An open-source, web-based math solver for solving multi-variable equilibrium problems in general chemistry (Ulrich, N., Spudich, T., Kowalski, E., Kalainoff, M.)

Abstract

We discuss our use of SageMathCell, a web-based, open-source math-solver, in conjunction with the systematic method, to solve problems involving multi-variable equilibrium reactions while avoiding the use of simplifying approximations. Students initially face a steep learning curve, but after about a week are able to solve fairly complex equilibrium problems in just minutes. We find this to be an excellent first introduction to coding, which students will likely encounter elsewhere in their academic or work careers. For those instructors who use traditional "ICE" tables but still want to give their students exposure to coding, we also provide a representative example of simple SageMath code.

Introduction

SageMathCell is a web interface for SageMath, an open-source mathematics software system that uses a Python-based language. Our students use SageMathCell to solve problems involving equilibrium reactions using the systematic method. The systematic method relies on an engineering model that incorporates the principles of charge and mole balance to allow one to write a series of algebraic equations to model chemical systems at equilibrium.[1,2,3,4] This method is an extremely versatile alternative to ICE tables as it allows one to solve for multiple unknowns simultaneously while avoiding simplifying approximations. For example, the systematic method accounts for the autoionization of water in problems involving weak acid and weak base equilibria; in contrast, most general chemistry textbooks arbitrarily avoid using dilute solutions of acids and bases and instead choose relatively concentrated solutions of acids and bases with comparatively large K values in order to ignore the contributions of the autoionization of water. Table 1 lists pH values of weak acid solutions calculated using both the systematic method and ICE tables. Acetic acid is a commonly used example in textbooks because it has a relatively high K_a value. Even at very dilute concentrations, the respective calculated pH values are equivalent up to three decimal places. The calculated pH values start to vary significantly, however, for an acid such as hypoiodous acid (K_a = 2.3x10^{-11}), even at fairly high concentrations.

Table 1. Comparison of pH values calculated via the systematic method and ICE tables.

Acid	K_a	$[HA]_{initial}$ (M)	Calculated pH (systematic method)	Calculated pH (ICE table)
Acetic acid	1.8×10^{-5}	1.00	2.373	2.373
		1.00×10^{-1}	2.875	2.875
		1.00×10^{-2}	3.382	3.382
		1.00×10^{-3}	3.902	3.902
		1.00×10^{-4}	4.464	4.464
		1.00×10^{-5}	5.145	5.145
Hypoiodous acid	2.3×10^{-11}	1.00	5.319	5.319
		1.00×10^{-1}	5.818	5.819
		1.00×10^{-2}	6.310	6.319
		1.00×10^{-3}	6.741	6.819
		1.00×10^{-4}	6.955	7.319
		1.00×10^{-5}	6.995	7.819

We first introduce our students to the systematic method and SageMath using problems involving a single equilibrium reaction. Below is a representative example of student work using the systematic method to solve a simple gas phase equilibrium problem with three variables (figure 1):

Example 1: A chemist adds 5.0 *M* hydrogen gas and 4.0 *M* iodine gas into a closed container. Determine the concentrations of all the gases once the system has achieved equilibrium.

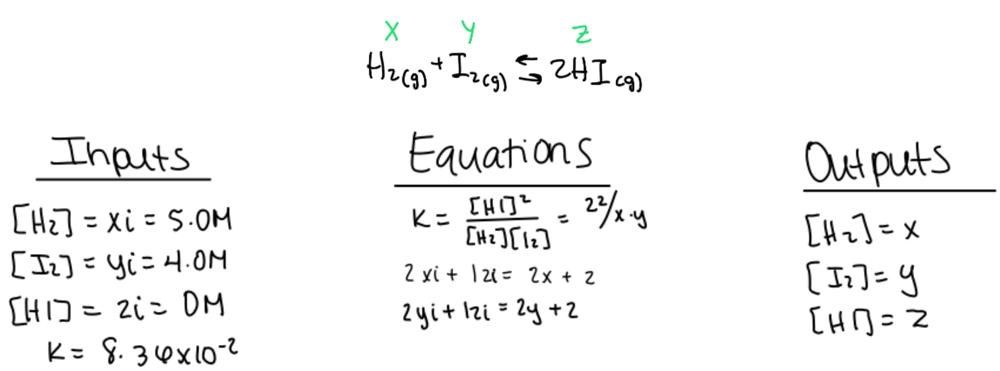

Figure 1. Student work using the systematic method to solve a simple equilibrium problem.

After assigning variables to each species, the student defined the input (initial) and output (equilibrium) concentrations, along with the known K_c value. Because there are three unknowns, the student then wrote three equations: the K_c expression and mole balance equations for hydrogen and iodine.

Below is the corresponding SageMathCell code for the above problem (Figure 2):

Type some Sage code below and press Evaluate.

```
var('x,y,z')
xi=5.0
yi=4.0
zi=0
K=8.35e-2
eq1=K==z^2/(x*y)
eq2=2*xi+zi==2*x+z
eq3=2*yi+zi==2*y+z
solns = solve([eq1,eq2,eq3],x,y,z, solution_dict=True)
[[s[x].n(30), s[y].n(30), s[z].n(30)] for s in solns]
```

Evaluate

Language: Sage

Share

```
[[5.7559445, 4.7559445, -1.5118891], [4.4359360, 3.4359360, 1.1281281]]
```

Help | Powered by SageMath

Figure 2. SageMathCell code to solve the equations developed in Figure 1.

Students are provided with the last two lines of code as a template (these lines tell SageMath to output solutions in decimal form). Students write the remainder of the code themselves by simply listing the variables and equations they wrote. SageMath often provides multiple solution sets, including any negative solutions, and students must choose the solution set that makes physical sense and report their answers in terms of the actual chemical species:

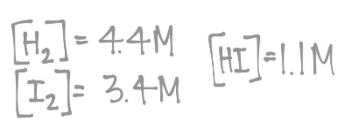

Figure 3. Solutions to Example 1 calculated from equations in Figure 1 and SageMathCell code in Figure 2.

It generally takes students about a week to become proficient at using SageMath. At first, students often make syntax errors because they forget to include coding elements like multiplication symbols, double equal signs, etc. Frustration levels are generally high during this week as this is usually the first time students have encountered coding, but this frustration usually subsides as students grow accustomed to using the software. Even given this initial steep learning curve, however, we find this to be an excellent first introduction to coding, which students will likely encounter at some other point in their academic or work careers.

Within 2-3 class periods, students are ready to move on to more complex equilibrium systems (*e.g.*, pH buffers and acid-base titrations), where they are required to account for each reaction occurring in aqueous solution. Below is a representative example of student work using the systematic method to solve a pH buffer problem (figure 4):

Example 2: Find the pH after adding 0.025 L of 1.20 M HCl to a 1.200 L buffer solution that consists of 1.3 M sodium phenolate (C_6H_5ONa) and 1.2 M phenol (C_6H_5OH) (pK_a of phenol = 10.00).

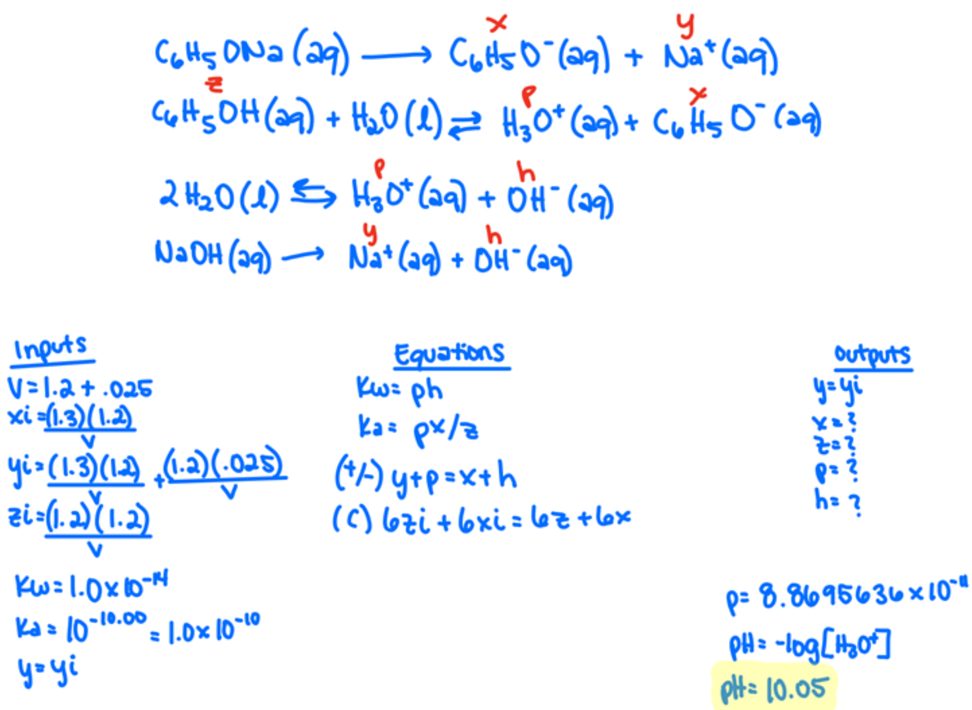

Figure 4. Student work using the systematic method to solve a more complicated buffer problem.

The student first wrote the equations for each reaction occurring in solution: the dissociation of a salt, the equilibrium reaction of a weak acid in water, the autoionization of water, and the dissociation of a strong base. The student then assigned variables to each species in solution, determined the input and output concentrations (while accounting for dilution), and wrote four equations to solve for the four unknowns.

Below is the corresponding SageMathCell code:

Type some Sage code below and press Evaluate.

```
var('x,z,p,h,pH')
V=1.2+0.025
xi=1.3*1.2/V
yi=1.3*1.2/V + 1.2*0.025/V
y=yi
zi=1.2*1.2/V
Kw=1.0e-14
Ka=1.0e-10
eq1=Ka==p*x/z
eq2=Kw==p*h
eq3=y+p==x+h
eq4=6*zi+6*xi==6*z+6*x
eq5=pH==-log(p,10)
solns = solve([eq1,eq2,eq3,eq4,eq5],x,z,p,h,pH, solution_dict=True)
[[s[x].n(30), s[z].n(30), s[p].n(30), s[h].n(30), s[pH].n(30)] for s in solns]
```

Evaluate

Language: Sage

Share

```
[[2.4491923,
  -0.00021274512,
  -8.6863378e-15,
  -1.1512331,
  14.061163 - 1.3643764*I],
 [1.2978464, 1.1511331, 8.8695636e-11, 0.00011274512, 10.052098]]
```

Help | Powered by SageMath

Figure 5. SageMathCell code to solve the equations developed in Figure 4.

This code allows for the direct calculation of the pH (the last solution in set 2).

This is as complex as the problems get for general chemistry students at West Point and Maryville. Students in Maryville's analytical chemistry courses use SageMath to solve for up to twenty unknowns simultaneously. Because SageMath can also complete dilution and pH calculations, students do not have to rely on multiple computing devices. Additionally, once students have written code for the first step of a multi-part problem, they can easily modify the code to incorporate any new parameters. This is especially useful in titration problems, where typically using ICE tables there is a significant amount of work associated with each different point in the titration. Using the systematic method, students can solve for the pH at multiple points in an acid-base titration in just a few minutes.

If you are loathe to relinquish ICE tables, but still want to expose your students to simple coding in a chemistry class, your students could still use SageMath to bypass any algebra involving successive approximations or the quadratic equation. Here is what the SageMath code would look like for example 1 (Figure 1) using ICE tables:

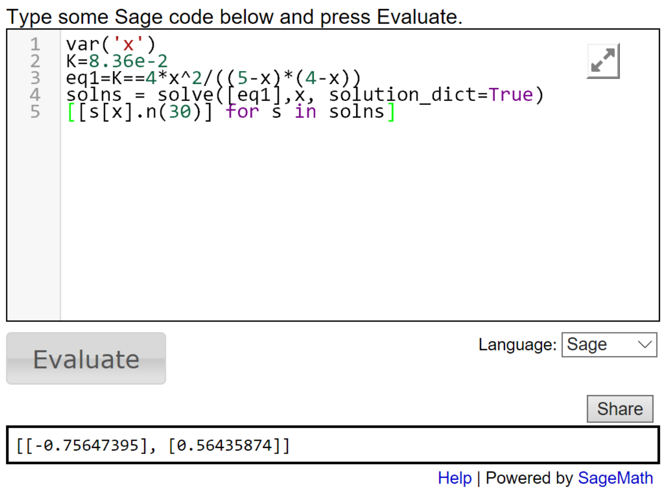

Figure 6. SageMathCell code to solve an ICE table for Example 1.

Your students could then use the above value of x (0.56 *M*) to calculate the actual equilibrium concentrations of each species (or you could even have SageMath do that work for you).

In conclusion, the systematic method, in conjunction with SageMath, are powerful tools for solving multi-variable problems involving chemical equilibrium reactions. Although students initially encounter a steep learning curve, with practice they can become proficient at solving even relatively complex equilibrium problems. This exposure to an engineering model for solving for multiple unknowns also helps students hone their problem-solving skills and provides students with a first experience in coding.

---

[1] de Levie, R. (1999). *Aqueous acid-base equilibria and titrations*. Oxford University Primers, v. 80. New York: Oxford University Press.

[2] Kalainoff, M., Lachance, R., Riegner, D., Biaglow, A. “A computer algebra approach to solving chemical equilibria in general chemistry.” *Primus: Problems, Resources & Issues in Mathematics Undergraduate Studies*; May/Jun2012, Vol. 22 Issue 4, p284-302, 19p

[3] Kalainoff, M., Ulrich, N., Kowalski, E., Spudich, T. *Chemical Equilibrium using the Systematic Method*, unpublished document.

[4] Baeza-Baeza, J. J., García-Álvarez-Coque, M. C. *J. Chem. Educ*. 88, *2*, 169-173.