

MATLAB-BASED TEACHING MODULES IN BIOCHEMICAL ENGINEERING

KILHO LEE, NOELLE K. COMOLLI, WILLIAM J. KELLY, AND ZUYI (JACKY) HUANG
Villanova University • Villanova, PA 19085

Mathematical models play an important role in biochemical engineering. For example, the models developed in the field of systems biology have been used to identify drug targets to treat pathogens such as *Pseudomonas aeruginosa* in biofilms.^[1] In addition, competitive binding models for chromatography processes have been developed to predict expanded bed breakthrough of a recombinant protein expressed in a high cell density fermentation.^[2] Numerical computing tools such as MATLAB can facilitate the development of these models in biochemical engineering. Compared to other computational tools and programming languages, such as C++ and Java, MATLAB is easier for students to learn since it has a large library of functions and toolboxes, including some developed in systems biology.^[3] MATLAB has been widely incorporated in advanced courses in certain engineering disciplines, such as mechanical engineering, to enhance the students' learning experience. For example, Simulink and MATLAB were introduced by Azemi and Yaz^[4] to analyze and solve systems involving non-linear differential equations in a mechanics course. Dabney and Ghorbel^[5] further demonstrated the integration of MATLAB and Simulink into engineering mechanics courses by providing MATLAB tutorial sessions, designing MATLAB/Simulink homework problems, and assigning a course project related to dynamical systems modeling.

In contrast, training in MATLAB is not extensively provided in advanced biochemical engineering courses at most U.S. universities.^[6] When MATLAB has been used, it has mainly been introduced to solve equations such as the Peng-Robinson equation of state^[7] or equilibrium-staged separations,^[8] to perform stochastic simulations with the Gillespie algorithm on a simple reversible chemical reaction,^[9] to acquire data and implement process control,^[10] to solve simple ODE equations for heat transfer experiment design,^[11] or to determine the stoichiometric coefficients for chemical equations.^[12] At Villanova, it was found that students enrolled in CHE8663 Systems Biology self-evaluated their MATLAB knowledge prior to taking the course as poor (1.4/5; refer to the Teaching Strategies section for survey detail). This is most likely due to students' limited exposure to MATLAB in biochemical engineering courses. Therefore, it is necessary to explore the training of biochemical engineers in MATLAB. This paper focuses on the use of MATLAB in systems biology, a rising new field requiring advanced mathematical modeling skills.

There are some textbooks^[13,14] and papers^[15] introducing general problem-solving instruction of MATLAB in biochemical engineering. However, the techniques essential to model complicated biochemical reaction networks—such as ODE model simulation, sensitivity analysis, and parameter estimation—have not been covered in these textbooks and papers. In addition, while some textbooks introduce the MATLAB programs for metabolic modeling,^[16] few textbooks in systems biology provide MATLAB implementation examples for modeling signal transduction pathways that describe how activities of intracellular molecules are adjusted to extracellular stimulations. In this work, we will use the Interleukin 6 (IL-6) pathway, which is extensively studied in systems biology, as an example to illustrate the application of MATLAB in a typical modeling procedure for biochemical reaction networks. The modules introduced in the example include: 1) the ODE model simulation; 2) identification of important parameters in the model via sensitivity analysis; 3) estimation of important model parameters from experimental data; and 4) evaluation of the influence of feedback loops on the system behavior. In addition, teaching strategies used by professors at Villanova University for both on-site and distance-learner students are provided. Surveys were conducted to evaluate the efficacy of these teaching strategies, and the results are also included.

Kilho Lee was a Master's student in the Department of Chemical Engineering at Villanova University. He developed metabolic modeling approaches to engineer *Escherichia Coli* for generating 3-hydroxypropionic acid from glycerol. He is now a Ph.D. student in the Department of Chemical Engineering at Ohio State University.

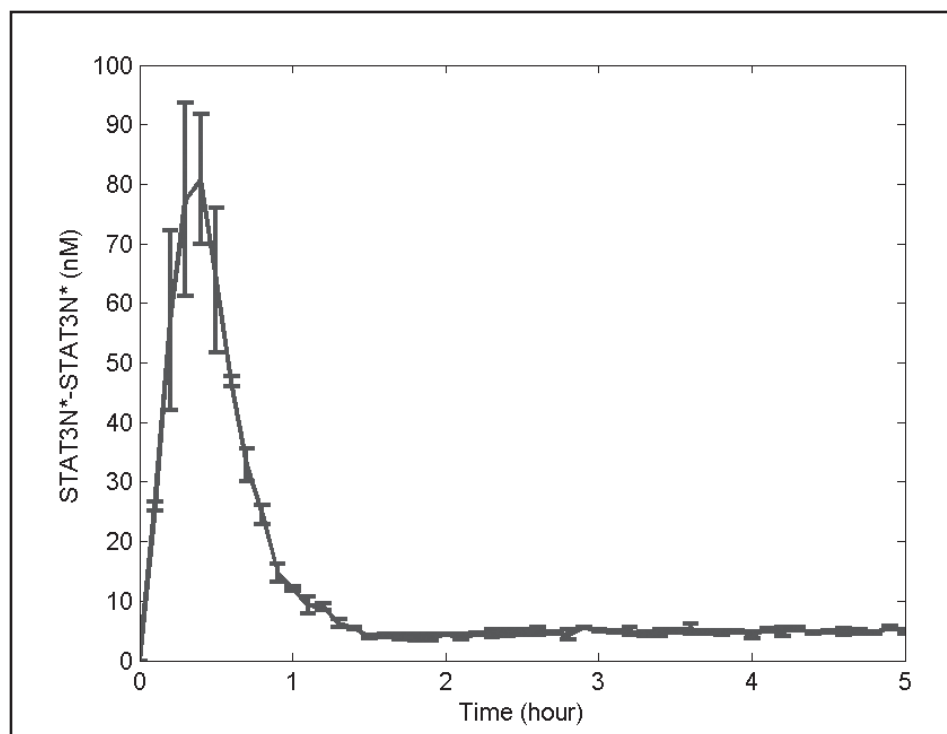
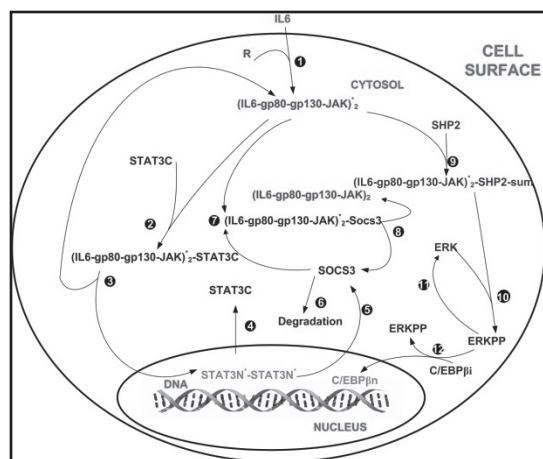
Noelle K. Comolli is an associate professor in the Department of Chemical Engineering at Villanova University. She is the founder and director of Biomaterials and Drug Delivery Lab at Villanova University. She is teaching courses Biomaterials and Drug Delivery.

William J. Kelly is an associated professor in the Department of Chemical Engineering at Villanova University. He is the founder and director of Biotechnology and Bioprocess Lab at Villanova University. He is teaching courses Biochemical Engineering I & II. Dr. Kelly worked for Merck Vaccines prior to starting at Villanova in 1999. His research includes upstream and downstream bioprocessing of biopharmaceuticals. He has focused recently specifically on the production of monoclonal antibodies and plasmid DNA (for gene therapy applications).

Zuyi (Jacky) Huang is an assistant professor in the Department of Chemical Engineering at Villanova University. He is the founder and director of the Biological and Environmental Systems Engineering Lab at Villanova University. He is teaching senior students Process Control and graduate students Systems Biology.

AN EXAMPLE FOR MATLAB TRAINING IN SYSTEMS BIOLOGY

Before introducing these MATLAB implementations, a general tutorial on MATLAB was given (refer to Supplemental Document 1), which covers the algebraic operation, script function development, flow and loop control, logical operation, and basic algorithms for searching and sorting. The introduction of MATLAB implementation starts with the instruction on simulating the ODE model for a simple enzymatic reaction (*i.e.*, $S + E \leftrightarrow SE \rightarrow P + E$). Once students have the basic knowledge of MATLAB programming skills, the more complex IL-6 signaling pathway is used as an example to



illustrate advanced modeling techniques like sensitivity analysis and parameter estimation. The example for IL-6 signaling is given below and the corresponding MATLAB codes are provided in Supplemental Document 2. This example and the MATLAB codes can be used by students to make their own programs to solve homework problems or design term projects.

MATLAB IMPLEMENTATION FOR MODELING IL-6 SIGNALING PATHWAY

Problem statement

The IL-6 signaling pathway describes the activation of a cascade of biochemical reactions within hepatocyte cells that is stimulated by cytokine IL-6, one of the major inflammatory mediators secreted by leukocyte cells around a wound. The outcome of IL-6 signaling is that transcription factor nuclear STAT3 dimer is regulated to tune the expression of some important proteins such as acute phase proteins to coordinate the response of the human body to the wound. The ODE model for a simplified IL-6 signaling pathway shown in Figure 1A is given. Three reduplicate datasets are provided for the concentration of the nuclear STAT3 dimer in HepG2 cells that are stimulated by 100 ng/ml IL-6 for five hours (Figure 1B). Please determine the four parameters in the ODE model that are the most important to nuclear STAT3 dimer dynamic, estimate these four parameters from the STAT3 datasets, and identify and study the influence feedback loops in the pathway have on nuclear STAT3 dimer behavior. The ODE model, the initially guessed value of model parameters, and the state initial values are given in Supplemental Document 2. The detail of the IL-6 signaling is given in Huang, *et al.*^[17]

Figures 1. The simplified IL-6 signaling pathway (A, top) and the dataset for STAT3N*-STAT3N* from liver cells stimulated by 100 ng/ml IL-6 (B, bottom). The IL-6 signaling consists of three modules: 1) the binding of IL-6 to its receptor to form the receptor complex; 2) the activation of JAK-STAT pathway, in which transcription factor STAT3N*-STAT3N* is activated and triggers the expression of SOCS3 that in turn inhibits JAK-STAT pathway; and 3) the activation of MAPK pathway which begins with the binding of SHP2 to the receptor complex and ends with the activation of transcription factor C/EBP β in the nucleus.

Solution

MATLAB-based ODE solving

The ODE model for the simplified IL-6 signaling pathway consists of 17 parameters and 14 differential equations, each of which quantifies the change rate of a molecule in the pathway. Based upon the differential equations, the implementation of MATLAB to simulate the ODE model consists of the following two steps:

- Develop a MATLAB function, *i.e.*, IL6_model.m, to represent the ODE model.


```
function dxdt= IL6_model(t, x, para, u, R)
% t - time; x - state variable vector;
para - parameter vector; u - IL- 6 concentration; R - receptor concentration
dxdt (1,1) = para (1) * u ^ 2 * R ^ 2 - para (2) * x (1) - para (3) * x (1) * x (2) + para (4) * x (3) + para (5) * x (3) - para (9) * x (1) * x (5) + para (10) * x (6) - para (12) * x (1) * x (7) + para (13) * x (8);
```
- Based on the function IL6_model.m, build a main function, *i.e.*, Main_simulation.m, to solve the ODE model using ODE solvers such as ode15s.


```
[t x]=ode15s(@IL6_model,[0 5*3600],x0,[],para,u,R);
% [0 5*3600] - simulation time span from 0 to 5 hours; x0 - initial values for all state variables; para - parameter value vector; u - IL-6 % concentration; R - receptor concentration
```

While only partial MATLAB codes are presented above, for IL6_model.m and Main_simulation.m, the completed MATLAB codes are provided in Supplemental Document 2. Solver ode15s was used here, as the IL-6 model was stiff, because the dynamic of C/EBP β was much faster than that of STAT3N*-STAT3N*. Another commonly used solver, ode45, can be used for non-stiff systems. Interested readers can refer to MATLAB's online documentation for more detail of the stiff equation solvers.

Sensitivity analysis

Sensitivity analysis can quantify the change of output, *i.e.*, STAT3N*-STAT3N* concentration over the change of the value of a single parameter. The sensitivity measure $s_{i,j}$ is typically defined by Eq. (1), in which the partial derivative of the output y_j over parameter p_i (*i.e.*, reaction rate constants) is normalized by the nominal values of p_i and y_j (*i.e.*, \bar{p}_i and \bar{y}_j , respectively).

$$s_{i,j} = \frac{\bar{p}_i}{\bar{y}_j} \frac{\partial y_j}{\partial p_i} \bigg|_{\bar{p}} \quad (1)$$

where the vector \bar{P} is a vector of nominal values of the 17 parameters in the model. The output of the system, *i.e.*, y_j , is quantified as the norm value of the output profile sampled at specific time points from the simulation result. In this example, STAT3N*-STAT3N* concentration is the only output. The MATLAB code (*i.e.*, Main_sensitivity_analysis.m), as well as the detailed explanation of the code, is provided in Supplemental Document 2. The four most important parameters identified from the sensitivity analysis are p_1 , p_3 , p_6 , and p_{12} . The reason for selecting the most important parameters for re-estimation is because the available data are quite limited for a model with 17 parameters.

Parameter estimation

The selected parameters are re-estimated from the dataset shown in Figure 1B using the non-linear least squares routine in MATLAB via the following steps.

- Develop an objective function, *i.e.*, objective.m, to quantify the sum of squared errors (SSE) between the model predicted output and the experimental data for a set of values of p_1 , p_3 , p_6 , and p_{12} .


```
function SSE = objective(para_est, data)
para(1)= para_est(1);
para(3) = para_est(2);
para(6) = para_est(3);
para(12) = para_est(4);
[t x] = ode15s(@IL6_model,[0 5*3600],x0,[], para, u, R);
y_model = interp1(t/3600,x(:,4),data.sample_t);% sample the simulation output at the time points with measurement.
y_exp = mean(data.STAT3,2)';
SSE = sum((y_model - y_exp).^2);
```
- On the basis of the objective function objective.m, build a main function, *i.e.*, Main_estimation.m, to estimate the values of the four parameters that minimize SSE.

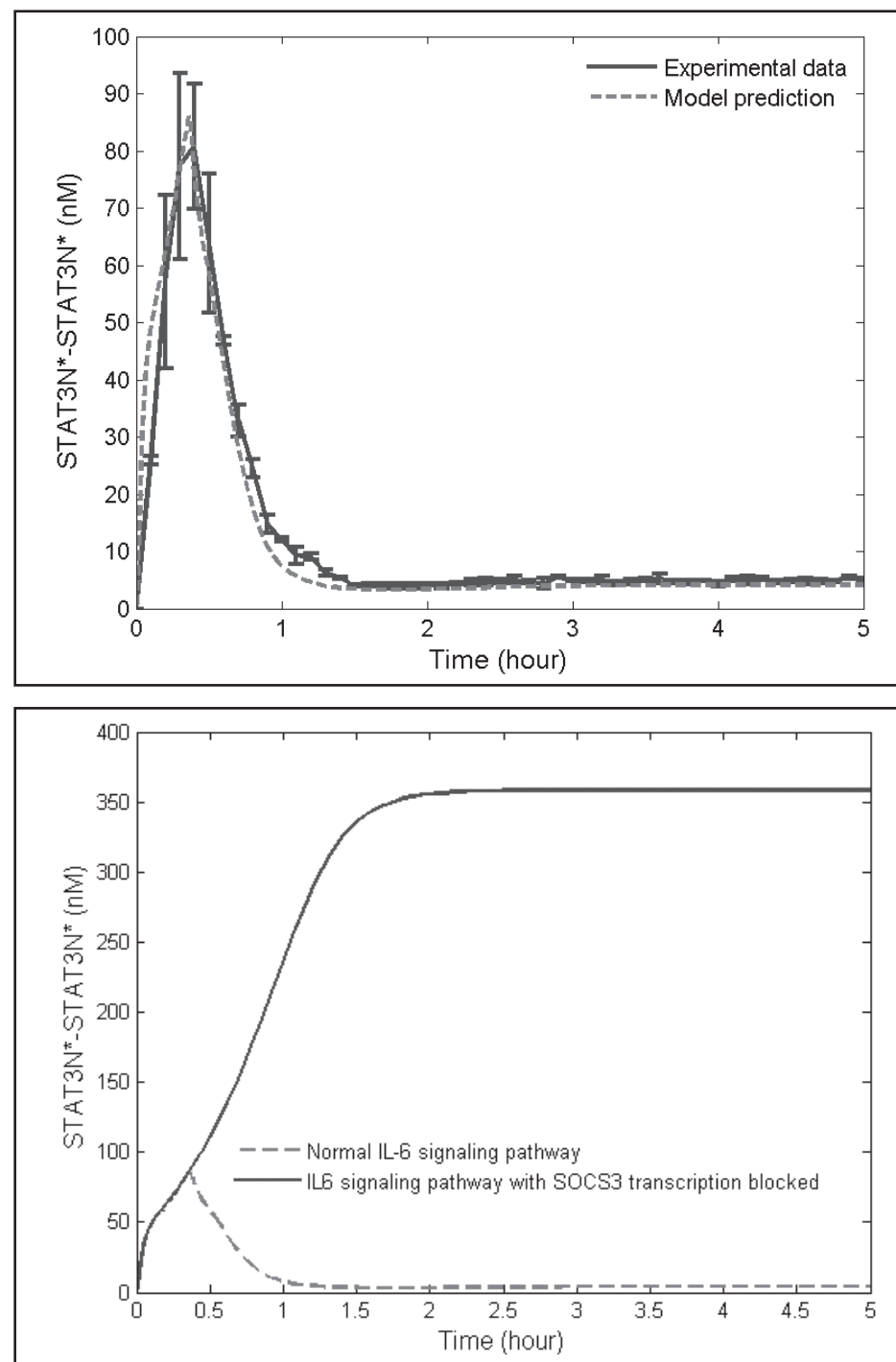

```
p0 = [0.0028 0.0068 0.0087 0.0006];
% p0-initially guessed values of the four parameters
[P_estimate, fval] = fmincon(@(p) objective(p, data),p0,[],[],[],[],[0 0 0 0],[1 1 1 1]);
% [0 0 0 0] and [1 1 1 1] are the lower and upper bounds of the parameter values
```

The MATLAB routine fmincon is used for the parameter estimation in this example, while lsqnonlin is another

commonly used routine. The fitting result for the re-estimated model is shown in Figure 2A.

Analysis of the impact from the feedback loop

The developed IL-6 model can act as an *in silico* platform to investigate the interactions among the reactions in the signaling pathway. Since SOCS3 is produced from JAK-STAT



Figures 2. Comparison of the model prediction for STAT3N*-STAT3N* to the experimental data (A, top) and the STAT3N*-STAT3N* profile with SOCS3 blocked (B, bottom). Since SOCS3 implements a negative feedback on JAK pathway, the inhibition of SOCS3 transcription enhances the concentration of STAT3N*-STAT3N*.

pathway and in turn inhibits JAK-STAT pathway (Figure 1A), it implements a negative feedback loop in the JAK-STAT pathway. The model is used to predict the STAT3N*-STAT3N* profile for liver cells with SOCS3 transcription inhibited. The result shown in Figure 2B explicitly illustrates the influence from the SOCS3 feedback loop on the dynamic of JAK pathway.

TEACHING STRATEGIES FOR TRAINING MATLAB

The aforementioned strategies introduced by Dabney and Ghorbel¹⁵ to train MATLAB were used as background material provided to the students. Specifically, ODE models, problem statements, and the corresponding MATLAB codes were provided to students in a handout. In-class examples such as the one shown in Figures 1 were reviewed and the students were required to type in all the MATLAB codes during class. These were sent to the instructor for review at the end of each class. The MATLAB codes were then used as templates by the students to solve homework problems. Students also completed a larger term project in which they identified a problem to solve, developed ODE models, and wrote all their own MATLAB codes.

Additional teaching strategies, which are listed in Table 1 and mainly include the student-instructor interactions in class via the question and answer section (Item 6 of Table 1) and the in-class assignment activity (Item 7 of Table 1), were also implemented. In addition, homework problem sets were posted before each lecture and due on the day following the next lecture (Item 8 of Table 1). This provided students additional time to discuss the homework problems with the instructor

Item	Survey of student's experience with MATLAB programming
1	Rate your understanding of MATLAB programming to model and analyze a biological system prior to taking CHE 8663.
2	Rate your understanding of MATLAB programming to model and analyze a biological system after taking CHE 8663.
3	Rate the usefulness of MATLAB programming for understanding/modeling complicated biological systems such as signaling pathways and metabolic networks.
Survey of the instructor's teaching strategies on MATLAB programming	
4	The instructor trained MATLAB programming using examples with real biochemical engineering application.
5	Reference MATLAB codes were provided for all examples in the lecture
6	At the end of each lecture, the instructor offered 30 to 45 minutes for students to interact with the instructor and ask him questions related to MATLAB programming.
7	In-class assignments were assigned for students to complete in the lecture.
8	Homework problem sets were posted before each lecture and due on the day following the next lecture.
9	Solutions of homework problems, especially the MATLAB codes, were given.
10	Each student is required to complete a term project in which he should apply MATLAB programming skills.

before finishing the homework assignment. A distance-learning section was also offered for this course (both in 2012 and 2013). Different from on-site learners, distance learners watched the recorded lectures at a time convenient to them and asked questions via email or phone. Distance learners were also expected to turn in their MATLAB codes for the examples shown during the lecture and complete all in-class assignments. One of the challenges mentioned in Azemi and Yaz⁴¹ was ensuring the accessibility of MATLAB. We addressed this challenge by making MATLAB available on students' personal computers. To fully integrate MATLAB into the course, the instructor used illustrative examples for each topic covered in the course throughout the entire semester. In particular, the instructor worked through several full examples of problems in relevant biochemical engineering applications (such as the example shown above) and provided reference MATLAB codes for all examples in the lecture. These examples were developed to engage students in the active learning of MATLAB.

At both the beginning and the end of the semester, a survey was conducted to evaluate the efficacy of our teaching strategies. Table 1 lists several items of the survey, while the entire survey is given in Supplemental Document 3. Most questions in the survey were designed to scale the student response to five levels in which five and one represent the highest/lowest ranking, respectively. Fifteen and seven students took the course Systems Biology (CHE8663) in the semesters of Fall, 2012 and 2013, respectively. Among them were seven distance learners. While the distribution of the student responses for each survey question is given in Supplemental Document 4, the mean value and standard-deviation (represented by the error bar) are plotted for each item in Figure 3, showing: 1) most students found MATLAB useful for understanding/modeling complicated biological systems such as signal transduction

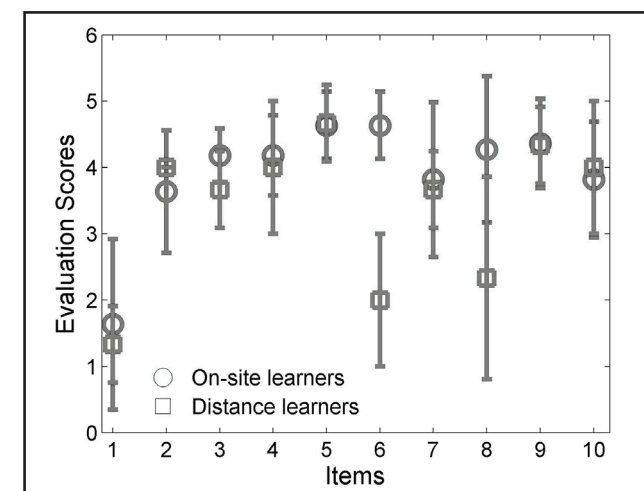


Figure 3. The evaluation scores for each item shown in Table 1. The score for each item ranges from 1 to 5, with the score of 5 representing the highest efficacy/understanding level. There are 14 onsite students and seven distance learners participating in this survey.

pathways; 2) most teaching strategies, *i.e.*, items 4 to 10, were effective for MATLAB training as most evaluation scores for them are close to or above 4 out of 5; 3) students' MATLAB programming skills have been enhanced from level 1.6/1.4 to level 3.7/3.9 for the on-site/distance learners after they took this course. Two sample Kolmogorov-Smirnov (KS) tests¹⁸ were applied to confirm that these improvements are statistically significant with very low P-value (*i.e.*, less than 10^{-3}). The two KS tests were then used to determine the statistical difference between the response of the on-site and distance learners in the 10 items shown in Table 1. The detail of all statistical analysis is provided in Supplemental Document 4. It turns out that the responses of these two groups of students are statistically similar with the exception of the response to Item 6. This can

be explained by the fact that the distance learners didn't have a chance to interact with the instructor in person or ask questions on the homework assignments in class. These student-instructor interaction strategies can be further improved by offering direct email or phone interaction for distance learners in class. In addition, it may be better to ask students to practice the programs once after each example, instead of offering students 30 minutes at the end of the class to re-write the programs. Each student was required to design a term project by him- or herself. Since some students were part-time Master's students not working in a field directly related to systems biology, it was challenging for them to find good term projects. This may explain why the evaluation scores for Item 10 in Figure 3 are slightly below 4. Open-ended suggestions given by students in the survey imply that designing good homework assignments is the most helpful component for MATLAB training. In addition to this survey, the teaching effectiveness was directly evaluated by the instructor via course-embedded assessment components such as homework, in-class assignments, exams, and the term project. These assessments, which are not shown here due to the space constraint, indicate that most students can use MATLAB to formulate and solve modeling problems in systems biology. The instructor also found that students had better performance in modeling biological systems when the instructor implemented MATLAB in teaching.

CONCLUSIONS

MATLAB-based modeling techniques are useful in biochemical engineering, however MATLAB training is not provided by most U.S. universities in this field. To address this, an example using IL-6 signaling was given in this paper to illustrate how MATLAB can be used to simulate an ODE model, determine important parameters for parameter estimation, and investigate the influence of feedback loops on the system's behavior. All these techniques are essential for modeling complicated reaction networks in biochemical engineering. It was found that students' MATLAB programming skills were enhanced significantly after they took the training described above. The teaching strategies developed at Villanova were introduced and evaluated by students. Survey results indicate that most of our teaching strategies worked better for on-site learners. Teaching strategies that can engage distance learners in the student-instructor interaction need to be further investigated. A possible solution may be pre-posting in-class modules and adding phone/email interactions during class with distance learners.

ACKNOWLEDGMENT

We appreciate Dr. Rees B. Rankin's help in polishing the manuscript.

REFERENCES

- Xu, Z., X. Fang, T.K. Wood, and Z. Huang, "A Systems-Level Approach for Investigating *Pseudomonas Aeruginosa* Biofilm Formation," *PLoS One*, **8**(2), e57050 (2013)
- Kelly, W., G. Kamguia, P. Mullen, A. Ubiera, K. Göklen, Z. Huang, and G. Jones, "Using a Two Species Competitive Binding Model to Predict Expanded Bed Breakthrough of a Recombinant Protein Expressed in a High Cell Density Fermentation," *Biotechnology and Bioprocess Eng.*, **18**(3), 546 (2013)
- Schmidt, H., and M. Jirstrand, "Systems Biology Toolbox for MATLAB: a Computational Platform For Research in Systems Biology," *Bioinformatics*, **22**(4), 514 (2006)
- Azemi, A., and E. Yaz, "Utilizing Simulink and MATLAB in a Graduate Nonlinear Systems Analysis Course," 26th Frontier in Education Conference, Salt Lake City, UT, 1996: p. 595- 599
- Dabney, J.B., and F.H. Ghorbel, "Enhancing an Advanced Engineering Mechanics Course Using Matlab and Simulink," *Int. J. Eng. Ed.*, **21**(5), 885 (2005)
- Lee, K., N. Comolli, V. Punzi, W.J. Kelly, and Z. Huang, "Teaching Tip: Results of a Survey on MATLAB & MathCAD Education in Bio-Chemical Engineering," *Chem. Eng. Educ.*, **48**(1), 59 (2014)
- Nasri, Z., and H. Binous, "Applications of the Peng-Robinson Equation of State Using MATLAB®," *Chem. Eng. Ed.*, **43**(2), 115 (2009)
- Binous, H., "Equilibrium-Staged Separations Using Matlab and Mathematica," *Chem. Eng. Ed.*, **42**(2), 69 (2008)
- Argoti, A., L.T. Fan, J. Cruz, and S.T. Chou, "Introducing Stochastic Simulation of Chemical Reactions Using the Gillespie Algorithm and Matlab: Revisited and Augmented," *Chem. Eng. Ed.*, **42**(1), 35 (2008)
- Ricker, N.L., "Using MATLAB/Simulink for Data Acquisition and Control," *Chem. Eng. Ed.*, **35**(4), 286 (2001)
- Garcia, J.G., R. Garcia, E. Aparicio, and M.J. Cocero, "MATLAB: A Powerful Tool For Experimental Design in Chemical Engineering," *Int. J. Eng. Ed.*, **21**(4), 676 (2005)
- Anderson, P.K., and G. Bjedov, "Chemical Stoichiometry Using MATLAB," in 26th Annual Frontiers in Education. 1996. p. 612-615
- Cutlip, M.B., and M. Shacham, *Problem Solving in Chemical and Biochemical Engineering With POLYMATH, Excel, and MATLAB*, 2nd ed., Prentice Hall PTR International Series in the Physical and Chemical Engineering Sciences, Upper Saddle River, NJ (2008)
- Recktenwald, G.W., *Numerical Methods With Matlab: Implementations and Applications*, Upper Saddle River, N.J.: Prentice Hall (2000)
- Mackenzie, J.G., and M. Allen, "Mathematical Power Tools: Maple, Mathematica, MATLAB, and Excel," *Chem. Eng. Ed.*, **32**(2), 156 (1998)
- Palsson, B.O., *Systems Biology: Simulation of Dynamic Network States*, Cambridge University Press (2011)
- Huang, Z.Y., Y.F. Chu, and J. Hahn, "Model Simplification Procedure For Signal Transduction Pathway Models: An Application to IL-6 Signaling," *Chem. Eng. Science*, **65**(6), 1964 (2010)
- Fasano, G., and A. Franceschini, "A Multidimensional Version of the Kolmogorov-Smirnov Test," *Monthly Notices of the Royal Astronomical Society*, **225**(1), 155 (1987)

SUPPLEMENTAL MATERIALS

Interested readers can contact Dr. Zuyi (Jacky) Huang (zuyi.huang@villanova.edu) to get the supplemental materials.

Document 1: A general tutorial on MATLAB

Document 2: The MATLAB codes for the IL-6 example

Document 3: The teaching survey questionnaire

Document 4: Statistical analysis of students' response to the teaching survey □