of Munich. Chair of Computational Modeling and Simulation. Available online at https://mediatum.ub.tum.de/doc/1578648/1578648.pdf.

Konis, K.; Gamas, A.; Kensek, K. (2016): Passive performance and building form: An optimization framework for early-stage design support. In *Solar Energy* 125, pp. 161–179. DOI: 10.1016/j.solener.2015.12.020.

Kristensen, M. H.; Petersen, S. (2016): Choosing the appropriate sensitivity analysis method for building energy model-based investigations. In *Energy and Buildings* 130, pp. 166–176. DOI: 10.1016/j.enbuild.2016.08.038.

Kucherenko, S.; Zaccheus, O. (2020): SobolGSA Software: Imperial College London. Available online at https://www.imperial.ac.uk/process-systems-engineering/research/free-software/sobolgsa-software/.

Ladybug Tools LLC (2023): Ladybug Tools Plugin for Grasshopper Documentation. Available online at https://discourse.ladybug.tools/pub/lbt-grasshopper-doc.

Laine, T.; Karola, A. (Eds.) (2007): Benefits of Building Information Models in Energy Analysis. Clima 2007 WellBeing Indoors. Available online at https://www.irbnet.de/daten/iconda/CIB8170.pdf.

Lawrence Berkeley National Laboratory (2013): Space Boundary Tool. Version SBT-1. Available online at https://simulationresearch.lbl.gov/projects/space-boundary-tool.

Maile T.; Fischer, M.; Bazjanac, V. (2007): Building Energy Performance Simulation Tools - a Life-Cycle and Interoperable Perspective. In *CIFE Working Paper*, Article #WP107. Available online at https://www.researchgate.net/publication/237621385_Building_Energy_Performance_Simulation_Tools_-_a_Life-Cycle_and_Interoperable_Perspective.

Mara, T. A.; Tarantola, S. (2008): Application of global sensitivity analysis of model output to building thermal simulations. In *Building Simulation* 1 (4), pp. 290–302. DOI: 10.1007/s12273-008-8129-5.

Marelli, S.; Sudret, B. (Eds.) (2014): UQLab: A Framework for Uncertainty Quantification in MATLAB. The 2nd International Conference on Vulnerability and Risk Analysis and Management (ICVRAM 2014). University of Liverpool, United Kingdom. Available online at https://ascelibrary.org/doi/10.1061/9780784413609.257.

Maučec, D.; Premrov, M.; Leskovar, V. Ž. (2021): Use of sensitivity analysis for a determination of dominant design parameters affecting energy efficiency of timber buildings in different climates. In *Energy for Sustainable Development* 63, pp. 86–102. DOI: 10.1016/j.esd.2021.06.003.

McNeel (2021): Grasshopper Hops Python servers. Available online at https://github.com/mcneel/compute.rhino3d/blob/7.x/src/ghhops-server-py/README.md.

Moradi, A.; Kavgic, M.; Costanzo, V.; Evola, G. (2023): Impact of typical and actual weather years on the energy simulation of buildings with different construction features and under different climates. In *Energy* Volume 270. DOI: 10.1016/j.energy.2023.126875.

Morris, M. D. (1991): Factorial Sampling Plans for Preliminary Computational Experiments. In *Technometrics* 33, pp. 161–174. DOI: 10.1080/00401706.1991.10484804.

National Renewable Energy Laboratory: EnergyPlus. Available online at https://energyplus.net/.

National Renewable Energy Laboratory (2023a): Building Energy Modeling. Available online at https://www.nrel.gov/buildings/building-energy-modeling.html.

National Renewable Energy Laboratory (2023b): OpenStudio. Available online at https://openstudio.net/.

Neale, J.; Shamsi, M. H.; Mangina, E.; Finn, D.; O'Donnell, J. (2022): Accurate identification of influential building parameters through an integration of global sensitivity and feature selection techniques. In *Applied Energy* 315, p. 118956. DOI: 10.1016/j.apenergy.2022.118956.

Nouri, M. A. (2023): Building energy performance simulation benchmarking by meta-model-based sensitivity assessment. Doctor. Rheinisch-Westfälischen Technische Universität Aachen. Available online at https://publications.rwth-aachen.de/record/962549/files/962549.pdf.

O'Donnell, J. T.; van Dessel, M.; Maile, T. (2020): BIM to Building Energy Performance Simulation: An Evaluation of Current Industry Transfer Processes. In Vincenzo Corrado, Enrico Fabrizio, Andrea Gasparella, Francesco Patuzzi (Eds.): Proceedings of Building Simulation 2019: 16th Conference of IBPSA. Building Simulation

2019. Rome, Italy, 9/2/2019 - 9/4/2019: IBPSA (Building Simulation Conference proceedings), pp. 92–99.

Office of Energy Efficiency & Renewable Energy (2022): Collaboration + Automation = Pollination. Available online at https://www.energy.gov/eere/buildings/articles/collaboration-automation-pollination.

Østergård, T.; Maagaard, S.; Lund Jensen, R. (Eds.) (2016): Thermal Comfort in Residential Buildings by the Millions -Early Design Support from Stochastic Simulations. 12th REHVA World Congress CLIMA 2016. Aalborg, Denmark.

Pang, Z.; O'Neill, Z.; Li, Y.; Niu, F. (2020): The role of sensitivity analysis in the building performance analysis: A critical review. In *Energy and Buildings* 209, p. 109659. DOI: 10.1016/j.enbuild.2019.109659.

Pianosi, F.; Beven, K.; Freer, J.; Hall, J. W.; Rougier, J.; Stephenson, D. B.; Wagener, T. (2016): Sensitivity analysis of environmental models: A systematic review with practical workflow. In *Environmental Modelling & Software* 79, pp. 214–232. DOI: 10.1016/j.envsoft.2016.02.008.

Pianosi, F.; Sarrazin, F.; Wagener, T. (2015): A Matlab toolbox for Global Sensitivity Analysis. In *Environmental Modelling & Software* 70 (1364-8152), pp. 80–85. DOI: 10.1016/j.envsoft.2015.04.009.

Pollination (2023): User Manual. Available online at https://docs.pollination.cloud/user-manual/.

Puy, A.; Lo Piano, S.; Saltelli, A.; Levin, S. (2021): sensobol: an R package to compute variance-based sensitivity indices. Available online at https://arxiv.org/pdf/2101.10103.pdf.

Qingsong, M.; Fukuda, H. (2016): Parametric Office Building for Daylight and Energy Analysis in the Early Design Stages. In *Procedia - Social and Behavioral Sciences* 216, pp. 818–828. DOI: 10.1016/j.sbspro.2015.12.079.

Ramaji, I. J.; Messner, J. I.; Mostavi, E. (2020): IFC-Based BIM-to-BEM Model Transformation. In *J. Comput. Civ. Eng.* 34 (3), Article 04020005. DOI: 10.1061/(ASCE)CP.1943-5487.0000880.

Razavi, S.; Sheikholeslami, R.; Gupta, H. V.; Haghnegahdar, A. (2019): VARS-TOOL: A toolbox for comprehensive, efficient, and robust sensitivity and uncertainty analysis. In *Environmental Modelling & Software* 112, pp. 95–107. DOI: 10.1016/j.envsoft.2018.10.005.

Reisinger, J.; Kovacic, I. (Eds.) (2019): BIM-based workflows for building energy modelling. International Conference on Sustainable Materials, Systems and Structures – SMSS 2019. Rovinj, Croatia. Available online at https://www.researchgate.net/publication/333045203_BIM-BASED_WORKFLOWS_FOR_BUILDING_ENERGY_MODELLING_-_A_VARIANT_STUDY.

Robert McNeel & Associates (2023): Rhino.Inside.Revit Guides. Available online at https://www.rhino3d.com/inside/revit/1.0/guides/.

Roudsari, M. (2021): Principles of Pollination. Pollination. Available online at https://www.pollination.cloud/blog/principles-of-pollination.

Ruano, M. V.; Ribes, J.; Seco, A.; Ferrer, J. (2012): An improved sampling strategy based on trajectory design for application of the Morris method to systems with many input factors. In *Environmental Modelling & Software* 37, pp. 103–109. DOI: 10.1016/j.envsoft.2012.03.008.

Schneider-Marin, P.; Harter, H.; Tkachuk, K.; Lang, W. (2020): Uncertainty Analysis of Embedded Energy and Greenhouse Gas Emissions Using BIM in Early Design Stages. In *Sustainability* 12 (7), p. 2633. DOI: 10.3390/su12072633.

Shahsavari, F.; Yan, W.; Koosha, R. (Eds.) (2019): A case study for sensitivity-based building energy optimization. ARCC 2019. Toronto, Canada. 1 volume. Available online at https://www.arcc-journal.org/index.php/repository/article/view/644.

Singh, M. M.; Geyer, P. (2019): Information requirements for multi-level-of-development BIM using sensitivity analysis for energy performance. In *Advanced Engineering Informatics* 43, p. 101026. DOI: 10.1016/j.aei.2019.101026.

Singh, M. M.; Singaravel, S.; Klein, R.; Geyer, P. (2020): Quick energy prediction and comparison of options at the early design stage. In *Advanced Engineering Informatics* 46, S. 101185. DOI: 10.1016/j.aei.2020.101185.

Singh, M. M.; Smith, I. F. C. (2023): Convolutional neural network to learn building-shape representations for early-stage energy design. In *Energy and AI* 14 (100293). DOI: 10.1016/j.egyai.2023.100293.

Spielhaupter, O. (2021): BIM to BEM Transformation Workflows: A Case Study Comparing Different IFC-Based Approaches. Master, Wien, Austria. Technischen Universität Wien. Available online at https://repositum.tuwien.at/handle/20.500.12708/18535, checked on 5/7/2023.

Tokarzewski, B. (2020): Performance Analysis of Integrated Building Systems. Coursework: Integrated Building Systems Simulation. The Bartlett School of Environment, Energy and Resources, University College London. London, England. Available online at https://github.com/Tokarzewski/UCL-SBDE-CW/blob/master/BENV0088%20-%20Integrated%20Building%20Systems%20Simulation/%5BBENV0088%5D_%5BFBXX7%5D_%5B05052020%5D.pdf.

Tong, C. (2015): PSUADE Reference Manual (v1.7). Livermore, CA (United States): Lawrence Livermore National Laboratory. Available online at https://github.com/LLNL/psuade.

Touloupaki, E.; Theodosiou, T. (2017): Performance Simulation Integrated in Parametric 3D Modeling as a Method for Early Stage Design Optimization—A Review. In *Energies* 10 (5), p. 637. DOI: 10.3390/en10050637.

Videau, P. (2023): Data Exchange Connectors for Dynamo and Grasshopper Now Available in Public Beta. With assistance of Autodesk Inc. Available online at https://blogs.autodesk.com/aec/2023/03/01/data-exchange-connectors-for-dynamo-and-grasshopper-now-available-in-public-beta/.

Wicaksono, D. (2016): Morris Screening Method. Available online at https://gsa-module.readthedocs.io/en/stable/implementation/morris_screening_method.html.

Wu, J.; Dubey, R.; Abualdenien, J.; Borrmann, A. (2022): Model healing: Toward a framework for building designs to achieve code compliance. In *Proc. of European Conference on Product and Process Modeling 2022*. Available online at https://mediatum.ub.tum.de/1155878?query=Model+Healing%3A+Toward+a+framework+for+building+designs+to+achieve+code+compliance&show_id=1687827&srcnodeid=1155878.

Zardo, P.; Ribeiro, A.; Mussi, A. (2019): BIM AND PARAMETRIC DESIGN APPLICATIONS FOR BUILDINGS' ENERGY EFFICIENCY: AN ANALYSIS OF PRACTICAL APPLICATIONS. In *Arquitetura Revista* 15 (2). DOI: 10.4013/arq.2019.152.02.

Appendix A

Case Study

A.1 Software environment

Table A-1: Software package used in the case study

Software Tool	Version
Autodesk Revit	v 24.2.0.63
Rhinoceros	v 7.34.23267.11001
Rhino.Inside.Revit	v 1.16.8620.27572
Ladybugs Tool for Grasshopper	v 1.7.0
OpenStudio	v 3.6.1+bb9481519e
EnergyPlus	v 23.1.0-87ed9199d4
SALib	v 1.4.7

A.2 Autodesk Revit

A.2.1 Elements thermal properties

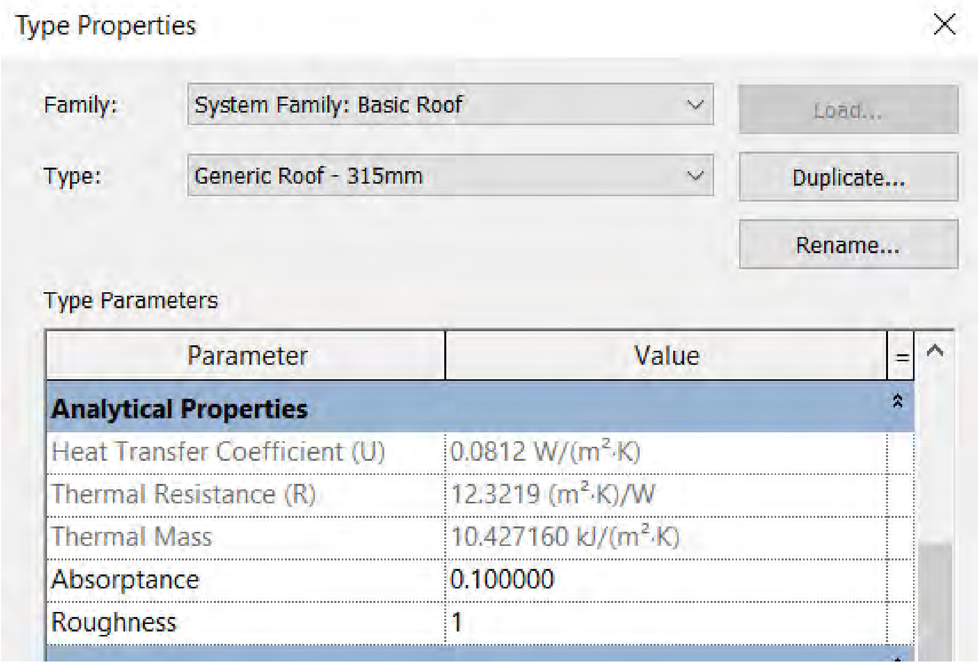


Figure A-2.1: Roof thermal properties

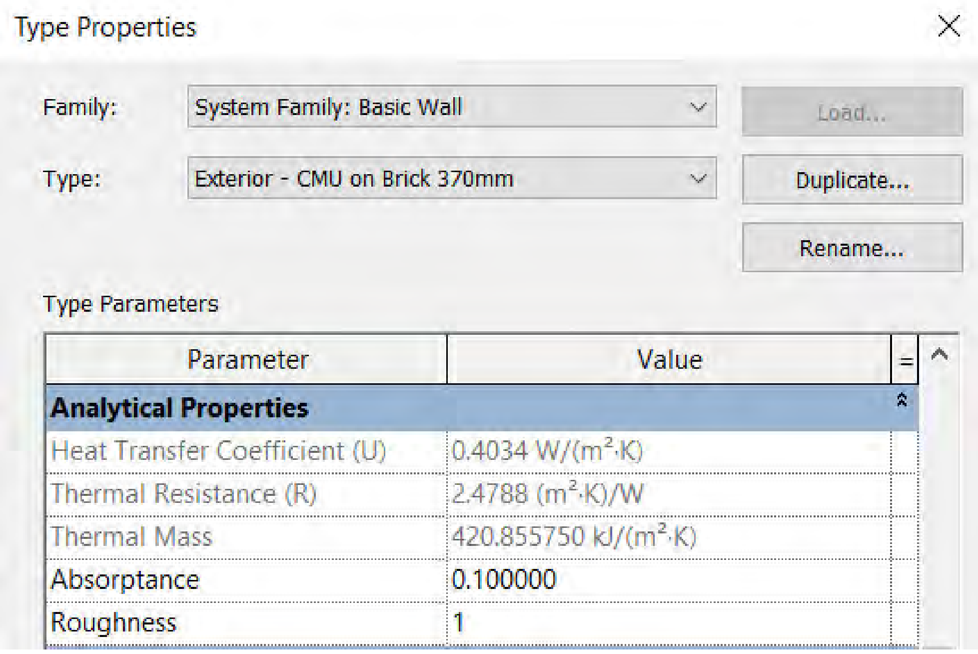


Figure A-2.2: Exterior wall thermal properties

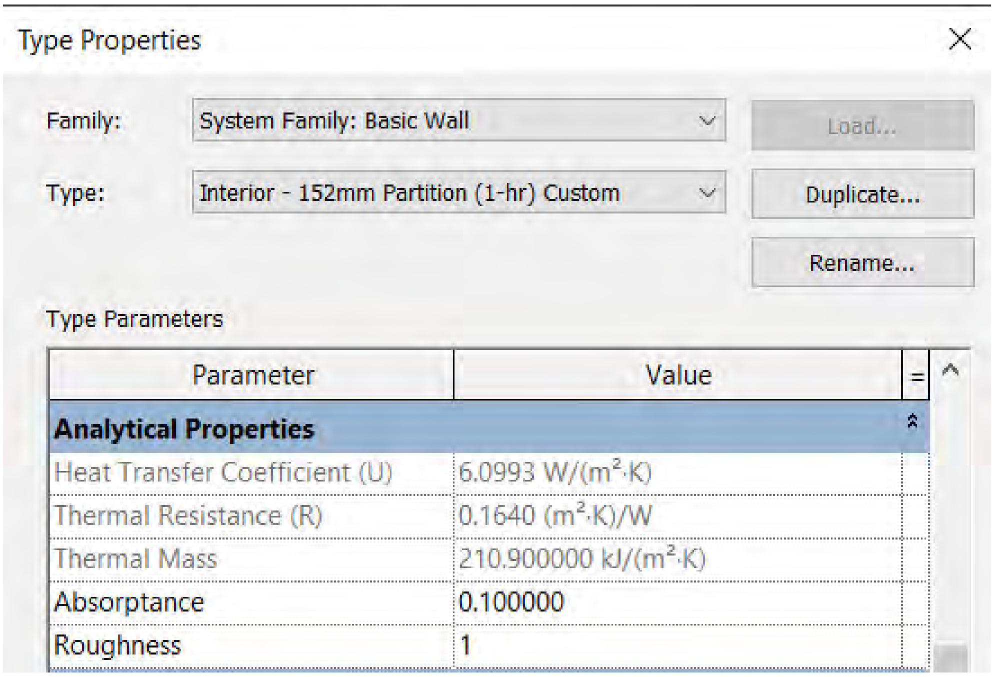


Figure A-2.3: Interior wall thermal properties

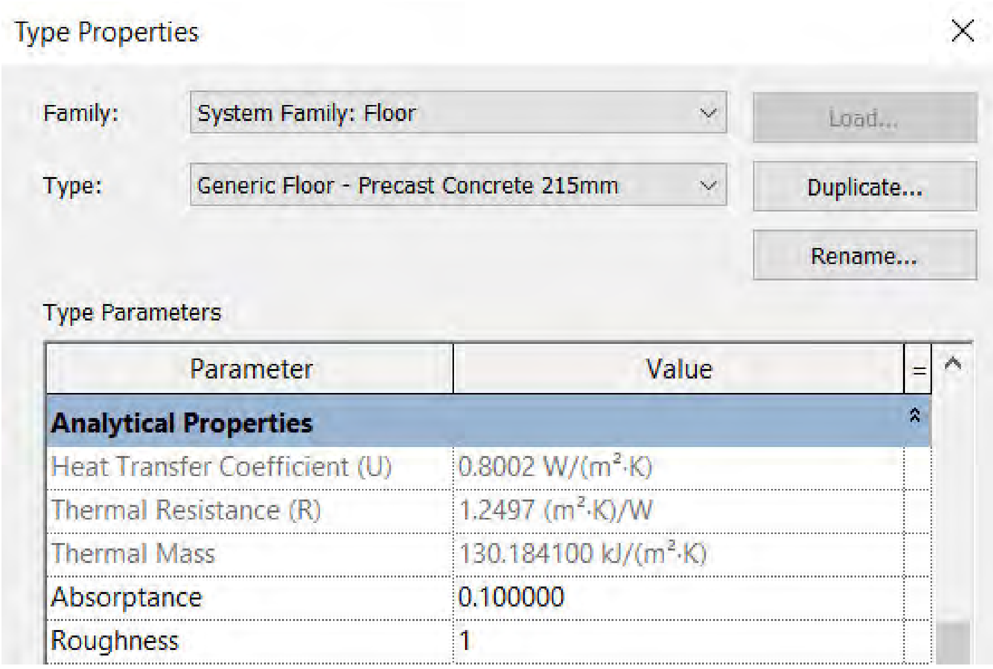


Figure A-2.4: Floor thermal properties

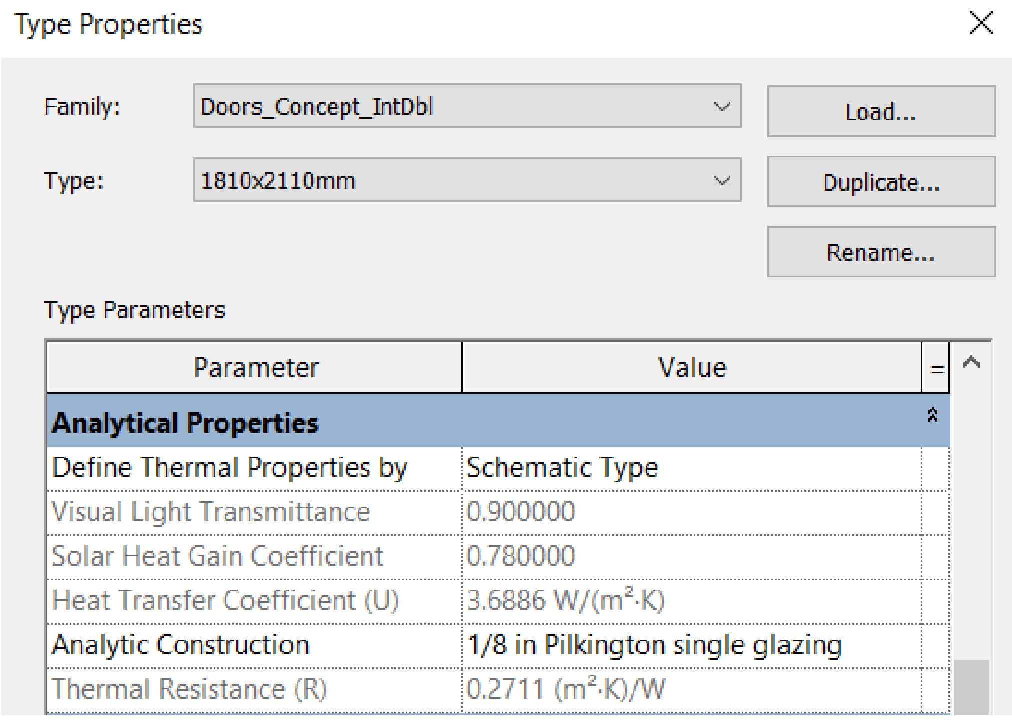


Figure A-2.5: Door thermal properties

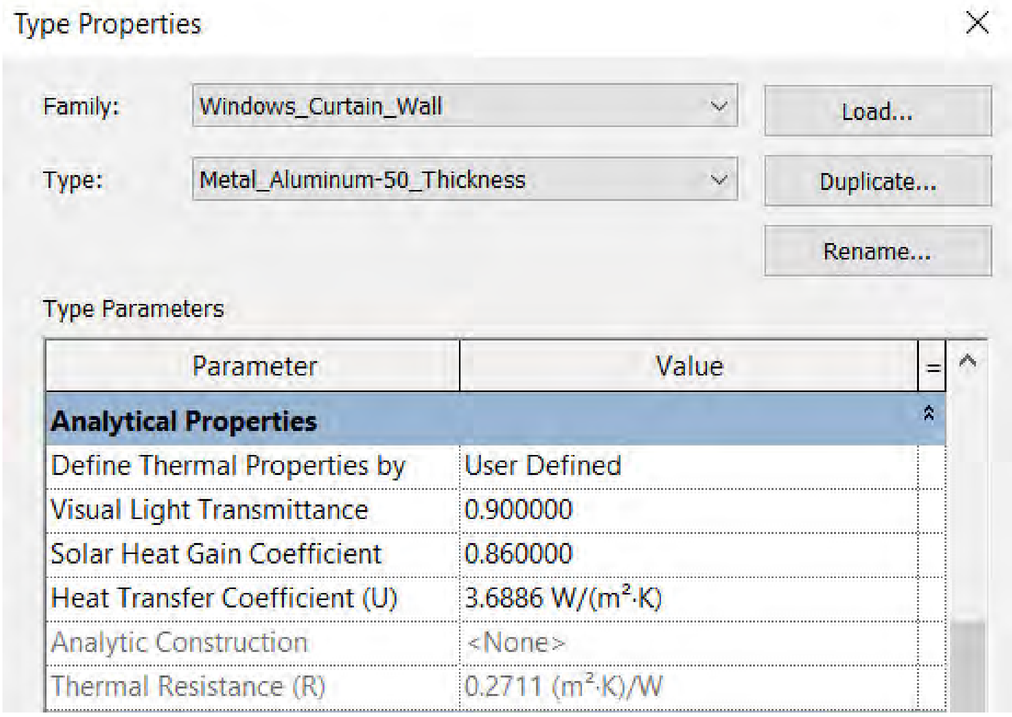


Figure A-2.6: Interior wall thermal properties

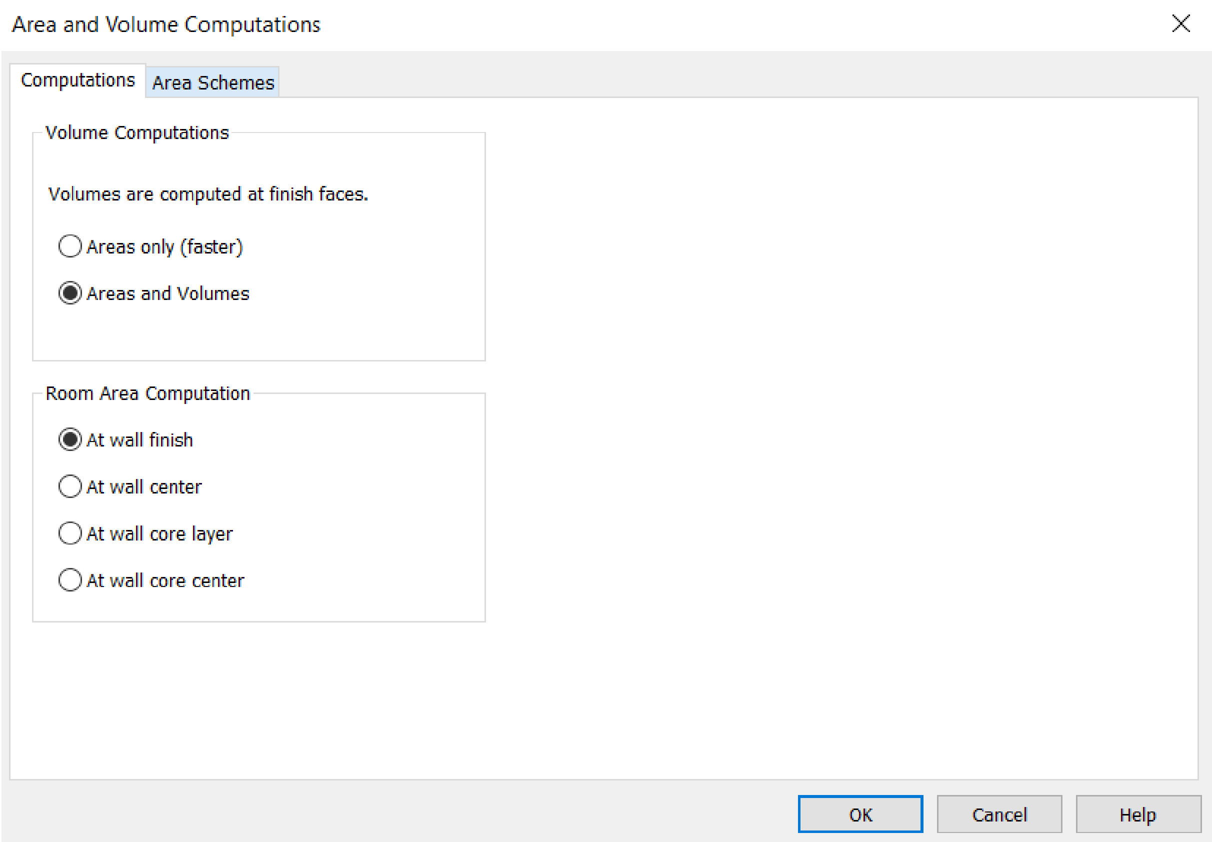
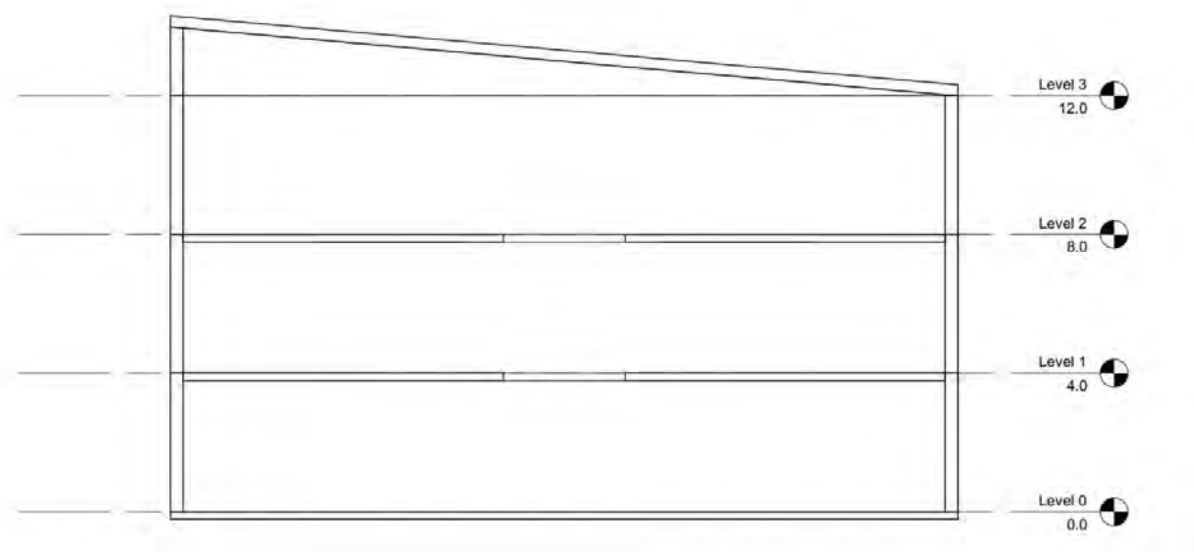


Figure A-2.7: Area and Volume Computations settings

A.2.2 BIM models



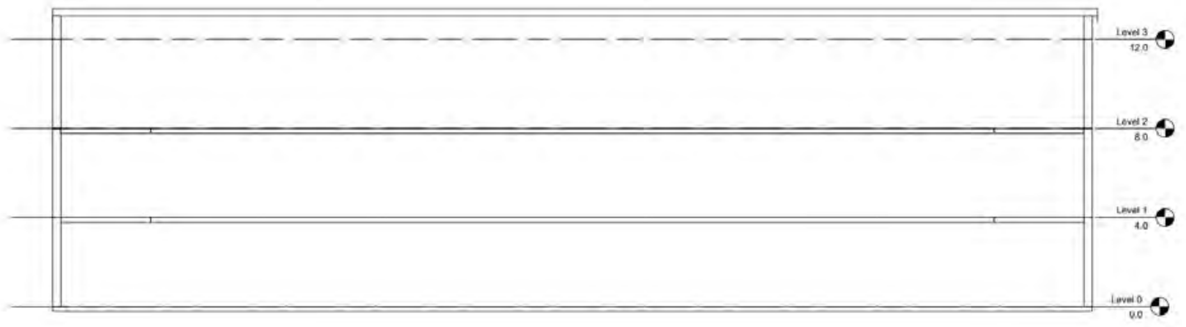


Figure A-2.8: Horizontal and vertical section of the BIM model at BDL 3

Figure A-2.9: Horizontal and vertical section of the BIM model at BDL 4

A.3 Rhino.Inside.Revit

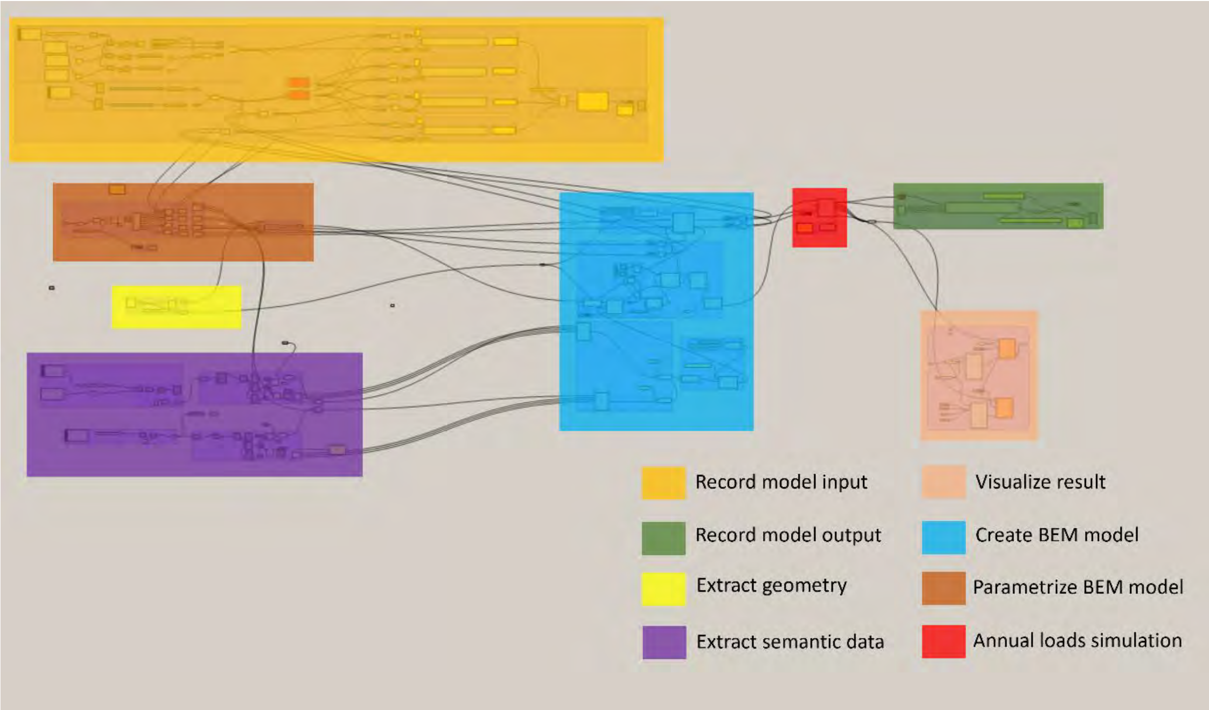


Figure A-3.1: Grasshopper script for the BDL 3 case

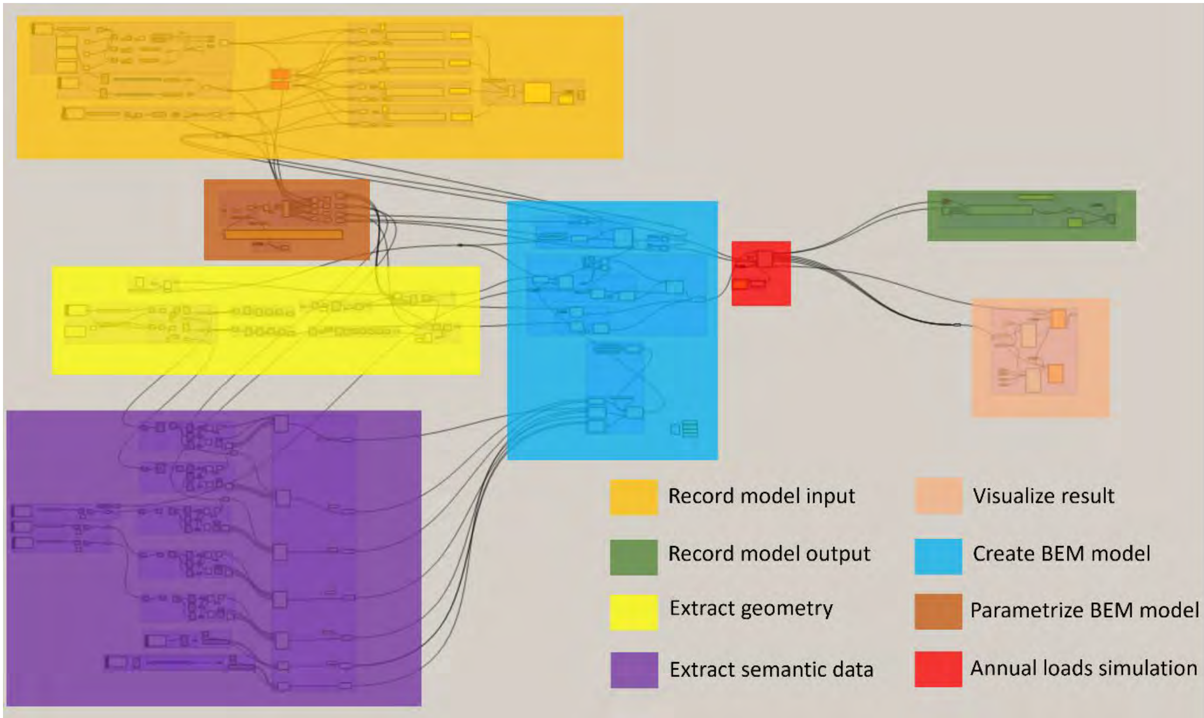


Figure A-3.2: Grasshopper script for the BDL 4 case

A.4 SALib

```python
from SALib.sample.morris import sample

from SALib.util import read_param_file
import numpy as np


problem1 = read_param_file(r"C:\Users\daovi\Documents\data\BDL3_m\parameter_file.txt")
param_values1 = sample(problem1, N=240, num_levels=4, optimal_trajectories=None)
print(param_values1.shape)

with open(r"C:\Users\daovi\Documents\data\Morris_nonop\BDL3\model_input.txt", 'w') as mi1:
    np.savetxt(mi1,param_values1)

problem2 = read_param_file(r"C:\Users\daovi\Documents\data\BDL4_m\parameter_file.txt")
param_values2 = sample(problem2, N=240, num_levels=4, optimal_trajectories=None)
print(param_values2.shape)

with open(r"C:\Users\daovi\Documents\data\Morris_nonop\BDL4\model_input.txt", 'w') as mi2:
    np.savetxt(mi2,param_values2)
```

Figure A-4.1: Python scripts for sampling of model inputs

```python
from SALib.analyze import morris
from SALib.util import read_param_file
from SALib.plotting.morris import horizontal_bar_plot
import matplotlib.pyplot as plt
import numpy as np

# Import data:
problem = read_param_file(r"C:\Users\daovi\Documents\data\Morris\BDL3\parameter_file.txt")
problem.pop('groups')

param_values = np.loadtxt(r"C:\Users\daovi\Documents\Jiabin_BDL3_1200-withresults\Jiabin_BDL3_1200\model_input.txt")
data = np.loadtxt(r"C:\Users\daovi\Documents\Jiabin_BDL3_1200-withresults\Jiabin_BDL3_1200\model_output.txt")

print("data ndim: ", data.ndim)
print("data shape:", data.shape)
print("data size: ", data.size)


# Perform Morris SA:
Y_0 = data[:,0]

Si_0 = morris.analyze(
    problem,
    param_values,
    Y_0,
    conf_level=0.95,
    print_to_console=True,
    num_levels=4,
    num_resamples=100,
)
```

```python
# Visualize results:
fig1, (ax1,ax2) = plt.subplots(1,2)
colors = ["blue", "orange", "red", "purple", "brown", "pink", "gray", "cyan", "olive"]
names = Si_0["names"]
sigma = Si_0["sigma"]
mu_star = Si_0["mu_star"]
Si_0["color"] = colors

for i in range(0,len(names)):
        out = ax1.scatter(mu_star[i], sigma[i], c=colors[i], marker="o", label = names[i])

ax1.set_ylabel(r"$\sigma (kWh/m^2)$", fontsize = 16)
ax1.set_xlabel(r"$\mu^\star (kWh/m^2)$", fontsize = 16)

ax1.set_xlim(0,)
ax1.set_ylim(0,)
x_axis_bounds = np.array(ax1.get_xlim())
(line1,) = ax1.plot(x_axis_bounds, x_axis_bounds, "k-", label = r"$\sigma / \mu^{\star} = 1.0$")
(line2,) = ax1.plot(x_axis_bounds, 0.5 * x_axis_bounds, "k--", label = r"$\sigma / \mu^{\star} = 0.5$")
(line3,) = ax1.plot(x_axis_bounds, 0.1 * x_axis_bounds, "k-.", label = r"$\sigma / \mu^{\star} = 0.1$")

fig1.legend(loc = 'lower center', ncol =4, fontsize = 12)
ax1.tick_params(axis='both', which='major', labelsize=12)

def sort_Si(Si, key, sortby="mu_star"):
    return np.array([Si[key][x] for x in np.argsort(Si[sortby])])

colors_sorted = sort_Si(Si_0, "color", sortby = "mu_star")
print(colors_sorted)

horizontal_bar_plot(ax2, Si_0, {"color" : colors_sorted}, sortby='mu_star')
ax2.set_xlabel(r"$\mu^\star (kWh/m^2)$", fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=12)
fig1.suptitle("Total Load Intensity $(kWh/m^2)$", fontsize=20)

plt.show()
```

Figure A-4.2: Python scripts for Morris SA in BDL 3 case

```python
from SALib.analyze import morris
from SALib.sample.morris import sample
from SALib.util import read_param_file
from SALib.plotting.morris import horizontal_bar_plot
import matplotlib.pyplot as plt
import numpy as np

# Import data:
problem = read_param_file(r"C:\Users\daovi\Documents\data\Morris\BDL4\parameter_file.txt")
problem.pop('groups')
print(problem)
param_values = np.loadtxt(r"C:\Users\daovi\Documents\BDL4_final_1190\model_input.txt")
data = np.loadtxt(r"C:\Users\daovi\Documents\BDL4_final_1190\model_output.txt")

# Perform Morris SA:
Y_0 = data[:,0]

Si_0 = morris.analyze(
    problem,
    param_values,
    Y_0,
    conf_level=0.95,
    print_to_console=True,
    num_levels=4,
    num_resamples=100,
)
```

```python
# Visualize results:
fig1, (ax1,ax2) = plt.subplots(1,2)

colors = ["blue", "orange", "red", "purple", "brown", "pink", "gray", "cyan", "olive"]
names = Si_0["names"]
sigma = Si_0["sigma"]
mu_star = Si_0["mu_star"]

Si_0["color"] = colors

for i in range(0,len(names)):
        out = ax1.scatter(mu_star[i], sigma[i], c=colors[i], marker="o", label = names[i])

ax1.set_ylabel(r"$\sigma (kWh/m^2)$", fontsize = 16)
ax1.set_xlabel(r"$\mu^\star (kWh/m^2)$", fontsize = 16)
ax1.set_xlim(0,)
ax1.set_ylim(0,)
x_axis_bounds = np.array(ax1.get_xlim())
(line1,) = ax1.plot(x_axis_bounds, x_axis_bounds, "k-", label = r"$\sigma / \mu^{\star} = 1.0$")
(line2,) = ax1.plot(x_axis_bounds, 0.5 * x_axis_bounds, "k--", label = r"$\sigma / \mu^{\star} = 0.5$")
(line3,) = ax1.plot(x_axis_bounds, 0.1 * x_axis_bounds, "k-.", label = r"$\sigma / \mu^{\star} = 0.1$")

fig1.legend(loc = 'lower center', ncol =4, fontsize = 12)
ax1.tick_params(axis='both', which='major', labelsize=12)

def sort_Si(Si, key, sortby="mu_star"):
    return np.array([Si[key][x] for x in np.argsort(Si[sortby])])
colors_sorted = sort_Si(Si_0, "color", sortby = "mu_star")
print(colors_sorted)

horizontal_bar_plot(ax2, Si_0, {"color" : colors_sorted}, sortby='mu_star')
ax2.set_xlabel(r"$\mu^\star (kWh/m^2)$", fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=12)
fig1.suptitle("Total Load Intensity $(kWh/m^2)$", fontsize=20)

plt.show()
```

Figure A-4.3: Python scripts for Morris SA in BDL 4 case

Appendix B

SA for BEPS

Table B-1: Definition of parameters used in previous studies

Group	Parameters	Symbol	Unit
Geometrical	Length	L	m
	Width	W	m
	Height	H	m
	Ratio for LengthA	rLenA	-
	Ratio for WidthA	rWidA	-
	Orientation	Ori	°
	Basement depth (if applicable)	-	m
Technical Specifications	Wall U-value	U_Wall	$W/m^2 °K$
	Ground Floor U-value	U_GFloor	$W/m^2 °K$
	Roof U-value	U_Roof	$W/m^2 °K$
	Infiltration	Infil	-
	Internal Wall	-	%
	Construction thickness External Wall	-	m
	Construction thickness Ground Slab	-	m
	Construction thickness Floor Slab	-	m
	Construction thickness Roof Slab	-	m
	Reinforcement	-	kg/m^3
Window Construction	Window U-value	U_Window	$W/m^2 °K$
	Window G-value	g_Window	%

	Window to Wall ratio North Window to Wall ratio West Window to Wall ratio South Window to Wall ratio East	WWR_N WWR_W WWR_S WWR_E	% % % %
Building Operation	Operating hours Light and Electrical Heat Gain	OpH L/EHG	h W/m^2
System Efficiency	Boiler Efficiency Chiller COP	B_Eff C_COP	- -

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Master-Thesis selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 16. December 2023

Viet Cuong Dao

Viet Cuong Dao

Bahnhofstraße 11

D-85375 Neufahrn bei Freising

vietcuong.dao@tum.de