

Research paper

AI SUPPORTED PYTHON PROGRAMMING IN ARCHITECTURAL FORM-FINDING PROCESSES

Mirjana Devetaković¹, Jelena Ivanović²,
Djordje Djordjević³, Djordje Mitrović⁴

Abstract

In recent years, artificial intelligence (AI) has become widely accessible, with powerful AI tools integrated into common ICT services and new AI-based applications emerging almost daily, often freely available. The Architecture, Engineering, Construction, and Operation (AECO) sector is no exception, as numerous studies highlight AI-enhanced applications supporting various aspects of the industry. While many focus on visualization, AI is also increasingly contributing to design and construction processes. This paper explores the process of architectural form generation using Rhino 3D, leveraging Python scripting for parametric geometric modeling. ChatGPT-3 was utilized to generate Python scripts compatible with the Rhino 3D Python Editor, benefiting from its ability to integrate mathematical concepts with programming logic and respond to textual programming prompts. Through this approach, a series of complex geometric forms was modeled, visually examined, and selectively 3D printed. Following an introduction to the relevant mathematical concepts and digital tools (ChatGPT, Python, and Rhino 3D), the paper details the methodology for guiding AI through parametric modeling. Since achieving the desired results is not always straightforward, specific strategies for controlled, purpose-driven interaction with ChatGPT are outlined. The study is illustrated with characteristic examples, and the generated architectural forms are analyzed, discussing the advantages and limitations of the approach. The paper concludes by examining the evolving technological skillset required for contemporary architectural design in the AI era.

Key words: AI, Python, Architecture, Geometry, Form-finding

¹ Dr, Associate Professor, University of Belgrade – Faculty of Architecture, Serbia, mirjana.devetakovic@arh.bg.ac.rs, 0000-0001-7928-0102

² Dr, Assistant Professor, University of Belgrade – Faculty of Architecture, Serbia, jelena.ivanovic@arh.bg.ac.rs, 0000-0001-8046-9703

³ Dr, Associate Professor, University of Belgrade - Faculty of Architecture, country, djordje@arh.bg.ac.rs, 0000-0002-8943-5398

⁴ Teach. Assistant, University of Belgrade – Faculty of Architecture, Serbia, djordje.mitrovic@arh.bg.ac.rs, 0000-0001-5156-6133

1. INTRODUCTION

Artificial intelligence and its wide usage is a phenomenon that could be examined in almost every aspect of the human activity in the digitalized world. It is present, sometimes invisible, in services ranging from simple email correspondence, to complex managing of urban life. The appearance of freely available, user oriented chat bots in recent years, additionally challenges fields like education, medicine, marketing, music creation, language assistance, data analysis, content writing and many others.

In this study, one professionally-controlled usage of AI is demonstrated. It relates mostly to an assistance to Python coding for the Rhino 3D software. The main framework of the study is geometric form-finding as a part of architectural design. The experimental work was completed in the summer 2023 and 3D printing was realized in spring 2024.

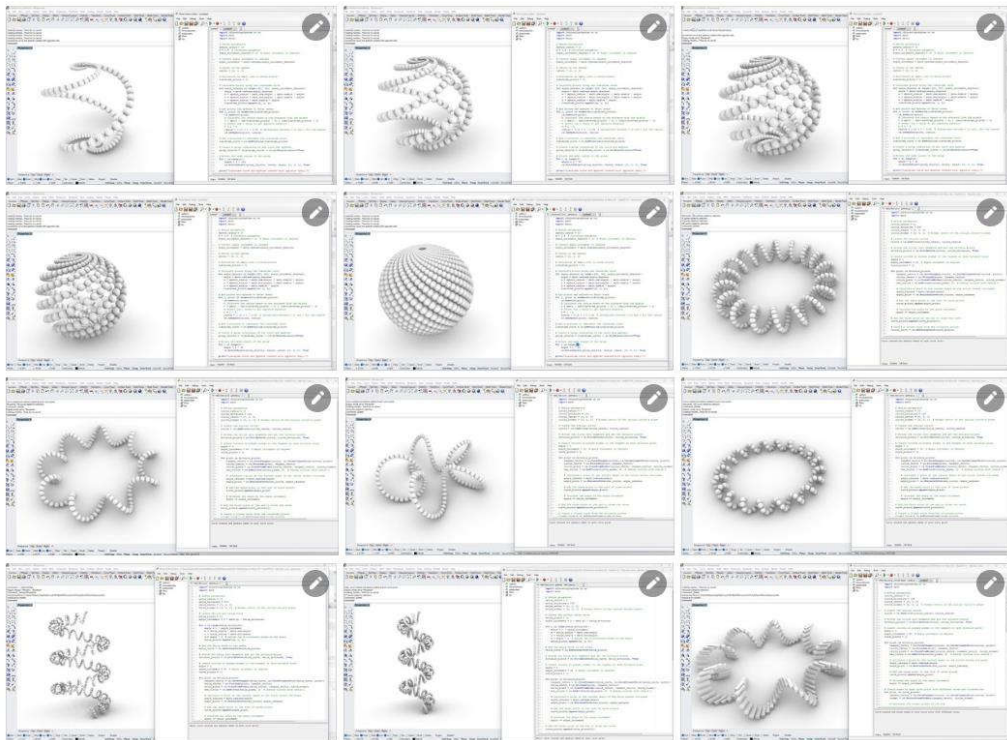


Figure 1 A set of geometric forms created by combining mathematics, AI, Python and Rhino 3D

In the study, a series of geometric forms have been created (Figure 1) and several different knowledge aspects were mutually used:

- Mathematical/geometry-wise (regarding parametrization of spatial curves and surfaces),
- AI-wise,
- Python-wise (regarding programming),
- Rhino 3D-wise (regarding modelling and visualization), and
- 3D printing-wise

Each of the knowledge aspects is analysed in the following sections.

In this research it is assumed that spatial curves play a crucial role in forming a tectonics of architectural form, defining often its structural elements, shell geometries and decorative parts.

Relying on mathematical knowledge with which architects are equipped, in this study, well known, parametrically defined curves have been examined, relying on their potential to contribute to architectural form-finding by their relatively unexpected appearance.

It is important to stress that the examination process relies on significant previous experiences of the authors in both fields of architectural 3D modelling and various aspects of coding.

2. MATHEMATICAL / GEOMETRY-WISE ASPECT

Differential geometry, serving as a mathematical discipline dedicated to the study of smooth shapes and spaces, empowers architects with the capacity to craft sleek, refined forms through its foundational principles. By employing differential equations, architects can precisely manipulate the geometry of structures and surfaces within digital environments, enhancing both the accuracy and efficiency of design iterations, ultimately leading to the creation of more refined and optimized architectural solutions.

In this study, a key focus lies in exploring the parametric representation of curves and surfaces, a method widely applied in computer graphics, CAD, and mathematical modelling, for describing shapes in three-dimensional space. This method expresses points on a curve or surface as functions of one or two parameters, respectively. To be more precise, a space curve can be described by a vector function of a single parameter t : $r(t) = (x(t), y(t), z(t))$, where t varies over some interval, while $x(t)$, $y(t)$ and $z(t)$ are functions that define the coordinates of points on the curve. A helix is a common example of a space curve whose equations can be given as: $x(t) = a \cos t$, $y(t) = a \sin t$ and $z(t) = bt$, where a is the radius of the helix, b is the pitch, and t is the parameter.

This mathematical concept is quite basic but sufficient for the subject of this study, and we certainly assume that architects are proficient in this knowledge, as well as knowing the realm and the aesthetics of space curves and surfaces [1], especially the ones widely used in architecture [2]. For a deeper understanding and exploration of more complex mathematical concepts that can also be applied in generating architectural form, we recommend a [3] and [4].

However, the essence of this work lies not in the application of complex mathematical theory, nor in deep familiarity with the next two tools discussed in the following sections, but rather in the synergy of understanding fundamental concepts from all three fields, enabling architects to optimally leverage their expertise in shape generation without the intervention of mathematicians and programmers, exploring, programming, and creating seemingly independently.

3. AI-WISE ASPECT

In this study, AI is used to assist in the process of Python coding for Rhino 3D. The AI chatbot was ChatGPT, version 3, which is freely available on the Internet.

A chatbot is defined as a software application or web interface that is designed to mimic human conversation through text or voice interactions [5] ChatGPT is one of the most well-known chatbots, developed by OpenAI Company. It appeared in the late 2022 and was dramatically quickly widespread [6].

The interaction with ChatGPT was called here “conversation” [7]. The conversations were conducted in English language.

The conversation with ChatGPT normally started with a mathematical/geometry introduction. This step was very important in terms of focus the chatbot on the desired area. Here are the examples of the introductory questions asked:

- “Could we talk this time about space curves?”
- “Could you, please, list the well-known space curves with their parametric representations?”
- “Could you give an example of space curve, represented parametrically (X, Y, Z)”
-

It is possible to say that, in this area, ChatGPT deals with a considerable body of reliable knowledge. Here’s an example of conducted a simple (helix-related) conversation (Figure 2):

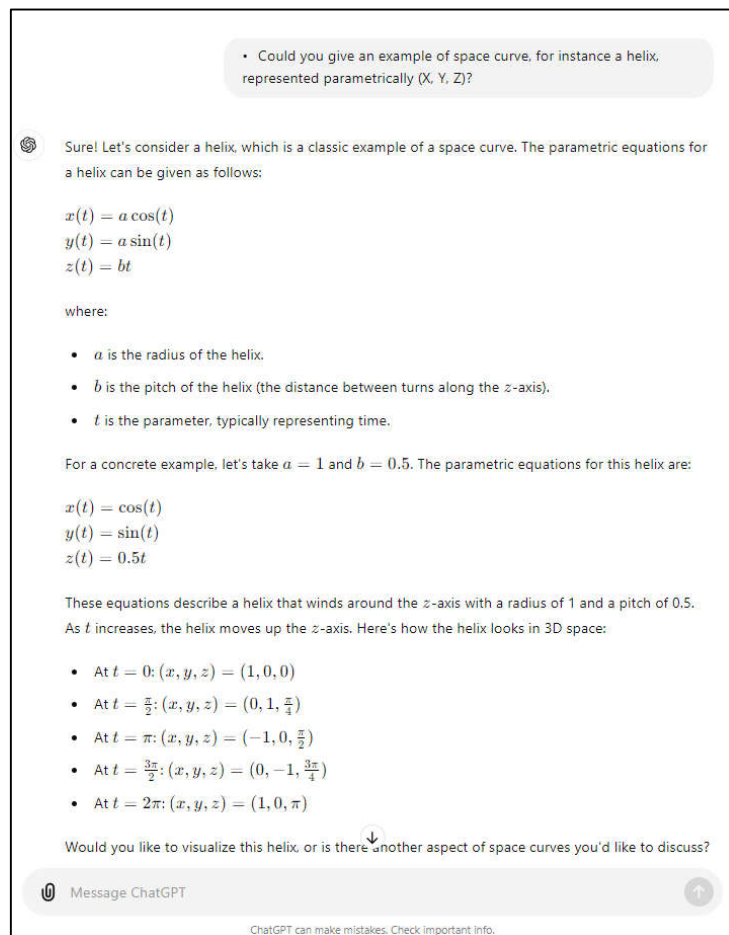


Figure 2 An example of a conversation with ChatGPT about a helix space curve, represented parametrically

After such a simple introduction and double checking the validity of the information provided, the conversation continued towards the Python for Rhino3D coding (Figure 3).



Figure 3 An example of a code created by ChatGPT that could be executed within Rhino 3D to generate a helix

In addition, ChatGPT gave an explanation of the created code (Figure 4). This helped very much to understand the structure and elements of the proposed code.

Explanation:**1. Parameters:**

- ``radius``: The radius of the helix.
- ``pitch``: The vertical distance between each turn of the helix.
- ``turns``: The total number of turns in the helix.
- ``points_per_turn``: The number of points used to define each turn of the helix.

2. Generating Points:

- The total number of points is calculated based on the number of turns and points per turn.
- A loop iterates to calculate the position of each point using the parametric equations of the helix.

3. Creating the Helix:

- ``rs.AddCurve(points)`` is used to create a curve from the list of points.

Steps to Run the Script:

1. Open Rhino 3D.
2. Open the Python scripting editor by typing ``EditPythonScript`` in the command line.
3. Copy and paste the above code into the editor.
4. Run the script by pressing the green "Run" button in the editor.

This will create a helix in the Rhino 3D workspace with the specified parameters. You can adjust the ``radius``, ``pitch``, ``turns``, and ``points_per_turn`` to create different helices.



Message ChatGPT



ChatGPT can make mistakes. Check important info.

Figure 4 An explanation of the created code used to model a helix in Rhino 3D

The coding process, however, is not always and necessarily easy and successful [8]. It needs to be pre-controlled, validated within the Rhino 3D software and if it works properly, a recommendation is to get ChatGPT back informed about the fact that the coding was satisfied (Figure 5). This step is very important, since, within one conversation, ChatGPT memorizes and learns from its own mistakes.

In some cases, the created script simply does not work properly. Then, ChatGPT should be informed about the error, in order it to refine the script in the next steps.

On the other hand, a script can be edited in the Rhino 3D Python Editor manually, and copied back to ChatGPT. In such situation, ChatGPT analyses the given script and returns

comments. These are two equal ways to teach or train the ChatGPT within the extended and repetitive conversation.

Finally, there are cases when ChatGPT persistently makes mistakes creating scripts that don't work. Such problems can be overcome by:

- Editing the script manually, according to available Python coding skills
- Starting the scripting process from the very beginning, redefining textual instructions
- Applying alternative parametric definitions

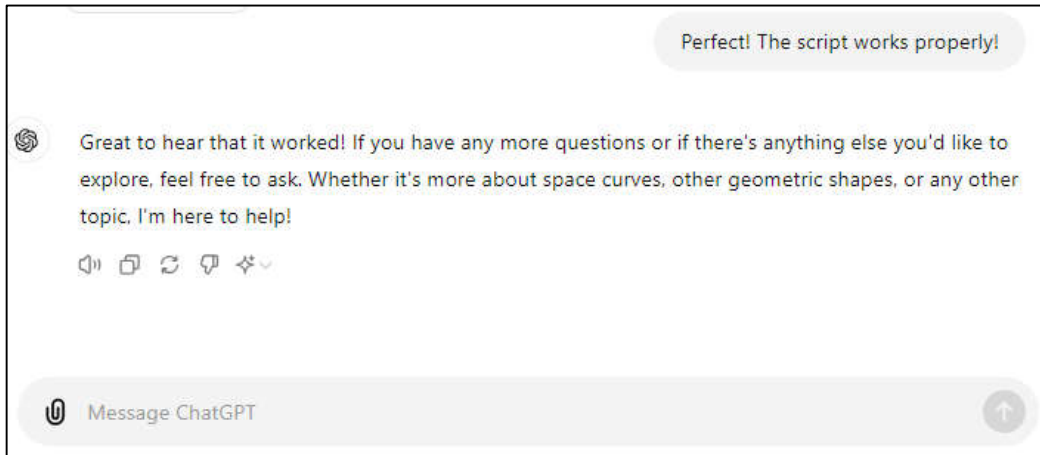


Figure 5 A part of the conversation with ChatGPT in which it was informed about the code that worked properly

4. PYTHON-WISE ASPECT

Python is a versatile and widely-used programming language known for its simplicity and readability. Its syntax is designed to be readable and straightforward, making it an excellent choice for beginners and experienced developers alike. It is an interpreted language which means code is typically executed line-by-line. This allow for quick testing and debugging [9].

Python is extensively used in many CAD applications, including Rhino 3D, a popular 3D computer graphics software. Python scripting in Rhino provides powerful tools for automating tasks, manipulating geometry, and creating custom functionality. RhinoScriptSyntax is a Python library that mirrors the functionality of RhinoScript, making it easier for users familiar with RhinoScript to transition to Python. It provides a straightforward API to access and manipulate Rhino objects and properties [10].

Here's an example of the Python script written with the assistance of ChattGPT that creates a multi-loxodrome form:

```
import rhinoscriptsyntax as rs
import math
import Rhino

# Define parameters
sphere_radius = 100
k = 0.5 # Curvature parameter
angle_increment_degrees = 5 # Angle increment in degrees
```



```
# Convert angle increment to radians
angle_increment = math.radians(angle_increment_degrees)

# Center of the sphere
center = [0, 0, 0]

# Initialize an empty list to store points
loxodrome_points = []

# Calculate points along the loxodrome curve
for angle_degrees in range(-600, 600, angle_increment_degrees):
    angle = math.radians(angle_degrees)
    x = sphere_radius * math.cos(angle) / math.cosh(k * angle)
    y = sphere_radius * math.sin(angle) / math.cosh(k * angle)
    z = sphere_radius * math.tanh(k * angle)
    loxodrome_points.append([x, y, z])

# Add points and spheres to Rhino scene
for i, point in enumerate(loxodrome_points):
    #rs.AddPoint(point)
    # Calculate the radius based on the distance from the middle
    t = abs(i - len(loxodrome_points) / 2) / (len(loxodrome_points) / 2)
    # Invert the t value to get opposite behavior
    t = 1 - t
    radius = 1 + t * 10 # Interpolate between 0.15 and 1 for the radius
    #rs.AddSphere(point, radius)

# Add a polyline to represent the loxodrome curve
loxodrome_curve = rs.AddPolyline(loxodrome_points)

# Create a group consisting of the curve and spheres
group_objects = [loxodrome_curve] + rs.AllObjects(select=True)

# Rotate and make copies of the group
for i in range(6):
    angle = i * 60
    rs.RotateObject(group_objects, center, angle, [0, 0, 1], True)

print("Loxodrome curve and spheres created with opposite radii.")
```

This script, executed in the Rhino 3D Python Editor, created the form shown in Figure 6.

Once created and executed, such script could be easily alternated in terms of changing characteristics parameters (in this case angle of rotation, number of rotated items, radii of spheres, etc.). For being able to manipulate geometric form that way, it is recommendable to well understand the structure and elements of the script [11].

Despite possible imperfections and errors in the Python coding for Rhino 3D, it is important to stress that AI represented here by ChatGPT, serves as an efficient programming assistant that returns considerable scripting results if it is professionally human-guided. It nevertheless saves enormous coding time and helps users that are not programmers to benefit from the Python functionalities [12].

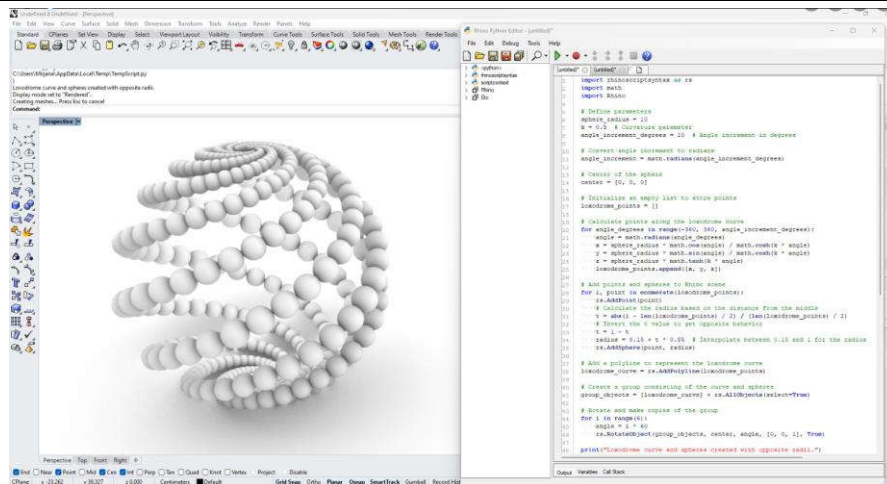


Figure 6 A multi-loxodrome form created from the script executed in the Rhino 3D Python Editor

5. RHINO 3D MODELLING, VISUALIZATION AND 3D PRINTING

To reach the aim of this investigation successfully, possession of Rhino 3D modelling skills is needed. Although the process of modelling is automated and entirely performed by the Python Editor component, it is necessary to carefully control the display, to choose appropriate viewpoints, etc. It is also desirable to well understand the process of Rhino3D modelling performed by the Python script, when some details ought to be alternated manually.

In this study the process of visualization entirely relied on the software's defaults (regarding scene material, light, ambient shadow...). The results obtained in that way, were satisfactory and affirming obtained geometric forms.



Figure 7 The 3D printed multi-loxodrome form

A rigid control of the model preceded the printing process, including the checking of its consistency, minimal dimensions, global dimension, etc. Once the digital model was fine-tuned and finalized, it was considered for 3D printing. As an example, only several models was exported in the .stl format and 3D printed. 3D prints were produced within the 3D IMPULS (Serbian National Laboratory for 3D printing) at the Faculty of Mechanical Engineering in Kraljevo [13].

6. CONCLUSIONS

This study demonstrates the potential of synergy between mathematics, AI, Python programming, Rhino3D, and 3D printing in the domain of complex geometric forms creation. This interdisciplinary approach not only enhances our ability to design intricate shapes, but also opens new horizons in various fields such as architecture, engineering, and digital art. The study indicates and goes parallel with other studies [14], that programming and usage of AI might be a set of very useful, even necessary skills for the architects in the near future.

The further research might be directed towards the question in what sense could this methodology be used to develop a generative system of emergent forms, in which the result is unknown but intended to be discovered.

The integration of mathematics, AI, Python, Rhino3D, and 3D printing signifies a transformative shift in the design and fabrication of complex geometric forms. This approach fosters innovation and efficiency, enabling the creation of structures that are both aesthetically pleasing and functionally robust.

The research presented is foreseen as part of incorporating the coding methods in architectural 3D modelling in both academic education and professional activity. It also tends to enrich the basic digital literacy of modern-day architects.

ACKNOWLEDGMENTS

This research has been funded by the Serbian Ministry of Science, Technological Development and Innovation, within the Project No. 200090 (University of Belgrade, Faculty of Architecture).

REFERENCES

- [1] E. H. Lockwood, **A Book of Curves**, Cambridge University Press, 1961.
- [2] S. Krasić, **Geometrijske površi u arhitekturi**, Građevinsko-arhitektonski fakultet, Univerzitet u Nišu, 2012.
- [3] T. Soshi, **Introduction to differential geometry of space curves and surfaces**, Kindle Edition, 2017.
- [4] A. Lastra, **Parametric Geometry of Curves and Surfaces: Architectural Form-Finding**, Mathematics and the built environment, Volume 5 ed., Birkhäuser, 2021.
- [5] **Chatbot**, "Chatbot Wikipedia," [Online]. Available: <https://en.wikipedia.org/wiki/Chatbot>. (17. 06. 2025.).

- [6] **ChatGPT**, "ChatGPT Wikipedia," 2024. [Online]. Available: <https://en.wikipedia.org/wiki/ChatGPT>. (17. 06. 2025.).
- [7] M. Hayak and U. Hyeok Ko, **Artificial Intelligence Supporting Independent Student Learning: An Evaluative Case Study of ChatGPT and Learning to Code**, Education Sciences, vol. 14, no. 2, p. 120, 24 01 2024.
- [8] C. E. A. Coello, M. N. Alimam and R. Kouatly, **"Effectiveness of ChatGPT in Coding: A Comparative Analysis of Popular Large Language Models,"** Digital, vol. 4, no. 1, pp. 114-125, 08 01 2024.
- [9] M. Kovačević, **Programming Basics in Python** (in Serbian language), 2 ed., Belgrade: Akademska misao, 2024.
- [10] **Rhino.Python Version 7 Guides**, Robert McNeel and Associates, 2024. [Online]. Available: <https://developer.rhino3d.com/guides/rhinopython/>. (17. 06. 2025.).
- [11] S. Tibbits, A. v. d. H. van der Harten and S. Baer, **Python for Rhinoceros 5**, McNeel, 2011.
- [12] C. A. G. da Silva, F. Negrelle Ramos, R. Veiga de Moraes and E. L. dos Santos, **"ChatGPT: Challenges and Benefits in Software Programming for Higher Education,"** Sustainability, vol. 16, no. 3, p. 1245, 01 02 2024.
- [13] **3D Impuls**, "3D Impuls," 2024. [Online]. Available: <https://www.3dimpuls.com/>. (17. 06. 2025.).
- [14] Henriques, G. C., Xavier, P. M., da Luca Silva, V., Bispo, L., Fraga, J. V., **"Computation for Architecture, hybrid visual and textual language: Research developments and considerations about the implementation of structural imperative and object-oriented paradigms"**, International Journal of Architectural Computing, 20(3), 673-678