JOURNAL OF CHEMICAL EDUCATION

pubs.acs.org/jchemeduc

Laboratory Experiment

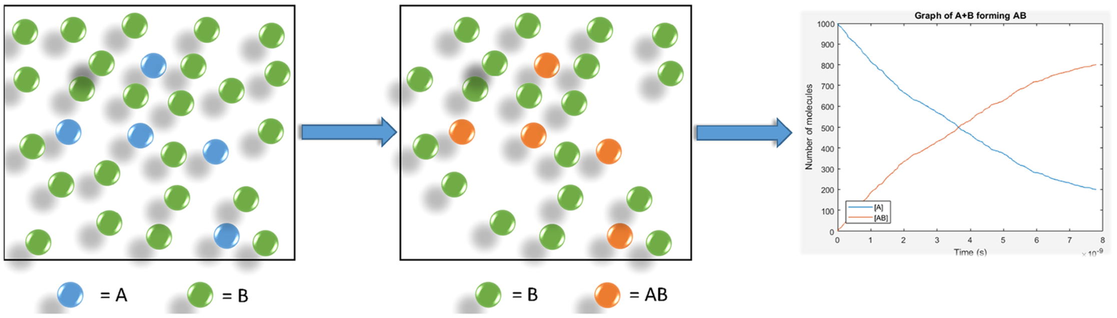
Developing the Chemist's Inner Coder: A MATLAB Tutorial on the Stochastic Simulation of a Pseudo-First-Order Reaction

Aidan A. E. Fisher*

Cite This: *J. Chem. Educ.* 2020, 97, 1476−1480

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information



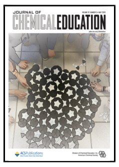
ABSTRACT: The drive in computational methods and more intuitive software has seen a rise in the number of publications in this area in recent years. Computational simulations can be found in many areas of science from computational biology and chemistry to fundamental physics. These may help synthetic chemists in their drug discovery endeavors and cosmologists predicting astronomical events. This paper is designed to equip chemists with a basic understanding of how loops and conditional statements can be used within both the MATLAB syntax and Excel spreadsheets to simulate dynamic processes. Many commercial software packages require little to no programming and often do not explicitly display the underlying calculations. While these programs are often very efficient for solving a specific problem, they are somewhat limited. Principally, this practical uses a stochastic simulation algorithm to generate the kinetic data of a reversible association reaction. The kinetic data is analyzed using the standard rate laws to highlight to students the effectiveness of this method. Ultimately, the skills developed in this practical will help students in future computational projects, where bespoke coding is necessary.

KEYWORDS: *Computational Chemistry, Kinetics, Physical Chemistry*

■ INTRODUCTION

The advent of more powerful computers and efficient software has, in recent years, led to a surge in publications which involve computer simulations. These simulations are important for building and testing scientific models in a quantitative fashion. Indeed, computational simulations have found application in areas ranging from stellar synthesis,[1] on an astronomical scale, to the modeling of nanoscale quantum dot kinetics[2] utilizing a stochastic simulation algorithm (SSA) such as the one presented in this paper. The stochastic simulation algorithm compared with standard Monte Carlo (MC) simulations is particularly well-suited to quantum dot kinetics primarily due to the fluctuating rate constant, which leads to slow computational times using conventional MC kinetics. It is evident that students wishing to pursue a more theory-based approach to chemistry must be comfortable with using and understanding coding ideas in order to simulate chemical systems. Furthermore, the logical thinking required for effective programming is an invaluable skill both in academia and beyond. Several projects exist in the current literature, which introduce students to the ideas of chemical kinetics employing software such as Excel,[3−5] which exploits the wide availability of Excel over more specialized software packages. This increased availability creates greater opportunity to reach out to a wider audience; an Excel file is included in the Supporting Information in this paper. While Excel is advantageous for an elementary understanding of input variables and relatively simple numerical manipulation, it is not as well-suited to the simulation of complex kinetics and rapidly becomes cumbersome to use for all but the simplest models. In addition to Excel-based practical experiments, there are examples of kinetics simulators,[6−9] which have been implemented both for elementary understanding and application to real systems for studying enzyme kinetics. While these are an excellent resource for students to learn key concepts and ideas,

Received: January 20, 2020
Revised: March 13, 2020
Published: April 17, 2020



ACS Publications

https://dx.doi.org/10.1021/acs.jchemed.0c00051

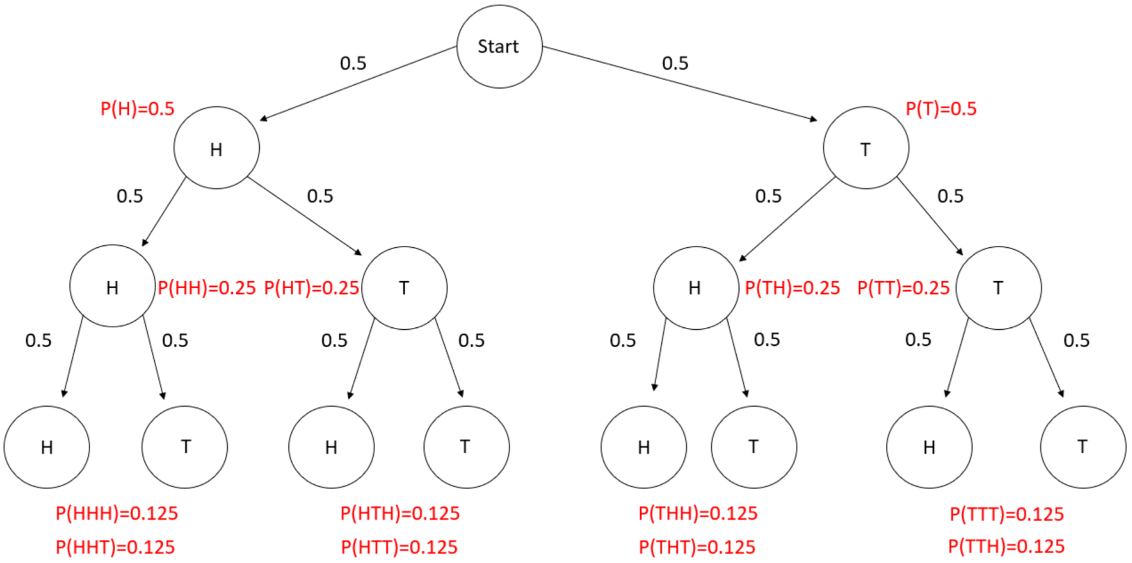


Figure 1. Simple Markovian probability tree diagram. An initial coin toss (start) yields a 50% probability of heads or tails. A second coin toss exhibits the same probability as the initial coin toss of a heads or tails outcome. However, the probability of throwing 2 heads in a row, $P(\mathrm{HH})$, is reduced to 25%. The system is Markovian in the sense that the previous result does not influence any future results. This idea is central to the stochastic simulation algorithm with the caveat that the probabilities of the forward and reverse reactions are not equally probable in the example given in this paper.

they do not offer any options for students to code their own kinetic data, which this paper attempts to develop using the MATLAB and Excel software.

Many physical science disciplines have successfully integrated programming and basic coding skills into their academic structure.[10−12] These methods of teaching are becoming extremely popular in many modern physics and engineering degrees, where some institutes devote entire modules to coding. It is important that, in order to remain competitive with other disciplines, coding skills should continue to be developed within the chemical sciences. This paper outlines the key steps required to successfully simulate the kinetics of a pseudo-first-order reaction using a stochastic simulation algorithm. In general, the code is highly adaptable and can be used to demonstrate other kinetic scenarios such as second-order reactions and non-integer-order reactions. Additionally, the code may be developed from a simple bimolecular process to show the kinetics of termolecular reactions. Several other authors have detected a shortage of coding-focused chemical simulations. To help launch this area, Pierce et al. developed a lab, where data analysis was performed using the MATLAB software.[13] More specifically, they developed a MATLAB analysis code to detect a biodiesel blend composition. The automation of such analyses is becoming more important both in academia and industry. It is important to emphasize here that the ability to perform basic coding is not simply limited to the design of kinetic models and interfacing code with standard lab equipment. More complex bespoke data acquisition can be performed by possessing a basic understanding of electronic components. Undergraduate projects developed by Wilczek-Vera and colleagues[14] focus on educating chemistry undergraduate students about standard electronic components (resistors, capacitors, etc.) with a view to signal processing and data acquisition. It is essential that this area continues to develop and thrive in the modern computational era.

■ EXPERIMENTAL SECTION

The simulation of a simple reversible association reaction was performed using the MATLAB software and in accordance with the scheme below:

$$\mathrm{A} + \mathrm{B} \rightleftharpoons \mathrm{AB}$$

This involved employing the Gillespie stochastic simulation algorithm.[15] The forward and reverse reactions are calculated in a probabilistic manner in accordance with the Markovian nature of the system. Simply put, the diffusing monomers and product associate and dissociate on a time scale which is much slower such that the system "forgets" its previous state. The concept of a Markov chain and its importance in the stochastic simulation algorithm is perhaps best visualized by introducing the familiar analogy of a fair coin toss illustrated by Figure 1. In this scheme, the chance of throwing an initial heads or tails is 50% for each case. The system in this example is Markovian in the sense that the probability of throwing a second heads or tails is again 50% for each case; i.e., the coin does not "remember" its previous state. However, the chance of throwing 2 heads in a row $P(\mathrm{HH})$ is only 25%. The stochastic simulation outlined in this paper exhibits the same Markovian nature as the coin toss example, but the probabilities of the forward and reverse reactions are not equally probable. This idea is explored in more detail below.

For the simple reversible association reaction of A and B to form product AB, the probability that monomer A has not reacted to form the product is given by ($A_{\mathrm{survival}}(t) = \exp(-R_{\mathrm{forward}}t)$), where R_{forward} is total rate toward A forming the product AB. This is particularly useful for the simulation of kinetics, where rates are highly distributed. The time-step until the next successful reaction is calculated from the random sampling of a logarithmic distribution function, which arises from taking the differential of the probability distribution function, $\tau = \int_0^\infty tR_{\mathrm{forward}} \exp(-R_{\mathrm{forward}}t)\ \mathrm{d}t$. In comparison, traditional MC methods are somewhat slow since the algorithm increments a defined time-step periodically regardless of reaction success. As a result, many computational cycles do not further the reaction progress and are wasteful in this respect.

This can become important in reactions where the kinetics are highly distributed, since conventional MC methods typically do not change the time-step size and so may become extremely inefficient at reaction rate extremes. In comparison, the sampling of the logarithmic distribution outlined above circumvents these issues as the logarithmic function scales with reaction rate. A more detailed discussion of these ideas is provided by Voter.[16]

This lab session was used with a chemistry undergraduate class with some limited experience of the MATLAB syntax from a previous MATLAB module.[17] Students worked individually with a lab script detailing the key points in the code with a laboratory demonstrator on hand to ensure that the students remain focused and do not become discouraged. The lab handout provided to the students is included in the Supporting Information.

Importantly, the code may be adapted to suit the level of the class i.e. a more prescriptive set of instructions for unfamiliar students or a minimal text script for more advanced users. Interestingly, the script can be complemented with laboratory real-world examples. The reduction of methylene blue using a ascorbic acid is a good example of this and may be tracked with UV−vis.[18] Of course several other reactions exhibiting a color change diagnostic to track the reaction kinetics may be used, where a literature search suggests the well-known iodine clock[19] and the reaction of food colorants with sodium hypochlorite.[20]

■ PROCEDURE AND RESULTS

In order to conform to MATLAB best practices, arrays are preallocated before beginning any coding. In this step array sizes are defined and populated with zeros to improve computational efficiency. This is achieved by pasting the following into MATLAB:

```
Rf=zeros(1,800);
Rr=zeros(1,800);
Rtot=zeros(1,800);
Pf=zeros(1,800);
Pr=zeros(1,800);
U=zeros(1,800);
X=zeros(1,800);
tau=zeros(1,800);
logA=zeros(1,800);
Linear=zeros(1,800);

A=zeros(1,800);
B=zeros(1,800);
AB=zeros(1,800);
t=zeros(1,800);
```

Several variables must now be defined. This is illustrated by the box at the top of Figure 2. These variables include the initial monomer quantities, [A] and [B]; product population, [AB]; rate constants for the forward, kf, and reverse, kr, reactions; and the initial time, t. This is defined by the annotated code below in MATLAB:

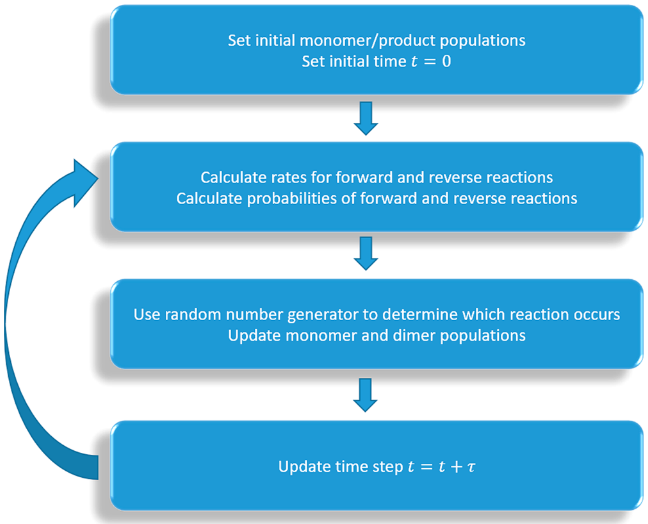
```
A(1)=1000;              %Number of A
B(1)=1000000;           %Number of B

AB(1)=0;                %Number of AB

kf=200;                 %Forward rate constant
kr=2;                   %Reverse rate constant

t(1)=0;                 %Initial time
```

The code then introduces students to a looping statement. This looping statement is illustrated by the curved arrow in Figure 2 linking the processes highlighted by boxes 2−4. Within the loop, the reaction rates and probabilities of both the forward and reverse reaction are calculated. This is achieved by typing the following in MATLAB:


Set initial monomer/product populations
Set initial time $t = 0$

Calculate rates for forward and reverse reactions
Calculate probabilities of forward and reverse reactions

Use random number generator to determine which reaction occurs
Update monomer and dimer populations

Update time step $t = t + \tau$

Figure 2. Outline of the Gillespie algorithm.[15] Step 1: The initial populations of monomers [A] and [B] are set as 1000 and 1×10^6, respectively. The initial population of product AB was zero, and the initial time was set as zero. Step 2: The forward and reverse reaction rates are calculated using the standard rate equations, assuming first-order kinetics with respect to both [A] and [B], and [AB]. The rate constant for the forward reaction is defined by kf = 200 molecule^{-1} s^{-1}, and the reverse reaction rate constant is given by kr = 2 s^{-1}. Step 3: A random number, defined by U in the code, was generated to stochastically determine which reaction occurs. The populations are updated depending on if the forward or reverse reaction occurred. Step 4: The time increment was updated by sampling from a logarithmic distribution. The process was repeated 800 times in this paper.

```
for L=1:800                         %Opens loop 800 iterations

        Rf(L)=kf*A(L)*B(L);         %Forward rate equation

        Rr(L)=kr*AB(L);             %Reverse rate equation

        Rtot(L)=Rf(L)+Rr(L);        %Total rate

        Pf(L)=Rf(L)/Rtot(L);        %Probability of forward reaction

        Pr(L)=Rr(L)/Rtot(L);        %Probability of reverse reaction
```

At time $t = 0$, the probability of the reverse reaction is zero since no product has yet formed. The probability of the forward reaction is unity. The starting material and product populations are subsequently updated following the forward reaction. In the second iteration of the loop, the product population is no longer zero, and there exists a small probability of the reverse reaction occurring, i.e., the dissociation of product AB into its constituent monomers. The probabilities of both the forward and reverse reactions are summed together, where the total probability equates to unity. A random number is generated within MATLAB to decide which reaction will occur. Making this decision introduces the students to the conditional statement and is shown below:

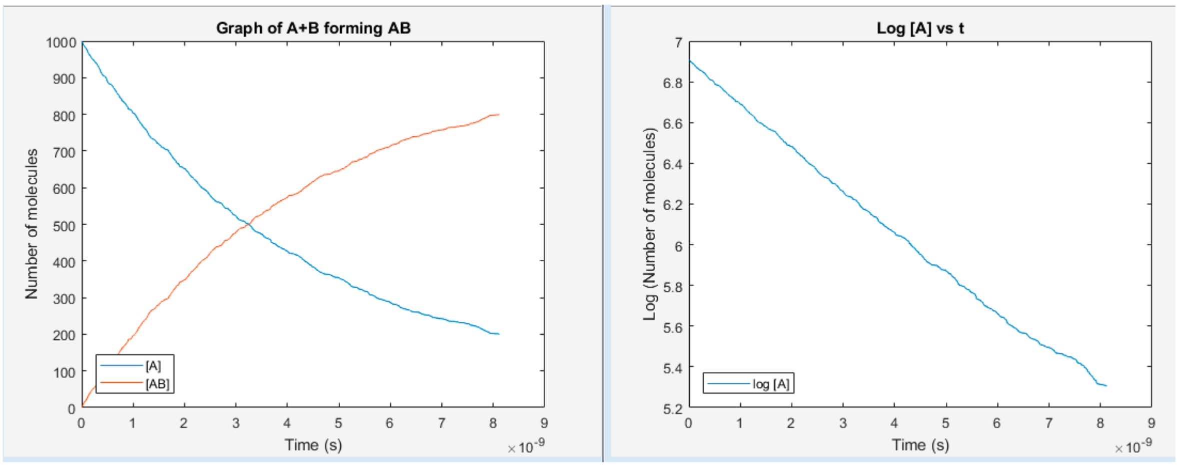


Figure 3. Results of the stochastic simulation for a simple reversible association reaction. Left: Change in the number of both the monomer [A] and product [AB]. Initially, the population of [A] was set to 1000, and the population of [AB] was set to zero. Initially, the reverse reaction (dissociation) is not possible, and the forward reaction (association) must occur. As the [AB] product forms, the reverse reaction becomes possible. The forward reaction, over the time scale presented, dominates since the rate constant is 100-fold larger. The population of [B] effectively remains constant for the reaction since the population of $[B] \gg [A]$ and is omitted for clarity. Right: Processed kinetic data. Here, kinetic data is presented by taking the ln of monomer [A] and plotting against time. Importantly, the gradient yields the pseudo-rate-constant 2×10^8 s^{-1}, where the intercept can be used to calculate the initial population of monomer [A].

```
U(L)=rand;                  %Generates random number [1,0] interval

Random=U(L);                %Assigns the random number to the variable

if Random <= Pf(L)          %Conditional statement


     A(L+1)=A(L)-1;    %Forward reaction occurs population of A decreases

     B(L+1)=B(L)-1;    %Forward reaction occurs population of B decreases

     AB(L+1)=AB(L)+1;  %Forward reaction occurs

else
     A(L+1)=A(L)+1;   %Reverse reaction occurs population of A increases

     B(L+1)=B(L)+1;   %Reverse reaction occurs population of B increases

     AB(L+1)=AB(L)-1; %Reverse reaction occurs
end
```

As more of the product forms, the reverse reaction becomes more likely in subsequent iterations of the loop. The final step in each loop iteration involves the calculation of the time-step until the next reaction. The time-step to the next reaction is calculated by sampling a logarithmic distribution as indicated by the final box in Figure 2 and as described above. This is achieved using the following:

```
X(L)=rand;                  %Generate random number in the [1,0] interval

tau(L)=-log(X(L))/Rtot(L); %Generate time step

t(L+1)=t(L)+tau(L);         %Add time step to previous time
```

Upon the successful simulation of the kinetics, the students may then perform typical rate law analyses on the data either within MATLAB or exported into Excel. The raw simulated data from 800 data points is shown in Figure 3. The large rate constant for the forward reaction dominates, on the time scale shown here, leading to more rapid association of A and B to product AB compared to dissociation to starting materials A and B. The population of [B] is omitted from Figure 3 for clarity and because the population does not change significantly. The raw data was processed in MATLAB to produce the expected linear relationship by using the integrated rate law for pseudo-first-order kinetics and is shown in Figure 3. In summary, student learning was fostered by the use of a lab handout, which included a detailed prelab reading section accompanied by useful links to the MATLAB website. While it was important that students familiarized themselves with the syntax and several key concepts prior to the session, most benefit from actually putting these ideas into practice by writing their own code. In a follow-up set of questions, the students were asked to validate if the gradient and intercepts for their processed integrated rate law plots agreed with the MATLAB input values. Students were also asked to modify their code to simulate pseudo-second-order kinetics inherently demonstrating some understanding of the coding ideas developed. The practical appeared to be well-suited for the undergraduate level taught, since all 27 participants successfully completed the practical, including all additional questions in the lab handout, indicating a good grasp on ideas developed. Importantly, this demonstration is highly adaptable and can be tailored to suit a variety of students' abilities and multiple chemical scenarios. The same algorithm is used to generate the kinetic data using the Excel software. This is given in the Supporting Information.

■ CONCLUSION

This practical allows students to explore the usefulness of coding by implementing looping and conditional statements. These functions are common to many programming languages and comprise a vital transferrable skill. The practical was successfully used in an undergraduate physical chemistry module, which explored phases, changes, and rates of chemical systems. The script can readily be modified to match students' computational expertise. For example, the script can be adapted to simulate more complex reaction pathways with multiple products for students who are comfortable with basic coding ideas. Beyond learning coding skills, this practical teaches students pros and cons of different algorithm models. Using a similar computational framework to that of this paper, the code may be adapted to implement a MC type simulation, where students may compare computational calculation times. In sum, this practical is a useful introductory tool to more complex coding features and fits well within typical undergraduate kinetics modules. It also helps to fill a void in current chemical education, where many undergraduate chemists can pass through their entire degree with no experience of coding.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available at https://pubs.acs.org/doi/10.1021/acs.jchemed.0c00051.

Developing the chemist's inner coder document (PDF, DOCX)
In-class lab handout (PDF, DOCX)
Kinetic data for SSA in Excel format (XLSX)
Kinetic data for SSA in MATLAB format (ZIP)

AUTHOR INFORMATION

Corresponding Author

Aidan A. E. Fisher − *University of Sussex, Department of Chemistry, Falmer BN1 9RH, United Kingdom;* orcid.org/0000-0003-2790-0749; Email: aidan.fisher@sussex.ac.uk

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jchemed.0c00051

Notes

The author declares no competing financial interest.

ACKNOWLEDGMENTS

The author would like to thank the students who participated in the use of this practical. The author also extends his thanks to Mark Osborne for the introduction to the SSA.

REFERENCES

(1) Pasetto, S.; Chiosi, C.; Kawata, D. Theory of stellar population synthesis with an application to N-body simulations. *Astron. Astrophys.* **2012**, *545*, A14.

(2) Osborne, M. A.; Fisher, A. Charge-tunnelling and self-trapping: common origins for blinking, grey-state emission and photoluminescence enhancement in semiconductor quantum dots. *Nanoscale* **2016**, *8* (17), 9272−9283.

(3) Loyson, P. Teaching Kinetics Using Excel. *J. Chem. Educ.* **2010**, *87* (9), 998.

(4) Muranaka, K. An Excel Program to Study First-Order Kinetics. *J. Chem. Educ.* **2002**, *79* (1), 135.

(5) Denton, P. Analysis of first-order kinetics using Microsoft Excel solver. *J. Chem. Educ.* **2000**, *77* (11), 1524−1525.

(6) Bentenitis, N. A convenient tool for the stochastic simulation of reaction mechanisms. *J. Chem. Educ.* **2008**, *85* (8), 1146−1150.

(7) Xie, Q.; Tinker, R. Molecular dynamics simulations of chemical reactions for use in education. *J. Chem. Educ.* **2006**, *83* (1), 77−83.

(8) Clark, A. G. Lucenz simulator: A tool for the teaching of enzyme kinetics. *Biochem. Mol. Biol. Educ.* **2004**, *32* (3), 201−206.

(9) González-Cruz, J.; Rodríguez-Sotres, R.; Rodríguez-Penagos, M. On the convenience of using a computer simulation to teach enzyme kinetics to undergraduate students with biological chemistry-related curricula. *Biochem. Mol. Biol. Educ.* **2003**, *31* (2), 93−101.

(10) Ibrahim, D. Engineering simulation with MATLAB: improving teaching and learning effectiveness. *Procedia Computer Science* **2011**, *3*, 853−858.

(11) Behrens, A.; Atorf, L.; Schwann, R.; Neumann, B.; Schnitzler, R.; Balle, J.; Herold, T.; Telle, A.; Noll, T. G.; Hameyer, K.; Aach, T. MATLAB meets LEGO Mindstorms—A freshman introduction course into practical engineering. *IEEE Trans. Educ.* **2010**, *53* (2), 306−317.

(12) Lyshevski, S. E. *Engineering and Scientific Computations Using MATLAB*; John Wiley & Sons: Hoboken, NJ, 2005; pp 1−5.

(13) Pierce, K. M.; Schale, S. P.; Le, T. M.; Larson, J. C. An Advanced Analytical Chemistry Experiment Using Gas Chromatography−Mass Spectrometry, MATLAB, and Chemometrics To Predict Biodiesel Blend Percent Composition. *J. Chem. Educ.* **2011**, *88* (6), 806−810.

(14) Antler, M.; Salin, E.; Wilczek-Vera, G. Teaching data acquisition. An undergraduate experiment in the advanced analytical chemistry laboratory. *J. Chem. Educ.* **2005**, *82* (3), 425−427.

(15) Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **1976**, *22* (4), 403−434.

(16) Voter, A. F. In *Radiation Effects in Solids*; Springer: Dordrecht, 2007; pp 1−23.

(17) Fisher, A. A. E. An Introduction to Coding with Matlab: Simulation of X-ray Photoelectron Spectroscopy by Employing Slater's Rules. *J. Chem. Educ.* **2019**, *96* (7), 1502−1505.

(18) Mowry, S.; Ogren, P. J. Kinetics of Methylene Blue Reduction by Ascorbic Acid. *J. Chem. Educ.* **1999**, *76* (7), 970−973.

(19) Creary, X.; Morris, K. M. A new twist on the iodine clock reaction: Determining the order of a reaction. *J. Chem. Educ.* **1999**, *76* (4), 530−531.

(20) Arce, J.; Betancourt, R.; Rivera, Y.; Pijem, J. The reaction of a food colorant with sodium hypochlorite: a student-designed kinetics experiment. *J. Chem. Educ.* **1998**, *75* (9), 1142−1144.