

Engineering Mathematics and Computing

Task 2: Coursework Assessment

Student Name: Sakariye Abiikar
KID: 2371673

Last Updated: Thursday 9th January, 2025, 13:21
Submission Deadline: Thursday 9th January, 2025, 5pm

Git Repo : <https://github.com/sakx7/mathcompuni2>

Part A

Mathematics

A.1 Q1

1. Find $\int \frac{1}{7x+6} dx$

$$\int \frac{1}{7x+6} dx$$

Use substitution, Let

$$u = 7x + 6$$

$$\frac{du}{dx} = 7 \quad \Rightarrow \quad dx = \frac{du}{7}$$

Substituting into the integral, we get

$$\int \frac{1}{7x+6} dx = \frac{1}{7} \int \frac{1}{u} du$$

This is a standard integral of the reciprocal

$$\frac{1}{7} \int \frac{1}{u} du = \frac{1}{7} \ln|u| + C$$

Finally, substituting back $u = 7x + 6$, we get

$$\boxed{\int \frac{1}{7x+6} dx = \frac{1}{7} \ln|7x+6| + C}$$

A.2 Q2

2. Find $\int \frac{x}{\sqrt{4-x^2}} dx$

$$\int \frac{x}{\sqrt{4-x^2}} dx$$

Use substitution, Let

$$u = 4 - x^2$$

$$\frac{du}{dx} = -2x \quad \Rightarrow \quad dx = -\frac{du}{2x}$$

Substituting into integral, we get

$$\int \frac{x}{\sqrt{4-x^2}} dx = \int \frac{x}{\sqrt{u}} \cdot \left(-\frac{du}{2x}\right) = -\frac{1}{2} \int \frac{1}{\sqrt{u}} du$$

This is a standard integral of the exponent/power

$$-\frac{1}{2} \int u^{-\frac{1}{2}} du = -\frac{1}{2} \cdot 2\sqrt{u} + C = -\sqrt{u} + C$$

Finally, substituting back $u = 4 - x^2$, we get

$$\boxed{\int \frac{x}{\sqrt{4-x^2}} dx = -\sqrt{4-x^2} + C}$$

A.3 Q3

3. Obtain the general solution of the equation $\frac{d^2y}{dx^2} - 18\frac{dy}{dx} + 81y = 0$

$$\frac{d^2y}{dx^2} - 18\frac{dy}{dx} + 81y = 0$$

We use the ansatz $y = e^{rx}$, substituting it, along with its corresponding derivatives:

$$r^2e^{rx} - 18re^{rx} + 81e^{rx} = 0$$

Dividing through by e^{rx} (which is never zero), we get the characteristic equation:

$$r^2 - 18r + 81 = 0$$

We can factorize the quadratic as:

$$(r - 9)(r - 9) = 0$$

This gives us a repeated root:

$$r_1 = r_2 = 9$$

For a second-order linear homogeneous differential equation with constant coefficients and a repeated root r , the general solution is:

$$y(x) = c_1e^{rx} + c_2xe^{rx}$$

Substituting $r = 9$ into the general solution, we get:

$$y(x) = c_1e^{9x} + c_2xe^{9x}$$

Factorising we get the final solution as:

$$\boxed{y(x) = e^{9x}(c_1 + c_2x)}$$

Interval of validity: Nothing really stands out, no singularities or undefined behaviours. Therefore, the interval of validity is:

$$(-\infty, \infty) = \{x \in \mathbb{R} \mid -\infty < x < \infty\}$$

This means the solution is valid for all real values of x . The constant C does not effect the solutions validity regardless of its value.

A.4 Q4

4. Find the particular solution of the differential equation $\frac{dy}{dx} + 3yx^3 = 0$, given $y(0) = 1$

$$\frac{dy}{dx} + 3yx^3 = 0$$

Separate the variables:

$$\frac{1}{y} dy = -3x^3 dx$$

Integrating both sides:

$$\int \frac{1}{y} dy = \int -3x^3 dx$$

$$\ln|y| = -\frac{3x^4}{4} + C$$

Exponentiating both sides:

$$\begin{aligned} y &= \exp\left(-\frac{3x^4}{4} + C\right) \\ &= C \exp\left(-\frac{3x^4}{4}\right) \end{aligned}$$

Apply the initial condition $y(0) = 1$:

$$1 = C \exp\left(-\frac{3(0)^4}{4}\right)$$

$$C = 1$$

Thus, the particular solution is:

$$\boxed{y(x) = e^{-\frac{3x^4}{4}}}$$

Interval of validity: Nothing really stands out, no singularities or undefined behaviours. Therefore, the interval of validity is:

Domain: $x \in \mathbb{R}$

Interval: $(-\infty, \infty)$

This means the solution is valid for all real values of x . The constant C does not effect the solutions validity regardless of its value

A.5 Q5

5. If $z = \frac{11+10j}{9-3j}$, express both $\frac{1}{z}$ and $z + \frac{1}{z}$ in the standard form $\alpha + \beta j$

$$z = \frac{11 + 10j}{9 - 3j}$$

$$z = \frac{(11 + 10j)(9 - 3j)}{(9 - 3j)(9 - 3j)}$$

Simplify the denominator:

$$\begin{aligned}(9 - 3j)(9 + 3j) &= 9^2 - (3j)^2 \\ &= 81 - (-9) \\ &= 81 + 9 = 90\end{aligned}$$

Simplify the numerator:

$$\begin{aligned}(11 + 10j)(9 + 3j) &= (11 \cdot 9) + (11 \cdot 3j) \\ &\quad + (10j \cdot 9) + (10j \cdot 3j) \\ &= 99 + 33j + 90j + 30j^2 \\ &= 99 + 123j + 30(-1) \\ &= 99 + 123j - 30 \\ &= 69 + 123j\end{aligned}$$

So:

$$\begin{aligned}z &= \frac{69 + 123j}{90} \\ &= \frac{69}{90} + \frac{123}{90}j \\ &= 0.7\bar{6} + 1.3\bar{6}j\end{aligned}$$

$$\frac{1}{z} = \frac{11 + 10j}{9 - 3j}$$

$$\frac{1}{z} = \frac{(9 - 3j)(11 - 10j)}{(11 + 10j)(11 - 10j)}$$

Simplify the denominator:

$$\begin{aligned}(11 + 10j)(11 - 10j) &= 11^2 - (10j)^2 \\ &= 121 - (-100) \\ &= 121 + 100 = 221\end{aligned}$$

The numerator is the conjugate of the previously calculated numerator, so:

$$(9 - 3j)(11 - 10j) = 69 - 123j$$

Thus:

$$\begin{aligned}\frac{1}{z} &= \frac{69 - 123j}{221} \\ &= \frac{69}{221} + \frac{-123}{221}j \\ &\approx \boxed{0.31 - 0.56j}\end{aligned}$$

now we can calculate $z + \frac{1}{z}$ easily

$$\begin{aligned}z + \frac{1}{z} &= \frac{69}{90} + \frac{123}{90}j + \frac{69}{221} + \frac{-123}{221}j \\ &= \left(\frac{69}{90} + \frac{69}{221}\right) + \left(\frac{123}{90} - \frac{123}{221}\right)j \\ &\approx \boxed{1.08 + 0.81j}\end{aligned}$$

in non approximate form this is $\frac{7153}{6630} + \frac{5371}{6630}j$

A.6 Q6

6. Obtain the general solution of $\frac{d^2y}{dx^2} - 2\frac{dy}{dx} - 48y = 5$

$$\frac{d^2y}{dx^2} - 2\frac{dy}{dx} - 48y = 5$$

Since this is **non-homogenous** we need to solve for the **complementary solution** (y_c) and then the **particular solution** (y_p) the general form of the solution the addition of these

$$y(t) = y_c(t) + y_p(t)$$

First lets solve the complementary solution, which solves the associated homogeneous equation:

$$\frac{d^2y}{dx^2} - 2\frac{dy}{dx} - 48y = 0$$

We use the ansatz $y = e^{rx}$, substituting it, along with its corresponding derivatives:

$$r^2e^{rx} - 2re^{rx} - 48e^{rx} = 0$$

Dividing through by e^{rx} (which is never zero), we obtain the characteristic equation:

$$r^2 - 2r - 48 = 0$$

We can factorize the quadratic as:

$$(r - 8)(r + 6) = 0$$

This gives us the roots:

$$r_1 = 8, \quad r_2 = -6$$

The general solution to a second-order linear homogeneous equation is given by:

$$y(x) = c_1e^{r_1x} + c_2e^{r_2x}$$

In so for our solution this is

$$y_c(x) = c_1e^{8x} + c_2e^{-6x}$$

Since the non-homogeneous term is a constant 5, we use the ansatz $y = A$, where A is a constant. Since A is a constant, its derivatives are zero, so:

$$-48A = 5$$

$$A = -\frac{5}{48}$$

Therefore, the particular solution is:

$$y_p(x) = -\frac{5}{48}$$

Now we have complementary solution (y_c) and the particular solution (y_p), The general solution to the non-homogeneous equation can be written as so:

$$y(x) = y_h(x) + y_p(x)$$

plugging in the solved values we have

$$y(x) = c_1e^{8x} + c_2e^{-6x} - \frac{5}{48}$$

Interval of validity: Nothing really stands out, no singularities or undefined behaviours. Therefore, the interval of validity is:

$$(-\infty, \infty) = \{x \in \mathbb{R} \mid -\infty < x < \infty\}$$

This means the solution is valid for all real values of x . The constant C does not effect the solutions validity regardless of its value.

A.7 Q7

7. Find $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ when $z = xy^4e^{2x}$

$$z = xy^4e^{2x}$$

$$\frac{\partial z}{\partial x} = \frac{\partial (xy^4e^{2x})}{\partial x}$$

Use the product rule:

$$\frac{\partial(u(x) \cdot v(x))}{\partial x} = v \frac{\partial u}{\partial x} + u \frac{\partial v}{\partial x}$$

let $u(x) = x$ and $v(x) = y^4e^{2x}$

$$\frac{\partial z}{\partial x} = \frac{\partial (xy^4e^{2x})}{\partial x} = y^4e^{2x} \frac{\partial(x)}{\partial x} + x \frac{\partial(y^4e^{2x})}{\partial x}$$

Easy to solve, just treat y as a constant and differentiate with respect to x :

$$\frac{\partial(x)}{\partial x} = 1, \quad \frac{\partial(y^4e^{2x})}{\partial x} = 2e^{2x}$$

now plug in we get

$$\begin{aligned} \frac{\partial z}{\partial x} &= y^4e^{2x} + xy^4(2e^{2x}) \\ &= e^{2x}y^4(2x + 1) \end{aligned}$$

Thus:

$$\boxed{\frac{\partial z}{\partial x} = e^{2x}y^4(2x + 1)}$$

$$z = xy^4e^{2x}$$

$$\frac{\partial z}{\partial y} = \frac{\partial (xy^4e^{2x})}{\partial y}$$

Treat x as a constant, we only differentiate y^4 :

$$\frac{\partial z}{\partial y} = \frac{\partial (xy^4e^{2x})}{\partial y} = xe^{2x} \frac{\partial(y^4)}{\partial y}$$

$$\frac{\partial(y^4)}{\partial y} = 4y^3$$

Thus:

$$\boxed{\frac{\partial z}{\partial y} = 4xy^3e^{2x}}$$

A.8 Q8

8. Integrate the function $\int \frac{x+9}{x(x+5)} dx$

$$\int \frac{x+9}{x(x+5)} dx$$

Split the integral:

$$\int \frac{x+9}{x(x+5)} dx = \int \frac{(x+5)+4}{x(x+5)} = \int \frac{1}{x} dx + 4 \int \frac{1}{x(x+5)} dx$$

The first integral is straightforward:

$$\int \frac{1}{x} dx = \ln|x| + C_1$$

For the second integral, rewrite:

$$\int \frac{1}{x(x+5)} dx = \int \frac{1}{x^2 \left(\frac{5}{x} + 1\right)} dx$$

Now use sub

$$u = \frac{5}{x} + 1$$

$$\frac{du}{dx} = -\frac{5}{x^2} \Rightarrow dx = -\frac{x^2}{5} du$$

Substituting into the integral, we get:

$$\int -\frac{x^2}{5x^2u} du = -\frac{1}{5} \int \frac{1}{u} du = -\frac{1}{5} \ln|u| + C_2$$

Substituting $u = \frac{5}{x} + 1 = \frac{5+x}{x}$ and then combining results yields:

$$\begin{aligned} \int \frac{x+9}{x(x+5)} dx &= \ln|x| + 4 \left(-\frac{1}{5} \ln \left| \frac{x+5}{x} \right| \right) + C \\ &= \ln|x| - \frac{4}{5} (\ln|x+5| - \ln|x|) + C \\ &= -\frac{4 \ln|x+5| - 9 \ln|x|}{5} + C \end{aligned}$$

Thus, the solution is:

$$\boxed{\int \frac{x+9}{x(x+5)} dx = \frac{9 \ln|x| - 4 \ln|x+5|}{5} + C}$$

A.9 Q9

9. Solve the equation $\frac{dy}{dt} + y \cot t = 5 \sin t$

$$\frac{dy}{dt} + y \cot t = 5 \sin t$$

Use integrating factor $\mu(t)$:

we get:

$$\mu(t) = e^{\int \cot t \, dt}$$

$$\frac{dy}{dt} \sin t + y \sin t \cot t = 5 \sin^2 t$$

Since this is a conventional integral, I shouldn't actually include it in the calculation because it should be widely recognised as $\ln|\sin t|$. Regardless, I'll demonstrate how to solve it quickly:

$$\frac{dy}{dt} \sin t + y \cos t = 5 \sin^2 t$$

The left side of the equation is nothing more than the product rule:

$$\int \cot t \, dt = \int \frac{1}{\tan t} \, dt = \int \frac{\cos t}{\sin t} \, dt$$

$$\frac{d}{dt}(y \sin t) = 5 \sin^2 t$$

Now use sub

Integrating both sides, we obtain:

$$u = \sin t$$

$$\frac{du}{dt} = \cos t \Rightarrow dx = \frac{1}{\cos t} \, dt$$

$$\int \frac{\cos t}{\sin t} \, dt = \int \frac{1}{u} \, du = \ln|\sin t| + C$$

so the int factor $\mu(t)$ is

$$\mu(t) = e^{\ln|\sin t|} = |\sin t|$$

$$\begin{aligned} y \sin t &= \int 5 \sin^2 t \, dt \\ &= \int 5 \cdot \frac{1 - \cos 2t}{2} \, dt \\ &= \frac{5}{2} \left(\int 1 \, dt - \int \cos 2t \, dt \right) \\ &= \frac{5}{2} \left(t - \frac{\sin 2t}{2} \right) + C \end{aligned}$$

Multiply both sides of the differential equation by $\mu(t)$

In so y as an explicit solution is:

$$y = \frac{5}{2 \sin t} \left(t - \frac{\sin 2t}{2} \right) + \frac{C}{\sin t}$$

Interval of validity: $\sin(t)$ is undefined where $\sin(t) = 0$, which occurs at $t = n\pi$, where n is any integer. Therefore, the interval of validity of the solution is:

$$\mathbb{R} \setminus \{n\pi : n \in \mathbb{Z}\} = \{t \in \mathbb{R} \mid t \neq n\pi \quad \forall n \in \mathbb{Z}\}$$

This means the solution is valid for all t being real, except for those values where $\frac{t}{\pi}$ is an integer. The constant C does not effect the solutions validity regardless of its value.

A.10 Q10

10. Evaluate the integral $\int_1^3 4xe^{4x} dx$

$$\int_1^3 4xe^{4x} dx$$

Use substitution, Let

$$u = 4x$$

$$\frac{du}{dx} = 4 \quad \Rightarrow \quad dx = \frac{1}{4} du$$

Now change of limits

$$4(3) = 12, \quad 4(1) = 4$$

The integral becomes

$$\frac{1}{4} \int_4^{12} ue^u dx$$

Use integration by parts $\int_b^a uv = [uv]_b^a - \int_b^a vu'$

$$u = x \quad dv = e^u$$

$$du = 1 \quad v = e^u$$

$$\frac{1}{4} \left([ue^u]_4^{12} - \int_4^{12} e^u \right)$$

$$\frac{1}{4} ([ue^u - e^u]_4^{12})$$

$$\frac{1}{4} ((12e^{12} - e^{12}) - (4e^4 - e^4)) \approx 4.4753 \times 10^5$$

$$\boxed{\int_1^3 4xe^{4x} dx \approx 4.4753 \times 10^5}$$

Part B

Computing

B.1 Q1

The boiling temperature of water, T_B , at various altitudes h , is given in the following table:

Altitude (m)	Boiling Temperature (°C)
0	100
2300	98.8
3000	95.1
6100	92.2
7900	90.0
10000	81.2
12000	75.6

Determine a linear equation in the form:

$$T_B = mh + b$$

that best fits the data. Use this equation to calculate the boiling temperature at 5000m. Additionally, make a plot of the data points and the equation.

main/q1.m

```
function applyPlotFormatting(ax, fig, titleText, xlabelText, ylabelText, legendText,
    filename)
    if ~isempty(titleText)
        title(ax, titleText);
    end
    if ~isempty(xlabelText)
        xlabel(ax, xlabelText);
    end
    if ~isempty(ylabelText)
        ylabel(ax, ylabelText);
    end
    fig.Color = 'w';
    ax.XGrid = 'on';
    ax.YGrid = 'on';
    ax.XMinorGrid = 'on';
    ax.YMinorGrid = 'on';
    ax.XMinorTick = 'on';
    ax.YMinorTick = 'on';
    ax.TickDir = 'out';
    ax.FontName = 'Calibri';
    ax.FontSize = 9;
    if ~isempty(legendText) && ~isequal(legendText, 'None')
        legend(ax, legendText, 'Location', 'best', 'Interpreter', 'latex');
    end
    if ~isempty(filename) && ~strcmp(filename, 'None')
        outputDir = 'graphs_images';
        savePath = fullfile(outputDir, [filename, '.jpeg']);
        if ~exist(outputDir, 'dir')
            [status, msg] = mkdir(outputDir);
            if ~status
                error('Unable to create the directory "%s": %s', outputDir, msg);
            end
        end
        try
            print(fig, savePath, '-djpeg');
```

```

        catch ME
            warning('MATLAB:%s', ME.identifier, ...
                ['Error saving image to "%s".\n' ...
                'Attempting to save in the current directory instead.\n' ...
                'Error details: %s'], savePath, ME.message);
            savePath = fullfile(pwd, [filename, '.jpeg']);
            print(fig, savePath, '-djpeg');
        end
    end
end

% Data
h = [0, 2300, 3000, 6100, 7900, 10000, 12000]; % Altitudes (m)
t_b = [100, 98.8, 95.1, 92.2, 90, 81.2, 75.6]; % Boiling temperatures (degrees)

% Perform linear regression
p = polyfit(h, t_b, 1);

% Extract slope and intercept
slope = p(1);
intercept = p(2);

% Display results
fprintf('The slope of the line (rate of change) is: %.4f\n', slope);
fprintf('The y-intercept of the line is: %.4f\n', intercept);

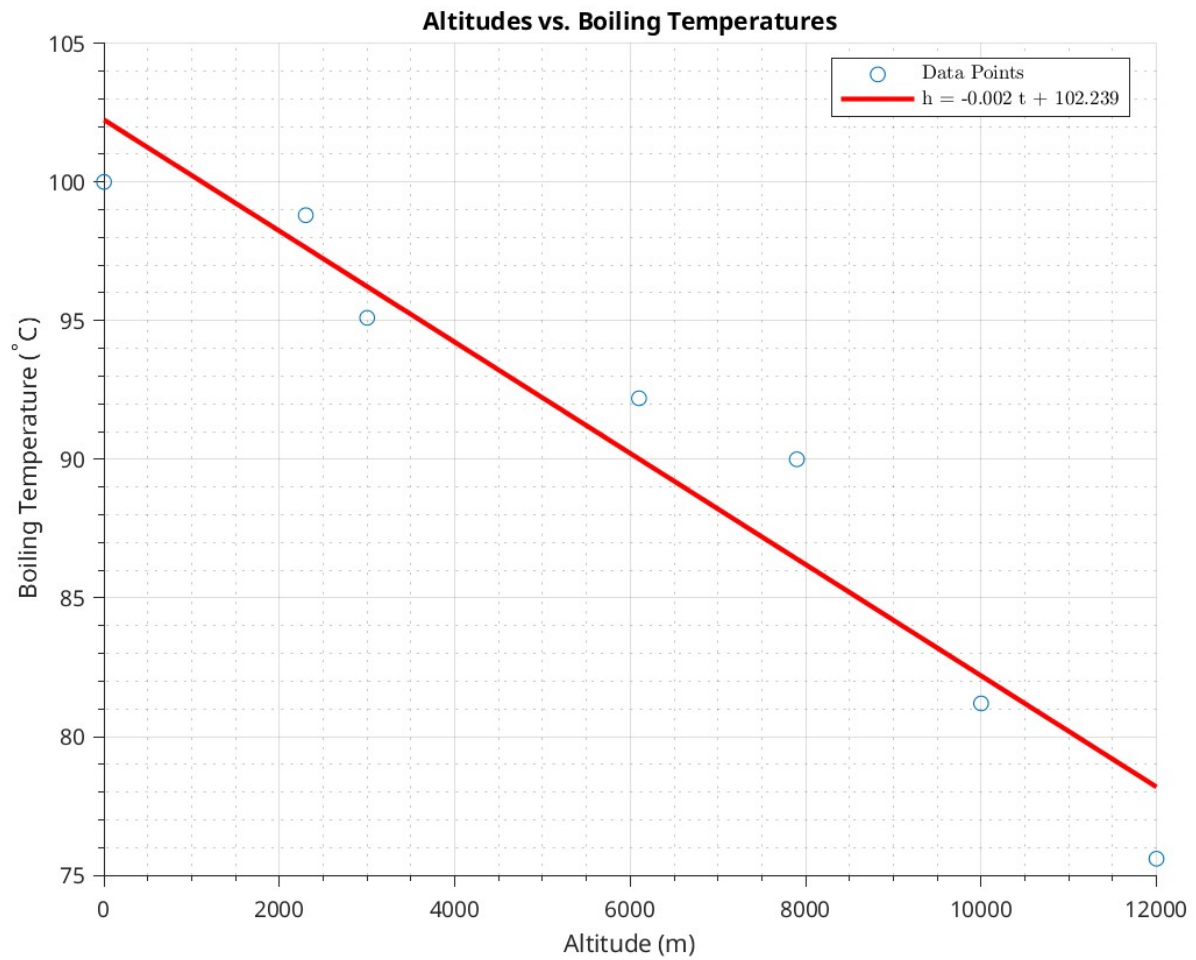
% Generate line of best fit
x_fit = linspace(min(h), max(h), 100);
y_fit = polyval(p, x_fit);

% Plot data and line of best fit
figure;
ax = gca;
fig = gcf;

scatter(h, t_b, 'o'); % Plot data points
hold on;
plot(x_fit, y_fit, '-r', 'LineWidth', 2); % Plot line of best fit

applyPlotFormatting( ...
    ax, ...
    fig, ...
    'Altitudes vs. Boiling Temperatures', ... % Title
    'Altitude (m)', ... % X-axis label
    'Boiling Temperature (^{\circ}C)', ... % Y-axis label
    {'Data Points', sprintf('h = %.3f t + %.3f', slope, intercept)}, ... % Legend
    entries
    'altitudes_vs_boiling_temperatures'); % Save filename

```



```
The slope of the line (rate of change) is: -0.0020  
The y-intercept of the line is: 102.2386
```

Hard to actually add input to this type (data type) of question.

B.2 Q2

The standard air density, \mathbf{D} , at different heights, \mathbf{h} , from sea level up to a height of 33 km is given below:

Height (km)	Density (kg/m ³)
0	1.2
3	0.91
6	0.66
9	0.47
12	0.31
15	0.19
18	0.12
21	0.075
24	0.046
27	0.029
30	0.018
33	0.011

Make the following four plots of the data points (\mathbf{D} as a function of \mathbf{h}) on the same figure:

1. Both axes with linear scale.
2. \mathbf{h} with log axis, \mathbf{D} with linear axis.
3. \mathbf{h} with linear axis, \mathbf{D} with log axis.
4. Both axes with log scale.

Based on the plots, choose a function (linear, power, exponential, or logarithmic) that best fits the data points and determine its coefficients. Plot the function and the points using linear axes.

main/q2.m

```
function applyPlotFormatting(ax, fig, titleText, xlabelText, ylabelText, legendText,
    filename)
    if ~isempty(titleText)
        title(ax, titleText);
    end
    if ~isempty(xlabelText)
        xlabel(ax, xlabelText);
    end
    if ~isempty(ylabelText)
        ylabel(ax, ylabelText);
    end
    fig.Color = 'w';
    ax.XGrid = 'on';
    ax.YGrid = 'on';
    ax.XMinorGrid = 'on';
    ax.YMinorGrid = 'on';
    ax.XMinorTick = 'on';
    ax.YMinorTick = 'on';
    ax.TickDir = 'out';
    ax.FontName = 'Calibri';
```

```

ax.FontSize = 9;
if ~isempty(legendText) && ~isequal(legendText, 'None')
    legend(ax, legendText, 'Location', 'best', 'Interpreter', 'latex');
end
if ~isempty(filename) && ~strcmp(filename, 'None')
    outputDir = 'graphs_images';
    savePath = fullfile(outputDir, [filename, '.jpeg']);
    if ~exist(outputDir, 'dir')
        [status, msg] = mkdir(outputDir);
        if ~status
            error('Unable to create the directory "%s": %s', outputDir, msg);
        end
    end
    try
        print(fig, savePath, '-djpeg');
    catch ME
        warning('MATLAB:%s', ME.identifier, ...
            ['Error saving image to "%s".\n' ...
            'Attempting to save in the current directory instead.\n' ...
            'Error details: %s'], savePath, ME.message);
        savePath = fullfile(pwd, [filename, '.jpeg']);
        print(fig, savePath, '-djpeg');
    end
end
end

% Data
h = [0 3 6 9 12 15 18 21 24 27 30 33];
D = [1.2 0.91 0.66 0.47 0.31 0.19 0.12 0.075 0.046 0.029 0.018 0.011];

% Combine the arrays into a 2xN matrix also known as zipping
data = [h; D];

% Create first figure
fig1 = figure;
fig1.Color = [1, 1, 1];

% Loop for subplots
for i = 1:4
    ax(i) = subplot(2, 2, i);
    L(i) = line(data(1, :), data(2, :), 'Parent', ax(i));

    % Configure line properties
    L(i).Color = [0.2, 0.7, 0];
    L(i).LineWidth = 3;
    L(i).LineStyle = '-';

    % Apply axis transformations based on subplot
    switch i
        case 1
            ax(i).XScale = 'linear';
            ax(i).YScale = 'linear';
            title = 'Linear Scale'; % Legend for linear axes
        case 2
            ax(i).XScale = 'log';
            ax(i).YScale = 'linear';
            title = 'Log-Linear Scale'; % Legend for log-linear axes
        case 3
            ax(i).XScale = 'linear';
            ax(i).YScale = 'log';
            title = 'Linear-Log Scale'; % Legend for linear-log axes
    end
end

```

```

        case 4
            ax(i).XScale = 'log';
            ax(i).YScale = 'log';
            title = 'Log-Log Scale'; % Legend for log-log axes
        end
        if i==5
            applyPlotFormatting(ax(i), fig1, title, '\ith \rm, km', '\itD\rm, kg/m^{3}', '', '
                ');
        else
            applyPlotFormatting(ax(i), fig1, title, '\ith \rm, km', '\itD\rm, kg/m^{3}', '', '
                'subplots');
        end
    end
end

% Adjust scales for specific subplots
ax(2).XScale = 'log';
ax(3).YScale = 'log';
ax(4).XScale = 'log';
ax(4).YScale = 'log';

% Fit data to exponential model
res = fit(data(1, :)', data(2, :)', 'exp1');

% fit requires one of the following:
% Curve Fitting Toolbox
% Model-Based Calibration Toolbox
% Predictive Maintenance Toolbox
% SimBiology
% Statistics and Machine Learning Toolbox

% Create second figure for the fitted curve
fig2 = figure;
fig2.Color = [1, 1, 1];

% Generate fitted line data
hplot = linspace(min(data(1, :)), max(data(1, :)), 1000);
Dplot = res.a * exp(res.b * hplot);

% Plot data and fitted curve
ax1 = subplot(1, 1, 1);

% Plot original data points
Y1 = line(data(1, :), data(2, :), 'Parent', ax1);
Y1.Color = 'r';
Y1.LineStyle = 'none';
Y1.LineWidth = 2;
Y1.Marker = 'o';
Y1.MarkerSize = 8;

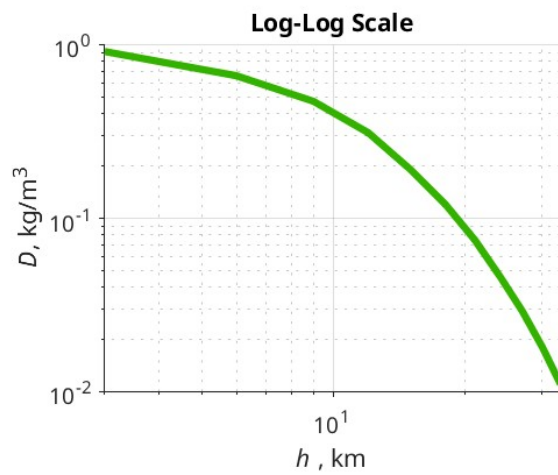
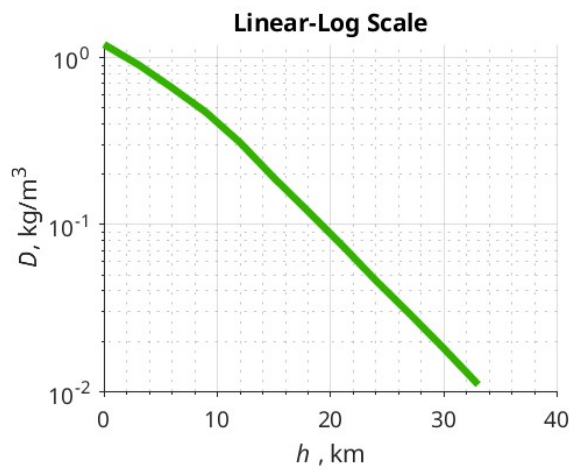
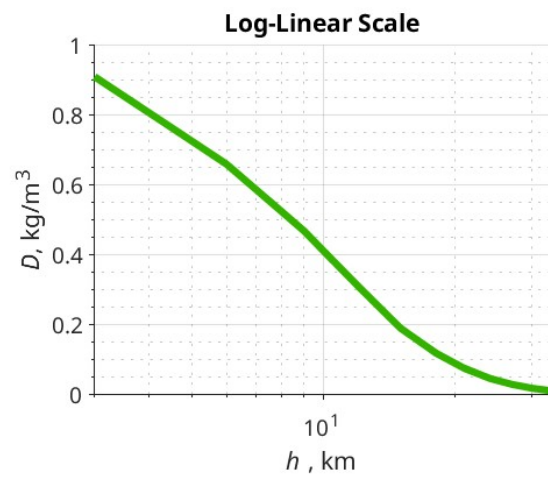
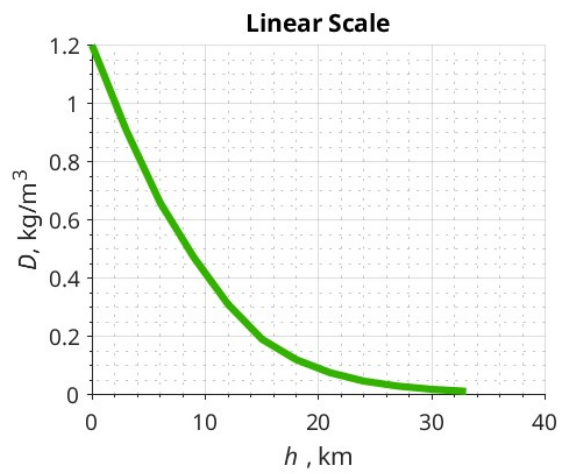
% Plot fitted curve
Y2 = line(hplot, Dplot, 'Parent', ax1);
Y2.Color = 'g';
Y2.LineStyle = '-';
Y2.LineWidth = 2;

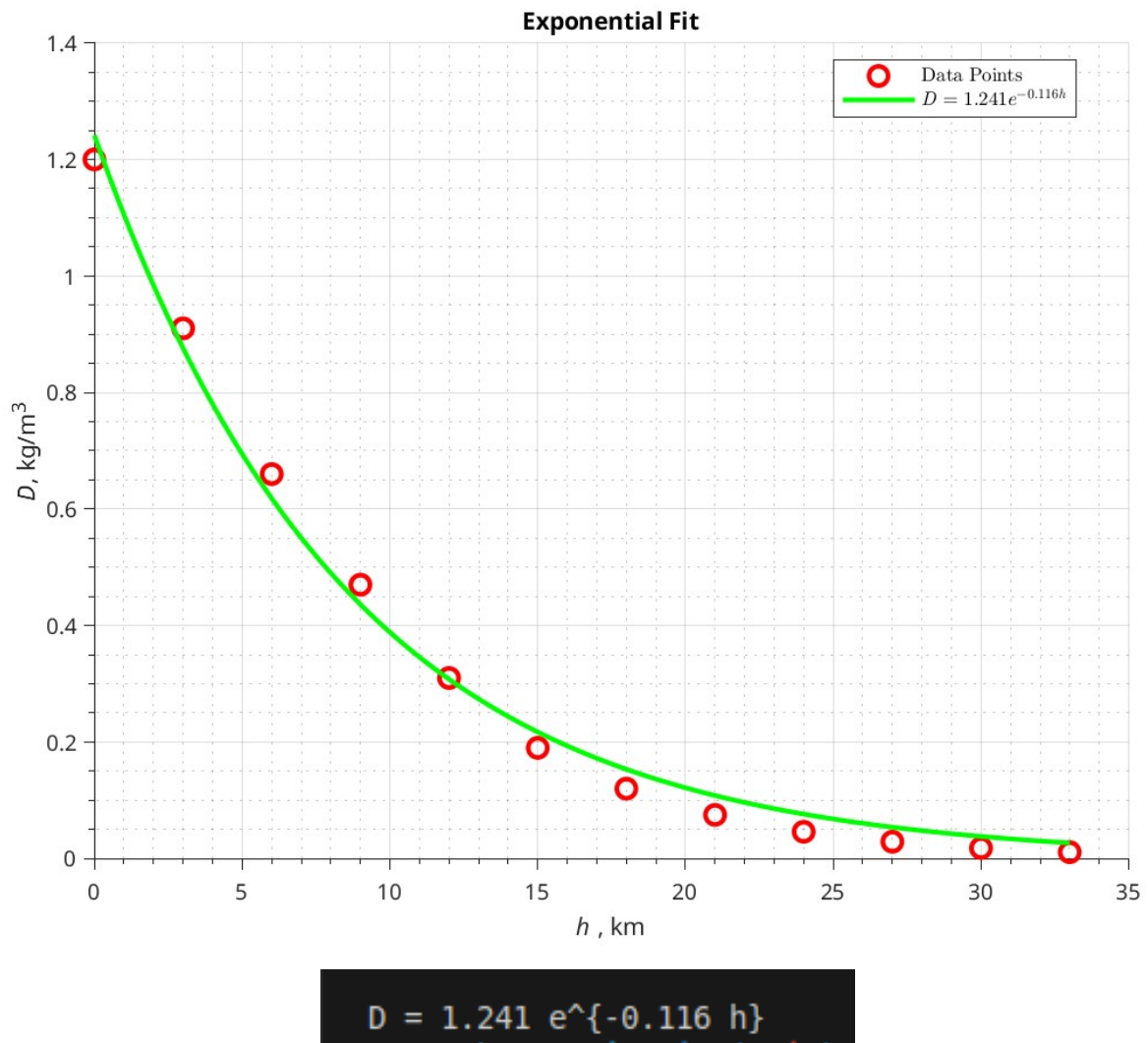
fprintf('D = %.3f e^{%.3f h}\n', res.a, res.b)

% Add title and labels using the applyPlotFormatting function
applyPlotFormatting(ax1, fig2, 'Exponential Fit', '\ith \rm, km', '\itD\rm, kg/m^{3}'

```

```
, ...
{'Data Points', sprintf('$\\mathit{D} = %.3f e^{%.3f \\mathit{h}}$', res.a, res.b
)}, 'fitted_curve');
```





Again hard to actually add input to this type (data type) of question.

B.3 Q3

Damped free vibrations can be modeled by a block of mass m attached to a spring and a dashpot as shown. From Newton's second law of motion, the displacement x of the mass as a function of time can be determined by solving the differential equation:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = 0$$

where k is the spring constant and c is the damping coefficient. If the mass is displaced from its equilibrium position and released, it will oscillate back and forth. The nature of the oscillations depends on the values of m , k , and c .

For the system shown, $m = 10 \text{ kg}$ and $k = 28 \text{ N/m}$. At $t = 0$, the mass is displaced to $x = 0.18 \text{ m}$ and released from rest. Derive expressions for the displacement $x(t)$ and velocity $v(t)$, considering the following cases:

1. $c = 3 \text{ Ns/m}$, $0 \leq t \leq 20 \text{ s}$,
2. $c = 50 \text{ Ns/m}$, $0 \leq t \leq 10 \text{ s}$.

For each case, plot $x(t)$ and $v(t)$ versus t (two plots on one page).

main/q3.m

```
function applyPlotFormatting(ax, fig, titleText, xlabelText, ylabelText, legendText,
    filename)
    if ~isempty(titleText)
        title(ax, titleText);
    end
    if ~isempty(xlabelText)
        xlabel(ax, xlabelText);
    end
    if ~isempty(ylabelText)
        ylabel(ax, ylabelText);
    end
    fig.Color = 'w';
    ax.XGrid = 'on';
    ax.YGrid = 'on';
    ax.XMinorGrid = 'on';
    ax.YMinorGrid = 'on';
    ax.XMinorTick = 'on';
    ax.YMinorTick = 'on';
    ax.TickDir = 'out';
    ax.FontName = 'Calibri';
    ax.FontSize = 9;
    if ~isempty(legendText) && ~isequal(legendText, 'None')
        legend(ax, legendText, 'Location', 'best', 'Interpreter', 'latex');
    end
    if ~isempty(filename) && ~strcmp(filename, 'None')
        outputDir = 'graphs_images';
        savePath = fullfile(outputDir, [filename, '.jpeg']);
        if ~exist(outputDir, 'dir')
            [status, msg] = mkdir(outputDir);
            if ~status
                error('Unable to create the directory "%s": %s', outputDir, msg);
            end
        end
    end
end
```

```

    end
    try
        print(fig, savePath, '-djpeg');
    catch ME
        warning('MATLAB:%s', ME.identifier, ...
            ['Error saving image to "%s".\n' ...
            'Attempting to save in the current directory instead.\n' ...
            'Error details: %s'], savePath, ME.message);
        savePath = fullfile(pwd, [filename, '.jpeg']);
        print(fig, savePath, '-djpeg');
    end
end
end

% Define symbolic variables for

syms x(t) c; % syms requires Symbolic Math Toolbox.

m = 10;
k = 28;
c1 = 3;
c2 = 50;
x0 = 0.18;
v0 = 0;

eq = m*diff(x,t,2) + c*diff(x,t) + k*x == 0;

vel = diff(x,t);

cond1 = x(0) == x0;
cond2 = vel(0) == v0;

Sol1 = dsolve(subs(eq,c, c1), [cond1, cond2]);
Sol2 = dsolve(subs(eq, c, c2), [cond1, cond2]);

tmin = 0;
tmax1 = 20;
tmax2 = 10;

fig = figure('Position', [100, 100, 800, 600]);

for i = 1:2
    ax(i) = subplot(2,1,i);
end

L1 = fplot(Sol1,[tmin, tmax1], 'Parent', ax(1));

L1.Color = [0.5, 0, 0.7];
L1.LineWidth = 3;
L1.LineStyle = '-';

L2 = fplot(Sol2,[tmin, tmax2], 'Parent', ax(2));
L2.Color = [0.5, 0, 0.7];
L2.LineWidth = 3;
L2.LineStyle = '-';

```

```
applyPlotFormatting(ax(1), fig, 'Displacement vs Time', 'Time, s', 'Displacement, m',  
                    '', '');  
applyPlotFormatting(ax(2), fig, 'Velocity vs Time', 'Time, s', 'Velocity, m/s', '', '  
displacement_velocity_plot');
```


B.4 Q4

The currents i_1 , i_2 , and i_3 in the circuit are described by the equation set:

$$\begin{pmatrix} 2R & -R & 0 \\ -R & 3R & -R \\ 0 & -R & 2R \end{pmatrix} \begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} v_1 \\ 0 \\ v_2 \end{pmatrix}$$

Here, v_1 and v_2 are applied voltages. The other two currents can be found as:

$$i_4 = i_1 - i_2, \quad i_5 = i_2 - i_3.$$

1. Use both the matrix inverse and division methods to solve for the currents in terms of R , v_1 , and v_2 .
2. Calculate the numerical values for the currents when $R = 1000\Omega$, $v_1 = 100\text{V}$, and $v_2 = 25\text{V}$.

main/q4.m

```
% Part A

% Define symbolic variables for resistance and voltages
syms R v_1 v_2 % syms requires Symbolic Math Toolbox.

% Create the node admittance matrix (NAM) for the circuit
NAM = [2*R -R 0; -R 3*R -R; 0 -R 2*R];
b = [v_1; 0; v_2];
I = sym('I', [3, 1]);

% Method 1: Solve system using division (Cramer's Rule)

% Calculate the determinant of the original matrix
delta_0 = det(NAM);
% Replace columns with b vector
for k = 1:3
    mNAM = NAM;
    mNAM(:, k) = b;
    delta_k = det(mNAM);
    I(k) = delta_k / delta_0;
end

% Add branch currents i4 and i5
I = [I; I(1)-I(2); I(2)-I(3)];

% Display results
disp('Currents using division (Cramer's Rule):')
for k = 1:length(I)
    fprintf('i%d = %s\n', k, char(I(k)))
end

% Method 2: Solve system using matrix inverse

% Use inverse function
NAM_inv = inv(NAM);
```

```

% Solution bro
I2 = NAM_inv*b;
% Add branch currents i4 and i5
I2 = [I2; I2(1)-I2(2); I2(2)-I2(3)];

% Display results from matrix inverse method
disp('Currents using matrix inverse:')
for k = 1:length(I2)
    fprintf('i%d = %s\n', k, char(I2(k)))
end

% Part B: Numerical computation with user-provided values

disp('-----')

% Get numerical values from user for resistance and voltages

Rs = input('Give me the R in ohms: ');
v1s = input('Give me the v_1 in volts: ');
v2s = input('Give me the v_2 in volts: ');

% Substitute numerical values into symbolic solutions and compute currents

I_num_c = vpa(subs(I, {R, v_1, v_2}, {Rs, v1s, v2s}), 6);
disp('Currents with given R and v_1, v_2 using Cramer''s Rule:')
for k = 1:5
    fprintf('i%d = %.6f\n', k, double(I_num_c(k)))
end

I_num_inv = vpa(subs(I2, {R, v_1, v_2}, {Rs, v1s, v2s}), 6);
disp('Currents with given R and v_1, v_2 using Matrix Inverse:')
for k = 1:length(I2)
    fprintf('i%d = %.6f\n', k, double(I_num_inv(k)))
end

```