

Eurasian  
Research Bulletin



# Using Python Libraries To Visualize Math Problems

**Kamalova Nilufar  
Ilkhomovna-Bukhara**

State University, Lecturer of the Department of Applied Mathematics and Programming Technologies, Doctor of Philosophy (PhD)

**Mustafoeva Oikoiz  
Soatilloevna**

Teacher of Mathematics and Computer Science of the 47th School of Gijduvon District.

**ABSTRACT**

The article is devoted to the study of the use of Python libraries for visual demonstration of mathematical problems in the educational process. The paper analyzes the capabilities of various libraries, such as Matplotlib, Seaborn, Plotly, Bokeh, and PyQtGraph, in terms of their applicability to visualize various mathematical concepts

**Keywords:**

Python, математика, Matplotlib, Seaborn, Plotly, Bokeh.

Modern technologies are being actively introduced into education, offering new opportunities for teaching and visualizing complex concepts. Mathematics, traditionally perceived as an abstract and complex discipline, can be significantly more accessible and engaging through the use of visual tools.

Python, with its rich set of libraries, provides unique opportunities for visually demonstrating mathematical problems. Libraries such as Matplotlib, Seaborn, Plotly, NumPy, and SciPy allow you to create interactive graphs, animations, simulate complex mathematical processes, and visualize algorithms.

Visualizing math problems increases students' understanding and engagement, promotes deeper understanding of concepts, and simplifies complex math calculations. The use of Python libraries makes visualization accessible and intuitive, which is especially

important in modern education that strives for innovation and practical applications.

The relevance of this topic is due to several factors:

- Visualization allows you to present complex mathematical concepts in an accessible form, improves understanding and memorization of the material.
- Interactive visualizations make learning more interesting and increase student motivation.
- Visualization contributes to the development of abstract thinking, analytical skills, and the ability to interpret data.
- Python allows you to create interactive learning materials that increase the effectiveness of learning.
- Python libraries can be used to create virtual labs where students can experiment with mathematical models.

- Visualization with Python can be used to present the results of scientific research, making them more accessible and understandable.

Thus, the use of Python libraries for visual demonstration of mathematical problems is a relevant and promising direction that can significantly increase the effectiveness of teaching mathematics.

There are many libraries in Python that are designed to visualize data. Each of them has its own advantages and disadvantages, which makes choosing the right library for a particular task non-trivial.

Python has many ready-made libraries for visualizing data. The most common of them are: **Matplotlib** - A basic library for visualization in Python. It provides extensive capabilities for creating static and dynamic graphs, charts, histograms, and other types of visualizations.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>Highly customizable and flexible.</li> <li>Excellent documentation and lots of examples.</li> <li>A large selection of available output formats.</li> </ul>	<ul style="list-style-type: none"> <li>May require more code to create complex graphs than some other libraries.</li> <li>The interface may seem less intuitive to beginners.</li> </ul>

**Seaborn** - Built on top of Matplotlib, provides a higher level of abstraction to create attractive and informative graphs.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>Simple and intuitive syntactic construction.</li> <li>Convenient functions for creating statistical graphs.</li> <li>Attractive default style.</li> </ul>	<ul style="list-style-type: none"> <li>Less customization flexibility compared to Matplotlib.</li> <li>Some types of graphs may be less convenient to create than with Matplotlib.</li> </ul>

**Plotly** - A powerful library for creating interactive visualizations, suitable for creating web applications.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>Interactive graphs with the ability to zoom, pan and highlight data.</li> <li>Support for various types of charts, including 3D charts.</li> <li>Integration with web frameworks.</li> </ul>	<ul style="list-style-type: none"> <li>Can be more difficult to use for beginners.</li> <li>Requires the installation of additional software for work.</li> </ul>

**Bokeh** is a library for creating interactive graphs, especially suitable for visualizing large data sets.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>High performance for processing large amounts of data.</li> <li>Convenient customization of interactive elements.</li> </ul>	<ul style="list-style-type: none"> <li>Less customization flexibility compared to Matplotlib.</li> </ul>

<ul style="list-style-type: none"> <li>Ability to create graphs for web browsers.</li> </ul>	<ul style="list-style-type: none"> <li>Some types of graphs may be less convenient to create than with Matplotlib.</li> </ul>
----------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------

**PyQtGraph** - A library for creating real-time graphs, suitable for the development of scientific applications.

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>High performance for real-time graph rendering.</li> <li>Ability to create complex graphs using the Qt framework.</li> <li>Support for various types of charts, including 3D charts.</li> </ul>	<ul style="list-style-type: none"> <li>Requires knowledge of the Qt framework.</li> <li>Less convenient for creating static graphs.</li> </ul>

Choosing the best library for data visualization depends on your specific tasks and requirements. Matplotlib and Seaborn are good choices to start with, while Plotly, Bokeh, and PyQtGraph provide great opportunities for creating interactive graphs and real-time visualizations.

It's important to experiment with different libraries and tools to find the most suitable solution for specific tasks. Let's look at practical examples of using Python libraries to solve mathematical problems.

Task: Plot the function  $y = \sin(x)$  and show its period, amplitude, and extremes.

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

```
x = np.linspace(-2*np.pi, 2*np.pi, 100)
y = np.sin(x)
```

```
plt.plot(x,
y)plt.xlabel("x")plt.ylabel("y")plt.title("Графи
к функции  $y = \sin(x)$ ")
```

```
# Показ периода
```

```
plt.axvline(x=0, color='r', linestyle='--',
label='Период')plt.axvline(x=2*np.pi,
color='r', linestyle='--')
```

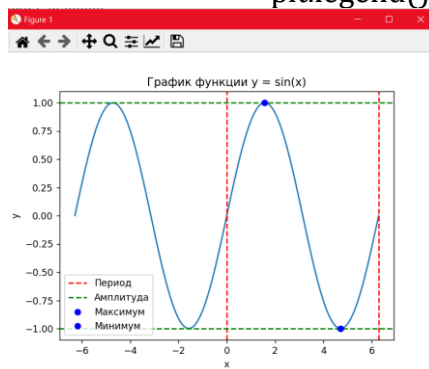
```
# Показ амплитуды
```

```
plt.axhline(y=1, color='g', linestyle='--',
label='Амплитуда')plt.axhline(y=-1, color='g',
linestyle='--')
```

```
# Показ экстремумов
```

```
plt.plot(np.pi/2, 1, 'bo',
label='Максимум')plt.plot(3*np.pi/2, -1, 'bo',
label='Минимум')
```

```
plt.legend()plt.show()
```



Drawing No. 1. Graphic function  $y = \sin(x)$

Task: To visualize the probability distribution for a normal distribution using a histogram and a density curve.

```

import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

# Generate random data from normal distribution
data = np.random.normal(loc=0, scale=1, size=1000)

# Build histogram
sns.histplot(data, kde=True, bins=30)

# Graph setting
plt.xlabel('X')plt.ylabel('Frequency')plt.title('Normal distribution')plt.show()

```

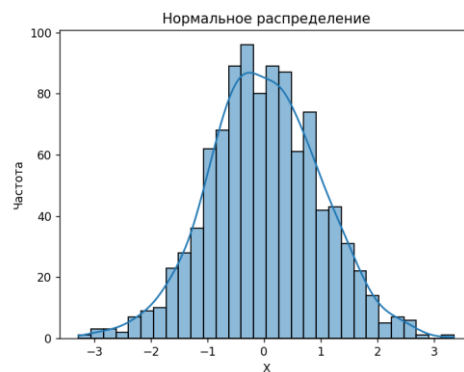


Figure No2. Probability distribution.

Objective: To visualize the surface of the function  $z = x^2 + y^2$  in three-dimensional space.

```

import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D

# Создание сетки значений x и y
x = np.linspace(-2, 2, 50)y = np.linspace(-2, 2, 50)X, Y = np.meshgrid(x, y)

# Вычисление значений функции
Z = X**2 + Y**2

# Создание фигуры и осей 3D
fig = plt.figure()ax = fig.add_subplot(111, projection='3d')

# Отрисовка поверхности
ax.plot_surface(X, Y, Z)

# Настройка осей
ax.set_xlabel('X')ax.set_ylabel('Y')ax.set_zlabel('Z')ax.set_title('Поверхность функции z = x**2 + y**2')plt.show()

```

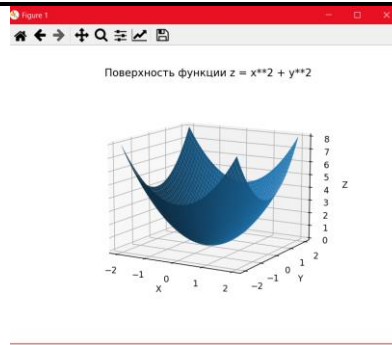


Figure No3. Surface of function  $z = x^2 + y^2$

Visualization in math education is playing an increasingly important role, contributing to a better understanding of abstract concepts and increasing student motivation. Visualization methods are actively developing using modern technologies and digital tools

	Classical methods	Development	Advantages	Tools
Static graphs and charts	Function graphs, diagrams, geometric shapes, drawings.	Use interactive tools to build graphs, add animations, rotate and scale objects.	Easy to understand, they allow you to analyze data and visualize dependencies.	Matplotlib, Seaborn, Plotly, Geogebra.
Animations and interactive models	Animations of geometric transformations, dynamic systems, and the movement of objects.	Creation of interactive simulations where the user can change parameters and observe changes, virtual reality.	They make the study of dynamic processes more visual, allow you to experiment and study phenomena.	Manim, Blender, WebGL.
3D visualization	Modeling of three-dimensional geometric shapes, body of rotation, surfaces.	The use of 3D printing, VR/AR to immerse yourself in three-dimensional space, the use of models of real objects.	It allows you to study geometric objects in real form, better understand their spatial properties.	Blender, Unity, Unreal Engine.
Information cards and graphs	Link diagrams, networks, decision tree.	Interactive maps with the ability to zoom, rotate, filter, visualize structural data.	They allow you to present large amounts of information in a structured form, to study the connections and dependencies between objects.	Gephi, Graphviz, Cytoscape.

Data visualization	Bar charts, scatter charts, boxes with mustaches.	Interactive dashboards, infographics, real-time data visualization.	It allows you to analyze data visually, identify trends and relationships.	Tableau, Power BI, Plotly Dash.
--------------------	---------------------------------------------------	---------------------------------------------------------------------	----------------------------------------------------------------------------	---------------------------------

#### Introduction into education:

- Creation of educational materials using visualization.
- Incorporating visuals into textbooks and workbooks.
- Use of interactive learning platforms.
- Development of special courses on the use of visualization tools.

#### Advantages of using visualization:

- Increasing students' interest in mathematics.
- Better understanding of abstract concepts.
- Development of spatial thinking and analytical skills.
- Creating more effective teaching methods.

#### Trends:

- Use artificial intelligence to automate visualization processes.
- Creating more interactive and immersive visual experiences.
- Development of new tools for visualization of data and structures.

The development of visualization methods in mathematics education opens up new opportunities for improving the learning process and increasing interest in mathematics. It's important to use modern technology and tools to create visuals that are more visual and memorable.

#### References:

1. INTEGRATION OF INTELLECTUAL TESTS IN THE TEACHING OF LILY K. PROGRAMMING LANGUAGES /Involta Scientific Journal. – 2022. – T. 1. – No. 8. – C. 37-45.
2. Kamalova N. I. Methodology of teaching the subject of parameter repetition operator. – 2022.
3. Kirnyaev M.M. Review of the Python programming language for solving

problems of mathematical modeling. 2016. No. URL: <https://cyberleninka.ru/article/n/obzor-yazyka-programmirovaniya-python-dlya-resheniya-zadach-matematicheskogo-modelirovaniya> (accessed: 31.05.2024).

4. Atrokhov K. G. et al. Python in the disciplines of the specialty "Computer Mathematics and System Analysis". – 2023.