Katie Izatt, Carson Ivory, and Johnny Peterson

MATH 2270-004

12 April 2019

Semester Final Project: Automatic Balancing Redox Reactions

Abstract:

Redox reactions are chemical reactions involving “ox”idation and “red”uction, or in other words, a transferring of electrons between atoms. Balancing these reactions can be highly difficult, but can be accomplished using linear algebra. Our objective is to create software capable of automatically and (hopefully) quickly balancing redox reactions. We will discuss possible tactics for accomplishing this purpose, and the various benefits or disadvantages of each.

Oxidation-Reduction reactions, more commonly known as “redox” reactions, are chemical reactions in which the oxidation states of the atoms involved are changing. While this is a simple definition, its generality allows it to encompass the majority of all chemical reactions. Combustion, acid-base, disproportionation, and single replacement reactions are all common types of redox reactions. Each type involves the oxidation of one or more atoms, and the complimentary reduction of other atoms throughout the reaction. The bonds between the atoms are broken and reformed as the atoms exchange their electrons, thereby changing their oxidation states. Since redox reactions encompass the majority of chemical reactions, whether simple or complex, they are extremely important to the world of chemistry. Chemists in nearly every field (and any students in high school or college general chemistry) come across them at some point.

They help to give a general definition and balancing method to the majority of chemical reactions.

Redox reactions are most often balanced using the half-reaction method. The first step in this method is identifying the oxidation and the reduction happening within the whole redox reaction. This is done by determining the oxidation states of each atom on the left side of the reaction, and then on the right side. This will show which atoms have a different oxidation state after the reaction. Oxidation is when an atom loses electrons, or in other words, when its oxidation state becomes more positive (electrons give an atom a negative charge). Reduction is the opposite. This is when an atom gains electrons, and its oxidation state "reduces", or becomes more negative. After it is determined which atoms are being oxidized, and which are being reduced, the associated "half-reactions" must be written.

To write a half-reaction, all of the atoms associated with the oxidation or reduction must be included. There may be some spectator ions (ions whose oxidation state did not change) left over, and these do not need to be included in the half reactions, but are necessary for the complete reaction. For each half reaction, write the associated reactants on the left side, and the products on the right. The same atoms should be present on each side, even if the reaction has not yet been balanced. The one exception to this rule is in the case of acid-base reactions. For acid-base reactions, the atoms are exchanging electrons as well as protons (H^+). However, there are often excess H^+ ions and H_2O molecules that are necessary to balance each half reactions, and these must be added in during the balancing.

Once each half-reaction is written, they must be balanced. This can be done by following basic chemical reaction balancing rules. First, balance the atoms. Before looking at anything

else, there must be the same number of each type of atom on each side, as matter can neither be created nor destroyed. In the case of acid-base reactions, this may involve adding H^+ ions or H_2O molecules to a certain side. Next, the charges must be balanced. Determine the overall charge of each side, and utilize excess electrons (e^-) to balance out the charges and make them equal. Finally, once both half-reactions are balanced, they must have the same number of electrons being transferred, or in other words, they should have excess electrons on opposite sides of the reactions, and these numbers must be the same so they cancel out. There cannot be 2 electrons on the left side, and 3 electrons on the other side. Instead, multiply the balanced half reactions by the appropriate numbers, in order to get these numbers to cancel out.

After this final balancing step, the half reactions can be added back together to show the complete reaction. Ensuring that everything on the left side stays on the left side, and vice versa, like molecules and the electrons can be cancelled out in order to simplify the total reaction.

The following are step-by-step examples of how to balance a redox reaction using the half-reaction method:

Balancing a Simple Redox Reaction:

Full reaction: $Al(s) + Cu^{2+}(aq) \rightarrow Al^{3+}(aq) + Cu(s)$

Full reaction with oxidation states written in parentheses:

$$Al^{(0)}(s) + Cu^{(+2)2+}(aq) \rightarrow Al^{(+3)3+}(aq) + Cu^{(0)}(s)$$

Oxidation half-reaction: $Al^{(0)}(s) \rightarrow Al^{(+3)3+}(aq)$

Reduction half-reaction: $Cu^{(+2)2+}(aq) \rightarrow Cu^{(0)}(s)$

*Even though in this example, the atoms are already balanced, the charges are not, because in the oxidation reaction, the charge on the left is 0, while the charge on the right is +3. In the reduction

reaction, the charge on the left is 2+, while the charge on the right is 0. Electrons will be used to fix this.

Balanced oxidation half-reaction: $Al^{(0)}(s) \rightarrow Al^{(+3)3+}(aq) + 3e^-$

Balanced reduction half-reaction: $2e^- + Cu^{(+2)2+}(aq) \rightarrow Cu^{(0)}(s)$

*Even though the half-reactions are balanced within themselves, they are not balanced with each other. In order for the electrons to cancel out, the oxidation reaction must be multiplied by 2, and the reduction reaction must be multiplied by 3.

Put them back together: $6e^- + 3Cu^{(+2)2+}(aq) + 2Al^{(0)}(s) \rightarrow 3Cu^{(0)}(s) + 2Al^{(+3)3+}(aq) + 6e^-$

*The electrons on each side will cancel

Balanced Full Reaction: $3Cu^{2+}(aq) + 2Al(s) \rightarrow 3Cu(s) + 2Al^{3+}(aq)$

Balancing a Redox Reaction in an Acidic Solution:

Full reaction: $MnO_4^-(aq) + I^-(aq) \rightarrow I_2(aq) + Mn^{2+}(aq)$

Full reaction with oxidation states written in parentheses:

$Mn^{(+7)}O_4^{(-2)-}(aq) + I^{(-1)-}(aq) \rightarrow I_2^{(0)}(aq) + Mn^{(+2)2+}(aq)$

Oxidation half-reaction: $I^{(-1)-}(aq) \rightarrow I_2^{(0)}(aq)$

Reduction half-reaction: $Mn^{(+7)}O_4^{(-2)-}(aq) \rightarrow Mn^{(+2)2+}(aq)$

*The number of iodine atoms in the oxidation must be balanced; in the reduction reaction, there is one manganese atom on each side, but there are four oxygens on the left, and none on the right. This will be remedied through the use of H_2O molecules and H^+ ions, which would be present in an acidic solution. Electrons will be added to balance charge.

Balanced oxidation half-reaction: $2I^{(-1)-}(aq) \rightarrow I_2^{(0)}(aq) + 2e^-$

Balanced reduction half-reaction: $5e^- + 8H^+(aq) + Mn^{(+7)}O_4^{(-2)-}(aq) \rightarrow Mn^{(+2)2+}(aq) + 4H_2O(l)$

*Though the half-reactions are balanced within themselves, they are not balanced with each other. In order for the electrons to cancel out, the oxidation reaction must be multiplied by 5, and the reduction reaction must be multiplied by 2.

Put them back together:

$$10e^- + 16H^+(aq) + 2Mn^{(+7)}O_4^{(-2)-}(aq) + 10I^{(-1)-}(aq) \rightarrow 2Mn^{(+2)2+}(aq) + 5I_2^{(0)}(aq) + 8H_2O(l) + 10e^-$$

*The electrons will cancel.

Balanced Full Reaction:

$$16H^+(aq) + 2Mn^{(+7)}O_4^{(-2)-}(aq) + 10I^{(-1)-}(aq) \rightarrow 2Mn^{(+2)2+}(aq) + 5I_2^{(0)}(aq) + 8H_2O(l)$$

While the half-reaction method of balancing is effective, redox reactions can be balanced using linear algebra. This is done through matrix row reduction. First, a coefficient matrix must be created to represent the coefficients in front of each chemical in the reaction. In order to simplify the transition from equation form to matrix form and vice versa, all of the coefficients on the right side of the chemical equation should be given a negative sign. This way, all of the coefficients may be treated as though on the left side of the equation, but the negative indicates those that were originally on the right. An essential point to keep in mind is that, no matter what type of redox reaction is being balanced, two additional columns need to be added to this coefficient matrix: one for H^+ ions, and one for H_2O molecules. In the case of balancing a reaction in acidic or basic solution, H^+ ions and/or H_2O molecules may need to be added as part of the balancing process. The initial values of these coefficients will almost always be zero. For each chemical, there will be a row in the matrix corresponding to the charge of the chemical, and each basic atom involved in the reaction.

Once this matrix of coefficients has been constructed, toolkit row operations must be used to row reduce the matrix to its reduced row echelon form. Then, write out each lead variable in terms of the free variables left in the RREF. Once this is done, the Strang's special solutions can be found. Using these, multiply each vector by the necessary scalar in order to get all of the entries into integer form. It should be noted that in the case of the Strang's special solutions, if a number is negative, it now indicates that that particular coefficient belongs on the left side of the equation, since it was moved from the left to the right when solving for Strang's special solutions. Once all of the vector entries are in integer form, the chemical equation can be rewritten with the new, balanced coefficients.

The following shows the previous given examples balanced using linear algebra:

Balancing a Simple Redox Reaction:

Full reaction: $Al(s) + Cu^{2+}(aq) \rightarrow Al^{3+}(aq) + Cu(s)$

Coefficient Matrix:

	Al (s)	Cu^{2+}(aq)	Al^{3+}(aq)	Cu (s)	H^+	H_2O
Charge (e^-)	0	2	3	0	1	0
Al	1	0	1	0	0	0
Cu	0	1	0	1	0	0
H	0	0	0	0	1	2
O	0	0	0	0	0	1

RREF Form of Coefficient Matrix:

1	0	0	2/3	0	0
0	1	0	1	0	0

0	0	1	-2/3	0	0
0	0	0	0	1	0
0	0	0	0	0	1

General Solution Equations: Ax=0

$$x_1 = -\frac{2}{3}x_4 \qquad x_2 = -x_4 \qquad x_3 = \frac{2}{3}x_4 \qquad x_4 = free$$

$$x_5 = 0 \qquad x_6 = 0$$

*in order to cancel out the fractions, set x_4=3

$$x_1 = -2 \qquad x_2 = -3 \qquad x_3 = 2 \qquad x_4 = 3$$

$$x_5 = 0 \qquad x_6 = 0$$

*due to manipulating the equations, the negative coefficients belong on the left side of the equation, and the positive coefficients belong on the right side of the equation

Final Balanced Reaction: $2Al(s) + 3Cu^{2+}(aq) \rightarrow 2Al^{3+}(aq) + 3Cu(s)$

Balancing a Redox Reaction in an Acidic Solution:

Full reaction: $MnO_4^-(aq) + I^-(aq) \rightarrow I_2(aq) + Mn^{2+}(aq)$

Coefficient Matrix:

	MnO_4^- (aq)	I^- (aq)	I_2 (aq)	Mn^{2+} (aq)	H^+	H_2O
Charge (e^-)	-1	-1	0	2	1	0
Mn	1	0	0	1	0	0
I	0	1	2	0	0	0
H	0	0	0	0	1	2
O	4	0	0	0	0	1

RREF Form of Coefficient Matrix:

1	0	0	0	0	1/4
0	1	0	0	0	5/4
0	0	1	0	0	-5/8
0	0	0	1	0	-1/4
0	0	0	0	1	2

General Solution Equations: Ax=0

$$x_1 = -\frac{1}{4}x_6 \qquad x_2 = -\frac{5}{4}x_6 \qquad x_3 = \frac{5}{8}x_6 \qquad x_4 = \frac{1}{4}x_6$$

$$x_5 = -2x_6 \qquad x_6 = free$$

*in order to cancel out the fractions, set x_6=8

$$x_1 = -2 \qquad x_2 = -10 \qquad x_3 = 5 \qquad x_4 = 2$$

$$x_5 = -16 \qquad x_6 = 8$$

*due to manipulating the equations, the negative coefficients belong on the left side of the equation, and the positive coefficients belong on the right side of the equation

Balanced Full Reaction:

$$2Mn^{(+7)}O_4^{(-2)-}(aq) + 10I^{(-1)-}(aq) + 16H^{+}(aq) \rightarrow 2Mn^{(+2)2+}(aq) + 5I_2^{(0)}(aq) + 8H_2O(l)$$

Ultimately, balancing redox reactions by hand is difficult, time-consuming, and error-prone. And yet redox reactions can be represented mathematically and balanced systematically, making it an ideal candidate for digital automation. This is precisely the sort of

problem for which computers were invented and for which software engineering as a problem-solving discipline excels. This is why we wrote a computer program capable of automatically balancing redox reactions. Our software is a valuable tool for making chemist's lives easier.

Our program reduces redox reactions using the linear algebra method. We considered several problem-solving strategies to accomplish this. One good option was to have our program interact directly with Maple (or similar mathematics software such as Matlab or Mathematica). There exists free code provided by the creators of the Maple software which makes this possible. A benefit of this strategy would be that we as programmers would not need to create any matrix handling code of our own, reducing the risk of errors. However, this strategy would make the program only useable by individuals having Maple, and therefore would lessen the portability of our program. Another solid option was to download a free open-source matrix operation code library. This strategy is smart because it would be just as easy for us to use as Maple, probably just as error-free, and it would not interfere with our program's portability. But we didn't choose this strategy because using an entire matrix library is unnecessary given that we only need to perform certain operations. We wanted to keep our program small. Thus, we decided to create our program by writing the actual matrix operations and row reduction algorithm ourselves in C++. We chose C++ because it is a very powerful programming language that enables easy but specific data manipulation. Writing all the code ourselves is not only challenging but risky because we as programmers are likely to make mistakes which the creators of the tools mentioned above did not. This didn't prove to be an issue though because we found many online examples of matrix handling algorithms on which we based our code.

The majority of our program's work is matrix row reduction. Here is a code snippet from our program:

```
void RowReduce(vector<vector<float>>& fMat)
{
	int lead = 0;
	int numRows = fMat.size();
	int numCols = fMat[0].size();
	int currentCol = 0;
	int maxRow;
	while (lead < numRows && currentCol < numCols)
	{
		// Swap the biggest absolute value entry to lead position
		maxRow = lead;
		for (int i = lead + 1; i < numRows; i++)
		{
			if (abs(fMat[i][currentCol]) > abs(fMat[maxRow][currentCol]))
			{
				maxRow = i;
			}
		}
		swapRows(fMat, maxRow, lead);
		// Scale
		if (fMat[lead][currentCol] != 0)
		{
			multiplyRow(fMat, lead, (1 / fMat[lead][currentCol]));
		}
		// Use row addition to get zero in other column entries.
		for (int i = 0; i < numRows; i++)
		{
			if (fMat[i][currentCol] != 0 && i != lead)
			{
				comboRows(fMat, lead, i, fMat[i][currentCol] * -1);
			}
		}
		 // Determine whether or not to increment lead
		if (fMat[lead][currentCol] == 1)
		{
			Lead++;
		}
		currentCol++;
	}
}
```

This code performs row reduction much the same way it is performed by hand. The only critical difference is that the lead entries are selected by choosing the largest absolute value in the column, so as to reduce arithmetic errors, since the computer can only represent a decimal point number with limited precision. The swap, combo, and multiply matrix operations are also just what one would expect, closely resembling the procedure followed by hand.

```
void swapRows(vector<vector<float>>& fMat, int source, int target)
{
	int numCols = fMat[0].size();
	float temp;
	for (int i = 0; i < numCols; i++)
	{
		temp = fMat[target][i];
		fMat[target][i] = fMat[source][i];
		fMat[source][i] = temp;
	}
}

void multiplyRow(vector<vector<float>>& fMat, int target, float mult)
{
	int numCols = fMat[0].size();
	for (int i = 0; i < numCols; i++)
	{
		fMat[target][i] *= mult;
	}
}

void comboRows(vector<vector<float>>& fMat, int source, int target, float mult)
{
	int numCols = fMat[0].size();
	for (int i = 0; i < numCols; i++)
	{
		fMat[target][i] += fMat[source][i] * mult;
	}
}
```

Much more complicated than the matrix reduction, converting the user's chemical reaction into a matrix and then back is surprisingly difficult. This cannot be easily described with code snippets since the process accounts for more than half of our code. The conversion involves tracking which elements the user's equation refers to, recognizing which numbers in the equations belong with which elements and molecules, assigning numbers to each element based on the entered molecules, and accounting for charge. Afterwords, the result must be translated back into a chemical equation, using details of the equation recorded while creating the matrix. Doing so requires processing the reaction many times, each time deciphering and recording a different aspect.

In conclusion, we are very satisfied with the results of our programming experiment. We have successfully written a program capable of automatically balancing redox reactions. The program is easy to use, runs efficiently, and produces minimal errors.