Balancing Chemical Equations: A Study in Linear Algebra

Stacie Barbarick and Alex Langfield
Red Rocks Community College
Linear Algebra, Fall 2019
Instructor: Adam Forland

29 September 2019

1 Introduction

A chemical equation is an expression that represents the reactants (substances that are altered) and the products (new substances that are created) of a chemical reaction. The Law of the Conservation of Mass dictates that mass of reactants and products must be equal and therefore the number of atoms per element remain constant from one side of the equation to the other.
To obtain this balanced chemical equation, whole number integers are calculated and applied to products and reactants, so these atom ratios equalize. Balancing chemical equations is a fundamental skill in any chemical laboratory application since correct ratios of reagents are necessary to obtain desired products.

Unfortunately, the traditional "guess and check" or algebra-based system that many chemists employ to balance such equations can become incredible difficult and time consuming as chemical equations become more complex. Until the early 1980s, many chemistry educators felt that it was impossible to balance the equations for some reactions by inspection or, possibly, or the arithmetic of the standard methods was "beyond the capabilities of most students". [1]
A different methodology for balancing these complex equations was proposed in 1978 by E.V. Krishnamurthy which, "consists in solving exactly for the rational (non-trivial) solutions of a system of linear, homogeneous algebraic equations whose coefficient matrix (called the reaction matrix) is singular, by using the concept of the generalized inverse of a matrix." [2] In fact, this matrix-based approach, when applied correctly, is capable of balancing chemical equations faster, and with more parameters than traditional methods. [3] Once a general program for the linear algebra is coded, anyone can simple input the elements of the chemical equation and quickly solve for the balancing coefficients.

2 An Example

A simple example of balancing a chemical equation using linear algebra is as follows. Take the thermal decomposition of potassium chlorate to produce potassium chloride and potassium perchlorate.
Source: https://chemiday.com/en/reaction/3-1-0-1127equation:

$$(x_1)KClO_3 \longrightarrow (x_2)KClO_4 + (x_3)KCl \quad (1)$$

We can assign each element to a row and each compound in the equation has an associated vector, denoting the weights of each element within that compound. Like this:

$$KClO_3 \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \longrightarrow KClO_4 \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix} + KCl \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad (2)$$

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} \longrightarrow x_2 \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix} + x_3 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad (3)$$

Now in linear combination form, all the terms can be moved to the left and set equal to zero:

$$x_1 \begin{bmatrix} 1 \\ 1 \\ 3 \end{bmatrix} - x_2 \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix} - x_3 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 0 \tag{4}$$

Which can be written as augmented matrix and row reduced. This will give us the balancing coefficients for the chemical equation.

$$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -1 & -1 \\ 3 & -4 & 0 \end{bmatrix} \xrightarrow{-R_2 + R_1} \begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \\ 3 & 4 & 0 \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} 1 & -1 & -1 \\ 0 & 0 & 0 \\ 3 & 4 & 0 \end{bmatrix} \xrightarrow{R_3 \Leftrightarrow R_2} \begin{bmatrix} 1 & -1 & -1 \\ 3 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} 1 & -1 & -1 \\ 3 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{-3R_1 + R_2} \begin{bmatrix} 1 & -1 & -1 \\ 0 & -1 & 3 \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} 1 & -1 & -1 \\ 0 & -1 & 3 \\ 0 & 0 & 0 \end{bmatrix} \xrightarrow{R_1 + R_2} \begin{bmatrix} 1 & 0 & -4 \\ 0 & 1 & -3 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

Solution Set:

$$\begin{cases} x_1 = -4 \\ x_2 = -3 \\ x_3 = free \quad variable \end{cases}$$

Therefore, the balanced chemical equation is:

$$4KClO_3 \longrightarrow 3KClO_4 + KCl \tag{9}$$

3 Collaborative Code:

The code was pulled directly from RRCC Linear Algebra class discussion on D2L.

Coding completed by Kevin Evers, Everett Oklar, Sophia Wyss, Holly Hammons, Alex Langfield and Brett Web, along with the help of Adam Forland.

Submitted by Stacie Barbarick and Alex Langfield

Annotated by Stacie Barbarick

This code will use linear algebra to find the most reduced coefficients of chemical reactants and products to balance a user input chemical reaction. Coding represented by blue text.

Python Code:

```
from sympy import *
```
This function will import the library from Python for symbolic computation.

```
import numpy as np
```
This command will import the libray numpy which allows the program to complete scientific computing, such as with arrays.

```
numElements = int(input("Please enter number of individual elements in the unbalanced equation (i.e. H2 + O2 = H2O has 2 elements, H and O)))
```
This command will require the user to input the number of individual elements and will define the number of rows in the matrix. The coding is defining the number of individual elements as an integer value that is an input from the user.

```
numReactants = int(input("Please input the number of reactants in the unbalanced equation: "))
```
This command will require the user to input the number of reactants from the unbalanced chemical

equation. The coding is defining the number of Reactants as an integer value that is an input from the user.

numProducts = int(input("Please input the number of products in the unbalanced equation:"))
This command will require the user to input the number of products from the unbalanced chemical equation. The coding is defining the number of Products as an integer value that is an input from the user.

numMolecules = numReactants + numProducts
This function will sum the total number of molecules in the reactants and products.

print(Please enter the number of atoms of each molecule in each element i.e. H2 + O2 = H2O There are 2 H atoms in molecule 1, 0 H atoms in molecule 2, and 2 H atoms in molecule 3. There are 0 O atoms in molecule 1, 2 O atoms in molecule 2, and 1 O atom in molecule 3.)
This code will provide an example to the user of how to enter data. The print command will display the text to the user.

equation = [ ]
This line of code creates an equation and the brackets create an empty list for the matrix to be built in. This empty list allows list formation that can be appended later.

for i in range(numElements):
This line creates a range for the elements or a row for each element and designates them as i elements.

elementRow = [ ]
Similarly this empty list will be used for the row to be built for each atom.

for j in range(numMolecules):
This line creates a column for each molecule and designates them as j molecules.

print("Please enter number of atoms of element", i + 1, "in molecule", j + 1, ": ", end="")
This print statement would not be accepted into the input function so it's printed separately-according to the programmer.

numAtoms = int(input())
This line defines the number of Atoms as an integer value and as an input.

if (j+1) ¡= numReactants:
elementRow.append(numAtoms)
This is an "if else" statement that tells the computer to execute a specific function if the first statement is true or false.

So this code will check to see if the number of molecules is larger by 1. The atoms of the reactants must remain the same, so this conditional statement will allow the matrix array to remain accurate.

else:elementRow.append(-numAtoms)
This is the else portion of the above conditional statement. It creates a function that if the above condition is not met, it will add negative atoms to the product side of the equation and subsequently to the matrix.

equation.append(elementRow)
After the computer has executed this if else condition, it will add the row to the empty list we created.

equation = Matrix(equation)
This command will convert the list into a matrix

balancedEquation = equation.rref()
RREF is the command for the computer to row reduce the matrix

print(balancedEquation)

This command will print the balanced equation for the user.

User input below in blue.

Please enter number of individual elements in the unbalanced equation (i.e. H2 + O2 = H2O has 2 elements, H and O): 3

Please input the number of reactants in the unbalanced equation: 1

Please input the number of products in the unbalanced equation: 2

Please enter the number of atoms of each molecule in each element i.e. H2 + O2 = H2O There are 2 H atoms in molecule 1, 0 H atoms in molecule 2, and 2 H atoms in molecule 3

There are 0 O atoms in molecule 1, 2 O atoms in molecule 2, and 1 O atom in molecule 3

Please enter number of atoms of element 1 in molecule 1 : 1
Please enter number of atoms of element 1 in molecule 2 : 1
Please enter number of atoms of element 1 in molecule 3 : 1
Please enter number of atoms of element 2 in molecule 1 : 1
Please enter number of atoms of element 2 in molecule 2 : 1
Please enter number of atoms of element 2 in molecule 3 : 1
Please enter number of atoms of element 3 in molecule 1 : 3
Please enter number of atoms of element 3 in molecule 2 : 4
Please enter number of atoms of element 3 in molecule 3 : 0
(Matrix([[1, 0, -4], [0, 1, -3], [0, 0, 0]]), (0, 1))

The output of this row reduction in Python shows agreement with the hand calculated balanced chemical equation in Section 2.

4 Conclusion

While the process of balancing chemical equations can be done by hand, there are far more productive, time-saving and free computing tools available. The example equation used above seems simplistic, however, it demonstrates the advantage gained by using technology in this application. A variation of the functioning code could be applied to any chemical equation, regardless of complexity. The small time investment in the creation of the code will be made up in dividends when it can be used quickly and easily in a countless number of applications. According to G.R Blakely, "the proper place for chemical reasoning is before the equation balancing process and after it, not during it". [3] With programs such as these in place, scientists will be able to spend more time engaged in such pursuits.

References

[1] Herndon, W. C. *On Balancing Chemical Equations: Past and Present. Journal of Chemical Education* 74(11), 1359. doi:10.1021/ed074p1359, 1997.

[2] Krishnamurthy, E. V. *Generalized matrix inverse approach for automatic balancing of chemical equations.* International Journal of Mathematical Education in Science and Technology, 9(3), 323–328. doi:10.1080/0020739780090310, 1978.

[3] Blakley, G. R. *Chemical equation balancing: A general method which is quick, simple, and has unexpected applications.* Journal of Chemical Education, 59(9), 728. doi:10.1021/ed059p728, 1982