BAREKENG: Journal of Mathematics and Its Applications
December 2024 Volume 18 Issue 4 Page 2531–2542
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss4pp2531-2542

PYTHON IN ORDINARY DIFFERENTIAL EQUATIONS LEARNING

Sigit Sugiarto[1*], John Nandito Lekitoo[2], Ratnah Kurniati MA[3]

[1,2,3]Department of Mathematics Education, Study Program Outside the Main Campus (PSDKU), Universitas Pattimura
Jln. Kampung Babar, Tiakur, 97442, Indonesia

*Corresponding author's e-mail: *sigith.sugiarto@gmail.com*

Article History:

Received: 25th, April 2024
Revised: 9th, June 2024
Accepted: 5th,August 2024
Published:14th, October 2024

Keywords:

Ordinary Differential Equations;
Python;
Jupyter Lab;
SymPy Library;
Symbolic Exact Solution.

ABSTRACT

Using software in mathematics learning can improve students' soft and hard mathematics skills at the high school and college levels. Therefore, using software in the learning process is important, including in learning Differential Equations. This research examines the use of the Python programming language with Jupyter Lab software and the SymPy library in solving ordinary differential equation problems symbolically in the Differential Equations course. The use of the Python programming language in Differential Equations learning includes solving linear ordinary differential equations of first-order, second-order, higher-order, and the Laplace transform. This research also examines the effect of using Python on the learning outcomes of differential equations of Mathematics Education Study Program students, Study Program Outside the Main Campus, Pattimura University. The population in this quantitative research is all students who programmed differential equations courses in the even semester of the 2023-2024 academic year as many as 19 students. The Python programming language can be used to solve differential equation problems symbolically easily, quickly, and accurately. In addition, using Jupyter Lab makes the process of solving differential equation problems easier and more interactive. Furthermore, t-test results show that the use of Python in learning differential equations can improve students' learning activities and learning outcomes. Using the Python programming language with Jupyter Lab software and the SymPy library can be developed to create teaching materials, textbooks, and reference books for Differential Equations courses.

How to cite this article:

S. Sugiarto, J. N. Lekitoo and R. Kurniati MA., "PYTHON IN ORDINARY DIFFERENTIAL EQUATIONS LEARNING," *BAREKENG: J. Math. & App.*, vol. 18, iss. 4, pp. 2531-2542, December, 2024.

Journal homepage:*https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article ·Open Access

1. INTRODUCTION

The development of Mathematics and Mathematics learning cannot be separated from technological developments such as software developments in the fields of mathematics and mathematics learning[1]. The use of software in mathematics learning over the last decade has continued to grow in various countries and published in indexed international journals[2]. Many teachers and researchers have developed software-based learning processes [3][4][5]. This is because the use of software in mathematics learning can improve students' mathematical soft skills and hard skills starting from middle school to college level [6][7]. Therefore, the use of software in learning mathematics is important, including in learning differential equations at the college level.

Students of Mathematics Education study program often have difficulty in determining the solution of differential equation problems. In addition, students also have difficulty in validating the solutions they have obtained. Thus, students need software assistance in learning differential equations. There is a lot of software that can be used to help the process of differential equations learning in solving problems analytically, including Matlab, Maple, and Mathematica. However, this software is paid software with relatively high costs, which is difficult for students and lecturers to purchase [8], even for a developing institution. This is also of course a burden for the Mathematics Education Study Program at the Southwest Maluku Regency Campus which is located in the Small Border Island area with the majority of students having a low economic level[9]. Therefore, in the process of differential equations learning, lecturers are required to be able to utilize affordable software or free software that can be used to solve differential equation problems.

Software has a very important role in finding analytical solutions to mathematics problems including differential equation problems. Several reasons why software has an important role in differential learning, namely: 1) Efficiency in calculations: Software can carry out complex mathematical calculations quickly and accurately. This allows students to explore a variety of analytical solutions in less time than if done manually; 2) Computational capabilities: Existing differential equation problems are often very complex and difficult to solve manually. The software can solve complex differential equation problems with appropriate numerical or analytical methods; and 3) Testing and verification of solutions: Software allows testing and verification of found solutions. By using software, solutions can be verified thereby increasing confidence in the results obtained.

One way to solve differential equation problems is to use the Python programming language (https://www.python.org/) with Jupyter Lab software (https://jupyter.org/) and the SymPy library (https://www. sympy.org/en/index.html). Python is a programming language that can be used for free and is open source. Python can be used and developed by users freely according to user needs. Python is available on various operating systems, namely Windows, Mac OS X, and Linux. Python can be used in many application domains, one of which is in the field of Scientific and Numerical [10]. The use of Jupyter Lab in solving differential equation problems provides ease of use and an interactive process. Furthermore, the SymPy library is a library that can be used to solve mathematical problems symbolically. In this case, the solution obtained is exact.

The Python programming language itself is nothing new for teachers and researchers. The Python programming language has been widely used in mathematical fields such as linear algebra, numerical methods [10][11], data analysis[12], data science [13][14], linear programming [15], dynamic system modeling [11][16], and so on which are available in learning books and research journals. However, the use of Python programming in solving differential equation problems to determine exact solutions symbolically has not been widely used. Therefore, it is important to research the use of the Python programming language in solving differential equation problems precisely to help the process of learning differential equations.

2. RESEARCH METHODS

This research examines how to use the SymPy library in Python to solve ordinary differential equations (ODE) in differential equations learning. This article will describe the use of the Python programming language in solving homogeneous and non-homogeneous linear ODE. Differential equations

is a mandatory subject for students of Mathematics, Mathematics Education, and other study programs. ODE learning for undergraduate students includes the following material: 1) first-order differential equations; 2) second-order homogeneous differential equations; 3) second-order homogeneous differential equations with variable coefficients; 4) second-order non-homogeneous differential equations; 5) higher-order homogeneous differential equations; 6) higher-order non-homogeneous differential equations; and the Laplace transform.

Furthermore, this research examines the effect of using Python on the learning outcomes of differential equations of Mathematics Education Study Program students, Study Program Outside the Main Campus, Pattimura University. The population in this quantitative research consisted of all students who programmed differential equations courses in the even semester of the 2023-2024 academic year as many as 19 students. The sampling technique used total sampling, by taking the entire population as samples in the study. The research design used is the One-Group Pretest-Postest Design type which is done by comparing the condition before treatment and the condition after treatment.

The data used in this study is the data of student learning outcomes in the differential equations course which is the result of the midterm exam and the final exam. The results of the midterm exam are used as pretest data, namely before the use of Python in learning. The results of the final semester exam are used as posttest data, namely after treatment with the use of Python in learning differential equations. The instrument used to collect data is a differential equation learning outcome test instrument in the form of a description test. Before use, the test instrument was first validated using expert validation, namely the panelist test.

Furthermore, the data obtained were then analyzed by descriptive statistical analysis and inferential statistics. Descriptive analysis in this study is needed to describe the learning outcomes through mean, median, mode, variance, minimum, and maximum. Inferential analysis is intended to test the hypothesis of the impact of using Python on student learning outcomes in differential equations courses. Before conducting hypothesis testing, researchers carried out the prerequisite analysis.

3. RESULTS AND DISCUSSION

The results of this study begin by describing how to use Python in solving differential equation problems. Followed by describing the effect of using Python on student learning outcomes of differential equations.

3.1 How to Use Python in Differential Equations

Python programming language with Jupyter Lab software and SymPy library can be used to solve ODE problems. The solution of ODE is either a symbolic solution or an exact solution. The use of the Python programming language with Jupyter Lab software and the SymPy library in solving ODE problems begins by calling the SymPy library and defining the symbols x and $y(x)$ as follows:

```
[1]:  from sympy import *
      init_printing()
[2]:  x = symbols('x')
[3]:  y = Function('y')(x)
```

Then, write the ODE that will be solved and show the solution which can be described as follows:

3.1.1 First-Order Differential Equations

The differential equations that will be described in this section are first-order ODE, including the Bernoulli equations.

Example 1. Solve the differential equation $y' = -xy^2$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]:  eq = Eq(y.diff(x),-x*y**2)
[5]:  eq
[6]:  sl = dsolve(eq,y)
[7]:  sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y' = -xy^2$, namely

$$y(x) = \frac{2}{C_1 + x^2}.$$

Example 2. (Bernoulli Equation) Solve the differential equation $y' + \frac{1}{x}y = xy^2$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]:  eq = Eq(y.diff(x)+(1/x)*y, x*y**2)
[5]:  eq
[6]:  sl = dsolve(eq,y)
[7]:  sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y' + \frac{1}{x}y = xy^2$, namely

$$y(x) = \frac{1}{x(C_1 - x)}.$$

Example 3. (Initial Value Problem) Find the general solution and specific solution of the differential equation $y^2y' = x$ and $y(0) = 1$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]:  eq = Eq(y**2*y.diff(x), x)
[5]:  eq
[6]:  sl = dsolve(eq,y)
[7]:  sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y^2y' = x$, namely

$$y(x) = \sqrt[3]{C_1 + \frac{3x^2}{2}}.$$

Moreover, to obtain the particular solution, the following command is used:

```
[8]:  na = {y.subs(x,0):1}
[9]:  sk = dsolve(eq,y,ics=na)
[10]: sk
```

Then, Python (Jupyter Lab) will immediately display the specific solution of the differential equation $y^2y' = x$ and $y(0) = 1$, namely

$$y(x) = \sqrt[3]{\frac{3x^2}{2} + 1}.$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve first-order ODE quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.2 Second-order Homogeneous Differential Equation

The differential equations defined in this section are second-order homogeneous linear ODE with constant coefficients.

Example 4. Solve the differential equation $y'' + y' - 6y = 0$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x)+y.diff(x)-6*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y'' + y' - 6y = 0$, namely

$$y(x) = C_1 e^{-3x} + C_2 e^{2x}.$$

Example 5. Solve the differential equation $4y'' - 4y' + y = 0$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(4*y.diff(x,x)-4*y.diff(x)+y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $4y'' - 4y' + y = 0$, namely

$$y(x) = (C_1 + C_2 x)e^{\frac{x}{2}}.$$

Example 6. (Initial Value Problem) Find the general solution and specific solution of the differential equation $y'' - 4y' + 5y = 0$ with $y(0) = -1$ and $y'(0) = 1$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x)-4*y.diff(x)+5*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y'' - 4y' + 5y = 0$, namely

$$y(x) = (C_1 \sin(x) + C_2 \cos(x))e^{2x}.$$

Then, to obtain the particular solution, the following command is used:

```
[8]: na = {y.subs(x,0):-1, y.diff(x).subs(x,0):1}
[9]: sk = dsolve(eq,y,ics=na)
[10]: sk
```

Then, Python (Jupyter Lab) will quickly display the specific solution of the differential equation $y'' - 4y' + 5y = 0$ with $y(0) = -1$ and $y'(0) = 1$, namely

$$y(x) = (3 \sin(x) - \cos(x))e^{2x}.$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve second-order homogeneous linear ODE with constant coefficients quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.3 Second-order Homogeneous Differential Equation with Variable Coefficients

Order reduction is one of several ways to solve second-order homogeneous linear ODE with variable coefficients in the differential equations subject. In addition, there is also the Cauchy–Euler equation.

Example 7. One of the solutions to the differential equation $x^2y''(x) - 2xy'(x) - 4y(x) = 0$ is $1/x$ for $x > 0$. Find another solution to the differential equation using order reduction.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(x**2*y.diff(x,x)-2*x*y.diff(x)-4*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the particular solution of the differential equation $x^2y''(x) - 2xy'(x) - 4y(x) = 0$, namely

$$y(x) = \frac{C_1}{x} + C_2x^4.$$

In this case another solution to the differential equation $x^2y''(x) - 2xy'(x) - 4y(x) = 0$ is $y(x) = C_2x^4$.

Example 8. (Cauchy–Euler Equation) Find the general solution and specific solution of the differential equation $x^2y'' - 4xy' + 6y = 0$, $y(1) = 0$, and $y'(1) = 1$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(x**2*y.diff(x,x)-4*x*y.diff(x)+6*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the general solution of the differential equation $x^2y'' - 4xy' + 6y = 0$, namely

$$y(x) = x^2(C_1 + C_2x).$$

To obtain the particular solution, the following command is used:

```
[8]: na = {y.subs(x,1):0, y.diff(x).subs(x,1):1}
[9]: sk = dsolve(eq,y,ics=na)
[10]: sk
```

Python (Jupyter Lab) will immediately display the particular solution of the differential equation $x^2y'' - 4xy' + 6y = 0$, $y(1) = 0$, and $y'(1) = 1$, namely

$$y(x) = x^2(x - 1).$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve second-order homogeneous linear ODE with variable coefficients quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.4 Second-order Non Homogeneous Differential Equation

The differential equation in this section is a second-order non-homogeneous linear ODE with constant coefficients.

Example 9. Solve the differential equation $y'' + y' - 2y = 4x + 18xe^x$.

Solution. The solution of ODEuses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x)+y.diff(x)-2*y,4*x+18*x*exp(x))
[5]: eq
```

[6]: sl = dsolve(eq,y)
[7]: sl

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y'' + y' - 2y = 4x + 18xe^x$, namely

$$y(x) = C_2e^{-2x} - 2x + (C_1 + 3x^2 - 2x)e^x - 1.$$

Example 10. Solve the differential equation $y'' = \frac{1}{x+1}$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

[4]: eq = Eq(y.diff(x,x), 1/(x+1))
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y'' = \frac{1}{x+1}$, namely

$$y(x) = C_1 + x(C_2 + \log(x+1)) + \log(x+1).$$

In this case, Python presents $\log x$ as $\ln x$. So, the solution in question is $y(x) = C_1 + x(C_2 + \ln(x+1)) + \ln(x+1)$.

Example 11. (More Complicated Problems) Solve the differential equation $y'' + 4y' + 13y = 3\sin(3x) - 5\cos(3x)$**[8]**.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

[4]: eq = Eq(y.diff(x,x)+4*y.diff(x)+13*y,3*sin(3*x)-5*cos(3*x))
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y'' + 4y' + 13y = 3\sin(3x) - 5\cos(3x)$, namely

$$y(x) = (C_1\sin(3x) + C_2\cos(3x))e^{-2x} - \frac{3\sin(3x)}{10} - \frac{7\cos(3x)}{20}.$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve second-order non-homogeneous linear ODE quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.5 Higher Order Homogeneous Differential Equation

The process of solving higher-order homogeneous ODE with constant coefficients is similar to solving second-order ODE with constant coefficients. Here are some examples of solving higher-order ODE with constant coefficients.

Example 12. Solve the differential equation $y''' - 2y'' - y' + 2y = 0$.

Solution. The solution of ODEuses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

[4]: eq = Eq(y.diff(x,x,x)-2*y.diff(x,x)-y.diff(x)+2*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y''' - 2y'' - y' + 2y = 0$, namely

$$y(x) = C_1 e^{-x} + C_2 e^{x} + C_3 e^{2x}.$$

Example 13. Solve the differential equation $y^{(4)} - 5y''' + 3y'' + 19y' - 30y = 0$.

Solution. The solution of ODEuses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x,x,x)-5*y.diff(x,x,x)+3*y.diff(x,x)+19*y.diff(x)-30*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y^{(4)} - 5y''' + 3y'' + 19y' - 30y = 0$, namely

$$y(x) = C_3 e^{-2x} + C_4 e^{3x} + (C_1 \sin(x) + C_2 \cos(x)) e^{2x}.$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve higher-order homogeneous linear ODE quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.6 Higher Order Non-Homogeneous Differential Equation

The process of solving higher-order non-homogeneous ODE with constant coefficients is similar to finding the solution of second-order ODE with constant coefficients. Here are some examples of solving higher-order non-homogeneous ODE with constant coefficients.

Example 14. Solve the differential equation $y^{(4)} - 5y''' + 5y'' + 5y' - 6y = 4e^{2x}$.

Solution. The solution of ODEuses the Python programming language (Jupyter Lab) proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x,x,x)-5*y.diff(x,x,x)+5*y.diff(x,x)+5*y.diff(x)-6*y,4*exp(2*x))
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y^{(4)} - 5y''' + 5y'' + 5y' - 6y = 4e^{2x}$, namely

$$y(x) = C_2 e^{-x} + C_3 e^{x} + C_4 e^{3x} + \left(C_1 - \frac{4x}{3}\right) e^{2x}.$$

Example 15. Solve the differential equation $y''' - 3y'' + y' + 5y = e^{2x} + \cos x$.

Solution. The solution of ODE uses the Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[4]: eq = Eq(y.diff(x,x,x)-3*y.diff(x,x)+y.diff(x)+5*y,exp(2*x)+cos(x))
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Furthermore, Python (Jupyter Lab) will immediately display the exact solution of the differential equation $y''' - 3y'' + y' + 5y = e^{2x} + \cos x$, namely

$$y(x) = C_3 e^{-x} + \left(C_1 \sin(x) + C_2 \cos(x) + \frac{1}{3}\right) e^{2x} + \frac{\cos(x)}{8}.$$

Besides being able to determine the final solution of an ODE, Python with the SymPy library is also able to solve ODE procedurally following the steps of a particular method. As an example, the following will describe the solution of a higher-order non-homogeneous linear ODE using the parameter variation method.

Example 16. Find the general solution of the differential equation $y''' - 2y'' - y' + 2y = 6e^{x}$ by using the parameter variation method.

Solution. The solution of ODE using the Python programming language (Jupyter Lab) and proceeds with the following steps and syntax:
First, determine the homogeneous solution ($y_h(x)$) with the following syntax:

```
[4]: eq = Eq(y.diff(x,x,x)-2*y.diff(x,x)-y.diff(x)+2*y,0)
[5]: eq
[6]: sl = dsolve(eq,y)
[7]: sl
```

Second, determine the value of $u_i{}'$ by solving the system of linear equations with the following syntax:

```
[8]: A = Matrix(([exp(-x),exp(x),exp(2*x)],[-exp(-x),exp(x),2*exp(2*x)],[exp(-
     x),exp(x),4*exp(2*x)]))
[9]: b = Matrix(([0],[0],[6*exp(x)]))
[10]: ua = A.LUsolve(b)
[11]: ua
```

Third, determine the value of u_i through integrating $u_i{}'$ with the following syntax:

```
[12]: u1 = integrate(ua[0], x)
[13]: u1
[14]: u2 = integrate(ua[1], x)
[15]: u2
[16]: u3 = integrate(ua[2], x)
[17]: u3
```

Fourth, write down the particular solution, i.e.:

$$y_p(x) = -\frac{3}{2}e^x - 3xe^x.$$

Afterward, write down the general solution of the differential equation, $y(x) = y_h(x) + y_p(x)$, as follows:

$$y(x) = C_1e^{-x} + C_2e^x + C_3e^{2x} - \frac{3}{2}e^x - 3xe^x.$$

Based on the examples above, it is clear that Python with the SymPy library can be used to solve higher-order non-homogeneous linear ODE by following the steps of a particular method quickly, easily, and accurately. In addition, the solution obtained is exact.

3.1.7 The Laplace Transform

The SymPy library in Python can also be used to solve problems related to Laplace transform and inverse Laplace transform.

Example 17. Find the $\mathcal{L}\{f(x)\}$ of the function $f(x) = 4 - 2x$.

Solution. Solution of Laplace transform using Python programming language (Jupyter Lab) and proceeds with the following syntax:

```
[2]: x, s = symbols('x s')
[3]: f = Function('f')(x)
[4]: f = 4-2*x
[5]: L = laplace_transform(f, x, s)
[6]: L
```

Moreover, Python (Jupyter Lab) will immediately display the Laplace transform of the function $f(x) = 4 - 2x$, which is

$$\mathcal{L}\{f(x)\} = \frac{4}{s} - \frac{2}{s^2}.$$

Example 18. Find $\mathcal{L}^{-1}\{f(x)\}$ of $\mathcal{L}\{f(x)\} = \frac{4}{s} - \frac{2}{s^2}$.

Solution. The solution of inverse Laplace transform using the Python programming language (Jupyter Lab) is solved by the following syntax:

```
[2]:  x, s = symbols('x s')
[3]:  L = Function('L')(s)
[4]:  L = 4/s-2/(s**2)
[5]:  f = inverse_laplace_transform(L, s, x)
[6]:  f
```

Furthermore, Python (Jupyter Lab) will immediately display the inverse Laplace transform of the $\mathcal{L}\{f(x)\} = \frac{4}{s} - \frac{2}{s^2}$, which is

$$f(x) = -2x\theta(x) + 4\theta(x).$$

It shows that $f(x) = -2x + 4$.

Based on the example above, it is clear that the SymPy library in Python can be used to solve Laplace transform and inverse Laplace transform problems quickly and accurately.

Python with Jupyter Lab software and SymPy library can be used to solve differential equation problems symbolically easily, quickly, and accurately. The use of Jupyter Lab allows the users to solve differential equation problems interactively. The existence of the SymPy library makes users do not need in-depth knowledge of programming languages. Python with Jupyter Lab software and SymPy library can be used by everyone because it can be obtained for free on their respective websites. In addition, Python is a popular programming language nowadays, so there are many references for users to utilize the Python programming language.

3.2 The Effect of Using Python in Learning Differential Equations

The data of students' differential equation learning outcomes both pretest and posttest were analyzed using descriptive statistical analysis and inferential statistics. The analysis was conducted to determine the effect of using Python on the learning outcomes of students' differential equations.

3.2.1 Descriptive Statistical Analysis

The data of students' differential equation learning outcomes both the pre-test and first post-test were analyzed using descriptive statistical analysis with the results can be seen in **Table 1**.

Table 1. Descriptive Analysis of Student Differential Equation Learning Outcomes

Descriptive Statistics	Pre-test	Post-test
Mean	64.12	75.00
Median	61.67	71.00
Mode	90.00	71.00
Variance	431.27	279.33
Minimum	20.00	35.00
Maximum	95.00	90.00

Table 1 shows that the mean, median, and mode of the pre-test data are 64.12, 61.67, and 90.00. Whereas the mean, median, and mode of the post-test data are 75.00, 71.00, and 71.00. Furthermore, the variance, minimum value, and maximum value of the pretest data are 431.27, 20.00, and 95.00, respectively. Meanwhile, the variance, minimum value, and maximum value of pretest data are 279.33, 35.00, and 90.00, respectively.

3.2.2 Inferential Statistical Analysis

Before conducting hypothesis testing, the prerequisite analysis test was conducted, which was a data normality test using the Kolmogorov-Smirnov test and a variance homogeneity test. The results of the data normality test showed that the pre-test and post-test data were both normally distributed. Meanwhile, the data homogeneity test showed that both data groups had homogeneous variances. The hypothesis test was then conducted using a paired t-test. The paired t-test results show that there is a significant difference between the pre-test and post-test data. In this case, the post-test score after treatment using Python in learning differential equations is better than before treatment.

Besides improving learning outcomes, observation results show that the use of Python can improve student learning activities in learning differential equations. This was indicated by the students who became more active in communicating their answers to differential equation problems in front of the class. In addition, the lecturer becomes easier in delivering learning materials and explaining concepts related to solving differential equations. Thus, the use of Python in learning differential equations can provide advantages for both students and the lecturer himself.

The use of Python in learning differential subjects can help teachers or lecturers in delivering lessons about solving differential equation problems. The use of software in the learning process can also increase students' motivation and interest in learning **[17][18]**. In addition, the use of software in the learning process can improve students' problem-solving skills, mathematical reasoning abilities, and mathematical proof abilities **[19][20]**. Furthermore, the use of Python with Jupyter Lab software and SymPy library is expected to improve problem-solving skills and learning outcomes in differential equation subjects, especially for students in small border islands.

4. CONCLUSIONS

Python with the SymPy library can be used to solve ordinary differential equation (ODE) problems quickly and accurately. Apart from that, the resulting solution is symbolically exact. Using the SymPy library makes it easy for students because its use does not require knowledge of programming languages. Furthermore, using Jupyter Lab makes learning and solving problems easier and more interactive. Therefore, it is important to develop teaching materials for differential equations learning by utilizing the Python programming language with the SymPy library and Jupyter Lab software or other software.

REFERENCES

[1] R. Kurniati and R. A. Ramly, "Development of Macromedia Flash Module in the Learning Media Course Faculty of Teacher Training and Education University of Pejuang Republik Indonesia," *MaPan J. Mat. dan Pembelajaran*, vol. 10, no. 2, pp. 366–384, 2022.

[2] R. Kurniati, S. Sugiarto, and Lestari, "Systematic Literature Review (Slr) of Technology in Mathematics Learning During the Last Decade," *MaPan J. Mat. dan Pembelajaran*, vol. 11, no. 2, pp. 386–402, 2023, doi: 10.24252/10.24252/mapan.2023v11n2a12.

[3] D. Hewitt, "Designing Educational Software: The Case of Grid Algebra," *Digit. Exp. Math. Educ.*, vol. 2, no. 2, pp. 167–198, 2016, doi: 10.1007/s40751-016-0018-4.

[4] N. Panorkou and D. Pratt, "Using Google SketchUp to Develop Students' Experiences of Dimension in Geometry," *Digit. Exp. Math. Educ.*, vol. 2, no. 3, pp. 199–227, 2016, doi: 10.1007/s40751-016-0021-9.

[5] G. Roorda, P. Vos, P. Drijvers, and M. Goedhart, "Solving Rate of Change Tasks with a Graphing Calculator: a Case Study on Instrumental Genesis," *Digit. Exp. Math. Educ.*, vol. 2, no. 3, pp. 228–252, 2016, doi: 10.1007/s40751-016-0022-8.

[6] W. Xu and F. Ouyang, "The application of AI technologies in STEM education: a systematic review from 2011 to 2021," *Int. J. STEM Educ.*, vol. 9, no. 1, 2022, doi: 10.1186/s40594-022-00377-5.

[7] Lestari, S. Sugiarto, and R. Kurniati, "Systematic Literature Review (Slr): Pemanfaatan Software Geogebra Dalam Pembelajaran Matematika," *J. Rev. Pendidik. dan Pengajaran*, vol. 6, no. 4, pp. 3275–3287, 2023, [Online]. Available: http://journal.universitaspahlawan.ac.id/index.php/jrpp/article/view/22627

[8] H. Amin, Zahedi, L. Enos, A. Ansoruddin, W. Wingkolain, and E. Efendi, "C PROGRAM AS A TOOL FOR THE TEACHING OF SECOND ORDER ORDINARY DIFFERENTIAL EQUATION," *BAREKENG J. Math. Its Appl.*, vol. 18, no. 1, pp. 205–212, 2024.

[9] Badan Pusat Statistik Kabupaten Maluku Barat Daya, "Statistik Daerah Kabupaten Maluku Barat Daya," Tiakur, 2023.

[10] S. Sugiarto, *Metode Numerik dengan Scilab dan Python*. Bandung: Alfabeta, 2023.

[11] J. Sundnes, *Solving Ordinary Differential Equations in Python*. Oslo: Simula SpringerBriefs on Computing, 2024.
[12] W. McKinney, *Python for Data Analysis*, Second Edi. Sebastopol: O'Reilly Media Inc, 2018.
[13] P. Bruce, A. Bruce, and P. Gedeck, *Practical Statistics for Data Scientists*, Second Edi. Sebastopol: O'Reilly Media Inc, 2020.
[14] J. Grus, *Data Science from Scratch*, Second Edi. Sebastopol: O'Reilly Media Inc, 2019.
[15] A. A. Aldino and M. Ulfa, "OPTIMIZATION OF LAMPUNG BATIK PRODUCTION USING THE SIMPLEX METHOD," *BAREKENG J. Ilmu Mat. dan Terap.*, vol. 15, no. 2, pp. 297–304, 2021.
[16] S. Sugiarto, *Pemodelan Matematika Penyebaran Penyakit Tuberkulosis*. Nganjuk: Dewa Publishing, 2024.
[17] T. Chao, J. Chen, J. R. Star, and C. Dede, "Using Digital Resources for Motivation and Engagement in Learning Mathematics: Reflections from Teachers and Students," *Digit. Exp. Math. Educ.*, vol. 2, no. 3, pp. 253–277, 2016, doi: 10.1007/s40751-016-0024-6.
[18] M. Teplá, P. Teplý, and P. Šmejkal, "Influence of 3D models and animations on students in natural subjects," *Int. J. STEM Educ.*, vol. 9, no. 1, 2022, doi: 10.1186/s40594-022-00382-8.
[19] A. Sokolowski, Y. Li, and V. Willson, "The effects of using exploratory computerized environments in grades 1 to 8 mathematics: a meta-analysis of research," *Int. J. STEM Educ.*, vol. 2, no. 1, 2015, doi: 10.1186/s40594-015-0022-z.
[20] B. Tamam and D. Dasari, "The use of Geogebra software in teaching mathematics The use of Geogebra software in teaching mathematics," *J. Phys. Conf. Ser.*, vol. 1882, no. 012042, pp. 1–6, 2021, doi: 10.1088/1742-6596/1882/1/012042.