Exploring Python's Role in Pure Mathematical Analysis

[1]Dr. Neeru Bala
Department of Mathematics, Dayanand College, Hisar-125001

[2]Ms. Shalu Rani
Department of Computer Science, Dayanand College, Hisar-125001
Affiliated to Guru Jambheshwar University of Science and Technology, Hisar.

Abstract-
In this paper, we studied the relationship between Python and pure mathematics. Both complement and enhance each other's capabilities. Python, a high-level programming language which is very popular for its simplicity, flexibility and readability. It is very easy for new learners to learn and understand its codes. It has become an essential tool in the field of pure mathematics. Mathematicians can apply their expertise in solving real-world problems through the application of Python. We discussed the capabilities and functionalities of Python for solving mathematical problems. Python's rich library provides a computational framework to tackle complex problems in pure mathematics and gain deeper insights. We also discussed in detail how Python's library functions accessible to quickly handle mathematical issues.

Keywords: Python, Pure Mathematics, versatility, computation.

Published in IJIRMPS (E-ISSN: 2349-7300), Volume 9, Issue 5, Sept - Oct 2021

Introduction:

In mathematical research, Python plays a vital role by providing a versatile and powerful computational platform that facilitates efficient exploration, analysis, and implementation of complex mathematical concepts and algorithms. Python stands out as an ideal language for quick learning and practical programming in real-world scenarios. Python is like a super-smart assistant for mathematicians, aiding them in various tasks such as solving equations, working with numbers, and even creating colorful charts to enhance data comprehension. Python, as a user-friendly programming language, is significantly transforming the landscape of mathematical research. Its simplicity and readability have excelled it accessible to mathematicians and empowered them to swiftly sum up the theoretical concepts into executable code. Python's versatility and rich libraries empower researchers to explore complex mathematical problems, conduct experiments, and visualize results seamlessly. The language's interactive nature, especially in tools and facilitates a more iterative and exploratory approach to problem-solving. Key strengths of Python in pure mathematics lie in its symbolic computing capabilities. Numerical computing libraries like NumPy and SciPy further extend Python's utility by providing robust tools for effective collaboration with arrays, matrices, and numerical algorithms. Mathematicians can seamlessly implement and experiment with numerical methods, simulations, and optimizations, thus introducing a computational dimension to their research.

Traditionally, Mathematics has been interchangeable with pen-and-paper proofs, abstract theories, and rigorous analytical thinking. Advancement of technology brought significant progress to research in pure mathematics. Python into the domain of pure mathematics has paved the way for exploration, computation, and visualization, revolutionizing the way mathematicians approach problems and gain insights.

In the context of pure mathematics, Python acts as a link between abstract mathematical concepts and their practical implementation.

Python's Contribution to Pure Mathematics:

Python has greatly enriched the field of pure mathematics by providing powerful libraries for arithmetic processing, symbolic computations and visual data analysis. Let's explore a few key points.

- **Computational Numerical Analysis:** Python offers powerful libraries such as NumPy and SciPy for numerical computations, empowers mathematicians to tackle complex equations, perform matrix operations, and conduct statistical analysis with exceptional precision and efficiency.
- **Symbolic Mathematics:** Libraries like SymPy offers help for mathematical symbolic reasoning and also help to manipulate algebraic expressions, perform symbolic equation solving and derive mathematical identities.
- **Graphical representation:** Python's matplotlib and Plotly libraries facilitate the creation of high-quality Information visualization, including plots, graphs, and dynamic visualizations of mathematical concepts, assisting in the investigation and communication of mathematical ideas.
- **Algebraic Computation and Number Theory:** Python libraries such as SageMath provide extensive tools for computational algebra and number theory, supporting research in areas such as prime factorization, modular arithmetic, and polynomial manipulation.

Mathematical Calculations in Python Programs:

The math module in Python provides a collection of mathematical operations such as functions for trigonometry functions, exponential functions, basic arithmetic operations, logarithmic functions, statistical functions and many more. We can also create mathematical functions as per our requirements. Using Python's math module, we can perform many mathematical calculations:

- **Basic Arithmetic Calculation using Python's Program:** Elementary arithmetic forms the foundation of mathematical calculations, encompassing fundamental operations involving numbers. As commonly understood, there are four basic operations: addition, subtraction, multiplication, and division. Python's Program for basic arithmetic calculation is as follows:

```
a = float(input('Enter the value of a  '))
b = float(input('Enter the value of b '))
sum = a + b
difference = a - b
product = a * b
division = a / b
print('Sum of ',a ,'and' ,b ,'is :',sum)
print('Difference of ',a ,'and' ,b ,'is :',difference)
print('Product of' ,a ,'and' ,b ,'is :',product)
print('Division of ',a ,'and' ,b ,'is :',division)
```

Its output is

```
Enter the value of a  12
Enter the value of b 2
Sum of  12.0 and 2.0 is : 14.0
Difference of  12.0 and 2.0 is : 10.0
Product of 12.0 and 2.0 is : 24.0
Division of  12.0 and 2.0 is : 6.0
```

- **Trigonometry Programs:** Trigonometry is a discipline within mathematics which focuses on the exploration of relationships between the sides of the triangle and angles of triangles. It includes the study of trigonometric functions such as sine, cosine, tangent, and their reciprocal functions, moreover their properties and applications within different areas such as engineering, survey, physics, astronomy and navigation.

Here, we are providing a snippet of few trigonometric functions:

```
import math
angle_deg = float(input('Enter the angle in Degrees '))
angle = math.radians(angle_deg)
ans = math.sin(angle)
print('sine of angle ',int(angle_deg), 'degree is', ans)
```

Its output is

```
Enter the angle in Degrees 90
sine of angle  90 degree is 1.0
```

- **Exponential and Logarithmic Functions:** In Python, we can work with exponential and logarithmic functions by using built-in functions which are available in the field of math module library. Here are examples demonstrating how to use exponential and logarithmic functions.

Program for finding the exponential value of a given number is as follows:

```
import math
x = float(input('Enter the value of x '))
ans=math.exp(x)
print('Exponential value of ', x, 'is' , ans)
```

Its output is

```
Enter the value of x 2
Exponential value of  2.0 is 7.38905609893065
```

Program for finding the logarithm value of a given number is as follows:

```
import math
x = float(input('Enter the value of x '))
ans=math.log(x)
print('Natural Logarithm value of ', x, 'is' , ans)
```

Its output is

```
Enter the value of x 10
Natural Logarithm value of  10.0 is 2.302585092994046
```

- **Power and square roots:** In Python, we can also calculate powers and square roots using various methods. Here are examples demonstrating how to compute powers and square roots.

The following is the program for computing the power of given number

```
import math
ans=math.pow(5,2)
print('Square of 5 is ', ans)
```

Its output is

```
Square of 5 is  25.0
```

The following is the program for determining the square root of a given number

```
import math
x = float(input('Enter the value of x '))
ans=math.sqrt(x)
print('Square root of ', x, 'is' , ans)
```

Its output is

```
Enter the value of x 121
Square root of  121.0 is 11.0
```

- **Conversion of Degrees to Radians:** In Python, by using functions available in math module library, we can convert degrees to radians. The conversion formula is:

$$\text{Radians} = \text{Degrees} \times \frac{\pi}{180}$$

Here's how we carrying out this conversion:

```
import math
angle_deg = float(input('Enter the angle in Degrees '))
angle_rad = math.radians(angle_deg)
print('Angle in degrees is ',angle_deg, '\nIts conversion in
    radians is ',angle_rad)
```

Its output is

```
Enter the angle in Degrees 90
Angle in degrees is  90.0
Its conversion in radians is  1.5707963267948966
```

- **Conversion of Radians to Degrees:** We can transform radians into degrees by Employing functions accessible in the math module. The conversion formula is:

$$\text{Degrees} = \text{Radians} \times \frac{180}{\pi}$$

Here is the method for executing this conversion

```
import math
angle_rad = float(input('Enter the angle in radians '))
angle_deg = math.degrees(angle_rad)
print('Angle in radians is ',angle_rad, '\nIts conversion in
    degrees is ',angle_deg)
```

Its output is

```
Enter the angle in radians 1.57079
Angle in radians is  1.57079
Its conversion in degrees is  89.99963750135457
```

- **GCD or HCF of Two Numbers:** The Greatest Common Divisor (GCD) of two Numbers represent the largest common divisor for a given set of numbers. It is also referred to as as The greatest common divisor (GCD) or the Greatest Common Factor (GCF). For example, the GCD of two numbers 12 and 20 is 4. Different applications of mathematics utilize GCD such as fractions, arithmetic problems and many more. In Python, we can find the GCD of two numbers using the math.gcd() function from the math module. Following is the example to demonstrate how to compute GCD:

```
import math
a = int(input('Enter the first number  '))
b = int(input('Enter the second number '))
ans = math.gcd(a, b)
print('GCD of ', a, 'and', b, 'is', ans)
```

Its output is

```
Enter the first number  70
Enter the second number 15
GCD of  70 and 15 is 5
```

These applications demonstrate the capability of programming language Python and its solid incorporation of mathematical concepts, turning it into a preferred option for a broad spectrum of mathematical computations, analyses, and simulations across multiple domains. For more advanced mathematical and numerical operations, you might want to explore additional libraries such as NumPy and SciPy. NumPy stands for Numerical Python. Travis Oliphant created NumPy in 2005, and it operates as an open-source project, freely available for use. SciPy stands for Scientific Python. SciPy is a Python library employed for scientific computation and statistical analysis. In 2001, Travis Oliphant, Eric Jones, and Pearu Peterson developed Python to create a complete scientific computing environment in Python.

Conclusion:

In conclusion, this paper analyzed the Influence of Python on pure mathematics. The language's clear-cut syntax, coupled with its well-built libraries and adaptability, facilitates mathematicians to bridge the gap between Formulation and computational analysis. In mathematical research, Python is providing a powerful toolkit for exploration, analysis, and communication in the constantly changing environment of pure mathematics.

REFERENCES:

1. Brown, M. C. (2001) Python: The Complete Reference. McGraw-Hill Professional.
2. Jones, E. Oliphant, T and Peterson, P. (2001). SciPy: OpenSource Scientific Tools for Python.
3. Millman, K. J., and Aivazis M. (2011). Python for Scientists and Engineers. Computing in Science & Engineering, 13(2), 9-12.
4. Oliphant, T. (2007). Python for Scientific Computing. Computing in Science & Engineering, 9(3), 10-20.
5. Smedt, T. D., and Daelemans, W. (2012). Pattern for Python. Journal of Machine Learning Research, 13, 2063-2067.
6. Srinath, K. R. (2017). Python - The Fastest Growing Programming Language. International Research Journal of Engineering and Technology, 04(12), 354-357.