Awarding Body: Arden University (Berlin)	
Programme Name: Maths for Data Science	
Module Name (and Part if applicable): COM7023	
Assessment Title: Maths for Data Science Portfolio	
Student Number: 24169956	
Tutor Name: Prof. Ali Vaisifard	
Word Count: 2870 Please refer to the Word Count Policy on your Module Page for guidance	

Table of Contents

Introduction

In this research, I look at a tiny mathematical model of **heat flow on a steel plate** that is utilized by the **EU Steel Preparation Agency.** The company hopes to use laser or plasma cutters to cut steel plates more effectively. They must comprehend how temperature affects the plate's surface to accomplish this. The cutting quality may be unsatisfactory, and the operation becomes less effective if the temperature is occasionally too high or too low.

There are four primary mathematical components to the problem. First, I solve a system of four equations using **linear algebra.** The approximate temperatures at four grid points on the plate are described by these equations. Second, I investigate how the temperature varies along a single plate edge using **calculus**. Since this can indicate where the material may be under higher thermal stress, I concentrate on the areas where the temperature varies most quickly. Third, I compute the likelihood that a temperature measurement error is greater than a given value using **probability**. Lastly, I use **hypothesis testing** to see whether the desired value of **100°C** is reached by the mean temperature at a particular location.

I use both **Python code** and mathematical reasoning for each section. Python lets me solve the problems and quickly visualize the results, while math demonstrates the fundamental concepts and principles. In data science, where both theory and computing are frequently required, this combination is typical. I go into how the findings fit together and what they might signify for the business at the conclusion of the report.

Linear Algebra – Heat Flow Across the Steel Plate

2.1 Problem Description

We are given a steel plate with four grid points on which the steady state temperatures are to be determined. Let x_1 , x_2 , x_3 and x_4 be these unknown temperatures. A set of four linear equations is used to simulate the heat flow. Every equation connects a single spot to both fixed boundary temperatures on the plate and its neighbours.

The system is:

$$4x_1 - x_2 - x_3 = 100$$

$$-x_1 + 4x_2 - x_4 = 100$$

$$-x_1 + 4x_3 - x_4 = 0$$

$$-x_1 - x_3 + 4x_4 = 0$$

Boundary circumstances, such as the plate's boundaries being held at such temperatures, are the source of the constant numbers **100 or 0** on the right. Coefficients like **4 and -1** show how each point is affected by heat from nearby points. The goal is to solve this system and find the numerical values of x_1 , x_2 , x_3 and x_4 , which estimates the temperature at each grid point in degrees Celsius.

2.2 Mathematical Model and Gaussian Elimination

To solve the system efficiently, I express it in matrix form as $\boldsymbol{Ax} = \boldsymbol{b}$, where

$$\boldsymbol{A} = \begin{bmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{bmatrix}, \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} 100 \\ 100 \\ 0 \\ 0 \end{bmatrix}$$

This gives the enhanced matrix:

$$[\boldsymbol{A} \quad | \quad \boldsymbol{b}] = \begin{bmatrix} 4 & -1 & -1 & 0 & | & 100 \\ -1 & 4 & 0 & -1 & | & 100 \\ -1 & 0 & 4 & -1 & | & 0 \\ 0 & -1 & -1 & 4 & | & 0 \end{bmatrix}$$

I transform this matrix into **upper triangular form using Gaussian elimination.** Eliminating the entries **(the diagonal elements)** beneath each pivot is the primary goal. For instance, the first pivot in the first row is **4**. By adding appropriate multiples of the first row to rows **2 and 3**, I may eliminate the **-1** entries beneath it.

In Practice, this means performing row operations such as:

$$\Rightarrow \boldsymbol{R_2 \leftarrow R_2 + \frac{1}{4}R_1}$$
$$\Rightarrow \boldsymbol{R_3 \leftarrow R_3 + \frac{1}{4}R_1}$$

The first column below the pivot **equals 0** because of these actions. To produce zeros beneath it, I then proceed to the second pivot and repeat the procedure. The matrix is gradually converted into an upper triangular form with zero values for all entries below the main diagonal. Once in this form, I can solve for $\boldsymbol{x_4}$ from the last row, then substitute upwards to find $\boldsymbol{x_3}$, $\boldsymbol{x_2}$, and $\boldsymbol{x_1}$.

I get the following result if I perform all the computations (or verify using Python):

$$\boldsymbol{x_1 = 37.5,\ \ x_2 = 37.5,\ \ x_3 = 12.5,\ \ x_4 = 12.5,}$$

2.3 Python Implementation and Numerical Solution

I can also use Python to solve the matrix equation rather than doing all the algebra by hand. I use the built-in linear algebra functions in the **NumPy** package. The pertinent code is shown in **Figure 2.3.1.**

```
import numpy as np

A = np.array([
    [4, -1, -1, 0],
    [-1, 4, 0, -1],
    [-1, 0, 4, -1],
    [0, -1, -1, 4]
], dtype=float)

b = np.array([100, 100, 0, 0], dtype=float)

x = np.linalg.solve(A, b)
print(x)

[37.5 37.5 12.5 12.5]
```

Figure 3.2.1: Python code and solution for the linear system.

This code creates **matrix A** and **vector b** and calls **np.linalg.solve** to find the **vector x.** The output is shown in **Figure 3.2.1.**

Point	**Temperature (°C)**
x_1	37.5
x_2	37.5
x_3	12.5
x_4	12.5

Table 2.3.1: Temperatures

2.4 Interpretation of the Temperatures

According to the numerical data, the lower grid points, $x\mathbf{3}$ and $x\mathbf{4}$, have temperatures of **12.5°C**, whereas the top two grid points, $x\mathbf{1}$ and $x\mathbf{2}$, have temperatures of **37.5°C.** According to this pattern, heat is concentrated closer to the areas of the plate that are associated with the **100°C** border conditions. Because they are farther from the hotter edges and affected by the boundary at **0°C**, the lower spots are colder.

This indicates that the steel plate does not reach a consistent temperature from a practical perspective. Certain areas are still much colder. Such temperature variations may have an impact on the quality of a laser or plasma cut. While cooler areas may demand more energy or result in uneven edges, hotter areas may be simpler to cut. Engineers can modify process parameters, like cutting speed or power, to get consistent outcomes throughout the entire plate by having a better understanding of these temperature distributions.

Calculus – Rate of Change of Temperature

3.1 Temperature Function Along the Edge

The temperature is modeled as a continuous function of position *x* along one edge of the plate:

$$T(x) = 100sin\left(\frac{\pi x}{10}\right)$$

where x is measured from **0 to 10** along the edge. The temperature variation from one end of the edge to the other is described by this function. The temperature rises from **zero to a maximum** in the middle and then returns to zero at the opposite end, according to the **sine curve.**

Finding the locations where temperature varies the fastest is the aim of this part, in addition to knowing the temperature at each location. In calculus, the derivative $T'(x)$ represents the rate of change. We can identify the locations where the temperature gradient is highest by examining the derivative; these locations may be crucial for material stress or cooling rates.

3.2 Derivative and Points of Fastest Change

To check how quickly the temperature changes according to the position, I differentiate $T(x)$ with respect to x. Using the **chain rule:**

$$T(x) = 100sin\left(\frac{\pi x}{10}\right)$$

Gives

$$T'(x) = 100 \cdot \cos\left(\frac{\pi x}{10}\right) \cdot \frac{\pi}{10} = 10\pi \cos\left(\frac{\pi x}{10}\right)$$

The temperature curve's slope at each location is revealed by the derivative. The temperature varies quickly when $|T'(x)|$ is large; it changes slowly or not at all when $|T'(x)|$ is small or zero.

The maximum possible value of $|cos(\theta)|$ is **1**. So, the largest value of $|T'(x)|$ can be:

$$\max|T'(x)| = 10\pi \cdot 1 = 10\pi \approx 31.42$$

This occurs when

$$\cos\left(\frac{\pi x}{10}\right) = \pm 1$$

Solution:

- $\cos\left(\frac{\pi x}{10}\right) = 1$ occurs when $\left(\frac{\pi x}{10}\right) = 2k\pi \Rightarrow x = 20k$
- $\cos\left(\frac{\pi x}{10}\right) = -1$ occurs when $\left(\frac{\pi x}{10}\right) = (2k+1)\pi \Rightarrow x = 10(2k+1)$.

In the interval $0 \leq x \leq 10$, the relevant values are $x = 0$ *and* $x = 10$. This indicates that the temperature fluctuates more quickly near the two ends of the edge.

3.3 Graphical Analysis Using Python

I plot the temperature function along the edge using Python to understand this behavior better. A straightforward Python script shown in Figure 3.3.1:

```python
import matplotlib.pyplot as plt

x = np.linspace(0, 10, 400)
T = 100 * np.sin(np.pi * x / 10)
Tp = 10 * np.pi * np.cos(np.pi * x / 10)

plt.plot(x, T)
plt.xlabel('Position x')
plt.ylabel('Temperature T(x)')
plt.title('Temperature along plate edge')
plt.grid(True)
plt.show()
```

Figure 3.3.1: Python code for T(x)' along plate edge

The graph shown in **Figure 3.3.2** displays a smooth sine wave that begins at **0°C at x=0**, reaches a maximum of **100°C at x=5**, and returns to **0°C at x=10**. The derivative confirmed that the curve is sharpest close to the ends.

I can also plot the derivative ***T' (x)*** if I'd like, but the main notion is that the steep portions of the curve correlate to high values of **|*T' (x)*|**.

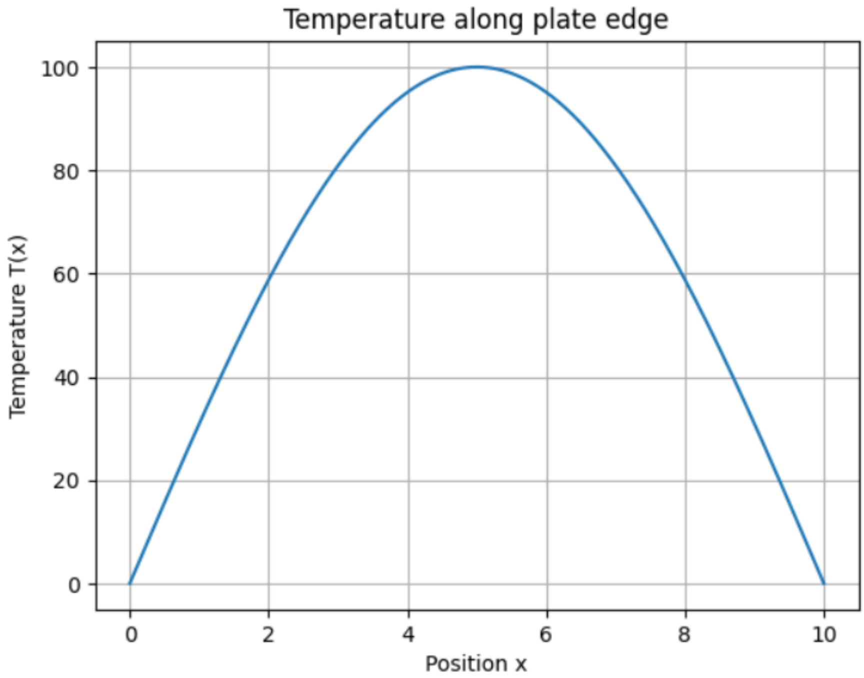

Figure 3.3.2: Graph of temperature along the plate edge.

3.4 Practical Implications

For the steel cutting process, it is helpful to know that the temperature varies most quickly around **x = 0 and x = 10**. Thermal stress in the material might result from abrupt temperature fluctuations. When the steel is cut or cooled, these tensions could cause distortion, cracks, or other flaws.

To lower the danger of damage, engineers may need to modify the cutting speed, power, or cooling flow if the cutting path passes through areas with extremely high gradients. On the other hand, the thermal conditions are more stable in the center area **around x = 5,** where the temperature is high but varies more slowly. For crucial cutting tasks, this area might be more advantageous.

This portion of the assignment demonstrates how calculus may be used to analyze patterns in a function and pinpoint significant locations in terms of data science. Plots and derivatives together provide a vivid picture of the physical system's behavior along a continuous dimension.

Probability – Size of Measurement Error

4.1 Modelling the Error as a Normal Distribution

There are errors in the temperature readings. The discrepancy between the measured and actual temperatures is represented by the **error term *E*.** It is assumed that the mistake in this assignment has a normal distribution with a mean of **0** and a standard deviation of **2°C**:

$$\boldsymbol{E \sim N(0,\ 2^2)}$$

This indicates that while the measurements are generally accurate, individual readings may differ somewhat from the actual value. The inaccuracy is usually relatively modest, but it can occasionally be rather big.

What is the likelihood that the inaccuracy will be larger than 3°C? This is frequently expressed as ***P* (|*E*| > 3).** The measurement system could not be dependable enough for accurate industrial applications if this probability is significant.

4.2 Standardization and Probability Calculation

I use the following to transform **error *E*** into a standard normal **variable *Z*** to get the **probability:**

$$\boldsymbol{Z = \frac{E - 0}{2}}$$

Regarding the **3°C** barrier,

$$\boldsymbol{P(E > 3) = P\left(Z > \frac{3}{2}\right) = P(Z > 1.5)}$$

The cumulative distribution function at **1.5** is roughly $\boldsymbol{\Phi(1.5) \approx 0.9332}$, According to ordinary normal tables or Python. Consequently,

$$\boldsymbol{P(E > 3) = 1 - \Phi(1.5) \approx 1 - 0.9332 = 0.0668}$$

If I want to know how big the error is in both directions (too high or too low), I figure out:

$$\boldsymbol{P(E > 3) = 2P(E > 3) \approx 2\ \times 0.0668 = 0.1336}$$

Therefore, there is a **6.7% probability** that the error is greater than **3°C** in the positive direction and a **13.4% probability** that the error is greater than **3°C** in either direction.

4.3 Interpretation for the Measurement System

For a substantial mistake **(greater than 3°C in size),** the probability of about **13%** is not very modest. This indicates that the inaccuracy may be greater than **3°C** in about **1 in 7 or 8** measurements. The steel cutting process's quality standards will determine whether or not this is acceptable.

Such a high likelihood of significant errors could be an issue if extremely precise temperature control is required. The business may then choose to add additional calibration procedures, take several measurements and average them, or enhance the sensors. This section of the assignment demonstrates how measurement system dependability can be evaluated using probability and the normal distribution. As shown in **Figure 4.3.1.**

```
from scipy.stats import norm

p_one = norm.sf(3, loc=0, scale=2)   # P(E > 3)
p_two = 2 * p_one                    # P(|E| > 3)

print("One-sided:", p_one)
print("Two-sided:", p_two)
```

```
One-sided: 0.06680720126885807
Two-sided: 0.13361440253771614
```

Figure 4.3.1: Python code: calculates the probabilities using scipy.stats.norm

Hypothesis Testing – Mean Temperature at a Point

5.1 Sample Data and Goal of the Test

The following eight temperature readings **(in °C)** are recorded by the company at a specific location on the steel plate:

$$\mathbf{99, 100, 98, 101, 97, 99, 100, 98}$$

At this moment, **100°C** is the target mean temperature. Whether these observed values support the goal mean or provide evidence that the true mean deviates from **100°C** is the matter at hand.

I use a one-sample **hypothesis test** on the mean to address this. I employ a **t-test** instead of a **z-test** because the sample size is **small** and the population standard deviation is unknown. As shown in **Table 5.1.1.**

Measure	***Value***
Sample size n	8
Sample mean $\bar{x}$	99
Sample standard deviation s	1.31 (approx.)

Table 5.1.1: Sample size, mean, and standard deviation for the recorded temperatures

5.2 Hypotheses and Choice of Test

Let $\boldsymbol{\mu}$ be the actual mean temperature at this moment. The **hypotheses** are:

- ⇒ Null hypothesis: $\boldsymbol{H_0: \mu = 100}$
- ⇒ Alternative hypothesis: $\boldsymbol{H_1: \mu \neq 100}$

This is a two-sided test since I want to know if the mean deviates from **100°C** in any direction, not just higher or lower.

I employ a one-sample **t-test** because the population standard deviation is unknown and $\boldsymbol{n = 8}$ is small. The test statistic is provided by:

$$\boldsymbol{t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}}$$

Where $\bar{x}$ is the sample mean, $\mu_0 = 100$ is the hypothesised mean, s is the sample standard deviation, and n is the sample size.

5.3 Manual Calculation of the t-Statistic

First, I've computed the sample mean:

$$\bar{x} = \frac{99 + 100 + 98 + 101 + 97 + 99 + 100 + 98}{8} = \frac{792}{8} = 99$$

Next, I've calculated the sample standard deviation s using the formula:

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

If I figure this out (or use Python), I get $s \approx 1.3093.$

Now, I compute the **t-statistic** using:

$$t = \frac{\bar{x} - 100}{\delta/\sqrt{n}} = \frac{99 - 100}{1.3093/\sqrt{8}} = \frac{-1}{1.3093/\sqrt{2.8284}} \approx \frac{-1}{0.4630} \approx -2.16$$

The degrees of freedom are:

$$df = n - 1 = 7$$

I evaluate the p-value for t = **– 2.16** and df = **7** to the significance level $\alpha = 0.05$ to conclude. The two-sided p-value using software or a t-table is roughly **0.068**.

Since $p \approx 0.068 > 0.05$, At the **5%** level, I do **not reject** the null hypothesis.

5.4 Python Confirmation of the Test

To confirm the manual calculations, I've used Python and the scripy.stats library as shown in **Figure 5.4.1.**

```
from scipy import stats

data = np.array([99, 100, 98, 101, 97, 99, 100, 98])
t_stat, p_value = stats.ttest_1samp(data, 100)

print("t-statistic:", t_stat)
print("p-value:", p_value)
```

```
t-statistic: -2.160246899469287
p-value: 0.06758329382336989
```

Figure 5.4.1: Python t-test output.

A one-sample **t-test** of the mean is carried out by this code. The result provides **the two-sided p-value** and the **t-statistic**. The t-statistic of **around -2.16** and the p-value of roughly **0.068** are in line with the manual results.

5.5 Decision and Practical Conclusion

The **p-value of roughly 0.068** is more than **0.05** at the **5% significance level**. In other words, the outcome is not statistically significant. Thus, I am unable to reject the null hypothesis $\boldsymbol{H_0: \mu = 100}$**.** In other words, there is insufficient data to draw the conclusion that the actual mean temperature differs from **100°C** based on this sample of **eight measurements**.

In practical terms, the measured average of **99°C** is close to the desired temperature of **100°C**. The slight discrepancy can just be the result of a measurement mistake and chance variance. This implies that the current temperature is about on target for the company. To make sure that the process stays stable over time, it could still be prudent to gather additional data or to routinely check this point because the sample size is rather small.

Discussion

I compiled the findings from the four mathematical components in this section and considered their implications for the steel cutting procedure.

I discovered that the four grid spots on the plate have different steady-state temperatures from the linear algebra section. There are two locations that are roughly **37.5°C** and two that are roughly **12.5°C**. This demonstrates a distinct temperature gradient brought on by the boundary circumstances. This means that different parts of the plate will respond to the same cutting parameters in different ways.

The math portion examined temperature variations along a single edge. The middle of the function $\boldsymbol{T(x) = 100 sin\left(\frac{\pi x}{10}\right)}$ is **100°C**, whereas the ends are **0°C.** The derivative demonstrated that the temperature changes most quickly around **x = 0 and x = 10**. Cutting close to these areas may require special caution since rapid changes might cause thermal stress and alter how the material expands or contracts.

Measurement inaccuracy was the main topic of the probability section. The probability that the mistake is more than **3°C** is roughly **13.4%** with $\boldsymbol{E \sim N(0,\ 2^2)}$. This is not insignificant. It implies that even in cases where the system is steady, individual readings may occasionally deviate significantly from the actual temperature. Instead of relying solely on one value for quality control, it might be preferable to utilize smoothing techniques or repeated measurements.

In the hypothesis testing phase, it was determined if the average temperature at a given location was **100°C**. There is insufficient evidence at the **5% level** to conclude that the true mean differs from **100°C with a sample mean of 99°C**, a t-statistic of around **-2.16**, and a p-value of roughly **0.068**. This implies that the process is **statistically near the goal.**

Overall, these findings demonstrate how data-driven analysis of an industrial process can be accomplished by combining **linear algebra, calculus, probability, and hypothesis testing.**

Conclusion

A simplified mathematical model of heat flow on a steel plate utilized in a cutting process has been examined in this research. I discovered that the steady-state temperatures at the grid points, which have values **of 37.5°C and 12.5°C**, are not uniform by solving a system of four linear equations. This demonstrates the plate's distinct temperature gradients, which may have an impact on cutting efficiency.

I examined the temperature change along a plate edge using calculus. The temperature varies more quickly at the edge's ends, according to the temperature function's derivative. These areas might need extra care during cutting because they are more susceptible to heat stress.

There is a discernible possibility of errors **greater than 3°C**, according to the probability analysis of measurement error. This emphasizes how crucial it is to comprehend sensor accuracy and take tactics like repeated measurements into account. There is some assurance that the process is operating as planned because the **one-sample t-test** on the mean temperature at one point indicated that the **true mean is not substantially different from the objective of 100°C at the 5% level.**

Overall, the assignment shows how Python and mathematical skills may be used to understand physical processes. In real-world data science and engineering problems, the same method can be used for larger datasets and more complicated systems.

References

1. Strang, G. and Herman, E. (2016) *Calculus Volume 1.* Houston, TX: OpenStax. Available at: https://openstax.org/books/calculus-volume-1/pages/1-introduction (Accessed: 17 November 2025).
2. Illowsky, B. and Dean, S. (2023) *Introductory Statistics 2e.* Huston, TX: OpenStax. Available at: https://openstax.org/books/introductory-statistics-2e/pages/1-introduction (Accessed: 17 November 2025).
3. NumPy Developers (2025) *NumPy Documentation.* Available at: https://numpy.org/doc/ (Accessed: 17 November 2025).
4. Virtanen, P. et al. (2020) 'SciPy 1.0: Fundamental algorithms for scientific computing in Python', *Nature Methods,* 17(3), pp. 261-272. Available at: https://www.nature.com/articles/s41592-019-0686-2 (Accessed: 17 November 2025).
5. Matplotlib Developers (2025) *Matplotlib: Visualization with Python.* Available at: https://matplotlib.org/ (Accessed: 17 November 2025).

Appendix

```
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
from scipy.stats import norm

A = np.array([
    [4, -1, -1, 0],
    [-1, 4, 0, -1],
    [-1, 0, 4, -1],
    [0, -1, -1, 4]
], dtype=float)
b = np.array([100, 100, 0, 0], dtype=float)
x = np.linalg.solve(A, b)
print(x)

x = np.linspace(0, 10, 400)
T = 100 * np.sin(np.pi * x / 10)
Tp = 10 * np.pi * np.cos(np.pi * x / 10)

plt.plot(x, T)
plt.xlabel('Position x')
plt.ylabel('Temperature T(x)')
plt.title('Temperature along plate edge')
plt.grid(True)
plt.show()


p_one = norm.sf(3, loc=0, scale=2)
p_two = 2 * p_one


print("One-sided:", p_one)
print("Two-sided:", p_two)


data = np.array([99, 100, 98, 101, 97, 99, 100, 98])
t_stat, p_value = stats.ttest_1samp(data, 100)


print("t-statistic:", t_stat)
print("p-value:", p_value)
```

GitHub Link: ***https://github.com/mirzasameer2000/Maths-for-Data-Science-Portfolio***