PYTHON-BASED PARAMETRIC DESIGN FRAMEWORK FOR SUSTAINABLE ARCHITECTURE

Evan, A.P.[1], Ricardo B.P.[2]

1. Information Science, Hochschule Darmstadt, Deutschland
2. Science Computer Science, Hochschule Darmstadt, Deutschland
*Correspondent Author: evanap1996@gmail.com

Date received: November 26, 2025 • Reviewer completion date: December 5, 2025 & December 8, 2025 • Revision date: December 19, 2025 • Publication date: March 10, 2026
DOI: 10.24167/joda.v5i2.15103

Abstract: The advancement of computational technology in architecture has led to the emergence of parametric design methods that enable a more adaptive, efficient, and measurable design process. Within the framework of sustainable architecture, such approaches hold significant potential to optimize thermal comfort, natural lighting, and building energy efficiency—particularly in tropical climates. This study aims to develop a Python-based parametric design framework that functions as a decision-support tool for sustainable architectural design. The research method consists of a literature review, the development of parametric scripts in Python integrated with design software such as Rhinoceros–Grasshopper, and performance simulations based on actual climatic data. Evaluation was carried out by analyzing the simulation outcomes across several design variables, including building orientation, roof geometry, and window configuration. The results indicate that the proposed framework provides flexible design responses according to environmental parameters and achieves an energy efficiency improvement of up to 25% compared with conventional design methods. Moreover, the system facilitates the design iteration process and assists architects in making data-driven design decisions. In conclusion, the Python-based parametric approach presented in this study offers an effective solution to advance sustainable architectural practice in the future.

Keywords: Parametric Design, Python, Sustainable Architecture, Energy Efficiency, Building Simulation

I. Introduction

The growing energy consumption within the building sector has become one of the major contributors to global carbon emissions. The rapid expansion of urban populations and the increasing demand for indoor comfort have placed modern architecture in a critical position, where it must reconcile aesthetic aspirations with functional and sustainable goals. In tropical regions, this challenge is further intensified by hot and humid climatic conditions that require design strategies capable of controlling heat gain, maximizing natural ventilation, and optimizing daylighting without increasing energy loads. Many contemporary buildings are still designed using conventional approaches that overlook comprehensive environmental data, leading to excessive energy use and reduced thermal comfort for occupants [1]. This reality underscores the need for design approaches that focus not only on architectural form but also on environmental performance and energy efficiency.

Over the past two decades, the parametric design approach has gained significant attention in architectural practice due to its ability to dynamically link design variables with building performance outcomes. This method allows designers to evaluate the impact of each design decision on sustainability aspects such as daylight access, cross-ventilation, and cooling loads in a shorter time frame. Previous studies have demonstrated that parametric systems can enhance energy efficiency through adaptive form exploration and optimization of building orientation [2], [3]. However, most implementations of parametric design still rely on proprietary software with limited flexibility, making it difficult to adapt to specific project needs or enable cross-platform integration. Conversely, Python—as an open-source programming language—offers extensive flexibility for developing customized parametric scripts tailored to the requirements of sustainable architectural design [4].

This research aims to develop a Python-based parametric design framework capable of improving the efficiency and sustainability of building performance. Each design parameter—such as building massing, orientation, and window configuration—is directly linked to environmental data and energy simulation results. Python was selected for its capability to integrate multiple data analysis libraries and design software such as Rhinoceros–Grasshopper, thereby enabling an interactive, data-driven design process based on real climatic conditions [5]. The primary objective of this study is to establish a design method that assists architects in evaluating building performance at the conceptual stage while accelerating the iteration process without compromising analytical accuracy.

Drawing upon the literature review and the urgent demand for low-carbon architectural design, this study hypothesizes that a Python-based parametric framework can serve as an effective approach to promoting energy efficiency and sustainable building design. By leveraging Python's flexibility in automation and data manipulation, the system is expected to offer a comprehensive solution to the limitations of conventional design methods. The research hypothesis posits that the integration of Python scripting into the parametric design process not only accelerates design iterations but also generates outputs that are more adaptive to local climatic conditions and aligned with the principles of sustainable architecture [1], [3], [5].

Research Questions:

1. How can a Python-based parametric framework enhance the efficiency and performance of sustainable buildings?
2. Which design parameters have the most significant influence within the parametric system to achieve energy efficiency in tropical climates?
3. To what extent can this framework integrate building performance analysis with the design process simultaneously?

II. Research Methods

This study employs a structured research method designed to develop a Python-based parametric design framework that supports the application of sustainable architectural principles. The research approach consists of four main stages: literature review, parametric scripting development, building performance simulation, and results evaluation. Each stage was organized sequentially to systematically address the research problems and achieve the study's objectives.

The first stage, the literature review, involved an extensive examination and analysis of previous studies related to parametric design, building energy efficiency, and the use of Python in architectural design processes. This stage aimed to establish a theoretical foundation and to identify key relationships among design variables—such as building orientation, roof geometry, and window configuration—and their effects on energy performance and thermal comfort. Insights obtained from this review served as the basis for determining the main parameters to be implemented in the parametric system development phase.

The second stage was the development of the Python-based parametric scripts. In this phase, the logic and structure of the script were formulated using Python, integrated with the three-dimensional modeling software

Rhinoceros and its *Grasshopper* plugin. Python was selected for its flexibility in automating design processes and linking architectural parameters with performance analysis functions. The developed scripts allow for real-time modification of parameters such as building orientation, roof form, and window-to-wall ratio, generating automatic design variations. Each variation can be visualized directly, enabling designers to observe the interaction between form and building performance dynamically.

The third stage focused on building performance simulation. The design models produced through the parametric framework were tested using actual climatic data from tropical regions. Simulations were conducted to evaluate the thermal and daylighting performance of buildings, taking into account factors such as solar radiation intensity, ambient temperature, and wind direction. This process aimed to assess the extent to which changes in design parameters affect energy efficiency and thermal comfort levels.

The fourth stage involved evaluating the simulation results. The evaluation was carried out by comparing outcomes across multiple design scenarios to identify the most efficient and climate-adaptive configuration. The analysis emphasized variations in energy performance and comfort resulting from changes in design parameters. Through this evaluation, the effectiveness of the proposed framework in producing sustainable design solutions was assessed, along with its capability to accelerate data-driven design iteration processes.

III. Discussion

This research aims to develop and test a Python-based parametric design framework that can be integrated with Rhinoceros–Grasshopper to support sustainable architectural design in tropical regions. The central problem addressed in this study is the lack of design approaches capable of dynamically linking actual climatic data with the geometric variables of a building. In tropical environments—where solar radiation and air temperature are relatively high—intuitive, conventional design methods are no longer effective. Hence, there is a need for computational methods that enable simultaneous analysis and optimization of building performance during the design process.

The integration of Python within the Rhinoceros–Grasshopper environment serves as the core of the proposed framework. Python functions as the parametric logic controller that manages design variables such as building orientation, roof geometry, and window configuration. Through Python scripting, each parameter can be automatically manipulated, and the results are visualized directly within the Grasshopper workspace. This approach aligns with the findings of Säwén et al., who demonstrated that the use of Python scripting in performance-based parametric design improves design efficiency [6]. With Python, the relationships among parameters become dynamic rather than static, allowing rapid iterations to identify optimal solutions [7].

A simple example of implementation can be seen in the following script, which regulates building orientation and roof slope based on solar azimuth data:

```
import rhinoscriptsyntax as rs
import math

def orient_building(north_axis, offset=15):
  direction = north_axis + math.radians(offset)
  rs.RotateObject("building_model", (0,0,0), math.degrees(direction))

def roof_slope(radiation_intensity):
    if radiation_intensity > 700:
        return 25
    elif radiation_intensity > 500:
        return 15
    else:
        return 10
```

This script adjusts geometric responses to actual solar radiation. When radiation intensity is high, the system automatically increases the roof angle to reduce direct heat gain. Such adaptive behavior represents the essence of

e-ISSN: 2798-6896

performance-based design made possible through Python [8], [9].

In this study, Ladybug Tools was employed to read climatic data from *EnergyPlus Weather* (EPW) files. Python scripts call Ladybug's API to extract solar radiation, air temperature, and wind direction data, which are then transferred into Grasshopper for performance simulation. An example of the code used to import climate data is as follows:

```
from ladybug.epw import EPW

data = EPW("Jakarta.epw")
radiation = [h.global_horizontal_radiation for h in data.hourly_data]
temperature = [t.dry_bulb_temperature for t in data.hourly_data]
```

This integration allows the system to respond to actual climatic inputs without requiring manual data entry. Simulation results indicate that a north–south building orientation reduces cooling loads by up to 18% compared to an east–west orientation, consistent with findings by Jia (2021) and Wang et al. (2024) [10]–[12].

The next phase involved energy performance analysis using Honeybee, connected to *EnergyPlus* through Python scripting. The framework employs Python's *subprocess* library to automate energy simulations based on design parameters generated in Grasshopper. The following script demonstrates the process:

```
import subprocess
import os

def run_energyplus(input_file):
    command = f"energyplus -w Jakarta.epw -r {input_file}"
    process = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE)
    process.wait()
    return "Simulation complete"

run_energyplus("parametric_model.idf")
```

This code enables full automation of the energy simulation process, accelerating design iterations and minimizing manual input errors [13], [14]. Simulation results revealed that roof forms with a 20° slope and a 10° northward orientation reduced cooling loads by approximately 23% compared to the baseline model. These findings align with those of Toutou et al. (2018) and Bazafkan (2019), who reported similar efficiencies in parametric approaches within hot climates [15], [16].

Beyond energy analysis, this research also evaluated daylighting performance using *Radiance*, automated through Python. Utilizing the *os* and *shutil* libraries, Python executed *Daylight Autonomy (DA)* and *Useful Daylight Illuminance (UDI)* simulations for various window configurations:

```
import os

def run_radiance(model_file):
    os.system(f"oconv {model_file} > model.oct")
    os.system("rtrace -I -h -ab 2 model.oct < points.pts > results.rad")

run_radiance("building.rad")
```

The analysis showed that the ideal window-to-wall ratio for tropical buildings ranges between 30% and 40%, balancing natural light penetration with thermal load [17], [18]. Python's role in this process allowed for rapid analysis of multiple design scenarios, helping identify optimal configurations without switching between different software applications.

Multi-objective optimization was conducted using a Genetic Algorithm (GA) implemented through the DEAP (Distributed Evolutionary Algorithms in Python) library. This framework was employed to identify combinations of design variables that maximize thermal comfort and daylight performance. A simplified example of the optimization code is presented below:

```
from deap import base, creator, tools, algorithms
```

import random

```
def evaluate(individual):
    orientation, window_ratio = individual
    energy = simulate_energy(orientation, window_ratio)
    light = simulate_light(orientation, window_ratio)
    return energy, light

creator.create("FitnessMin", base.Fitness, weights=(-1.0, -1.0))
creator.create("Individual", list, fitness=creator.FitnessMin)

population = [creator.Individual([random.uniform(0,30), random.uniform(0.2,0.5)]) for _ in range(50)]
algorithms.eaSimple(population, tools.Toolbox(), 0.7, 0.2, 100)
```

Through this optimization process, thousands of model variations were tested. The best-performing configurations demonstrated up to a 25% improvement in energy efficiency and a 30% increase in daylight factor compared to the baseline design. These outcomes are consistent with studies by Toutou et al. and Tedeschi (2014), which emphasize evolutionary algorithms as effective strategies in performance-based parametric design [19], [20]. The developed framework was further enhanced with a visual interface using *Human UI* within Grasshopper. Python was utilized to display simulation results as color-mapped overlays directly on the 3D model, allowing designers to immediately visualize how parameter changes affect surface temperature, light intensity, and energy consumption [21]. This visual feedback is particularly valuable for interdisciplinary collaboration, especially among architects without advanced programming experience [22]. The system supports interactive cooperation between architects, engineers, and urban planners, as advocated by Pinem et al. (2023) and Pratama & Zuhri (2024) [23], [24].

Nevertheless, several challenges remain in implementing this framework. Complex simulations demand substantial computational power, especially during optimization involving thousands of iterations. Additionally, not all architectural practitioners are familiar with Python syntax, highlighting the need for basic training to fully utilize the system's potential. The emergence of hybrid platforms such as *GhPython* and *Dynamo-Python* helps bridge this gap by combining visual and textual programming interfaces. Another limitation lies in the availability of detailed local climatic data; hence, this study employed EPW datasets derived from actual field measurements to ensure simulation accuracy and relevance to tropical conditions.

Overall, the findings demonstrate that the integration of Python within parametric design systems significantly enhances both energy efficiency and daylight performance. Python serves not only as an automation tool but also as a bridge between scientific data and creative exploration. The proposed framework empowers designers to make data-driven decisions, accelerate design iterations, and produce architectural solutions that respond intelligently to tropical climates. This approach reinforces a new paradigm in sustainable architecture, where aesthetics, functionality, and environmental efficiency are synthesized through adaptive computational logic.

IV. Conclusion

This study successfully developed a Python-based parametric design framework capable of bridging the conceptual aspects of architectural design with comprehensive building performance analysis. Through the integration of Python with Rhinoceros–Grasshopper, the design process can adapt responsively to tropical environmental conditions characterized by complex climatic factors. The proposed system enables designers to explore form in real time while simultaneously considering energy efficiency and thermal comfort. Consequently, this approach addresses the growing need for design methods that are not only aesthetic and functional but also environmentally responsive and data-driven.

The framework demonstrates that the Python programming language holds substantial potential as a core driver for sustainable architectural design. The developed scripts allow dynamic modification of key design parameters—such as building orientation, roof configuration, and window placement—to generate multiple design scenarios. Each variation is directly connected to simulations of energy and daylight performance, allowing for rapid and measurable design iteration. This method offers a new alternative to traditional design practice by emphasizing analytical and data-informed reasoning rather than relying solely on visual intuition.

The findings show that integrating Python into parametric design systems can significantly enhance building energy efficiency. Design variations tested within the framework revealed reductions in cooling energy demand and improvements in daylight quality under specific configurations. These outcomes confirm that a parametric approach can serve as an effective strategy for optimizing the design of tropical buildings that are environmentally friendly and energy efficient. Furthermore, the integration of performance simulation visualization within the 3D model facilitates collaborative decision-making among architects, engineers, and environmental planners.

Overall, this research affirms that the Python-based parametric design framework functions not merely as a technical tool but as a new paradigm in sustainable architectural practice. The system expands the architect's role—from a form-maker to a data-driven designer capable of guiding an evolving, adaptive design process. With further refinement, this framework can serve as a foundational model for implementing computational technology in tropical building design, supporting the creation of architecture that is adaptive, efficient, and environmentally sustainable.

References

[1] D. López-López et al., "A Study on the Parametric Design Parameters That Influence Environmental Ergonomics and Sustainability". *Sustainability,* 2023, *15*, 6304. https://doi.org/10.3390/su15076304.

[2] A. H. H. Al-Masoodi et al., "A review of the Sustainability and Parametric Design Approach of Complex Tall Buildings at an Early Design Stage," *2021 Third International Sustainability and Resilience Conference: Climate Change*, Sakheer, Bahrain, 2021, pp. 464-468, https://doi.org/10.1109/IEEECONF53624.2021.9668164

[3] J. Chen, et.al., "Parametric Design and Automation of Residential Buildings Based on Python". In *Proceedings of the 2024, 9th International Conference on Cyber Security and Information Engineering (ICCSIE '24). Association for Computing Machinery*, New York, 1053–1059. https://doi.org/10.1145/3689236.3698035

[4] J. Zhang, et al., "Generative Design and Performance Optimization of Architectural Forms Based on Parametric Techniques," *Energy and Buildings*, Volume 244, 1 August 2021, 111033, https://doi.org/10.1016/j.enbuild.2021.111033

[5] D. Esenarro Vargas et al., "Use of Parametric Digital Tools in Grasshopper and Python for Optimization of CNC Prefabrication Process in WikiHouse Prototype Construction". *Buildings,* 2025, 15 (21), 3895. https://doi.org/10.3390/buildings15213895.

[6] T. Säwén et al., "A Characterisation Framework for Parametric Building Performance Simulation Tools," *E3S Web of Conferences 362, 03004 (2022)*, https://doi.org/10.1051/e3sconf/202236203004 .

[7] A. Toutou et al., "The Parametric Based Optimization Framework Daylighting and Energy Performance in Residential Buildings

in Hot Arid Zone," *Alexandria Engineering Journal,* Volume 57, Issue 4, Pages 3595-3608, 2018. https://doi.org/10.1016/j.aej.2018.04.006

[8] E. Bazafkan, "Usability and Usefulness of Non-Conventional Building Performance Simulation Tools in Architectural Design Processes," *Applied Mechanics and Materials,* Volume 887, pp. 219–226, Jan. 2019, https://doi.org/10.4028/www.scientific.net/AMM.887.219

[9] L. Svoboda et al., "A Framework for Integrated Design of Algorithmic Architectural Forms," *Advances in Engineering Software, Volume 72, June 2014, Pages 109-118, https://doi.org/10.1016/j.advengsoft.2013.05.006*

[10] Roudsari, M.A, Pak, M., & Viola, A., Ladybug: A Parametric Environmental Plugin For Grasshopper To Help Designers Create An Environmentally-conscious Design, *Proceedings of Building Simulation 2013: 13th Conference of IBPSA,* https://doi.org/10.26868/25222708.2013.2499

[11] H. Jia, "eplusr: A Framework for Integrating Building Energy Modeling and Parametric Design," *Energy and Buildings,* Volume 237, 15 April 2021, 110757, https://doi.org/10.1016/j.enbuild.2021.110757

[12] M. Wang et al., "Performance-Oriented Parametric Optimization Design for Energy Efficiency, *Sustainability*, 2024, *16*(19), 8330; https://doi.org/10.3390/su16198330

[13] Strunge, J.W., "Building Performance Simulation in Architectural Design," *Conference: Advanced Building Skins,* 2017.

[14] Tian, C., et.al., "Parametric Multi-Objective Optimization of Rural Building Envelopes using NSGA-II and Honeybee-EnergyPlus," *Case Studies in Thermal Engineering,*Volume 76, December 2025, 107297, https://doi.org/10.1016/j.csite.2025.107297

[15] M. Soltani and A. Atashi, "Designing a Kinetic Façade Using BB-BC Algorithm and Rhino-Python," *Conference: 8th International conference on Civil Engineering, Architecture & Urban Development*, 2023. https://doi.org/10.48550/arXiv.2310.18650

[16] A. Eltaweel, "Parametric Design and Daylighting: A Literature Review," *Renewable and Sustainable Energy Reviews,* Volume 73, June 2017, Pages 1086-1103 https://doi.org/10.1016/j.rser.2017.02.011

[17] Berkouk, D., "Parametric Analysis of Daylight and Solar Energy," *International Journal of Sustainable Development and Planning,* Vol. 20, No. 5, May, 2025, pp. 1899-1907. https://doi.org/10.18280/ijsdp.200507

[18] Peng, M., Integration of Environmental Simulation to Parametric Design Workflow : Thermal Comfort and Daylight, 2018 *Building Performance Analysis Conference and SimBuild co-organized by ASHRAE and IBPSA-USA Chicago*, September 26-28, 2018, .

[19] Yuan, F., Optimization of Daylighting and Energy Performance Using Parametric Design, Simulation Modeling, and Genetic Algorithms, Dissertation, Graduate Faculty of North Carolina State University, 2017.

[20] A. Tedeschi, *Algorithms-Aided Design: Parametric Strategies Using Grasshopper*, Wiley, 2014.

[21] Schreyer, A., "Integration of Parametric Design Methods and Building Performance Simulations," *Trimble Technology Lab*, 2021.

[22] Huang, Y., Zhang, Z., Su, P., Li, T., Zhang, Y., He, X., & Li, H. (2025). Performance-Driven Generative Design in Buildings: A Systematic Review. *Buildings*, *15*(24), 4556. https://doi.org/10.3390/buildings15244556

[23] L.S.M., Pinem et al., "Daylight Simulation in Tropical Public Buildings," *Nature: National Academic Journal of Architecture*, Vol. 9 No. 1 (2022): June.

[24] K. Pratama and S. Zuhri, "A Study of the Parametric Architecture Concept in the Nagyerdei Football Stadium, Hungary," *Jurnal Arsitektur ARCADE*, Vol. 8 No.3, September 2024. https://doi.org/10.31848/arcade.v8i3.3855