European Journal of Interdisciplinary Research and Development
Volume- 25 March - 2024
Website: www.ejird.journalspark.org ISSN (E): 2720-5746

PERFORMING COMPLEX MATHEMATICAL CALCULATIONS IN PYTHON USING THE SYMPY LIBRARY

Raupova M. H.
Chirchik State Pedagogical University
r.mokhinur@gmail.com

Xasanova M. F.
Sophomore student of Chirchik State Pedagogical University
xasanovamohichehra668@gmail.com

Abstract:
This article describes the use and importance of Python's Sympy library. For users new to Python, here are some helpful tips on how to install and use the SymPy library. A number of composed mathematical examples are implemented in Python using the SymPy library.

Keywords: SymPy, Python, integrate, differential, dsolve.

Introduction

The Python programming language is one of the most popular and widely used programming languages known for having more than 137,000 libraries, and several libraries are widely used for performing mathematical operations and data analysis. A Python library is a collection of modules that contain functions and classes that can be used by other programs to perform various tasks. Top libraries useful for working with math in Python include Math Module, NumPy, SymPy, Pandas, Scipy, MatplotLib, Scikit Learn, and more. The SymPy library allows you to create 2D and 3D plots of integrals, differential equations, discrete mathematics, geometry, and mathematical functions. In this article, we will look at several mathematical tasks using the SymPy library. First, when working with Python libraries on a computer, we download one of the latest versions of PyCHarm, Visual Studio Code, Spyder, Sublime text, or Python, an IDE for the Python programming language, and download the latest version of this program to the Python website (https:/ /www.python.org/downloads/) and enter the "Installation (Dowloads)" section and install the operating system suitable for our computer. Then we access the SymPy library using the term pip install sympy in a terminal or command line.
Once the process is complete, we can check the installation.

```
Terminal   Local
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Установите последнюю версию PowerShell для новых функций и улучшения! https://aka.ms/PSWindows

(venv) PS C:\Users\HP\PycharmProjects\SymBOlic> pip install sympy
```

So we can see that SymPy version 1.12 is loaded on our computer.
Now let's look at some examples using SymPy library functions.

Example 1. Calculate the limit: $\lim_{x\to 0}\frac{\sqrt{x^2+9}-3}{x^2}$

Solving: $\lim_{x\to 0}\frac{\sqrt{x^2+9}-3}{x^2} = \lim_{x\to 0}\frac{\sqrt{x^2+9}-3}{x^2}\cdot\frac{\sqrt{x^2+9}+3}{\sqrt{x^2+9}+3} = \lim_{x\to 0}\frac{\left(\sqrt{x^2+9}\right)^2-(3)^2}{x^2\cdot\left(\sqrt{x^2+9}+3\right)} = \lim_{x\to 0}\frac{1}{\sqrt{x^2+9}+3} = \frac{1}{6}$

Answer: $\frac{1}{6}$

Software code:

```
from sympy import*
#            We            import            Sympy            library            variables
x=Symbol('x')
# We choose the symbolic variable as x
y=limit((sqrt(x**2+9)-3)/(x**2), x, 0)
# We enter the limit function, we enter the expression given to its 1st argument, the variable to its 2nd argument, and the desired number to its 3rd argument
print(y)
# This line prints the value of the expression
```

Example 2. Find the 4th $y^{(100)}$ - derivative of the function: $y = x\cdot shx$.

Solving: $y = x\cdot shx$

$$y^{(1)} = (x \cdot shx)^{(1)} = shx + x \cdot chx$$

$$y^{(2)} = (shx + x \cdot chx)^{(2)} = 2chx + x \cdot shx$$

$$y^{(3)} = (2chx + x \cdot shx)^{(3)} = 3shx + x \cdot chx$$

....

....

$$y^{(100)} = (99shx + x \cdot chx)^{(100)} = 100chx + x \cdot shx$$

Answer: $y^{(100)} = x \cdot shx + 100chx$

Software code:


```
from sympy import *
# In this line, we call the SymPy library by the name SymPy.
x = symbols('x')
# to enter variables, we enter 'x' through the symbol function
n=int(input("n="))
# to find the n(arbitrary integer)-order derivative of the given expression
y = diff(x*sinh(x), x,n)
# Using the diff (y, x, n) function, we enter the expression given to the 1st argument, the variable to the 2nd argument, and the order of the derivative to the 3rd argument.
print(y)
# This array prints the value of the expression.
```

Example 2. Calculate the $\int \frac{1}{\cos x} dx$.

Solving:

$$\int \frac{1}{\cos x} dx = \int \frac{\cos x}{\cos x \cdot \cos x} dx = \int \frac{\cos x}{\cos^2 x} = \left(\sin^2 x = 1 - \cos^2 x\right) = \int \frac{\cos x}{1 - \sin^2 x} dx = \begin{pmatrix} \sin x = t \\ \cos x dx = dt \end{pmatrix} =$$

$$= \int \frac{du}{1 - t^2} = -\int \frac{1}{(1-t)\cdot(1+t)} dt = -\frac{1}{2} \int \left(\frac{1}{t-1} - \frac{1}{t+1} \right) dt = \frac{1}{2} \ln \frac{t+1}{t-1} + c = \frac{1}{2} \ln \left| \frac{t+1}{t-1} \right| + c = \frac{1}{2} \ln \left| \frac{\sin x + 1}{\sin x - 1} \right| + c$$

Answer: $\frac{1}{2} \ln \left| \frac{\sin x + 1}{\sin x - 1} \right| + c$

Software code: We calculate the integral using the *Integrate ()* function.

```
# First, we import the necessary modules:
from sympy import Symbol, integrate, cos
# We enter the variable in the integral through Symbol
x = Symbol('x')
# We enter the expression in our integral:
s = 1 / cos(x)
# We call the Integrate function:
integral = integrate (s, x)
# we print the result stored in the integral variable
print(integral)
```

The SymPy library has additional functions for working with indefinite integrals, multiple integrals, and improper integrals.

Example 4. Solve the differential equation: $xdy-(x-2y)dx=0$

Solving:

$$xdy-(x-2y)dx\,/\vdots\,dx=0\Rightarrow x\frac{dx}{dy}-(x-2y)=0\Rightarrow x\cdot y'-(x-2y)=0\Rightarrow y=tx\Rightarrow y'=t+t'x$$

$$\Rightarrow \ln(1+t)=\ln(x\cdot c)\Rightarrow 1+\frac{y}{x}=xc\Rightarrow y=x(xc-1)$$

$$\Rightarrow x\cdot(t+t'\cdot x)-(x+2tx)=0\Rightarrow x\cdot t+t'\cdot x^2-x-2tx=0\Rightarrow t'x^2=x(1+t)\Rightarrow\frac{t'}{1+t}=\frac{1}{x}\Rightarrow\frac{dt}{1+t}=\frac{dx}{x}$$

Answer: $y=x(xc-1)$

Software code: Sympy's **dsolve ()** function can be used to solve ordinary differential equations.

```
# we can import the necessary functions in this line
from sympy import symbols, Function, Eq, dsolve
x = symbols('x')
# we enter x as a variable using the symbol function
y = Function('y')(x)
# We select "y" as a function through the Function function.
```

```
diffeq = Eq(x*y.diff(x),x+2*y)
```

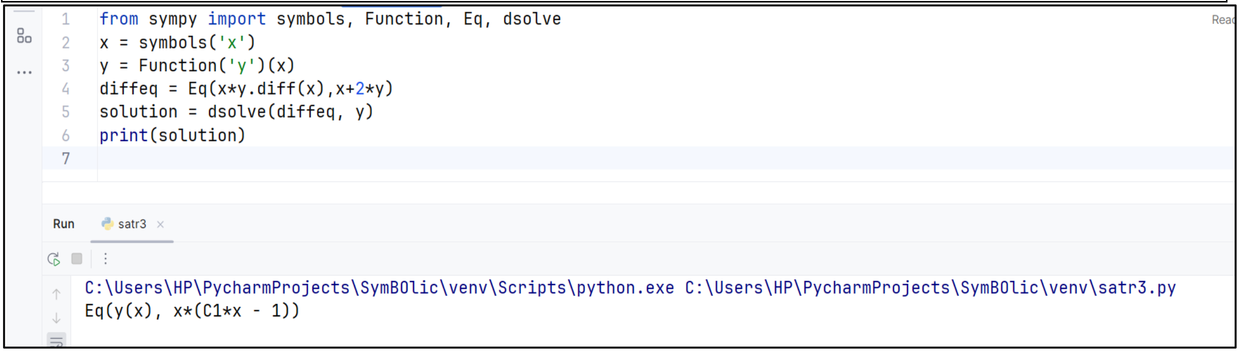
*# The Eq function in this line represents mathematical equations. x*y.diff(x) represents the derivative of this function with respect to the variable, i.e.* $\frac{dy}{dx}$*, and x+2*y represents the sum x*y'(x)=x+2*y(x)*

```
solution = dsolve(diffeq, y)
```

 # The dsolve function finds analytical solutions to differential equations.

```
print(solution)
```

 # Displays the result



Example 5. Divide by multiples: $(x+y)^6$

Solving: We can solve this problem using Pascal's triangle or the binomial theorem. Using the theorem of binomials, we can form

$$(x+y)^n = \sum_{k=0}^{n} C_n^k \cdot a^{n-k} \cdot b^k$$

$$C_n^k = \frac{n!}{k!(n-k)!} \Rightarrow (x+y)^6 = C_6^0 \cdot x^{6-0} \cdot y^0 + C_6^1 \cdot x^{6-1} \cdot y^1 + C_6^2 \cdot x^{6-2} \cdot y^2 + C_6^3 \cdot x^{6-3} \cdot y^3 + C_6^4 \cdot x^{6-4} \cdot y^4 +$$

$$+C_6^5 \cdot x^{6-5} \cdot y^5 + C_6^6 \cdot x^{6-6} \cdot y^6 = x^6 + 6x^5y + 15x^4y^2 + 20x^3y^3 + 15x^2y^4 + 6xy^5 + y^6 .$$

Software code:

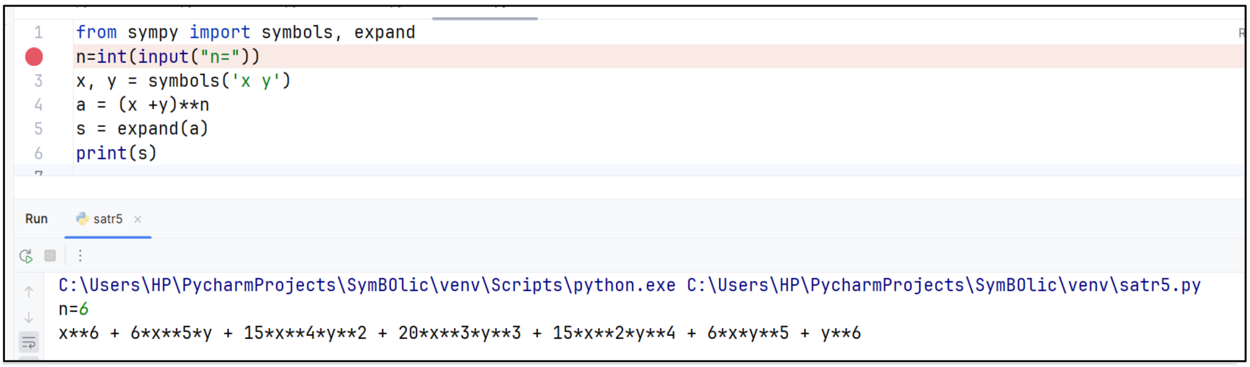
```
from sympy import symbols, expand
```

 # This line imports the symbols and expand functions from the sympy module

```
n=int(input("n="))
```

*# We insert n in the expression (x+y)**n to expand to an arbitrary integer level*

```
x, y = symbols('x y')
```



```
# Variables are included in this code
a = (x +y)**n
# expression is included
s = expand(a)
# The expand function expands the expression
print(s)
# prints the expression
```

Here are some examples of many features of the SymPy library. For mathematics, the SymPy library is a powerful library for simplifying expressions, solving equations, inequalities, matrix and linear algebra operations, derivative and integral calculus. The importance of the SymPy library in education provides an opportunity to integrate mathematical operations into the Python programming language. The SymPy library allows you to simplify mathematical examples, create books and electronic textbooks. Mathematicians can collaborate with programmers to develop algorithms for new SymPy functions. The SymPy library is convenient and useful for teaching mathematics, and its versatility makes it a valuable tool for many branches of mathematics.

References

1. Ashurova M.M. (2021) Methodology of creating and using didactic materials for students using the Python programming language "Economics and society" pp. 50-54.
2. Kuzmina M. A. (2017) The role of independent work in teaching a foreign language of students of technical university" Bulletin of MSRU. Series: Pedagogics 108-114 p.
3. https://datascientest.com/en/python-the-most-popular-programming-language
4. Musayev H., Soliev B. Public, Protected, Private Members In Python //Potomki Al-Fargani. - 2023. - T. 1. – no. 1. – S. 43-46.
5. SH. M. X., M. S. A. Rules for writing a program in Python // Theory of the latest scientific research. - 2023. - t. 6. – no. 4. - p. 113-119
6. Ashurova M.M. (2021) Python dasturlash tili yordamida o'quvchilarga o'quv- didaktik materiallar yaratish va ulardan foydalanish metodikasi "Экономика и социум" 50-54 betlar.
7. Kuzmina M. A. (2017) The role of independent work in teaching a foreign language of students of technical university" Bulletin of MSRU. Series: Pedagogics 108-114 b.
8. https://datascientest.com/en/python-the-most-popular-programming-language
9. Musayev X., Soliev B. Public, Protected, Private Members In Python //Потомки Аль-Фаргани. – 2023. – Т. 1. – №. 1. – С. 43-46.
10. SH. M. X., M. S. A. pythonda dastur yozish qoidalari //so 'ngi ilmiy tadqiqotlar nazariyasi. – 2023. – т. 6. – №. 4. – с. 113-119.