SCHEME : K

Name : _____

Roll No. : _____ Year : 20___ 20___

Exam Seat No. : _____

LABORATORY MANUAL FOR

MATHEMATICS FOR MACHINE LEARNING (314320)

COMPUTER ENGINEERING GROUP

MAHARASHTRA STATE BOARD OF
TECHNICAL EDUCATION, MUMBAI
(Autonomous) (ISO 9001: 2015) (ISO/IEC 27001:2013)

VISION

To ensure that the Diploma level Technical Education constantly matches the latest requirements of Technology and industry and includes the all-round personal development of students including social concerns and to become globally competitive, technology led organization.

MISSION

To provide high quality technical and managerial manpower, information and consultancy services to the industry and community to enable the industry and community to face the challenging technological & environmental challenges.

Quality Policy

We, at MSBTE are committed to offer the best-in-class academic services to the students and institutes to enhance the delight of industry and society. This will be achieved through continual improvement in management practices adopted in the process of curriculum design, development, implementation, evaluation and monitoring system along with adequate faculty development programmes.

Core Values

MSBTE believes in the following:

Skill development in line with industry requirements.

Industry readiness and improved employability of Diploma holders.

Synergistic relationship with industry.

Collective and Cooperative development of all stakeholders.

Technological interventions in societal development.

Access to uniform quality technical education.

A Practical Manual
for

MATHEMATICS FOR MACHINE LEARNING

(314320)

Semester– (IV)

"K-SCHEME"

(AI/AN/DS)

Maharashtra State
Board of Technical Education, Mumbai
(Autonomous) (ISO-9001-2008) (ISO/IEC 27001:2013)

Maharashtra State
Board of Technical Education, Mumbai

(Autonomous) (ISO-9001-2008) (ISO/IEC 27001:2013)

4th Floor, Government Polytechnic Building, 49, Kherwadi,
Bandra (East), Mumbai -400051.

Maharashtra State
Board of Technical Education

Certificate

This is to certify that Mr. / Ms. …….. Roll No……. of Fourth Semester of Diploma in ……of Institute ……
(Code……) has completed the term work satisfactorily in course Mathematics for Machine Learning **(314320)** for the academic year 20……to 20…..... as prescribed in the curriculum.

Place …… Enrollment No……
Date: …..... Exam Seat No. ……..

Course Teacher **Head of the Department** **Principal**

Seal of the Institute

Preface

The objective of all engineering laboratories or field experience in the technical education system is to help students acquire the critical competencies and skills that businesses demand. In light of this, MSBTE developed the cutting-edge "K" Scheme curricula for engineering diploma programs, emphasizing outcome-based learning and the National Education Policy 2020 (NEP2020). As a result, a sizable portion of the program is dedicated to practical work. This demonstrates how crucial laboratory work is in helping teachers, instructors, and students understand that every minute of lab time must be used efficiently to create these outcomes rather than being spent on pointless tasks. Consequently, each practical has been created to operate as a "vehicle" to advance this industry in order to ensure the successful implementation of this outcome-based curriculum. It is challenging to teach practical skills using only the "chalk and duster" activity. Because of this, the "K" scheme laboratory manual creation team focused on the outcomes when designing the practical rather than following the long-standing custom of doing the practical to "verify the theory" (which may turn out to be a by-product along the way).

This lab manual is intended to support all parties involved, particularly the students, instructors, and teachers, in helping the students achieve the pre-established objectives. It is required of every student to read through the relevant practical process in its entirety and comprehend the bare minimum of theoretical background related to the practical at least one day in advance of the practical. As a crucial starting point for carrying out the practical, each exercise in this manual starts with establishing the competency, industry-relevant skills, course outcomes, and practical outcomes. The skills, that students will acquire from the process outlined there, together with the necessary safety measures to be followed, will subsequently be made clear to them. These will enable them to apply the knowledge and abilities to solve real-world problems in their professional lives.

Matrices, Partial Differentiation, Vectors and Tensors, Numerical Differentiation, Integration and Linear Programming are foundation of machine learning. These Math concepts plays a crucial role in understanding how models learn from data and optimize their performance. Concepts like vectors, matrices, and matrix operations are essential for understanding data representations, transformations, and model computations. Vector calculus for multivariable systems is useful in understanding how changes in multi-dimensional spaces affect outcomes. Partial derivatives describes changes of quantities with respect to another. After going through these learning experiences students will be able to implement mathematical concepts using Python Programming which will enhance the knowledge and skills to use the methodology for solving real world problems of various domains using Machine Learning.

The team responsible for developing the Practical manual would like to express its gratitude to MSBTE for taking the lead in developing and implementing the curriculum. Additionally, the team recognizes the valuable contributions made by individual course experts who have been directly or indirectly involved in the development of the "K" scheme curriculum and the laboratory manual. It is impossible to claim perfection in this laboratory manual, even though every effort has been made to verify it for errors, especially because this is the first edition. Any such mistakes and recommendations for enhancements are quite appreciated and can be brought to our attention.

Lab Manual Development Team

Programme Outcomes (POs) to be achieved through Practical of this Course

Following POs are expected to be achieved through the practicals of the Mathematics for Machine Learning course.

PO1 : **Basic and Discipline specific knowledge:** Apply knowledge of basic mathematics, science and engineering fundamentals and engineering specialization to solve the engineering problems.

PO2 : **Problem analysis:** Identify and analyze well-defined engineering problems using codified standard methods.

PO3 : **Design/ development of solutions:** Design solutions for well-defined technical problems and assist with the design of systems components or processes to meet specified needs.

PO4 : **Engineering Tools, Experimentation and Testing:** Apply modern engineering tools and appropriate technique to conduct standard tests and measurements.

PO5 : **Engineering practices for society, sustainability and environment:** Apply appropriate technology in context of society, sustainability, environment and ethical practices.

PO6 : **Project Management:** Use engineering management principles individually, as a team member or a leader to manage projects and effectively communicate about well-defined engineering activities in diverse and multidisciplinary fields.

PO7 : **Life-long learning:** Ability to analyze individual needs and engage in updating in the context of technological changes.

List of Industry Relevant Skills-

The following industry relevant skills of the competency 'Apply Mathematics to solve real-world problems using AI/ML concept and principles to enhance decision-making, design and innovation with precision and efficiency are expected to be developed in you by undertaking the practical of this laboratory manual.

- Develop critical thinking and enhance decision making skills.
- Develop skills related to Data Analysis, Prediction and optimizing designs.
- Develop skills to use machine learning libraries and tools effectively.

Practical- Course Outcome matrix

Course Outcomes (COs)

CO1 - Use partial differentiation concept to obtain optimal solution.
CO2 - Implement matrix concept to solve real life problems.
CO3 - Build programs to implement basic operations based on vectors and tensors.
CO4 - Evaluate numerical differentiation and integration functions.
CO5 - Apply the linear programming problem concept to obtain optimal solution.

S. No.	Laboratory Practical Titles	CO1	CO2	CO3	CO4	CO5
1	Write a program to compute partial derivative.	√	-	-	-	-
2	* Write a program to find maximum and minimum value of the function for two variables.	√	-	-	-	-
3	Write a program to find maximum and minimum value of the function for three variables.	√	-	-	-	-
4	Write a program to find a) Elementary row and column transformations using Python loops. b) Rank of a matrix.	-	√	-	-	-
5	* Write a program to find inverse of a matrix by elementary transformation.	-	√	-	-	-
6	* Write a program to solve system of linear equations.	-	√	-	-	-
7	Write a program to calculate eigen values and eigen vector for given matrix of order 2.	-	√	-	-	-
8	Write a program to calculate eigenvalues and eigen vector for given matrix of order 3.	-	√	-	-	-
9	* Write a program to implement algebra of vectors like addition, subtraction and scalar multiplication.	-	-	√	-	-
10	* Write a program to implement vectors operations like dot product, cross product and scalar triple product.	-	-	√	-	-
11	Write a program to implement basic algebraic operations on tensors like addition, subtraction.	-	-	√	-	-
12	* Write a program to evaluate numerical differentiation for the given data.	-	-	-	√	-
13	Write a program to evaluate numerical integration using Trapezoidal rule for the given data.	-	-	-	√	-
14	* Write a program to evaluate numerical integration using Simpson's one third rule for the given data.	-	-	-	√	-
15	* Write a program to implement simplex method for 2 equations in 2 variables.	-	-	-	-	√

Guidelines to Teachers

1. There will be two sheets of blank pages after every practical for the student to report other matters (if any), which is not mentioned in the printed practical.
2. For difficult practical if required, teacher could provide the demonstration of the practical emphasizing of the skills which the student should achieve.
3. Teachers should give opportunity to students for hands-on after the demonstration.
4. Assess the skill achievement of the students and COs of each unit.
5. One or two questions ought to be added in each practical for different batches. For this teacher can maintain various practical related question banks for each course.
6. For effective implementation and attainment of practical outcomes, teacher ought to ensure that in the beginning itself of each practical, students must read through the complete write-up of that practical sheet.
7. During practical, ensure that each student gets chance and takes active part in taking observations/ readings and performing practical.
8. Teacher ought to assess the performance of students continuously according to the MSBTE guidelines

Instructions for Students

1. For incidental writing on the day of each practical session every student should maintain a ***dated log book*** for the whole semester, apart from this laboratory manual which s/he has to ***submit for assessment to the teacher*** in the next practical session.
2. For effective implementation and attainment of practical outcomes, in the beginning itself of each practical, students need to read through the complete write-up including the practical related questions and assessment scheme of that practical sheet.
3. Student ought to refer the reference books, lab manuals and e-learning material etc.
4. Student should not hesitate to ask any difficulties they face during the conduct of practical.

Content Page

List of Practical and Progressive Assessment Sheet

S. No	Laboratory Practical Titles	Page No.	Date of performance	Date of submission	FA PR marks (25)	Dated sign. of teacher	Remarks (if any)
1	Write a program to compute partial derivative.						
2	* Write a program to find maximum and minimum value of the function for two variables.						
3	Write a program to find maximum and minimum value of the function for three variables.						
4	Write a program to find a) Elementary row and column transformations using Python loops. b) Rank of a matrix.						
5	* Write a program to find inverse of a matrix by elementary transformation.						
6	* Write a program to solve system of linear equations.						
7	Write a program to calculate eigen values and eigen vector for given matrix of order 2.						
8	Write a program to calculate eigenvalues and eigen vector for given matrix of order 3.						
9	* Write a program to implement algebra of vectors like addition, subtraction and scalar multiplication.						
10	* Write a program to implement vectors operations like dot product, cross product and scalar triple product.						
11	Write a program to implement basic algebraic operations on tensors like addition, subtraction.						
12	* Write a program to evaluate numerical differentiation for the given data.						
13	Write a program to evaluate numerical integration using Trapezoidal rule for the given data.						
14	* Write a program to evaluate numerical integration using Simpson's						

S. No	Laboratory Practical Titles	Page No.	Date of performance	Date of submission	FA PR marks (25)	Dated sign. of teacher	Remarks (if any)
	one third rule for the given data.						
15	* Write a program to implement simplex method for 2 equations in 2 variables.						
Total							

Note: To be transferred to Proforma of CIAAN-2023.

Note : Out of above suggestive LLOs -

- **'*' Marked Practical's (LLOs) Are mandatory.**
- **Minimum 80% of above list of lab experiment are to be performed.**
- **Judicial mix of LLOs are to be performed to achieve desired outcomes.**

Practical No.1

Write a program to compute partial derivative.

I. Practical Significance

Calculating the first order, second order and mixed order partial derivative of various types of functions like polynomial, trigonometric, exponential functions of two variables help to compute gradients in optimization algorithms like gradient descent. This helps in minimizing a loss function to train models also it helps to solve many real-world problems.

II. Industry/Employer Expected Outcome (s)

The Practical is expected to develop the following skills for the Industry/Employer 'Evaluate partial fraction of various function and use it in advanced algebra topics involved in research and development'.

III. Course Level Learning Outcome (CO)

CO1: Use Partial Differentiation concept to obtain optimal solutions.

IV. Laboratory Learning Outcome(s)

LLO 1.1 Find partial derivative of first order, second order and mixed order using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Positive Attitude towards problem-solving and applying it optimization in machine learning.
- Maintain tools and equipment.
- Cultivate ethical Practices for Model optimization.
-

VI. Minimum Theoretical Background

Partial Derivatives

Partial derivatives are used in multivariable calculus to measure how a function changes as one of its input variables changes, while keeping the other variables constant. For a function (f(x, y)), the partial derivatives with respect to (x) and (y) are denoted as: $\frac{\partial f}{\partial y \partial x}$

Compute Partial Derivatives in Python

Download the latest version of Python then download Anaconda where spyder is used as code editor.

To compute partial derivatives in Python, you can use the SymPy library, which is a powerful tool for symbolic mathematics.

Steps to be followed:

1. Install SymPy
2. Define the Function and Variables**:**

```
# Define the variables
```

```
# Define the function
```

3. Compute Partial Derivatives**:**

```
# Partial derivative with respect to x
partial_x = sp.diff(f, x)
print(f"∂f/∂x: {partial_x}")
# Partial derivative with respect to y
partial_y = sp.diff(f, y)
print(f"∂f/∂y: {partial_y}")
Full Code with Example:
```

This shows the partial derivatives of the function (f(x, y)) with respect to (x) and (y).

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to find the partial derivative of Polynomial function.
2. Write a program to find the partial derivative of Trigonometric function
3. Write a program to find the partial derivative of logarithmic function.
4. Write a program to find the partial derivative of exponential function.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a python program to find the partial derivative of homogeneous function.
2. Write a python program to verify the Euler's theorem for homogeneous function.
3. Write a python program to find the order of function
4. Write a python program to find the partial derivative without using Sympy Library.

(Space for answers)

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/how-to-calculate-and-plot-the-derivative-of-a-function-using-matplotlib-and-python/
2. https://www.adventuresinmachinelearning.com/mastering-derivatives-solving-calculus-problems-with-pythons-sympy/
3. https://www.youtube.com/watch?v=YFS8EHKR2Fw
4. https://www.geeksforgeeks.org/partial-derivatives-practice-problems/
5. https://www.youtube.com/watch?v=aojkrNYfHrY
6. https://www.tutorialspoint.com/scipy/index.htm

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical no. 2

Write a program to find the maximum and minimum value of the function for two variables.

I. Practical Significance

Calculating the maximum and minimum value of a function helps in finding the optimal values for x and y that minimize or maximize the function. The same principles apply when dealing with real-world data like maximizing profit, minimizing cost, or finding the best configuration for a system.

II. Industry/Employer Expected Outcome (s)

The Practical is expected to develop the following skills for the Industry/Employer 'Evaluate partial fraction of various function and use it in advanced algebra topics involved in research and development'.

III. Course level Learning Outcome (CO)

CO1 - Use partial differentiation concept to obtain optimal solution.

IV. Laboratory Learning Outcome (s)

LLO 2.1 Find maximum and minimum value of the function for two variables using Python programming.

V. Relative Affective Domain related Outcome (s)-

- Built Positive Attitude towards problem-solving and applying it optimization in machine learning.
- Maintain tools and equipment.
- Cultivate ethical Practices for Model optimization.

VI. Minimum Theoretical Background:

Algorithm to Find Maxima and Minima

1. Find the values of x and y using **fx = 0** and **fy = 0** [**NOTE**: fx and fy are the partial double derivatives of the function with respect to x and y respectively.]
2. The obtained result will be considered as **stationary/turning points** for the curve.
3. Create 3 new variables r,t, and s.
4. Find the values of r,t and s using **r=fx, t=fy, s=fx**
5. **If** $(rt - s^2)$**|(stationary pts.)>0 (Maxima/Minima) exists**
6. **If** $(rt - s^2)$**|(stationary pts.)**<0 (No Maxima/Minima)/(Saddle point)

7. **If** $(rt - s^2)$**|(stationary pts.)** = 0, then the case is doubtful and needs further investigation.
8. If **r=fx >0 (Minima)**
9. If **r=fx <0 (Maxima)**

Steps:

1. import numpy as np
   from scipy.optimize import minimize

2. Define the function to minimize
3. Initial guess for the variables
   initial_guess = [0, 0]
4. Use scipy's minimize function to find the minimum
   result_min = minimize(func, initial_guess)
5. Display the results for minimum
   if result_min.success:
      min_point = result_min.x
      min_value = result_min.fun
      print(f"Minimum value occurs at x = {min_point[0]}, y = {min_point[1]}")
      print(f"Minimum value of the function is {min_value}")
   else:
      print("Optimization failed to find the minimum.")
6. To find the maximum, negate the function and minimize
   def func_max(X):
      return -func(X)
7. Use scipy's minimize function to find the maximum
   result_max = minimize(func_max, initial_guess)
8. Display the results for maximum
   if result_max.success:
      max_point = result_max.x
      max_value = -result_max.fun # Negate the result to get the original maximum value
      print(f"Maximum value occurs at x = {max_point[0]}, y = {max_point[1]}")

```
    print(f"Maximum value of the function is {max_value}")
else:
    print("Optimization failed to find the maximum.")
```

VII. Resources Required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Computer System	Any Desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating System	Windows/Linux/Mac/Ubuntu	One computer system for each student	
3	Software	Python, Anaconda Navigator with updated version of Spyder, etc.	One computer system for each student	

VIII. Exercise

1. Write the definition of Maxima Minima and Saddle point.
2. Write program to define simple functions and find the partial derivatives of defined functions.
3. Write a program to find second-order derivatives after finding the second-order derivatives, write a function that accepts any two-variable function and returns its second-order partial derivatives along with an analysis of the stationary points and classification as maxima, minima, or saddle points.

(Space for answers)

……
……
……
……
……
……
……
……
……
……
……
……

IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to find the maxima and minima of given function.
2. Write a python program to find the critical points and saddle points of function.

(Space for answers)

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/how-to-find-extrema-of-multivariable-functions/
2. https://srivastavayushmaan1347.medium.com/unveiling-the-peaks-and-valleys-exploring-maxima-and-minima-in-calculus-with-python-b0340c9cf76d
3. https://btechfirstyearnotes.blogspot.com/2020/02/maxima-and-minima.html

XI. Assessment Scheme

XII. Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Process Related (15)**	

Practical No.3

Write Program to find maximum and minimum value of the function for three variables.

I. Practical Significance

In Machine Learning, the optimization of functions with multiple variables plays a crucial role in tasks such as training models, hyperparameter tuning, and improving model performance.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Get better-performing, scalable, and efficient models that help drive business objectives, reduce risks, and provide valuable insights.'

III. Course Level Learning Outcome (CO)

CO1: Use Partial Differentiation concept to obtain optimal solutions.

IV. Laboratory Learning Outcome(s)

LLO 3.1 Find maximum and minimum value of the function for three variables using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Positive Attitude towards problem-solving and applying it optimization in machine learning.
- Maintain tools and equipment.
- Cultivate ethical Practices for Model optimization.

VI. Minimum Theoretical Background

Partial differentiation is commonly used to find the maximum or minimum of a function by analyzing its gradient. The gradient consists of the partial derivatives of the function with respect to each variable. This process is typically done using optimization algorithms like Gradient Descent. To find the maximum and minimum values of a function for three variables using partial differentiation, we can use the following steps:

1. **Define the function**: We define a function f(x,y,z).
2. **Compute partial derivatives**: Compute the partial derivatives with respect to each variable (x, y, and z).
3. **Set the gradient to zero**: To find the critical points (where the gradient is zero), we solve for the values of x, y, and z.
4. **Check for maxima or minima**: Use the second derivative test or other methods to check whether the critical points correspond to maxima or minima.

Example:

Let's consider the following example function:

f(x,y,z)=−(x2+y2+z2)+10f(x, y, z) = -(x^2 + y^2 + z^2) + 10f(x,y,z)=−(x2+y2+z2)+10

This function has a maximum at the origin (0, 0, 0), and we will find both the maximum and minimum using partial differentiation.

```
import sympy as sp

# Define the symbols (variables)
x, y, z = sp.symbols('x y z')

# Define the function f(x, y, z)
f = x**2 + y**2 + z**2 - 6*x - 4*y + 3*z

# Calculate the partial derivatives with respect to x, y, and z
f_x = sp.diff(f, x)
f_y = sp.diff(f, y)
f_z = sp.diff(f, z)

# Display the partial derivatives
print("Partial derivative with respect to x:", f_x)
print("Partial derivative with respect to y:", f_y)
print("Partial derivative with respect to z:", f_z)

# Set the partial derivatives equal to zero to find the critical points
solutions = sp.solve([f_x, f_y, f_z], (x, y, z))

# Display the critical points
print("\nCritical points:", solutions)

# Evaluate the second partial derivatives to classify the critical points
f_xx = sp.diff(f_x, x)
f_yy = sp.diff(f_y, y)
f_zz = sp.diff(f_z, z)
f_xy = sp.diff(f_x, y)
f_xz = sp.diff(f_x, z)
f_yz = sp.diff(f_y, z)

# Calculate the determinant of the Hessian matrix at the critical points
Hessian = f_xx * f_yy * f_zz + 2 * f_xy * f_xz * f_yz - f_xx * f_yz**2 - f_yy * f_xz**2 - f_zz * f_xy**2
```

```
# Display the second partial derivatives
print("\nSecond partial derivatives:")
print("f_xx:", f_xx)
print("f_yy:", f_yy)
print("f_zz:", f_zz)
print("f_xy:", f_xy)
print("f_xz:", f_xz)
print("f_yz:", f_yz)

# Evaluate the Hessian determinant at the critical points
for point in solutions:
    hessian_val = Hessian.subs({x: point[0], y: point[1], z: point[2]})
    print(f"\nHessian determinant at point {point}: {hessian_val}")

    # If the Hessian determinant is positive, it indicates a local minimum or maximum.
    if hessian_val > 0:
        print(f"At point {point}, it's a local minimum or maximum.")
    elif hessian_val < 0:
        print(f"At point {point}, it's a saddle point.")
    else:
        print(f"At point {point}, the test is inconclusive.")
```

- **Define the function**: The function f(x,y,z)f(x, y, z)f(x,y,z) is defined symbolically using `sympy`.
- **Partial derivatives**: The partial derivatives with respect to xxx, yyy, and zzz are computed using `sp.diff()`.
- **Solve for critical points**: We solve the system of equations where the gradient (the vector of partial derivatives) is equal to zero using `sp.solve()`.
- **Second derivative test (Hessian matrix)**: The second-order partial derivatives are computed and arranged into a Hessian matrix. The determinant of the Hessian matrix is used to classify the critical points as maxima, minima, or saddle points.
    - If the determinant is positive and the diagonal elements are positive, the point is a local minimum.
    - If the determinant is negative, the point is a local maximum.
    - If the determinant is zero, the point is indeterminate (saddle point).

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write Program to find maximum and minimum value of the function for three variables.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a Python program to find the critical points of the given function.
2. Write a Python program to classify the critical points of the given function.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/python-program-maximum-three/
2. https://www.youtube.com/watch?v=h5rWayDF9rU
3. https://www.youtube.com/watch?app=desktop&v=2mtEE4rN4Wk&t=663s
4. https://realpython.com/python-min-and-max/

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total(25)**	

Practical no. 4

Write a program to find

a) Elementary row and column transformations using Python loops.
b) Rank of a matrix.

I. Practical Significance

Matrix principles are fundamental to AI/ML engineering. Implementing these concepts in Python programming will enhance skills and critical thinking to make use of it to solve real world engineering-based challenges.

II. Industry/Employer Expected Outcome

The Practical is expected to develop the following skills for the Industry/Employer 'Implement the concept of reduced row echelon form using python programming to calculate the rank of a matrix.'

III. Course level Learning Outcome (CO)

CO2 - Implement matrix concept to solve real life problems.

IV. Laboratory Learning Outcome

LLO 4.1 Calculate the rank of a matrix by elementary transformation using Python Programming.

V. Relative Affective Domain related Outcome (s)-

- Follow safety practices.
- Cultivate a positive and proactive attitude towards Matrix Mathematics.
- Follow ethical practices.

VI. Minimum Theoretical Background

Elementary Row Transformations

a) **Row Swapping**: Interchanging two rows.
b) **Row Multiplication**: Multiplying a row by a non-zero scalar.
c) **Row Addition**: Adding a multiple of one row to another row.

Elementary Column Transformations

a) **Column Swapping:** Interchanging two columns.
b) **Column Multiplication:** Multiplying a column by a non-zero scalar.
c) **Column Addition**: Adding a multiple of one column to another column.

Example Python code:

```
import numpy as np

def swap_rows(matrix, row1, row2):
    """Swap two rows of a matrix."""
    matrix[row1], matrix[row2] = matrix[row2], matrix[row1]

def multiply_row(matrix, row, scalar):
"""Multiply a row by a scalar."""
    matrix[row] = matrix[row] * scalar


def add_rows(matrix, row1, row2, scalar):
    """Add a multiple of one row to another row."""
    matrix[row1] = matrix[row1] + scalar * matrix[row2]

def rref(matrix):
    """Convert the matrix to reduced row echelon form (RREF)."""
    rows, cols = matrix.shape
    row = 0  # Start with the first row

    for col in range(cols):
        if row >= rows:
            break

        # Find the pivot row and swap it with the current row
        if matrix[row, col] == 0:
            for r in range(row + 1, rows):
                if matrix[r, col] != 0:
                    swap_rows(matrix, row, r)
                    break

        # If the pivot is still 0, move to the next column
        if matrix[row, col] == 0:
            continue

        # Normalize the pivot row (make pivot equal to 1)
        multiply_row(matrix, row, 1 / matrix[row, col])
```

```
        # Make all other entries in the column equal to 0
        for r in range(rows):
            if r != row:
                add_rows(matrix, r, row, -matrix[r, col])

        row += 1  # Move to the next row

    return matrix

# Example matrix
matrix = np.array([[1, 2, -1, 3],
                   [2, 4, 0, 2],
                   [1, 3, 1, 4]], dtype=float)

print("Original Matrix:")
print(matrix)

# Finding the RREF
rref_matrix = rref(matrix.copy())  # Copying the matrix to keep the original intact

print("\nReduced Row Echelon Form (RREF):")
print(rref_matrix)
```

Sample Output:

```
Original Matrix:
[[ 1.  2. -1.  3.]
 [ 2.  4.  0.  2.]
 [ 1.  3.  1.  4.]]

Reduced Row Echelon Form (RREF):
[[ 1.  0. -5.  1.]
 [ 0.  1.  2.  1.]
 [ 0.  0.  0.  0.]]
```

VII. Resources Required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Computer System	Any Desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating System	Windows/Linux/Mac/Ubuntu	One computer system for each student	
3	Software	Python, Anaconda Navigator with updated version of Spyder, etc.	One computer system for each student	

VIII. Exercise

1. Write a program to find the addition of two matrix.
2. Write a program to find the subtraction of two matrix
3. Write a program to find the scalar multiplication of two matrix.
4. Write a algorithm to find reduced row echelon form of the given matrix manually using Python loops.
5. Write a definition of rank of the matrix using normal form and triangular form.

(Space for answers)

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to find the order of given matrix
2. Write a python program to find the transpose of the given matrix.
3. Write a python program to find row echelon form of the given matrix using sympy.
4. Write a python program to find rank of the given matrix using sympy.

(Space for answers)

X. References/Suggestions for further Reading

1. Math for Data Science — Lecture 02 Elementary Row Operations, RREF + Rank via Python | by Aakash Goel | Math for data science | Medium
2. Elementary Operations on Matrices - GeeksforGeeks
3. Loops in Python with Examples - Python Geeks
4. Program for Rank of Matrix - GeeksforGeeks
5. Higher Engineering Mathematics by H. K. Dass, Er. Rajnish Verma.

XI. Assessment Scheme

Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Total (25)**	

Practical no. 5

Write a program to find inverse of a matrix by elementary transformation.

I. Practical Significance

Matrix principles are fundamental to AI/ML engineering. Implementing these concepts in Python programming will enhance skills and critical thinking to make use of it to solve real world engineering-based challenges.

II. Industry/Employer Expected Outcome

The Practical is expected to develop the following skills for the Industry/Employer 'Implement the concept Inverse of the matrix using python programming to find the solution of system of linear equations.

III. Course level Learning Outcome (CO)

CO2 - Implement matrix concept to solve real life problems.

IV. Laboratory Learning Outcome

LLO 5.1 Calculate the inverse of a matrix by elementary transformation using Python programming.

V. Relative Affective Domain related Outcome

- Follow safety practices.
- Cultivate a positive and proactive attitude towards Matrix Mathematics.
- Follow ethical practices.

VI. Minimum Theoretical Background

Inverse of a Matrix by Elementary Operation:

The Gaussian Elimination method is also known as the row reduction method and it is an algorithm that is used to solve a system of linear equations. It is usually understood as a sequence of operations performed on the corresponding matrix of coefficients.

This algorithm is used to find:

- The rank of a matrix.
- The determinant of a matrix.
- The inverse of a matrix.

Matrix Operations:

- Interchanging/swapping two rows.
- Multiplying or dividing a row by a positive integer.
- Adding or subtracting a multiple of one row to another.

Using these operations, we can modify a matrix and find its inverse.

The steps involved are:

- **Step 1:** Create an identity matrix of n x n.
- **Step 2:** Perform row or column operations on the original matrix (A) to make it equivalent to the identity matrix.
- **Step 3:** Perform similar operations on the identity matrix too.

- import numpy as np
- Swap rows i and j in matrix A
- Multiply row i by a scalar in matrix A
- Add scalar * row j to row i in matrix A
- Find the inverse of matrix A using elementary row transformations

```
def inverse_matrix(A):
    """Find the inverse of matrix A using elementary row transformations."""
    n = len(A)
    # Create an augmented matrix [A | I]
    I = np.identity(n)
    augmented_matrix = np.hstack((A, I))

    for i in range(n):
        # Make the pivot element A[i, i] to be 1
        if augmented_matrix[i, i] == 0:
            for j in range(i + 1, n):
                if augmented_matrix[j, i] != 0:
                    row_swap(augmented_matrix, i, j)
                    break

        pivot = augmented_matrix[i, i]
        row_multiply(augmented_matrix, i, 1 / pivot)
```

- Make all elements below and above the pivot element to be 0

```
for j in range(n):
        if j != i:
                factor = augmented_matrix[j, i]
                row_add_multiple(augmented_matrix, j, i, -factor)
```

- The right half of the augmented matrix is now the inverse of A

```
return augmented_matrix[:, n:]
```

VII. Resources Required

Sr. No.	Name of Resource	Specification	Quantity	Remarks
1	Computer System	Any Desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating System	Windows/Linux/Mac/Ubuntu	One computer system for each student	
3	Software	Python, Anaconda Navigator with updated version of Spyder, etc.	One computer system for each student	

VIII. Exercise

1. Write at least 4 methods to calculate inverse of the matrix.
2. Write program for all methods to find the inverse of matrix written in above question.
3. Write a program to find the inverse of a 2x2 matrix using the adjoint method.

(Space for answers)

……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……
……

IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to create matrix.
2. Write a python program to find the determinant of given matrix.
3. Write a python program to perform the matrix multiplication.

(Space for answers)

X. References/Suggestions for further Reading

1. Inverse of a Matrix by Elementary Operations - Matrices | Class 12 Maths - GeeksforGeeks
2. Elementary Operations on Matrices - GeeksforGeeks
3. How to inverse a matrix using NumPy - GeeksforGeeks
4. How to invert a matrix or nArray in Python?
5. Invertible matrix - Wikipedia
6. NumPy Matrix Operations (With Examples)
7. Higher Engineering Mathematics by H. K. Dass, Er. Rajnish Verma.

XI. Assessment Scheme

Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Total (25)**	

Practical no. 6

Write a program to solve system of linear equations.

I. Practical Significance

Solving systems of equations with Python is invaluable for simulations, predictions, and optimization problems, streamlining complex calculations and improve decision-making.

II. Industry/Employer Expected Outcome (s)

The Practical is expected to develop the following skills for the Industry/Employer 'Implement the concept Inverse of the matrix using python programming to find the solution of system of linear equations.

III. Course level Learning Outcome (CO)

CO2 - Implement matrix concept to solve real life problems.

IV. Laboratory Learning Outcome (s)

LLO 5.1 Calculate the inverse of a matrix by elementary transformation using Python programming.

V. Relative Affective Domain related Outcome

- Follow safety practices.
- Cultivate a positive and proactive attitude towards Matrix Mathematics.
- Follow ethical practices.
-

VI. Minimum Theoretical Background

A system of linear equations involves multiple linear equations that share common variables. Each equation can represent a line, plane, or higher-dimensional surface, depending on the number of variables involved. The solution to the system is the set of variable values that satisfy all equations simultaneously, usually corresponding to points of intersection among the lines or planes represented by these equations.

```
import numpy as np

def solve_system_of_equations():
    # Coefficient matrix (A) where each row represents an equation
    A = np.array([[2, 3],
                  [4, -1]])

    # Right-hand side (b), representing the constants in the equations
```

```
b = np.array([5, 3])

# Solving the system of linear equations using np.linalg.solve
try:
    solution = np.linalg.solve(A, b)
    print("Solution of the system:")
    print(f"x = {solution[0]}")
    print(f"y = {solution[1]}")
except np.linalg.LinAlgError as e:
    print("Error:", e)

# Run the function to solve the system
  solve_system_of_equations()
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

1. Write at least 4 methods to calculate inverse of the matrix.
2. Write program for all methods to find the inverse of matrix written in above question.
3. Write a program to solve a system of linear equations using Cramer's rule in 2 and 3 variables.
4. Write a program to solve a system of linear equations using Inverse Method in 2 and 3 variables.
5. Write a program to solve a system of linear equations using Gauss Jordan Method in 2 and 3 variables.

(Space for answers)

……

……

……

……

IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to create matrix.
2. Write a python program to find the determinant of given matrix.
3. Write a python program to perform the matrix multiplication.

(Space for answers)

X. References/Suggestions for further Reading

1. Higher Engineering Mathematics by H. K. Dass, Er. Rajnish Verma.
2. Gauss Elimination Method Python Program (With Output)
3. Python Program for Gauss-Seidel Iteration Method with Output
4. How to Solve a System of Equations in Python (3 Examples)
5. https://www.codesansar.com/numerical-methods/python-program-inverse-matrix-using-gauss-jordan.htm

XI. Assessment Scheme

Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Total (25)**	

Practical no. 7

Write a program to calculate Eigen values and Eigen vector for given matrix of order 2.

I. Practical Significance

Finding the Eigen values and Eigen vectors through the python programming will enhance the student's capacity to actual implementation of mathematical concepts in machine learning models.

II. Industry/Employer Expected Outcome

The Practical is expected to develop the following skills for the Industry/Employer 'Implement the concept of Eigen values and Eigen vectors of the matrix using python programming to find the engineering related solutions.'

III. Course level Learning Outcome (CO)

CO2 - Implement matrix concept to solve real life problems.

IV. Laboratory Learning Outcome (s)

LLO 7.1 Calculate eigen-values and eigenvectors for the given matrix of order 2 using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Positive Attitude towards problem-solving and applying it in machine learning optimization.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Definition: Given a square n×n matrix A, we say that a nonzero vector v is an eigenvector of A if there is a scalar λ such that

$$Av = \lambda v$$

The scalar λ is called the eigenvalue associated to the eigenvector v.

Step-by-Step Algorithm to find the eigen value and Eigen vectors for order 2 matrix

1. Given a 2x2 matrix

2. Find the characteristic equation

The characteristic equation is obtained by solving the determinant of (A - λI), where (λ) is an eigenvalue and (I) is the identity matrix.
$\det(A-\lambda I)=0$

3. Solve the characteristic equation

4. Find the eigenvectors

Follow the below steps for program-

import numpy as np

Create a matrix 'm' using np.mat() function

Display the original matrix 'm'

Compute the eigenvalues and eigenvectors of the matrix 'm' using np.linalg.eig() function

Display the eigenvalues and eigenvectors of the said matrix

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

1. Write the definition of characteristics polynomial of matrix and program to find characteristics polynomial of the given matrix.
2. Write important properties of Eigen values.

(Space for answers)

……

……

……

……

IX. IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to find the determinant of given matrix of order 2.
2. Write a python program to to perform the matrix multiplication of order 2.
3. Write a python program to program to find characteristics polynomial of the given matrix of order 2.

(Space for answers)

X. References/Suggestions for further Reading

1. Higher Engineering Mathematics by H. K. Dass, Er. Rajnish Verma.
2. https://www.miloriano.com/eigenvalues-and-eigenvectors-in-python-algorithm/
3. https://blog.finxter.com/5-best-ways-to-calculate-eigenvalues-and-eigenvectors-with-scipy-in-python/
4. https://www.geeksforgeeks.org/how-to-compute-the-eigenvalues-and-right-eigenvectors-of-a-given-square-array-using-numpy/
5. https://www.adventuresinmachinelearning.com/mastering-eigenvalues-and-eigenvectors-with-numpy-linalgeig/

XI. Assessment Scheme

Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Total (25)**	

Practical no. 8

Write a program to calculate Eigen values and Eigen vector for given matrix of order 3.

I. Practical Significance

Finding the Eigen values and Eigen vectors through the python programming will enhance the student's capacity to actual implementation of mathematical concepts in machine learning models.

II. Industry/Employer Expected Outcome

The Practical is expected to develop the following skills for the Industry/Employer 'Implement the concept of Eigen values and Eigen vectors of the matrix using python programming to find the engineering related solutions.'

III. Course level Learning Outcome (CO)

CO2 - Implement matrix concept to solve real life problems.

IV. Laboratory Learning Outcome (s)

LLO 8.1 Calculate eigen-values and eigen-vectors for the given matrix of order 3 using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Positive Attitude towards problem-solving and applying it in machine learning optimization.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Definition: Given a square $n \times n$ matrix A, we say that a nonzero vector v is an eigenvector of A if there is a scalar λ such that

$$\mathbf{Av = \lambda v}$$

The scalar λ is called the eigenvalue associated to the eigenvector v.

Step-by-Step Algorithm to find the eigen value and Eigen vectors for order 3 matrix

1. Given a 3x3 matrix

2. Find the characteristic equation

The characteristic equation is obtained by solving the determinant of (A - λI), where (λ) is an eigenvalue and (I) is the identity matrix.

$\det(A-\lambda I)=0$

3. Solve the characteristic equation

4. Find the eigenvectors

Follow below steps for program-

import numpy as np

define Coefficient matrix (A) for the 3 equations

Solve the system of linear equations using np.linalg.solve

Run the function to solve the 3x3 system

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

1. Write the relation between Eigen values and Inverse of a Matrix.
2. Write some practical applications of Eigen values and Eigen vectors in the various fields of engineering.

(Space for answers)

……

……

……

……

IX. Practical related Questions

Note: Below given are the few sample questions for reference. Teacher must design more such questions to ensure the achievement of identified CO.

1. Write a python program to find the determinant of given matrix of order 3
2. Write a python program to perform the matrix multiplication of order 3.
3. Write a python program to program to find characteristics polynomial of the given matrix of order 3.

(Space for answers)

X. References/Suggestions for further Reading

1. Higher Engineering Mathematics by H. K. Dass, Er. Rajnish Verma.
2. https://www.miloriano.com/eigenvalues-and-eigenvectors-in-python-algorithm/
3. https://blog.finxter.com/5-best-ways-to-calculate-eigenvalues-and-eigenvectors-with-scipy-in-python/
4. https://www.geeksforgeeks.org/how-to-compute-the-eigenvalues-and-right-eigenvectors-of-a-given-square-array-using-numpy/
5. https://www.adventuresinmachinelearning.com/mastering-eigenvalues-and-eigenvectors-with-numpy-linalgeig/

XI. Assessment Scheme

Performance Indicators		Weightage
Project related (15 Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10 Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (25 Marks)		**100%**

Marks Obtained			Dated Signature of Teacher
Process Related (15)	**Product related (10)**	**Total (25)**	

Practical No.9

Write a program to implement algebra of vectors like addition, subtraction and scalar multiplication.

I. Practical Significance

In machine learning, vectors are used to represent data points and features. Efficient vector operations are essential for tasks like calculating gradients (in optimization), manipulating datasets, or transforming data in high-dimensional spaces.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Demonstrate strong mathematical and programming skills, optimization abilities, familiarity with data manipulation, and problem-solving.'

III. Course Level Learning Outcome (CO)

CO3: Build programs to implement basic operations based on vectors and tensors.

IV. Laboratory Learning Outcome(s)

LLO 9.1: Implement algebra of vectors using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Positive Attitude towards problem-solving and applying it in machine learning optimization.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

A vector is an ordered list of numbers that can represent anything from physical quantities (like displacement or velocity) to features in machine learning.

Basic vector operations:

- Addition: Add corresponding elements of two vectors.
- Subtraction: Subtract corresponding elements of one vector from another.
- Scalar Multiplication: Multiply each element of a vector by a scalar (a single number).

Program Steps:

1. **Create Functions for Each Operation**: We will create three functions—one for each operation. These will be implemented using simple loops (manual method) and also with NumPy (for efficiency).
2. **Error Handling**: Ensure that both vectors are of the same length for addition and subtraction operations.
3. **Input**: Prompt the user for vectors and the scalar.
4. **Process Operations**: Perform the vector operations inside the functions and return the results.

Program Implementation:

Create Functions for Each Operation

```
# Function to add two vectors
def vector_addition(v1, v2):
    if len(v1) != len(v2):
        raise ValueError("Vectors must be of the same length for addition")
    return [v1[i] + v2[i] for i in range(len(v1))]

# Function to subtract two vectors
def vector_subtraction(v1, v2):
    if len(v1) != len(v2):
        raise ValueError("Vectors must be of the same length for subtraction")
    return [v1[i] - v2[i] for i in range(len(v1))]

# Function to multiply a vector by a scalar
def scalar_multiplication(v, scalar):
    return [scalar * v[i] for i in range(len(v))]
```

Set up the main part of the program where you interact with the user.

Note: For larger vectors or matrices, you can use **NumPy** to perform these operations more efficiently. Install NumPy using `pip install numpy` and modify the program.

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to implement algebra of vectors like addition, subtraction and scalar multiplication.

2. Write a python program to Calculate the magnitude of given vector and unit vector perpendicular to given vector:
   a. Create function to calculate the magnitude of a vector.
   b. Create function to calculate the unit vector.
   c. Create function to calculate a perpendicular vector in 2D.

(Space for answers)

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a python program to find angle between two vectors using scalar (dot) product.
2. Write a python program to calculate direction ratio and direction cosines of a vector.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/program-dot-product-cross-product-two-vector/
2. https://www.youtube.com/watch?v=fqxWBj0cmDY
3. https://www.youtube.com/watch?v=ZKWGYAYpd4c
4. https://learn.codesignal.com/preview/lessons/4605

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.10

Write a program to implement vectors operations like dot product, cross product and scalar triple product.

I. Practical Significance

Implementing vector operations using Python allows users to work with vectors in various real-world applications like engineering, physics, computer graphics, robotics, and machine learning, offering a straightforward way to handle vector-based calculations.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Provide accuracy, efficiency, scalability, and applicability while solving real-world problems.'

III. Course Level Learning Outcome (CO)

CO3: Build programs to implement basic operations based on vectors and tensors.

IV. Laboratory Learning Outcome(s)

LLO 10.1 Implement vectors operations using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built responsibility sense for Data Representation and Transformation.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Vectors:

- A vector is an ordered list of numbers, often represented as a column or row in an array. In ML, vectors represent data points, features, or directions in space.
- Each element of a vector corresponds to a feature or a coordinate in a vector space.

Dot Product (Scalar Product):

- The dot product between two vectors a=[a1,a2,...,an] and b=[b1,b2,...,bn] is calculated as:

$$a \cdot b = a_1b_1 + a_2b_2 + \cdots + a_nb_n$$

Cross Product (Vector Product):

- The cross product is only defined in **three-dimensional space (3D)** and produces a vector perpendicular to both input vectors. For vectors a=[a1,a2,a3] and b=[b1,b2,b3], the cross product is:

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}$$

Step 1: Importing the NumPy Library

Step 2: Implementing Dot Product

```
def dot_product(a, b):
    return np.dot(a, b)
```

Step 3: Implementing Cross Product

```
def cross_product(a, b):
    return np.cross(a, b)
```

Step 4: Implementing Scalar Triple Product

```
def scalar_triple_product(a, b, c):
    return np.dot(a, np.cross(b, c))
```

Step 5: Test the Implementations with Sample Data:

```
# Sample vectors
a = np.array([1, 2, 3])
b = np.array([4, 5, 6])
c = np.array([7, 8, 9])

# Test Dot Product
dot_result = dot_product(a, b)
print(f"Dot Product of a and b: {dot_result}")

# Test Cross Product
cross_result = cross_product(a, b)
print(f"Cross Product of a and b: {cross_result}")

# Test Scalar Triple Product
scalar_triple_result = scalar_triple_product(a, b, c)
print(f"Scalar Triple Product of a, b, and c: {scalar_triple_result}")
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

3. Write a program to implement Cosine Similarity Using Dot Product.
4. Write a program to implement the Cross Product in 3D Vector Geometry to find the normal vector to the surface of a triangle defined by three 3D points.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

3. Write a python program to implement **Scalar Triple Product for Volume.**
(Given three 3D vectors, compute the volume of the parallelepiped formed by the vectors using the scalar triple product. Write a Python function to implement this and apply it to a set of vectors representing a 3D shape.)

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.3blue1brown.com/topics/linear-algebra
2. https://numpy.org/doc/
3. https://www.youtube.com/watch?v=jH62KTvmvt0
4. https://www.geeksforgeeks.org/program-dot-product-cross-product-two-vector/

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.11

Write a program to implement basic algebraic operations on tensors like addition, subtraction.

I. Practical Significance

Tensors are used in a wide range of applications, including image processing, natural language processing, physics simulations, and data analysis. Implementing basic tensor operations helps build a deeper understanding of tensor mathematics and is a crucial skill and gives a practical experience that prepares users for working with advanced libraries and creating optimized algorithms.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Write optimized code for tensor operations to solve real-world problems.'

III. Course Level Learning Outcome (CO)

CO3: Build programs to implement basic operations based on vectors and tensors.

IV. Laboratory Learning Outcome(s)

LLO 11.1 Implement basic algebraic operations on tensors using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Analytical thinking and know Ethical considerations when working with data and technology.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

A tensor is a multi-dimensional generalization of scalars, vectors, and matrices. It can be thought of as an array with more than two dimensions:

Scalar: 0-dimensional tensor.
Vector: 1-dimensional tensor.
Matrix: 2-dimensional tensor.
Higher-dimensional tensors: More than 2 dimensions.

Step 1: Import the NumPy Library

Step 2: Define Two Tensors

Define two tensors (multi-dimensional arrays) to perform operations on it. Here define 2D arrays (matrices), but the operations are applicable to tensors of any dimension.

Define two 2D tensors (matrices)

```
tensor1 = np.array([[1, 2, 3],
                    [4, 5, 6],
                    [7, 8, 9]])
tensor2 = np.array([[9, 8, 7],
                    [6, 5, 4],
                    [3, 2, 1]])
```

Step 4: Implement Tensor Addition

Perform element-wise addition of the two tensors using the np.add( ) function:

```
# Add the two tensors element-wise
tensor_sum = np.add(tensor1, tensor2)
print("Result of Tensor Addition:")
print(tensor_sum)
```

Step 5: Implement Tensor Subtraction

Use np.subtract( ) function to subtract one tensor from the other element-wise.

```
# Subtract tensor2 from tensor1 element-wise
tensor_diff = np.subtract(tensor1, tensor2)
print("Result of Tensor Subtraction:")
print(tensor_diff)
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to implement basic algebraic operations on tensors like addition, subtraction.
2. Write a program to implement perform scalar multiplication on a tensor.

(Space for answers)

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a program to implement element-wise multiplication and division on tensors using np.multiply( ) and np.divide( ).
2. Write a python program to handle tensors of different shapes.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/python-operations-on-numpy-arrays/
2. https://www.analyticsvidhya.com/blog/2021/05/mathematical-operations-in-python-with-numpy/
3. https://www.youtube.com/watch?v=NjIgKQWn3Rk

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.12

Write a program to evaluate numerical differentiation for the given data.

I. Practical Significance

Numerical differentiation is widely used in scenarios where you need to find the rate of change or slope of a function but don't have an explicit mathematical formula for it. Like in Physics and Engineering and also in Economics to compute the rate of change of various economic factors like inflation, stock price changes, etc.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Handle various data sets and differentiate functions even when the underlying model is not explicitly known.'

III. Course Level Learning Outcome (CO)

CO4 - Evaluate numerical differentiation and integration functions.

IV. Laboratory Learning Outcome(s)

LLO 12.1 Find numerical differentiation for the given data using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Analytical thinking and know Ethical considerations when working with data and technology.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Numerical differentiation is the process of approximating the derivative of a function using discrete data points. Here we will implement two common methods for numerical differentiation:

1. Forward Difference Method
2. Backward Difference Method

Step 1: Import required Libraries:

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

Step 2: Prepare the Data:

- Define a set of x-values and the corresponding f(x)-values. These can be from an experimental dataset or defined functions.
- Ensure the data points are equally spaced, as the methods rely on this assumption.

```
# Example data (x, f(x)) values
 x_values = np.array([0, 1, 2, 3, 4, 5]) # x values
```

```
f_values = np.array([0, 1, 4, 9, 16, 25]) # f(x) = x^2 values

# Step size (h)
h = x_values[1] - x_values[0]
```

Step 2: Define the Differentiation Methods:

- Implement the forward, backward, and central difference methods in Python.

```
# Forward Difference Method
def forward_difference(x, f, h):
    return (f[1:] - f[:-1]) / h

# Backward Difference Method
def backward_difference(x, f, h):
    return (f[1:] - f[:-1]) / h
```

Step 3: Calculate the Derivatives:

- Apply the three methods to calculate the derivative for each point in the data set.

```
# Compute derivatives using different methods
forward_derivatives = forward_difference(x_values, f_values, h)
backward_derivatives = backward_difference(x_values, f_values, h)
central_derivatives = central_difference(x_values, f_values, h)

# Print computed derivatives
print("Forward Differences:", forward_derivatives)
print("Backward Differences:", backward_derivatives)
print("Central Differences:", central_derivatives)
```

Step 4: Plot the Results:

- Use Matplotlib to plot the computed derivatives and compare the performance of the three methods.

```
# Plotting the results
plt.figure(figsize=(10, 6))

# Plot forward difference derivatives
plt.plot(x_values[1:], forward_derivatives, label='Forward Difference', marker='o', linestyle='--')

# Plot backward difference derivatives
```

```
plt.plot(x_values[1:], backward_derivatives, label='Backward Difference', marker='x', linestyle='-.')
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to evaluate numerical differentiation for the given data.
2. Write a program to implement Central Difference Method for the given data.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. **Traffic Flow Analysis:** Write a python program for Traffic Flow analysis.
   Consider traffic speed data collected over time (in minutes) at various checkpoints. Compute the rate of change in traffic speed (i.e., the acceleration or deceleration) at each point using numerical differentiation methods.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/newton-forward-backward-interpolation/
2. https://www.youtube.com/watch?v=sYGV_IznA-k
3. https://patrickwalls.github.io/mathematicalpython/differentiation/differentiation/

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.13

Write a program to evaluate numerical integration using Trapezoidal rule for the given data.

I. Practical Significance

The Trapezoidal Rule is computationally efficient, making it ideal for large datasets where you might have hundreds or thousands of data points. The Trapezoidal Rule offers a quick way to approximate integrals without needing to explicitly solve complex functions.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Produce accurate, reliable and actionable insights that can be used for decision-making in the workplace.'

III. Course Level Learning Outcome (CO)

CO4 - Evaluate numerical differentiation and integration functions.

IV. Laboratory Learning Outcome(s)

LLO 13.1 Find numerical integration using Trapezoidal rule for the given data using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Built Analytical thinking and know Ethical considerations when working with data and technology.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

The **Trapezoidal Rule** is a numerical method used to estimate the integral of a function by approximating the area under the curve of the function using trapezoids. This method is particularly useful when the function is represented by discrete data points, such as sensor readings, stock prices, or experimental data. The **Trapezoidal Rule** formula is as follows:

$$I \approx \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]$$

Where:

- I is the estimated integral.
- h is the width of the subintervals (i.e., the difference between consecutive x-values).
- f(x0), f(x1), …, f(xn) are the function values at the given data points.

The Trapezoidal Rule assumes that the function is approximately linear between each pair of consecutive points.

Step 1: Set Up Python Environment

Make sure Python is properly installed and ready to use on your machine.

1. Install any required libraries (e.g., numpy, matplotlib for visualization)

Step 2: Import Required Libraries

In your Python code, import the necessary libraries for calculations and visualization:

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

- **numpy** is used for numerical calculations, especially for arrays and mathematical operations.
- **matplotlib** is used for visualizing the data and the estimated integral.

Step 3: Define the Trapezoidal Rule Function

Write a Python function to apply the **Trapezoidal Rule** to a set of data points. The function will take arrays of x-values and corresponding y-values (data points) as input and return the estimated integral.

```
def trapezoidal_rule(x, y):

    h = x[1] - x[0]             # Assuming uniform spacing between x values
    integral = (h / 2) * (y[0] + 2 * np.sum(y[1:-1]) + y[-1])
    return integral
```

Step 4: Input Data

You can either manually input the data or read it from a file (e.g., CSV or Excel).

Option 1: Manual Data Entry If the data is small, you can input it directly as arrays:

```
x = np.array([0, 1, 2, 3, 4])        # Example x-values
y = np.array([1, 2, 3, 4, 5])        # Corresponding y-values
```

Option 2: Read Data from File (CSV) If the data is in a CSV file, you can use numpy or pandas to read it. Here's an example using numpy:

```
# Assume data.csv has two columns: x-values and y-values
```

```
data = np.genfromtxt('data.csv', delimiter=',')
x = data[:, 0]
y = data[:, 1]
```

Step 5: Calculate the Integral Using the Trapezoidal Rule

Now that you have the data, pass the x and y arrays to the function you created:

```
integral = trapezoidal_rule(x, y)
print(f"Estimated integral using Trapezoidal Rule: {integral}")
```

Step 6: Visualize the Data and the Trapezoidal Approximation

It's helpful to visualize the data points and the trapezoids formed during integration. Use matplotlib to plot the data points and trapezoidal approximation.

```
# Plot the data points
    plt.plot(x, y, 'bo-', label='Data points')

# Draw trapezoids
    for i in range(len(x)-1):
        plt.plot([x[i], x[i+1]], [y[i], y[i+1]], 'r-', lw=2)  # Trapezoid sides
        plt.fill([x[i], x[i+1], x[i+1], x[i]], [0, 0, y[i+1], y[i]], 'r', alpha=0.2)  # Fill trapezoid

    plt.xlabel('x')
    plt.ylabel('y')
    plt.title('Trapezoidal Rule Approximation')
    plt.legend()
    plt.show()
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to evaluate numerical integration using Trapezoidal rule for the given data.
2. Use data from sensors, financial data, or environmental data for the above program.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a python program to estimate the total rainfall over the day using the Trapezoidal Rule. (Given hourly rainfall data)
2. Write a python program to estimate the total distance traveled by integrating velocity over time. (Given velocity data)

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.geeksforgeeks.org/trapezoidal-rule-for-approximate-value-of-definite-integral/
2. https://www.youtube.com/watch?v=nrfbPTA9G00
3. https://www.tutorialspoint.com/modelling-the-trapezoidal-rule-for-numerical-integration-in-python

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.14

Write a program to evaluate numerical integration using Simpson's one third rule for the given data.

I. Practical Significance

Numerical integration using Simpson's One-Third Rule allows for precise, efficient, and automated solutions to integrals that are common in both academic and real-world scenarios. This method provides a high level of accuracy and can be applied to a wide range of problems in science, engineering, and applied mathematics.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Solve real-world problems that involve integration.'

III. Course Level Learning Outcome (CO)

CO4 - Evaluate numerical differentiation and integration functions.

IV. Laboratory Learning Outcome(s)

LLO 14.1 Find numerical integration using Simpson's one third rule for the given data using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Develop a deeper analytical mindset and built logical reasoning skills.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Numerical integration is a technique used to approximate the integral of a function, particularly when an analytical solution is not available. Simpson's One-Third Rule is a popular method for such approximations. It is based on approximating the integrand by a second-degree polynomial over intervals and is particularly accurate for smooth functions. Simpson's One-Third Rule is given by:

$$I = \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=1,3,5,\ldots}^{n-1} f(x_i) + 2\sum_{i=2,4,6,\ldots}^{n-2} f(x_i)\right]$$

1. **Define the Simpson's Rule function**:
   - Accept parameters: function f, bounds a and b, and number of subintervals n.
   - Ensure that n is even (if not, increment n by 1).
2. **Compute the integral approximation**:
   - Calculate the step size h

- Compute the sum of terms as per Simpson's rule:
  - Add the values at the endpoints f(a) and f(b).
  - Sum the function values at odd and even indices within the interval, applying the coefficients 4 and 2 respectively.
- Multiply the final sum by h/3.

3. **Test your program with a simple function** such as $f(x)=x^2$ or $f(x)=\sin(x)$.
   1. **Choose a function** for testing, such as:
      - $f(x)=x^2$
      - $f(x)= \sin(x)$
      - $f(x)=e^x$
   2. **Specify the integration bounds** and number of subintervals:
      - For example, integrate x^2 from 0 to 2 with 6 subintervals.
   3. **Run the program and compare the result** with the exact integral if known.

4. Observations and Results

1. **Record the results** of the numerical integration. Compare the approximation with the exact integral value if available.
2. **Evaluate the accuracy** by varying the number of subintervals n. Generally, as n increases, the result becomes more accurate.

```
def simpsons_one_third_rule(func, a, b, n):
    # Ensure n is even
    if n % 2 == 1:
        n += 1

    # Calculate the width of each subinterval
    h = (b - a) / n

    # Compute the sum using Simpson's One-Third Rule
    integral = func(a) + func(b)  # f(a) + f(b)

    for i in range(1, n, 2):  # Odd indices: multiply by 4
        integral += 4 * func(a + i * h)

    for i in range(2, n-1, 2):  # Even indices: multiply by 2
        integral += 2 * func(a + i * h)

    integral *= h / 3  # Final scaling factor

    return integral

# Example function to integrate
```

```
def f(x):
    return x**2  # Function to integrate: x^2

# Define limits of integration
a = 0  # Lower limit
b = 2  # Upper limit
n = 6  # Number of subintervals (must be even)

# Call the Simpson's One-Third Rule function
result = simpsons_one_third_rule(f, a, b, n)

print(f"Approximate integral result: {result}")
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to evaluate numerical integration using Simpson's one third rule for the given data.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a python program for numerical integration using Simpson's One-Third Rule for any function.
2. Write a python program to evaluate the error in Numerical Integration.
3. Write a python program to compare the Trapezoidal Rule with the Simpson's Rule for numerical integration.

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

X. References/Suggestions for further Reading

1. https://www.youtube.com/watch?v=YdYd_0a5mGE
2. https://www.geeksforgeeks.org/composite-simpsons-rule-to-evaluate-definite-integral/
3. https://www.vlab.andcollege.du.ac.in/phySc/mpPhy/simpson1by3/simpson13-main.html
4. https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter21.04-Simpsons-Rule.html

XI. Assessment Scheme

Performance Indicators		Weightage
Process related (15Marks)		**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (10Marks)		**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	

Practical No.15

Write a program to implement simplex method for 2 equations in 2 variables.

I. Practical Significance

The Simplex method is widely used to solve Linear Programming Problems (LPP), which involve optimizing a linear objective function, subject to linear constraints. In real-world scenarios, businesses and organizations often need to optimize resources (like production, transportation, etc.) to minimize costs or maximize profits.

II. Industry/Employer Expected Outcome (s)

This practical is expected to develop the following skills for the industry/Employer 'Write code using Simplex method to solve complex mathematical problems.'

III. Course Level Learning Outcome (CO)

CO5 - Apply the linear programming problem concept to obtain optimal solution.

IV. Laboratory Learning Outcome(s)

LLO 15.1 Find optimal solution of linear programming problems by applying simplex method using Python programming.

V. Relative Affective Domain related Outcome(s)-

- Develop a deeper analytical mindset and built logical reasoning skills.
- Maintain tools and equipment.
- Appreciate the utility and relevance of mathematics.

VI. Minimum Theoretical Background

Linear Programming is used to find the optimal solution to problems that involve a linear objective function and linear constraints. The Simplex method is one of the most widely used algorithms for solving LP problems. For two variables, the process involves iterating over feasible solutions to find the one that optimizes the objective function (either maximizing or minimizing).

Steps in Simplex Method (for two variables):

1. **Formulate the problem**:
    - Maximize or minimize the objective function.
    - Define the constraints in terms of inequalities.
2. **Convert inequalities to equations**:
    - Convert "≤" inequalities into equalities by adding slack variables.
3. **Set up the initial Simplex tableau**:
    - The tableau is a matrix that represents the system of equations for the LP problem.

- The first row contains the coefficients of the objective function.
- The rest of the rows represent the constraints.

4. **Perform iterations**:
    - Identify the pivot column (entering variable) by finding the most negative value in the last row (for maximization problems).
    - Identify the pivot row (leaving variable) by calculating the ratio of the right-hand side (RHS) to the pivot column.
    - Perform row operations to update the tableau until the optimal solution is found (when all values in the objective function row are non-negative).

```
import numpy as np

def pivot(tableau, row, col):
    # Perform pivot operation on the tableau
    pivot_element = tableau[row, col]
    tableau[row, :] = tableau[row, :] / pivot_element
    for i in range(tableau.shape[0]):
        if i != row:
            tableau[i, :] = tableau[i, :] - tableau[i, col] * tableau[row, :]

def simplex_method(tableau):
    # Iterate to find the optimal solution
    while np.any(tableau[-1, :-1] < 0):        # Continue until no negative values in the
objective row
        # Find pivot column (most negative element in objective row)
        pivot_col = np.argmin(tableau[-1, :-1])

        # Find pivot row using minimum ratio rule (RHS / pivot column value)
        ratios = tableau[:-1, -1] / tableau[:-1, pivot_col]
        ratios[ratios <= 0] = np.inf  # Ignore non-positive ratios
        pivot_row = np.argmin(ratios)

        # Perform pivot operation
        pivot(tableau, pivot_row, pivot_col)

    # Return the optimal value and the corresponding values of variables
    return tableau[-1, -1], tableau[:-1, -1]

# Define the initial tableau (same as above)
tableau = np.array([[1, 1, 1, 0, 4],
```

```
        [1, 0, 0, 1, 3],
        [-3, -2, 0, 0, 0]], dtype=float)

    # Run the Simplex method
    optimal_value, solution = simplex_method(tableau)

    print(f"Optimal value of Z: {optimal_value}")
    print(f"Optimal values of variables (x1, x2): {solution[:2]}")
```

VII. Resources Required

Sr. No	Name Resource	Specification	Quantity	Remarks
1	Computer System	Any desktop or laptop computer with basic configuration	One computer system for each student	
2	Operating system	Windows/LINUX	One for each computer system	
3	Software	Any Python IDE	One for each computer system	

VIII. Exercise

(Use blank space for answers or attach more pages if needed)

1. Write a program to implement simplex method for 2 equations in 2 variables.
2. Write a Python Program to Implement the Simplex Method for Minimization Problem (2 variables).

(Space for answers)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

IX. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more. Such questions to ensure the achievement of identified CO.

1. Write a Python Program to Implement the Simplex Method for 3 Equations in 3 Variables.
2. Write a Python program to solve a transportation problem using the Simplex method.

(Space for answers)

X. References/Suggestions for further Reading

1. https://www.javatpoint.com/simplex-algorithm-in-python
2. https://www.geeksforgeeks.org/simplex-algorithm-tabular-method/
3. https://www.youtube.com/watch?v=SVgMPjAqiEw

XI. Assessment Scheme

	Performance Indicators	Weightage
	Process related (15Marks)	**70%**
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
	Product related (10Marks)	**30%**
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
	Total (25Marks)	**100%**

Marks Obtained			Dated signature of Teacher
Process Related (15)	**Product Related (10)**	**Total (25)**	