SOME EFFICIENT MATHEMATICAL PROGRAMMING TECHNIQUES FOR BALANCING EQUATIONS OF COMPLEX CHEMICAL REACTIONS

Mumtaz Yousaf1, Muhammad Mujtaba Shaikh2, Abdul Wasim Shaikh3

1 Institute of Mathematics and Computer Sciences, University of Sindh, Jamshoro, Pakistan
2 Department of Basic Sciences and Related Studies, Mehran University of Engineering and Technology, Jamshoro, Pakistan
3 Institute of Mathematics and Computer Sciences, University of Sindh, Jamshoro, Pakistan

1 mumtaz848@yahoo.com, 2 mujtaba.shaikh@faculty.muet.edu.pk, 3 wasim.shaikh@usindh.edu.pk

Corresponding Author: Muhammad Mujtaba Shaikh

https://doi.org/10.26782/jmcms.2020.10.00004

Abstract

The equations of chemical reactions usually describe the breakup of some desired or consequent products and the breakup of reactants used in chemical reactions. Usually, the equations in skeleton form are unbalanced, and a deeper analysis requires the balanced form which is not quite easy for complex reactions. In instances, the balancing can be done quickly with hit and trial and simple logic. In such cases, the trials are found not attractive, although they are helpful at a simple level at an advanced level they become more tough and unpredictable. For complex cases, many mathematical techniques can be used for balancing equations of chemical reactions. In this study, some efficient mathematical techniques are suggested which can be more suitable from all perspectives to balance chemical equations and to provide a case by case recommendations for the practitioners. Particularly, we suggest and utilize the linear algebra Gauss elimination (LA-GE) and the linear programming two-phase (LP-2P) approaches to successfully for chemical equation balancing. A number of chemical equations have been taken from literature to see the performance of both approaches. The advantages and disadvantages of both approaches are discussed, mainly with the computer programming in MATLAB and TORA systems, and an exhaustive comparison based on floating point operations (FLOPS) is carried out. The recommendations will prove fruitful for the practitioners for using efficient and yet simpler mathematical programming techniques for the balancing of equations of chemical reactions in the future.

Keywords: Chemical reactions, Mathematical programming; Linear algebra; Gauss elimination; Linear programming, Applied Chemistry, Mathematical Chemistry.

Mumtaz Yousaf et al
I. Introduction

Every era has an advantage in its characteristics to do things in a more efficient way [II],[XII]. Consequently, the researchers of the present time focus to propose exhaustive mathematical models than usual for physical systems [XXVII],[XVI], develop more efficient and accurate methods for the solution than the conventional ones and suggest best programming and software mechanisms as implementation platforms for the methods [XXI],[X],[XV]. The concepts in applied sciences and engineering research are based upon the core contents established in natural sciences [XXIV],[XXII],[III],[XIX],[XXVII]. For example, in Chemistry while studying stochiometric relationships and to produce so many things through different methods and processes: the combustion problem [II], in analyzing yield of Phthalic Anhydride [XX], Colebrook’s equation [XXV],[XXVI], CO₂ emission modeling [XXVIII], solving equations to estimate thermal properties of hydrocarbons [X], etc.

The chemical equations have great importance in the field of Chemistry in different department ways, and equation balancing is a unique problem which in the initial stage arises while examining, in the chemical reaction, the substance which we use in the process to get the result of new formation and produce the desired substance, respectively reactant and product. The desired equation should be studied in the form of stoichiometry which is observed during the chemical reaction, and is in real one of the fundamental parts of Mathematical chemistry – deals with the relation of balance which has a worth in the process and procedure of the structure of the chemical equation. When this is in the form of a feasible as well as a heavenly process, the results of equations often remain the same since taking it as a chemical reaction [XVII],[VI],[IV]. While the problem of balancing the equations of chemical reactions [XVII],[VI],[IV] is one of the fundamental concepts, a number of attempts have been made to suggest diagram [VI], empirical [XVII],[VI],[IV], mathematical [XXIII],[VIII] and computer [XIV] methods to balance the equations in different ways. Traditionally, the trial and error method is adopted on simpler chemical reactions, whereas due to highly complex balancing paradigms resulting from more advanced and hybrid chemical reactions, some more accurate mathematical techniques are needed which are also easily programmable to automate the process.

The modifications can be based upon the techniques of Linear programming and Algebra, since the balancing chemical equations yield a linear system of equations Linear programs with polynomial time [XI] with its remarkable extensions [V],[VII] are important in this regard. The extensions are basically updates of the simplest form of an algorithm by Dantzig [XXIV] at the start of 1950 with exponentially increasing time. On the other hand, some methods free from trial and error and deterministic in nature can be used depending on Hermite normal form and matrix inverses [XIII],[XX],[XIX], and use the non-trivial unconstrained equation to balance problem on a homogenous algebraic process linearly.

Meanwhile, in [XXXI], some methods were used in the form of matrix and conversion to reduced echelon form to make it possible to control the problem of any chemical equation which could be used and beneficial for industrial chemistry to make the things in an appropriate way. Such type of method was found for the ability

*Mumtaz Yousaf et al*
of the students of every efficiency level. In such study, MATHEMATICA and MAPLE software was used for this problem to solve it completely with its proper system to do it in a comprehensive way so that it might achieve its objectives of balancing the equation properly. In [XIX], integer linear programming and integer non-linear programming discussed about algorithm and complexity which is based on floating point arithmetic with rational arithmetic and functions of multiple molecules with residue arithmetic to make the process complete. But, some functions were found unsuitable due to intermediate number growth with the set of prime based residue arithmetic that was required to fulfill it with the product also some failures due to the division of zero. MATLAB and LINDO software was used in making the process of the product with the technique of branch and bound which proved suitable during the whole process with the use of features.

We propose and suggest two mathematical techniques for chemical equation balancing which are computer friendly and are easily implemented using MATLAB and TORA software. The algorithms of proposed methods, their implementation on several test equations with comments on their complexity are discussed. The results and recommendations will prove fruitful for the practitioners for using efficient and yet simpler mathematical programming techniques for the balancing of equations of chemical reactions in the future.

II. Material and Methods

The basic terminology concerning the equations of chemical reactions, mathematical models, algorithms of suggested techniques and complexity formulas are presented in this section.

Basic Terminologies

A process in which one or more substances, the reactants, are converted to one or more different substances, the product, is called a chemical reaction. A chemical equation is the symbolic representation of a chemical reaction in the form of symbols and formulae. An unbalanced equation (sometimes also called a skeleton equation) is one in which the number of atoms for at least one element on each side of the arrow is unequal. A balanced chemical equation is an equation for a chemical reaction in which the number of atoms for each element in the reaction and the total charge is the same for both the reactants and the products. For example, equation (1) is a skeleton form, and (2) is the corresponding balanced form of a chemical reaction in [VIII].

\[
\text{K}_4\text{Fe(CN)}_6 + \text{H}_2\text{SO}_4 + \text{H}_2\text{O} \rightarrow \text{K}_2\text{SO}_4 + \text{FeSO}_4 + 3(\text{NH}_4)_2\text{SO}_4 + \text{CO} \quad (1)
\]

\[
\text{K}_4\text{Fe(CN)}_6 + 6\text{H}_2\text{SO}_4 + 6\text{H}_2\text{O} \rightarrow 2\text{K}_2\text{SO}_4 + \text{FeSO}_4 + 3(\text{NH}_4)_2\text{SO}_4 + 6\text{CO} \quad (2)
\]

Mathematical Models

We discuss the mathematical models for the equations of chemical reactions in matrix form, in two different contexts: the Linear algebra (LA) and Linear programming (LP) paradigms. The coefficients of all terms in the equation are first assigned unknown names of variables that are to be determined. For example, (1) by assigning the unknown coefficients takes the form:

Mumtaz Yousaf et al
\[ K_4 \text{Fe(CN)}_6 + x_2 \text{H}_2 \text{SO}_4 + x_3 \text{H}_2 \text{O} \rightarrow x_4 \text{K}_2 \text{SO}_4 + x_5 \text{FeSO}_4 + x_6 (\text{NH}_4)_2 \text{SO}_4 + x_7 \text{CO} \]  

(3)

By using the elemental balances on both sides of the equation, we get a system of homogeneous equations, and in the context of the LA paradigm, it may be written as:

\[ A \mathbf{x} = \mathbf{0} \]  

(4)

Where the coefficient matrix \( A \) contains the difference in the number of atoms of elements in the equation. Rows in \( A \) represent elements, whereas columns represent the total number of terms (reactants and products) in the equation. The vector \( \mathbf{x} \) comprises of the unknown coefficients needed to balance the equation. It is imperative to get non-zero and integer values for the variables as the solution of a homogeneous system (4). The terms \( A \) and \( \mathbf{x} \) for (1) using (3) after elemental balances are, respectively:

\[
A = \begin{bmatrix}
K & 4 & 0 & 0 & -2 & 0 & 0 & 0 \\
Fe & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
C & 6 & 0 & 0 & 0 & 0 & 0 & -1 \\
N & 6 & 0 & 0 & 0 & 0 & -2 & 0 \\
H & 0 & 2 & 2 & 0 & 0 & -8 & 0 \\
S & 0 & 1 & 0 & -1 & -1 & -1 & 0 \\
O & 0 & 4 & 1 & -4 & -4 & -4 & -1 
\end{bmatrix}
\]

(5)

and \( \mathbf{x} = [x_1x_2x_3x_4x_5x_6x_7]^T \).

In a general LP problem, we seek to minimize a linear cost function \( Z \), which is the sum of all coefficients to be determined as in (7), overall \( n \)-dimensional non-zero vectors \( \mathbf{x} \) (as in (9) subject to a set of linear equality and inequality constraints as well as integrality restrictions on some or all of the variables in \( \mathbf{x} \) as given in (8)-(9).

\[
\text{minimize} \quad Z = \mathbf{c}^T \mathbf{x}
\]

(7)

\[
\text{subject to} \quad A \mathbf{x} = \mathbf{b}
\]

(8)

\[
\mathbf{x} \in \mathbb{R}^+
\]

(9)

For the mathematical LP model for the reaction (1), the definitions of \( A \) and \( \mathbf{x} \) are the same, and additionally, we have:

\[ \mathbf{b} = [0000000]^T \]

(10)

and \[ \mathbf{c} = [1111111]^T \]

(11)

An optimal solution, say of (1), in terms of the mathematical LP model (7)-(9) is a feasible solution where the objective function reaches its minimum value – for example, the least cost.

**Suggested Algorithms**

Mathematical methods can be utilized to reach the solutions of corresponding LA and LP formulations of the balancing problem of chemical equations. We suggest two algorithms that are sufficient to easily balance any typical, large and complex chemical reaction.

*Mumtaz Yousaf et al*
equation of the chemical reaction. We suggest the Gauss elimination method and two-phase method in the context of LA and LP mathematical models discussed in II (B), respectively labelled as LA-GE and LP-2P methods. Moreover, the methods have been programmed and successfully executed for a number of equations of different chemical reactions in MATLAB and TORA software, respectively. The results are discussed for some reactions in section III.

**Linear Algebra – Gauss Elimination (LA–GE)**

The computational steps are described as:

1. Assign variable coefficients to the unknowns in the equation of chemical reaction, as in (3).
2. Equate the number of atoms of different elements on the reactant and product sides of the equation.
3. Develop a homogeneous system of equations in the unknowns, i.e. system (4).
4. Reduce the coefficient matrix \( A \) into an echelon form using elementary row operations.
5. Find non-trivial solutions of the system, and decide which ones will desirably represent the coefficients in the balanced form of the given equation.

**Linear Programming – Two-phase (LP–2P)**

1. Convert the original problem into standard form.
2. Setup Phase-I problem by:
   (a) Add artificial variable where it is needed
   (b) Create a Phase-I objective function to minimize \( Z' = \sum R \)
   (c) Convert new objective function into standard form.
   (d) Don’t forget the non-negativity constraints
3. Setup tableau and legitimate.
4. Find the optimal solution for Phase-I.
5. The original problem is infeasible if the optimal solution of Phase-I has any of the artificial variables as non-zero. Otherwise end Phase-I, and begin Phase-II.
6. Replace the original objective function, legitimate and find the optimal solution by regular simplex iterations.

**Comparison mechanism of the complexity of an algorithm**

Extending the calculations in [XXX], the numbers of floating point operations in the Gauss-Elimination methods becomes:

\[
\text{FLOPS}_{\text{LA-GE}} = \text{FLOPS}_{\text{Echelon}} + \text{FLOPS}_{\text{Back Substitution}} \tag{12}
\]

Mumtaz Yousaf et al
So that, $Flops_{LA-GE} = \left\lceil \frac{1}{3} m^3 + m^2 \right\rceil$, (13) and $\lceil \cdot \rceil$ denotes the Ceiling function.

Using the procedure of [VIII], the numbers of floating point operations in the LP Two-phase (LP-2P) method per iteration can be formulated as:

$$Flops_{LP-2P} = \left\lceil \frac{1}{2} m^2 + mn + (m - p)^2 + 2(m - p) + 4m - 1 \right\rceil / \text{iteration} \quad (14)$$

where, $m =$ number of elements/rows
$n =$ number of coefficients/columns
$p =$ number of pivot elements in LP-2P method.

III. Test Equations and Software Implementation

For performance evaluations of LA-GE and LP-2P algorithms for balancing the equations of chemical reactions, the following test equations in skeleton form with complexity level from moderate to higher have been used from [IV], [VI],[VII], [XI], [XIII], [XIX],[XX],[XXXI].

1. KClO$_3$ $\rightarrow$ KCl + O$_2$ (15)
2. CH$_4$ + O$_2$ $\rightarrow$ CO$_2$ + H$_2$O (16)
3. BaO$_2$ + Al $\rightarrow$ Ba + Al$_2$O$_3$ (17)
4. CaCN$_2$ + H$_2$O $\rightarrow$ CaCO$_3$ + NH$_3$ (18)
5. K$_4$Fe (CN)$_6$ + H$_2$SO$_4$ + H$_2$O $\rightarrow$ K$_2$SO$_4$ + FeSO$_4$ + (NH$_4$)$_2$SO$_4$ + CO (19)
6. K$_4$Fe(CN)$_6$ + KMnO$_4$ + H$_2$SO$_4$ $\rightarrow$ KHSO$_4$ + Fe$_2$(SO$_4$)$_3$ + MnSO$_4$ + HNO$_3$ +CO$_2$+H$_2$O (20)

$$\begin{bmatrix} 1 & -1 & 0 \\ 1 & -1 & 0 \\ 3 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (21)$$

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 4 & 0 & 0 & -2 \\ 0 & 2 & -2 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (22)$$

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 2 & 0 & 0 & -3 \\ 0 & 1 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (23)$$

Mumtaz Yousaf \textit{et al}
\[
\begin{bmatrix}
1 & 0 & -1 & 0 \\
1 & 0 & -1 & 0 \\
2 & 0 & 0 & -1 \\
0 & 2 & 0 & -3 \\
0 & 1 & -3 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]  
(24)

\[
\begin{bmatrix}
4 & 0 & 0 & -2 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -1 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & -1 \\
6 & 0 & 0 & 0 & 0 & -2 & 0 \\
0 & 2 & 2 & 0 & 0 & -8 & 0 \\
0 & 1 & 0 & -1 & -1 & -1 & 0 \\
0 & 4 & 1 & -4 & -4 & -4 & -1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]  
(25)

\[
\begin{bmatrix}
4 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
6 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 4 & 4 & -4 & -12 & -4 & -3 & -2 & -1 \\
0 & 0 & 2 & -1 & 0 & 0 & -1 & 0 & -2 \\
0 & 0 & 1 & -1 & -3 & -1 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
x_5 \\
x_6 \\
x_7 \\
x_8 \\
x_9 \\
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]  
(26)

The test equations (15)-(20) in a form consistent with the mathematical model of LA (4) can be written as (21)-(26), respectively. Equations (21)-(26) are also the starting constraints in the LP formulation (8) of (15)-(20) along with non-negativity restrictions (9), whereas the objective function is defined in (7), and the vectors \(c\) and \(b\) are respectively one dimensional unit and zero vectors of same size \(n\).

For the full resolution implementation of the LA-GE approach we have utilized MATLAB software. Using the MATLAB environment, the matrix \(A\) can be defined and using the built-in feature ‘\text{ref}(A)’ leads to the echelon form of \(A\), and using the backward substitution the desired non-trivial solution is sought; the number of FLOPs are calculated using (13).

\section*{IV. Results and Discussion}

Using the LA-GE approach, the solution has been successfully acquired for equations (15)-(20) utilizing the matrix forms (21)-(26), and the implementation details have been summarized in Table 1. A possible concern with the LA-GE method lies in its searching strategy to find the desired non-trivial solution of the system (4) since in addition to the conventional method, where the non-zero solutions are required, it is also necessary that the solutions be positive integers. Thus, in some cases, we may need multiplying factors in the end to get the complete balanced equation. For example, as in Table 1, the multiplying factors of 2, 3 and 5 are needed to get desired balanced form of equations (15), (17) and (20), respectively.
The LP-2P algorithms have been successfully implemented in the TORA software, where the end user can select from a list of many algorithms for linear programming. The coefficients in the objective function (7), constraints (8)-(9), and bounds for search region, and if any of the variables is restricted, are to be supplied in form of the input grid, for example, we have described the input grid for the complex chemical equation (20) in Fig. 1. Similarly, all other equations (15)-(19) have been defined in form of Fig. 1. Finally, the LP-2P algorithms find the desired minimum solution after some iterations of phases I and II as mentioned in steps 1-6 of section II-C-(b). For example equation (20), the final solution and detailed values of entering, leaving and artificial variables arrived at the tenth iteration are presented in Figs.2-3. Due to a higher number of variables (i.e, 9), the final iteration solution is represented as split in Figs. 2-3. The values in the last column pertaining to the unblocked variables represent the optimal minimal solution. Unlike, the LA-GE algorithm, the desired solution for equation (20) and all others have been directly obtained by the LP-2P algorithm, but the latter is prone to the selection of appropriate lower and upper bounds. Implementation details of the LP-2P algorithms for balancing the chemical equations (15)-(20) are summarized in Table 2.

### Table 1: LA-GE implementation details

| Test equation | LA-GE implementation details | Non-trivial solution for variables $x_i$ |
|---------------|-----------------------------|-----------------------------------------|
|               | Row operation used | Free variable | Scaling factor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| (15)          | 4              | 2ND           | 2 | 1 | 1 | 3/2 | - | - | - | - | - | - |
| (16)          | 4              | 3RD           | NA | 1 | 2 | 1 | 2 | - | - | - | - | - |
| (17)          | 3              | 3RD           | 3 | 1 | 4/3 | 1 | 2/3 | - | - | - | - | - |
| (18)          | 6              | 3RD           | NA | 1 | 3 | 1 | 2 | - | - | - | - | - |
| (19)          | 17             | 1ST           | NA | 1 | 6 | 6 | 2 | 1 | 3 | 6 | - | - |
| (20)          | 20             | 5th           | 5 | 2 | 122/5 | 299/5 | 162/5 | 1 | 122/5 | 12 | 12 | 188/5 |

Therefore, the equations (15)-(20) in the final balanced form by LA-GE and LP-2P algorithms are given as (27)-(32).

\[
\begin{align*}
2\text{KClO}_3 &\rightarrow 2\text{KCl} + 3\text{O}_2 \quad (27) \\
\text{CH}_4 + 2\text{O}_2 &\rightarrow \text{CO}_2 + 2\text{H}_2\text{O} \quad (28) \\
3\text{BaO}_2 + 4\text{Al} &\rightarrow 3\text{Ba} + 2\text{Al}_2\text{O}_3 \quad (29) \\
\text{CaCN}_2 + 3\text{H}_2\text{O} &\rightarrow \text{CaCO}_3 + 2\text{NH}_3 \quad (30) \\
\text{K}_4\text{Fe(CN)}_6 + 6\text{H}_2\text{SO}_4 + 6\text{H}_2\text{O} &\rightarrow 2\text{K}_2\text{SO}_4 + \text{FeSO}_4 + 3(\text{NH}_4)_2\text{SO}_4 + 6\text{CO} \quad (31) \\
10\text{K}_4\text{Fe(CN)}_6 + 122\text{KMnO}_4 + 299\text{H}_2\text{SO}_4 &\rightarrow 162\text{KHSO}_4 + 122\text{MnSO}_4 + 60\text{HNO}_3 + 60\text{CO}_2 + 188\text{H}_2\text{O} \quad (32)
\end{align*}
\]

*Mumtaz Yousaf et al*
Mumtaz Yousaf et al

Fig. 1. LP-2P input grid for equation (20) in TORA software

Fig. 2. Final iteration (a) of LP-2P algorithm for equation (20) in TORA

Fig. 3. Final iteration (b) of LP-2P algorithms for equation (20) in TORA
Table 2: LP-2P implementation details

| Test equation | LP-2P implementation details | Optimal minimal solution for variables $x_i$ |
|---------------|-------------------------------|---------------------------------------------|
|               | Iteration | Coefficient | Pivot | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| (15)          | 4 | 3 | 2 | 2 | 3 | - | - | - | - | - | - | - |
| (16)          | 5 | 4 | 3 | 1 | 2 | 1 | 2 | - | - | - | - | - |
| (17)          | 5 | 4 | 3 | 3 | 4 | 3 | 2 | - | - | - | - | - |
| (18)          | 5 | 4 | 3 | 1 | 3 | 1 | 2 | - | - | - | - | - |
| (19)          | 9 | 7 | 6 | 1 | 6 | 6 | 2 | 1 | 3 | 6 | - | - |
| (20)          | 10 | 9 | 8 | 10 | 122 | 299 | 162 | 5 | 122 | 60 | 60 | 188 |

On the basis of FLOPs, as per the calculation procedure described in equations (13)-(14) for LA-GE and LP-2P algorithms, respectively, both algorithms have been compared and the results are displayed in Fig. 4.

It is immediate to realize that the LP-2P approach uses a higher number of floating point operations to reach the optimal solution of the minimization problem as compared to the LA-GE approach, and hence to balance the equations of chemical reactions.

While FLOPs in the LP-2P algorithm were observed to be higher than in LA-GE for all equations (15)-(20), the former can still safely be preferred due to the ease of software implementation and the later for manual calculations. It is also worthwhile to mention that the LP-2P method besides being a software friendly approach in the TORA environment is sensitive to the selection of lower bounds for the variables in order to attain the desired non-zero integer solutions. A similar sensitivity can be observed in the LA-GE algorithm to seek the desired solution out of infinitely many possible non-zero solutions since a few variables are free variables. The LA-GE can be safely recommended to be used manually as well as using the software implementation in MATLAB, whereas the manual implementation of the LP-2P algorithms due to a higher number of iterations and higher computational overload is not suggested. Regardless of the concern that an algorithm can be implemented manually or by using software, the main concern of the researchers is to anyhow reach to the desired solution, as it is one of the fundamental steps in understanding the exact relationship and quantities of the chemical elements in a chemical reaction but not the only objective. The accurate results by the software are nevertheless achieved besides some concerns. The FLOPs comparison is helpful in understanding that for which algorithm, computer implementation is the only way-out for the researchers to balance the equations of chemical reactions. For the LP-2P technique the software environment is recommended for its efficient use nowadays as it takes higher FLOPs as compared to the LA-GE algorithms, but also gives the desired solution in the appropriate form as compared to the LA-GE algorithm.

_Mumtaz Yousaf et al_
V. Conclusion

Two different efficient algorithms were suggested and thoroughly analyzed for balancing equations of chemical reactions. An exhaustive comparison of both methods on six test equations from literature was conducted. The computational procedure, advantages and disadvantages of both approaches were examined. It was found that the LP-2P approach in TORA software was a more direct and reliable technique than the LA-GE approach, and resulted in the desired solution directly using a computer program. In terms of FLOPs, the LA-GE approach was better than the LP-2P approach.

VI. Acknowledgement

Authors are highly indebted to their institutes for providing facilities for the successful conduct of this research.

Conflict of Interest:

There was no relevant conflict of interest in this paper.

References

I. A. Harshavardhan, Syed Nawaz Pasha, Sallauddin Md, D. Ramesh, “TECHNIQUES USED FOR CLUSTERING DATA AND INTEGRATING CLUSTER ANALYSIS WITHIN MATHEMATICAL PROGRAMMING”, J.Mech.Cont.& Math. Sci., Vol.-14, No.-6, November - December (2019) pp 546-557

Mumtaz Yousaf et al
II. Abro, Hameer Akhtar, and Muhammad Mujtaba Shaikh. "A new time-efficient and convergent nonlinear solver." Applied Mathematics and Computation 355 (2019): 516-536.

III. Bhan, Veer, Ashfaque Ahmed Hashmani, and Muhammad Mujtaba Shaikh. "A new computing perturb-and-observe-type algorithm for MPPT in solar photovoltaic systems and evaluation of its performance against other variants by experimental validation." Scientia Iranica 26, no. Special Issue on machine learning, data analytics, and advanced optimization techniques in modern power systems [Transactions on Computer Science & Engineering and Electrical Engineering (D)] (2019): 3656-3671.

IV. Blakley, G. R. "Chemical equation balancing: A general method which is quick, simple, and has unexpected applications." (1982): 728.734.

V. Chang, Soo Y., and Katta G. Murty. "The steepest descent gravitational method for linear programming." Discrete Applied Mathematics 25, no. 3 (1989): 211-239.

VI. Crosland, Maurice P. "The use of diagrams as chemical 'equations' in the lecture notes of William Cullen and Joseph Black." Annals of Science 15, no. 2 (1959): 75-90.

VII. Dantzig, G. B. "Linear Programming and Extensions, Princeton, Univ." Press, Princeton, NJ (1963).

VIII. Das, S. C. "A mathematical method of balancing a chemical equation." International Journal of Mathematical Education in Science and Technology 17, no. 2 (1986): 191-200.

IX. Hu, Jian-Feng, and Ping-Qi Pan. "An efficient approach to updating simplex multipliers in the simplex algorithm." Mathematical programming 114, no. 2 (2008): 235-248.

X. Hussain, Bilal, and Muhammad Ahsan. "A Numerical Comparison of Soave Redlich Kwong and Peng-Robinson Equations of State for Predicting Hydrocarbons’ Thermodynamic Properties." Engineering, Technology & Applied Science Research 8, no. 1 (2018): 2422-2426.

XI. Karmarkar, Narendra. "A new polynomial-time algorithm for linear programming." In Proceedings of the sixteenth annual ACM symposium on Theory of computing, pp. 302-311. 1984.

XII. Khoso, Amjad Hussain, Muhammad Mujtaba Shaikh, and Ashfaque Ahmed Hashmani. "A New and Efficient Nonlinear Solver for Load Flow Problems." Engineering, Technology & Applied Science Research 10, no. 3 (2020): 5851-5856.

Mumtaz Yousaf et al
XIII. Krishnamurthy, E. V. "Generalized matrix inverse approach for automatic balancing of chemical equations." *International Journal of Mathematical Education in Science and Technology* 9, no. 3 (1978): 323-328.

XIV. Kumar, David D. "Computer applications in balancing chemical equations." *Journal of Science Education and Technology* 10, no. 4 (2001): 347-350.

XV. Lemita, Abdallah, Sebti Boulahbel, and Sami Kahla. "Gradient Descent Optimization Control of an Activated Sludge Process based on Radial Basis Function Neural Network." *Engineering, Technology & Applied Science Research* 10, no. 4 (2020): 6080-6086.

XVI. Massan, Shafiq-ur-Rehman, Asim Imdad Wagan, and Muhammad Muitaba Shaikh. "A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem." *Applied Soft Computing* 90 (2020): 106176.

XVII. McNaught, Alan D. *Compendium of chemical terminology*. Vol. 1669. Oxford: Blackwell Science, 1997.

XVIII. Memon, Ali Asghar, Muhammad Muitaba Shaikh, Syed Sabir Hussain Bukhari, and Jong-Suk Ro. "Look-up Data Tables-Based Modeling of Switched Reluctance Machine and Experimental Validation of the Static Torque with Statistical Analysis." *Journal of Magnetics* 25, no. 2 (2020): 233-244.

XIX. Moore, John T. *Elements of linear algebra and matrix theory*. No. 512.897 M6. 1968.

XX. Rao, T. Mahadeva, K. Subramanian, and E. V. Krishnamurthy. "Residue arithmetic algorithms for exact computation of g-inverses of matrices." *SIAM Journal on Numerical Analysis* 13, no. 2 (1976): 155-171.

XXI. Sarosh, Ali, Arshad Hussain, Erum Pervaiz, and Muhammad Ahsan. "Computational Fluid Dynamics (CFD) Analysis of Phthalic Anhydride's Yield Using Lab Synthesized and Commercially Available (V2O5/TiO2) Catalyst." *Eng. Technol. Appl. Sci. Res* 8, no. 2 (2018): 2821-2826.

XXII. Sen, S. K., and E. V. Krishnamurthy. "Numerical Algorithms: Computations in Science and Engineering." *Affiliated East-West Press*, 2001 (2001).
XXIII. Sen, Syamal K., Hans Agarwal, and Sagar Sen. "Chemical equation balancing: An integer programming approach." *Mathematical and Computer Modelling* 44, no. 7-8 (2006): 678-691.

XXIV. Shahani, Zulfiqar Ali, Ashfaq Ahmed Hashmani, and Muhammad Mujtaba Shaikh. "Steady state stability analysis and improvement using eigenvalues and PSS." *Engineering, Technology & Applied Science Research* 10, no. 1 (2020): 5301-5306.

XXV. Shaikh, Muhammad Mujtaba, and Asim Imdad Wagan. "A new explicit approximation to Colebrook’s friction factor in rough pipes under highly turbulent cases." *International Journal of Heat and Mass Transfer* 88 (2015): 538-543.

XXVI. Shaikh, Muhammad Mujtaba, and Asim Imdad Wagan. "A sixteen decimal places' accurate Darcy friction factor database using non-linear Colebrook’s equation with a million nodes: A way forward to the soft computing techniques." *Data in brief* 27 (2019): 104733.

XXVII. Soomro, Abdul Sattar, Gurudeo Anand Tularam, and Muhammad Mujtaba Shaikh. "A comparison of numerical methods for solving the unforced van der Pol’s equation." *Mathematical Theory and Modeling* 3, no. 2 (2013): 66-78.

XXVIII. Soomro, Majid Ali, Sheeraz Ahmed Memon, Muhammad Mujtaba Shaikh, and Azizullah Channa. "Indoor air CO2 assessment of classrooms of educational institutes of hyderabad city and its comparison with other countries." In *AIP Conference Proceedings*, vol. 2119, no. 1, p. 020014. AIP Publishing LLC, 2019.

XXIX. Shahadat Ali, H. M., M. A. Habib, M. Mamun Miah, M. Ali Akbar, “A Modification of the Generalized Kudryashov Method for the System of Some Nonlinear Evolution Equations”, J.Mech.Cont.& Math. Sci., Vol.-14, No.-1, January-February (2019) pp 91-109

XXX. Strang, G. "Introduction to Applied Math. 1986." Wellesley-Cambridge Press, Wellesley, MA.

XXXI. Y. Hari Krishna; Reddy, G Venkata Ramana; Praveen, JP; Balancing Chemical Equations By Using Matrix Algebra’, World Journal Of Pharmacy And Pharmaceutical Sciences ,6(2), 994-999, 2017

*Mumtaz Yousaf et al*