Scilab for Chemical Engineers

TSEC - Online Certificate Course

SAMPLE PROBLEMS

Aditya Ganesh Date: 02 - 02 - 2025

1 Vectors, Matrices, Polynomials

1.1 Heat Exchange Network

You have a matrix that represents the temperature of different streams (4 in number) in a heat exchange network. The matrix is:

$$\texttt{Temperature\_Matrix} = \begin{bmatrix} 100 & 200 & 300 & 400 \\ 150 & 250 & 350 & 450 \\ 200 & 300 & 400 & 500 \end{bmatrix}$$

a) Extract the sub-matrix representing the temperatures of the first two streams.
b) Calculate the average temperature of each stream.
c) If the streams need to be rearranged in reverse order, how would the temperature matrix look like?

```
// Given Temperature_Matrix
Temperature_Matrix = [100 200 300 400; 150 250 350 450; 200
   300 400 500];

// (a) Extract the sub-matrix representing the temperatures
   of the first two streams.
Sub_Matrix = Temperature_Matrix(:, 1:2);

// (b) Calculate the average temperature of each stream.
Average_Temperature = mean(Temperature_Matrix, "r");
```

```

// (c) Reverse the order of the streams.
Reversed_Matrix = Temperature_Matrix(:, $:-1:1);
```

1.2 Reactor Feed Composition

Consider a matrix representing the composition of different feed streams entering a reactor: [Components = columns; Streams = rows]

$$\texttt{Composition\_Matrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.4 & 0.3 & 0.3 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$

a) Extract the composition of the second feed stream.
b) Determine the total composition of each component across all feed streams.
c) Add a new feed stream with composition $[0.3, 0.3, 0.4]$ to the matrix.

```
// Given Composition_Matrix
Composition_Matrix = [0.2 0.3 0.5; 0.4 0.3 0.3; 0.1 0.2 0.7];

// (a) Extract the composition of the second feed stream.
Second_Stream_Composition = Composition_Matrix(2, :);

// (b) Determine the total composition of each component.
Total_Composition = sum(Composition_Matrix, "r");

// (c) Add a new feed stream.
New_Feed_Stream = [0.3 0.3 0.4];
Updated_Composition_Matrix = [Composition_Matrix;
   New_Feed_Stream];
```

1.3 Reaction Rate Constants

You have the following polynomial equation representing the rate constants of a chemical reaction:

$$k(T) = 2T^2 + 3T + 1$$

a) Calculate the rate constant for $T = 300$ K.
b) Generate a vector of rate constants for $T = [250, 300, 350, 400]$.
c) Find the derivative of the polynomial and evaluate it at $T = 300$ K.

```
// Given polynomial equation
k_T = poly([2 3 1], 'x', 'coeff');

// (a) Calculate the rate constant for T = 300 K.
k_300 = horner(k_T, 300);

// (b) Generate a vector of rate constants.
T = [250 300 350 400];
Rate_Constants = horner(k_T, T);

// (c) Find the derivative and evaluate at T = 300 K.
k_T_derivative = derivat(k_T);
k_300_derivative = horner(k_T_derivative, 300);
```

1.4 Mass Balance in Mixing Tank

A mixing tank has three inlet streams [columns] with the following flow rates (in kg/h):

$$\texttt{Flow\_Matrix} = \begin{bmatrix} 10 & 20 & 30 \\ 15 & 25 & 35 \\ 20 & 30 & 40 \end{bmatrix}$$

a) Extract the flow rates of the first inlet stream.
b) Calculate the total flow rate for each inlet stream.
c) If the flow rates of the inlet streams are doubled, how would the flow matrix look like?

```
// Given Flow_Matrix
Flow_Matrix = [10 20 30; 15 25 35; 20 30 40];

// (a) Extract the flow rates of the first inlet stream.
First_Inlet_Stream = Flow_Matrix(:, 1);

// (b) Calculate the total flow rate for each inlet stream.
Total_Flow_Rate = sum(Flow_Matrix, "r");
```

```

// (c) Double the flow rates.
Doubled_Flow_Matrix = 2 * Flow_Matrix;
```

1.5 System of Linear Equations in Chemical Reactions

Solve the following system of linear equations representing the stoichiometry of a chemical reaction:

$$2A + B = 3C$$
$$A + 3B = 2C$$

a) Formulate the system of linear equations in matrix form.
b) Solve the system using matrix operations.
c) Extract the sub-matrix representing the coefficients of A and B.

```
// (a) Formulate in matrix form
A = [2 1; 1 3];
C = [3; 2];

// (b) Solve the system using matrix operations.
Solution = A \ C;

// (c) Extract the sub-matrix representing coefficients of A
   and B.
Sub_Matrix_AB = A(:, 1:2);
```

1.6 Distillation Column Design

A distillation column has multiple stages [rows], and you have a matrix representing the concentration of a component at each stage:

$$\texttt{Concentration\_Matrix} = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 \end{bmatrix}$$

a) Extract the sub-matrix representing the concentration at the first two stages.
b) Calculate the average concentration at each stage.
c) Transpose the matrix and interpret the result.

```
// Given Concentration_Matrix
Concentration_Matrix = [0.1 0.2 0.3; 0.4 0.5 0.6; 0.7 0.8
   0.9];

// (a) Extract sub-matrix for first two stages.
Sub_Matrix_Stages = Concentration_Matrix(1:2, :);

// (b) Calculate average concentration at each stage.
Average_Concentration = mean(Concentration_Matrix, "c");

// (c) Transpose the matrix.
Transposed_Matrix = Concentration_Matrix';
```

1.7 Chemical Kinetics

Consider a vector representing the concentrations of a reactant over time:

$$\texttt{Concentration\_Vector} = [0.1, 0.15, 0.2, 0.25, 0.3]$$

a) Slice the vector to extract the concentrations at the first three time points.
b) Calculate the rate of change of concentration between each time point.
c) Plot the concentration vs. time.

```
// Given Concentration_Vector
Concentration_Vector = [0.1 0.15 0.2 0.25 0.3];

```

```
// (a) Slice the vector.
Sliced_Vector = Concentration_Vector(1:3);

// (b) Calculate rate of change.
Rate_of_Change = diff(Concentration_Vector);

// (c) Plot concentration vs. time.
plot(1:length(Concentration_Vector), Concentration_Vector, '-
    o');
xlabel('Time');
ylabel('Concentration');
title('Concentration vs. Time');
```

1.8 Gas Mixture Composition

You have a matrix representing the composition of a gas mixture in different tanks [rows]:

$$\texttt{Gas\_Composition\_Matrix} = \begin{bmatrix} 0.2 & 0.3 & 0.5 \\ 0.4 & 0.3 & 0.3 \\ 0.6 & 0.2 & 0.2 \end{bmatrix}$$

a) Extract the composition of the gases in the second tank.
b) Determine the total composition of each gas component across all tanks.
c) Normalize the composition of the gases in each tank.

```
// Given Gas_Composition_Matrix
Gas_Composition_Matrix = [0.2 0.3 0.5; 0.4 0.3 0.3; 0.6 0.2
    0.2];

// (a) Extract composition of gases in second tank.
Second_Tank_Composition = Gas_Composition_Matrix(2, :);

// (b) Determine total composition of each gas.
Total_Gas_Composition = sum(Gas_Composition_Matrix, "r");

// (c) Normalize the composition.
Row_Sum = sum(Gas_Composition_Matrix, 2);
```

```
Normalized_Composition = Gas_Composition_Matrix ./ Row_Sum(:,
    ones(1, size(Gas_Composition_Matrix, 2)));
```

1.9 Optimization of Reaction Conditions

You have a polynomial equation representing the yield of a chemical reaction as a function of temperature:

$$Y(T) = -0.01T^2 + 1.2T - 5$$

a) Calculate the yield for $T = 100$ °C.
b) Generate a vector of yields for $T = [80, 90, 100, 110, 120]$.
c) Find the temperature at which the yield is maximized.

```
// Given polynomial equation
Y_T = poly([-0.01 1.2 -5], 'x', 'coeff');

// (a) Calculate the yield for T = 100 $^\circ$C.
Yield_100 = horner(Y_T, 100);

// (b) Generate a vector of yields.
Temperatures = [80 90 100 110 120];
Yields = horner(Y_T, Temperatures);

// (c) Find temperature at which yield is maximized.
Y_T_derivative = derivat(Y_T);
Max_Yield_Temperature = roots(Y_T_derivative);
```

1.10 Wastewater Treatment

A wastewater treatment plant has multiple treatment stages [rows], and you have a matrix representing the pollutant concentration at each stage:

$$\texttt{Pollutant\_Matrix} = \begin{bmatrix} 50 & 40 & 30 \\ 60 & 50 & 40 \\ 70 & 60 & 50 \end{bmatrix}$$

a) Extract the sub-matrix representing the pollutant concentration at the first two stages.
b) Calculate the average pollutant concentration at each stage.
c) Determine the reduction in pollutant concentration from the first to the last stage.

```
// Given Pollutant_Matrix
Pollutant_Matrix = [50 40 30; 60 50 40; 70 60 50];

// (a) Extract sub-matrix for first two stages.
Sub_Matrix_Stages = Pollutant_Matrix(1:2, :);

// (b) Calculate average pollutant concentration.
Average_Pollutant_Concentration = mean(Pollutant_Matrix, "c")
    ;

// (c) Determine reduction from first to last stage.
Reduction = Pollutant_Matrix($,:) - Pollutant_Matrix(1,:);
```

2 Loops, Conditionals, Cases

2.1 Temperature Distribution in a Reactor

A chemical reactor is divided into 10 equally spaced segments. The temperature at each segment needs to be calculated based on the following formula:

$$T_i = T_{i-1} + \Delta T$$

where T_i is the temperature at segment i and $\Delta T(= 10)$ is a constant temperature difference. The temperature at the first segment is given ($T_1 = 100$). Write a Scilab script using a `for` loop to compute the temperature at each segment.
Hint: Use a `for` loop to iterate through the segments and compute the temperature.

```
// Initial temperature at the first segment
T1 = 100;
delta_T = 10;
n = 10;

// Initialize the temperature array
```

```
T = zeros(1, n+1);
T(1) = T1;

// Calculate the temperature at each segment
for i = 2:n+1
    T(i) = T(i-1) + delta_T;
end

// Display the temperature distribution
disp(T);
```

2.2 Concentration Profile in a Plug Flow Reactor

In a plug flow reactor, the concentration of a reactant changes along the length of the reactor. The concentration at each point can be calculated using the following equation:

$$C_i = C_{i-1}(1 - k\Delta x)$$

where C_i is the concentration at point i, k is the reaction rate constant $(= 0.1)$, and Δx is the distance between points $(= 0.01)$. The initial concentration (C_1) is given $(= 1)$. The total domain length $L = 1$. Write a Scilab script using a `while` loop to calculate the concentration profile along the reactor.
Hint: Use a `while` loop to compute the concentration at each point until the end of the reactor is reached.

```
// Initial concentration and parameters
// Input parameters
C1 = 1;
L = 1;
dx = 0.01;
k = 0.1;
Nx = L/dx;

oneminuskdx = 1 - k*dx;

// Initialize the concentration array
C = zeros(Nx);
```

```
xs = zeros(Nx);
C(1) = C1;
i = 2

// Either this loop
while i <= Nx
    C(i) = C(i-1)*oneminuskdx;
    xs(i) = xs(i-1) + dx;
    i = i + 1;
end

// Or this loop for calculating the concentration profile
   along the reactor
while %t
    C(i) = C(i-1)*oneminuskdx;
    xs(i) = xs(i-1) + dx;
    i = i + 1;
    if i > Nx then
        break;
    end
end

// Plotting the concentration profile
plot(xs,C)
```

2.3 Batch Reactor Simulation with Nested Loops

Simulate the concentration of a reactant in a batch reactor over time for different reaction rate constants. The concentration change over time is given by:

$$C(t + \Delta t) = C(t) - k \cdot C(t) \cdot \Delta t$$

where Δt is the time step $(= 0.01)$ and k is the reaction rate constant $(= [0.1, 0.2, 0.3])$. Write a Scilab script using nested loops to calculate the concentration over time for different values of k.
Hint: Use an outer `for` loop to iterate over different values of k, and an inner `for` loop

to simulate the concentration change over time.

```
// Parameters
C1 = 1;
k = [0.1, 0.2, 0.3];
dt = 0.01;
T = 10.0;
Nt = T/dt;

// Initialize the concentration matrix
C = zeros(Nt+1, length(k));
C(1,:) = C1*ones(1,length(k));

// Simulate the concentration over time for different k
    values
for j = 1:length(k)
    for i = 2:Nt+1
        C(i,j) = C(i-1,j) - k(j)*C(i-1,j)*dt
    end
end
```

2.4 Cooling of a Tank with Conditional Statements

A tank is being cooled by circulating coolant. The temperature of the tank (T) decreases at a rate dependent on the coolant flow rate (F). The temperature decrease can be modeled as:

$$T(t + \Delta t) = T(t) - F \cdot (T(t) - T_{\text{coolant}}) \cdot \Delta t$$

If the temperature is high ($T > 60$), use a different cooling rate constant. Write a Scilab script using conditionals and loops to simulate the cooling process over time. ($T_i = 100; T_{\text{coolant}} = 25, F_{\text{high}} = 6; F_{\text{low}} = 3; \text{Time} = 10;\ \Delta t = 0.01$)
Hint: Use an `if-else` statement inside a loop to handle different cooling rate constants based on the temperature.

```
// Parameters
T_initial = 100;
T_coolant = 25;
F_high = 6;
```

```
F_low = 3;
delta_t = 0.01;
time = 0:delta_t:10;

// Initialize the temperature array
T = zeros(1, length(time));
T(1) = T_initial;

// Simulate the cooling process
for i = 2:length(time)
    if T > 60 then
        F = F_high;
    else
        F = F_low;
    end
    T(i) = T(i-1) - F * (T(i-1) - T_coolant) * delta_t;
end

% // Display the temperature profile
% disp(T);
%
```

2.5 Reaction in a CSTR with Case Statements

In a Continuous Stirred Tank Reactor (CSTR), the concentration of the reactant is affected by the feed concentration, flow rate, and reaction rate constant. Write a Scilab script using a `select-case` statement to handle different scenarios where the feed concentration or flow rate changes at different time intervals. The concentration change over time is given by:

$$C(t + \Delta t) = C(t) - k \cdot C(t) \cdot \Delta t$$

$C_0 = 1; k = 0.1; F = 2; \Delta t = 0.01; \text{Time} = 10$. For the first 10 timesteps $F = 1$, the following 40 timesteps $F = 3$ and the remaining timesteps $F = 2$

Hint: Use a `select-case` statement to handle different cases and a loop to simulate the concentration changes over time.

```
// Parameters
```

```
C0 = 1;
k = 0.1;
F = 2;
delta_t = 0.01;
time = 0:delta_t:10;

// Initialize the concentration array
C = zeros(1, length(time));
C(1) = C0;

// Simulate the concentration changes over time with case
    statements
for i = 2:length(time)
    select i
    case 1:11
        F = 1; // Different flow rate for the first interval
    case 12:51
        F = 3; // Different flow rate for the second interval
    else
        F = 2; // Default flow rate
    end
    C(i) = C(i-1) - k * C(i-1) * delta_t;
end

// Display the concentration profile
disp(C);
```