Problem1

A lab group is combining different amounts of 3 salts to obtain various weights of the final product. They combine 2 mol of salt A, 3 mol of salt B, and 1 mol of salt C to make 312.1g of the final product. Then, they combine 1 mol of salt A, 2 mol of salt B, and 1 mol of salt C to make 216.7g of the final product. Finally, the lab group combines 1 mol of salt A, 1 mol of salt B, and 2 mol os salt C to make 264g of the final product. Find the molar mass of each salt.

Solution

The equations are:

$$\begin{aligned} 2a + 3b + c &= 312.1 \\ a + 2b + c &= 216.7 \\ a + b + 2c &= 264 \end{aligned}$$

a) Gauss Jordan

$$\begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 1 & 2 & 1 & | & 216.7 \\ 1 & 1 & 2 & | & 264 \end{bmatrix}$$

$$R_2 - R_3 \begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 0 & 1 & -1 & | & -47.3 \\ 1 & 1 & 2 & | & 264 \end{bmatrix}$$

$$R_1 - 2R_3 \begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 0 & 1 & -1 & | & -47.3 \\ 0 & 1 & -3 & | & -215.9 \end{bmatrix}$$

$$R_2 - R_3 \begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 0 & 1 & -1 & | & -47.3 \\ 0 & 0 & 2 & | & 168.6 \end{bmatrix}$$

$$R_1 - 3R_2 \left[\begin{array}{ccc|c} 2 & 0 & 4 & 454 \\ 0 & 1 & -1 & -47.3 \\ 0 & 0 & 2 & 168.6 \end{array}\right]$$

$$R_1 - 2R_3 \left[\begin{array}{ccc|c} 2 & 0 & 0 & 116.8 \\ 0 & 1 & -1 & -47.3 \\ 0 & 0 & 2 & 168.6 \end{array}\right]$$

Divide through R_1 and R_3 by 2

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 58.4 \\ 0 & 1 & -1 & -47.3 \\ 0 & 0 & 1 & 84.3 \end{array}\right]$$

$$R_2 + R_3 \left[\begin{array}{ccc|c} 1 & 0 & 0 & 58.4 \\ 0 & 1 & 0 & 37 \\ 0 & 0 & 1 & 84.3 \end{array}\right]$$

a = 58.4

b = 37

c = 84.3

b) Gaussian Elimination

Our goal is to get the matrix to row echelon form

$$\left[\begin{array}{ccc|c} 2 & 3 & 1 & 312.1 \\ 1 & 2 & 1 & 216.7 \\ 1 & 1 & 2 & 264 \end{array}\right]$$

$$R_2 - R_3 \left[\begin{array}{ccc|c} 2 & 3 & 1 & 312.1 \\ 0 & 1 & -1 & -47.3 \\ 1 & 1 & 2 & 264 \end{array}\right]$$

$$R_1 - 2R_3 \begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 0 & 1 & -1 & | & -47.3 \\ 0 & 1 & -3 & | & -215.9 \end{bmatrix}$$

$$R_2 - R_3 \begin{bmatrix} 2 & 3 & 1 & | & 312.1 \\ 0 & 1 & -1 & | & -47.3 \\ 0 & 0 & 2 & | & 168.6 \end{bmatrix}$$

From the bottom:

$$\frac{2c}{2} = \frac{168.6}{2}$$
$$c = 84.3$$

$$b - c = -47.3$$
$$b - 84.3 = -47.3$$
$$b = 37$$

$$2a + 3b + c = 312.1$$
$$2a + 111 + 84.3 = 312.1$$
$$a = 58.4$$

```
from sympy import Matrix, symbols, solve

def gauss_jordan_elimination(matrix):
    # Convert the matrix to the reduced row-echelon form using Gauss-Jordan elimination
    reduced_matrix, pivot_columns = matrix.rref()

    return reduced_matrix

def print_solutions(reduced_matrix):
    # Extract the solutions from the reduced row-echelon form
    variables = symbols('a b c')
    num_variables = len(variables)

    for row in reduced_matrix.tolist():
        # Check if the row is all zeros except the last element
        if sum(row[:-1]) == 0 and row[-1] != 0:
            print("No solution exists.")
            return

    for row in reduced_matrix.tolist():
        # Find the leading entry (pivot) in the row
        pivot_column = next((i for i, val in enumerate(row[:-1]) if val != 0), None)

        if pivot_column is not None:
            # Extract the variable and its corresponding value
            variable = variables[pivot_column]
            value = round(row[-1], 1)

            # Print the individual solution
            print(f"{variable}: {value}")
        else:
            # If pivot_column is None, it means the row is all zeros
            print("Infinite solutions exist.")
            return

# Define your matrix as a list of lists
matrix_data = [
    [2, 3, 1, 312.1],
    [1, 2, 1, 216.7],
    [1, 1, 2, 264]
```

```
]

# Create a sympy Matrix from the matrix data
matrix = Matrix(matrix_data)

# Perform Gauss-Jordan elimination
result_matrix = gauss_jordan_elimination(matrix)

# Print the result
print("Original Matrix:")
print(matrix)
print("\nReduced Row-Echelon Form:")
print(result_matrix)

# Print individual solutions
print("\nIndividual Solutions:")
print_solutions(result_matrix)
```

```
Original Matrix:
Matrix([[2, 3, 1, 312.100000000000], [1, 2, 1, 216.700000000000], [1, 1, 2, 264]])

Reduced Row-Echelon Form:
Matrix([[1, 0, 0, 58.4000000000001], [0, 1, 0, 37.0000000000000], [0, 0, 1, 84.3000000000000]])

Individual Solutions:
a: 58.4
b: 37.0000000000000
c: 84.3
```

Problem2

A chemistry student is trrying to make $100ml$ of a 26% acid solution from a 10% solution, a 20% solution, and a 40% solution. Unfortunately, the lab is out of 20%, so the student uses a 25% solution, and ends up with $100ml$ of 28% solution. What volume of each solution did the student use?

Solution

The following are the equations derived from the question:

$$0.1x + 0.2y + 0.4z = 26$$
$$0.1x + 0.25y + 0.4z = 28$$
$$x + y + z = 100$$

a) Gauss Jordan Elimination

$$\left[\begin{array}{ccc|c} 0.1 & 0.2 & 0.4 & 26 \\ 0.1 & 0.25 & 0.4 & 28 \\ 1 & 1 & 1 & 100 \end{array}\right]$$

$$R_1 - R_2 \left[\begin{array}{ccc|c} 0.1 & 0.2 & 0.4 & 26 \\ 0 & -0.05 & 0 & -2 \\ 1 & 1 & 1 & 100 \end{array}\right]$$

$$R_1 - 0.1R_3 \left[\begin{array}{ccc|c} 0.1 & 0.2 & 0.4 & 26 \\ 0 & -0.05 & 0 & -2 \\ 0 & 0.1 & 0.3 & 16 \end{array}\right]$$

$$2R_2 + R_3 \left[\begin{array}{ccc|c} 0.1 & 0.2 & 0.4 & 26 \\ 0 & -0.05 & 0 & -2 \\ 0 & 0 & 0.3 & 12 \end{array}\right]$$

$$R_1 + 4R_2 \left[\begin{array}{ccc|c} 0.1 & 0 & 0.4 & 18 \\ 0 & -0.05 & 0 & -2 \\ 0 & 0 & 0.3 & 12 \end{array}\right]$$

$$3R_1 - 4R_3 \left[\begin{array}{ccc|c} 0.3 & 0 & 0 & 6 \\ 0 & -0.05 & 0 & -2 \\ 0 & 0 & 0.3 & 12 \end{array}\right]$$

Divide through the rows by the value of the leading term:

$$3R_1 - 4R_3 \begin{bmatrix} 1 & 0 & 0 & | & 20 \\ 0 & 1 & 0 & | & 40 \\ 0 & 0 & 1 & | & 40 \end{bmatrix}$$

b) Gaussian Method

The goal is to get the matrix to row echelon form using the steps below:

$$\begin{bmatrix} 0.1 & 0.2 & 0.4 & | & 26 \\ 0.1 & 0.25 & 0.4 & | & 28 \\ 1 & 1 & 1 & | & 100 \end{bmatrix}$$

$$R_1 - R_2 \begin{bmatrix} 0.1 & 0.2 & 0.4 & | & 26 \\ 0 & -0.05 & 0 & | & -2 \\ 1 & 1 & 1 & | & 100 \end{bmatrix}$$

$$R_1 - 0.1R_3 \begin{bmatrix} 0.1 & 0.2 & 0.4 & | & 26 \\ 0 & -0.05 & 0 & | & -2 \\ 0 & 0.1 & 0.3 & | & 16 \end{bmatrix}$$

$$2R_2 + R_3 \begin{bmatrix} 0.1 & 0.2 & 0.4 & | & 26 \\ 0 & -0.05 & 0 & | & -2 \\ 0 & 0 & 0.3 & | & 12 \end{bmatrix}$$

From the bottom:

$$\frac{0.3z}{0.3} = \frac{12}{3}$$
$$z = 40$$

$$\frac{-0.05y}{-0.05} = \frac{-2}{-0.05}$$
$$y = 40$$

$$0.1x + 0.2y + 0.4z = 26$$

Substitute the values of y and z:

$$0.1x + 0.2(40) + 0.4(40) = 26$$
$$\frac{0.1x}{0.1} = \frac{2}{0.1}$$
$$x = 20$$

```
from sympy import Matrix, symbols, solve

def gauss_jordan_elimination(matrix):
    # Convert the matrix to the reduced row-echelon form using Gauss-Jordan elimination
    reduced_matrix, pivot_columns = matrix.rref()

    return reduced_matrix

def print_solutions(reduced_matrix):
    # Extract the solutions from the reduced row-echelon form
    variables = symbols('x y z')
    num_variables = len(variables)

    for row in reduced_matrix.tolist():
        # Check if the row is all zeros except the last element
        if sum(row[:-1]) == 0 and row[-1] != 0:
            print("No solution exists.")
            return

    for row in reduced_matrix.tolist():
        # Find the leading entry (pivot) in the row
        pivot_column = next((i for i, val in enumerate(row[:-1]) if val != 0), None)

        if pivot_column is not None:
            # Extract the variable and its corresponding value
            variable = variables[pivot_column]
            value = round(row[-1], 0)

            # Print the individual solution
            print(f"{variable}: {value}")
        else:
```

```
        else:
            # If pivot_column is None, it means the row is all zeros
            print("Infinite solutions exist.")
            return

# Define your matrix as a list of lists
matrix_data = [
    [0.1, 0.2, 0.4, 26],
    [0.1, 0.25, 0.4, 28],
    [1, 1, 1, 100]
]

# Create a sympy Matrix from the matrix data
matrix = Matrix(matrix_data)

# Perform Gauss-Jordan elimination
result_matrix = gauss_jordan_elimination(matrix)

# Print the result
print("Original Matrix:")
print(matrix)
print("\nReduced Row-Echelon Form:")
print(result_matrix)

# Print individual solutions
print("\nIndividual Solutions:")
print_solutions(result_matrix)
```

```
Original Matrix:
Matrix([[0.100000000000000, 0.200000000000000, 0.400000000000000, 26], [0.100000000000000, 0.250000000000000, 0.400000000000000, 28], [1, 1, 1

Reduced Row-Echelon Form:
Matrix([[1, 0, 0, 20.0000000000000], [0, 1, 0, 40.0000000000000], [0, 0, 1, 40.0000000000000]])

Individual Solutions:
x: 20.0000000000000
y: 40.0000000000000
z: 40.0000000000000
```

Problem3

$$x\frac{df(x)}{dx} + f(x) - f(x)^2 = 0$$

```
from sympy import Function, dsolve, Eq, Derivative, symbols

# Define the variable and the function
x = symbols('x')
f = Function('f')

# Define the ordinary differential equation
ode = Eq(x * Derivative(f(x), x) + f(x) - f(x)**2, 0)

# Solve the ODE
solution = dsolve(ode)

# Print the solution
print("Solution:")
print(solution)
```

```
Solution:
Eq(f(x), -1/(C1*x - 1))
```