Q4

a)

Create function c(x,t,C0,D,u,kd)

Input : distance x, time t, parameters C0,D,u,kd

Output : COD concentration c(x,t)

```
c = @(x,t,C0,D,u,kd)(C0./sqrt(4*pi*D.*t)).*exp(-kd*t).*exp(-((x-u*t
.^2)./(4*D*t));
```

b)

```
% initialize parameters:
C0 = 142; mg/L
D = 0.01, m^2/s
u = 0.2, m/s
kd = 0.001, 1/s
x = 0:10:1000, m
t = 0:60:7200, s (0h → 2h, step 1 min)

% Display for verification:
fprintf('Parameters initialized:\nC0 = %.1f mg/L, D=%.3f m^2/s, u=%.
1f m/s, k=%.4f 1/s\n', C0, D, u, k);
fprintf('x range: %.0f to %.0f m (%d points)\n', min(x), max(x), length(x))
fprintf('t range: %.1f to %.1f hours (%.0f to %.0f s, %d points)\n', min(t_hours
max(t_hours), min(t), max(t), length(t));
```

c)

```
% Compute COD concentration over meshgrid
[X, T] = meshgrid(x,t);
C = COD_func(C0, D, u, k, X, T);

% Create 3D surface plot
figure;
surf(X, T/3600, C);
shading interp;
colorbar;
title('3D surface Plot of COD concentration over distance and Time',...
```

```
'FontSize', 14);
Xlabel ('Distance x(m)', 'FontSize', 12);
Ylabel ('Time t(hours)', 'FontSize', 12);
zlabel ('COD concentration C(mg/L)', 'FontSize', 12);
grid on
view (45, 30)

% Add text annotation (e.g., peak location)
[~, idx] = max (C(:)); [iy, ix] = ind2sub(size (C), idx);
annotation ('textbox', [0.2, 0.7, 0.3, 0.1], 'String', sprintf ('Peak C...
= %.1f mg/L at x = %.0f m, t = %.2f h', C(iy, ix), x(1, ix), T(iy, 1)/3600),
'FitBoxToText', 'on', 'BackgroundColor', 'yellow', 'EdgeColor', 'black');
```

d)

```
t_fixed = 30 * 60; % 30 minutes in seconds
x_short = linspace (0, 200, 200); % Distance 0 to 200m
D_values = [0.001, 0.01, 1];
colors = {'r--', 'b-', 'g:--'};
legends = {'D = 0.001 m^2/s', 'D = 0.01 m^2/s', 'D = 1 m^2/s'};

% plot
figure;
hold on;
for i = 1: length (D_values)
    D_temp = D_values (i);
    C_temp = COD_func (C0, D_temp, u, k, x_short, t_fixed * ones (size...
(x_short)));
    plot (x_short, C_temp, colors{i}, 'LineWidth', 2);
end
hold off;

title ('COD Concentration Vs. Distance for Different Dispersion Coefficients
(t=30min)', 'FontSize', 14);
Xlabel ('Distance x(m)', 'FontSize', 12);
Ylabel ('COD Concentration C (mg/L)', 'FontSize', 12);
```

```
legend (legends, 'Location', 'best', 'Fontsize', 10);
grid on;
Xlim([0,200]);
Ylim([0,max(COD_func(Co,max(D_values),u,k,X_short,t_fixed*ones(
size(X_short)))*1.1]);
```

0)

```
t_slider = 1800;
X_slider = linspace(0,200,200);
Dmin = 0.0001; D_max = 2; D_init = 0.01;

fig = figure('Position',[100,100,600,500]);
ax = axes('Parent',fig,'Position',[0.1,0.2,0.8,0.7]);

% initial plot
C_init = COD_func(Co,D_init,u,k,X_slider,t_slider*ones(size(X_slider)));
plot(ax, X_slider, C_init, 'b', 'LineWidth',2);
title(ax, sprintf('COD concentration vs. Distance (D=%.4f m^2/s)', D_init),
'Fontsize', 14);
Xlabel(ax, 'Distance x(m)', 'Fontsize', 12);
ylabel(ax, 'COD concentration C(mg/L)', 'Fontsize',12);
grid(ax, 'on');
Xlim(ax, [0,200]);

% slider
Slider = uicontrol('Parent', fig, 'Style','slider','Position', [100,50,400,20],...
    'Min', D_min, 'Max', D_max, 'Value', D_init, 'SliderStep', [0.01,0.1],...
    'Callback', @updatePlot);

% Text label for current D
txt = uicontrol('Parent', fig, 'Style', 'text', 'Position', [50,75,100,20],...
    'String', sprintf('D=%.4f', D_init), 'Fontsize', 10);

function updatePlots(src,~)
D_current = get(src,'Value');
```

```
set(txt, 'String', sprintf('D = %.4f', D_current));

C_new = COD_func(C0, D_current, u, k, x_slider, t_slider * ones(size(x_
slider)));
cla(ax);
plot(ax, x_slider, C_new, 'b-', 'LineWidth', 2);
title(ax, sprintf('COD concentration vs. Distance (D = %.4f m^2/s, t = 30 min)', D_Curren
'FontSize', 14);
xlabel(ax, 'Distance x (m)', 'FontSize', 12);
ylabel(ax, 'COD concentration C (mg/L)', 'FontSize', 12);
grid(ax, 'on');
xlim(ax, [0, 200]);
ylim(ax, [0, max(C_new) * 1.1]);
drawnow; % update display
end
```

NO5

a, % Load data from Excel sheets
disciplinesNames = readtable('Engineering_Firm.XLSX', 'Sheet', 'disciplines');
performanceMetrics = readtable('Engineering_Firm.XLSX', 'Sheet', 'Performance_data');

b, % Loop through each discipline to check budget compliance
for i = 1:height(disciplinesNames)
disciplineName = disciplinesNames.A{i};
budget = performanceMetrics.Budget_USD(strcmp(performanceMetrics...
.Discipline, disciplineName))
actualcost = performanceMetrics.Actual_cost_USD(strcmp(performanceMetrics
.Discipline, disciplineName));
costOverrun = actualcost - budget;
If actualcost > budget
fprintf('Discipline: %s exceeded the budget by $%.2f.\n', discipline...
Name, costOverrun);
else
fprintf('Discipline: %s stayed within budget by $%.2f.\n', disciplineName,
abs(costOverrun));
end
end

C, % concatenate tables and calculate resourceutilizationRate
CombinedData = innerjoin(disciplinesNames, performanceMetrics, 'Keys', 'Discipline');
CombinedData.ResourceutilisationRate = CombinedData.Resources_used./...
CombinedData.Time_Token;

d % Save to Excel
Writetable(combinedData, 'Engineering_Firm.XLSX', 'Sheet', 'Resources ...
Utilization Rate');

e, Yes Version control systems like Git help track changes in code. The core concepts include:

Repositories: used for storage for project files

commits: snapshots of changes

Branches: independent lines of development.

To create a remote repository on GitHub:

1. Create a new repository on GitHub
2. Initialize a Git repository locally (git init).
3. Add files (git add .).
4. Commit changes (git commit -m "Initial commit").
5. Link local repo to GitHub (git remote add origin <URL>).
6. Push to GitHub (git push -u origin master/main).