Flavio Manenti, Gintaras V. Reklaitis (Eds.), Proceedings of the 34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering (ESCAPE34/PSE24), June 2-6, 2024, Florence, Italy
 http://dx.doi.org/10.1016/B978-0-443-28824-1.50584-6

Interactive Coding Templates for Courses and Undergraduate Research Using MATLAB Live Scripts

Ashlee N. Ford Versypt*, Carley V. Cook, Austin N. Johns

Department of Chemical and Biological Engineering, University at Buffalo, The State University of New York, 507 Furnas Hall, Buffalo, NY, 14228, USA
ashleefv@buffalo.edu

Abstract

Undergraduate students in core chemical engineering courses spend a significant amount of time solving problems. For courses or research experiences early in undergraduate study, students generally have not yet taken advanced mathematics, numerical methods, or programming courses, making it challenging to address realistic problems without such tools. The resources provided in this paper aim to enable students, who are still in the early parts of their curriculum, to solve realistic problems in their coursework or research with the aid of faculty-provided interactive coding templates built in the MATLAB live script format. These files combine executable MATLAB code, formatted explanatory text and equations, images, and code output directly in a single file. Here, these MATLAB live scripts are referred to more generically as interactive coding templates because they could alternatively be provided in other coding languages (such as Jupyter Notebooks for Python or Julia). The paper details a set of interactive coding templates for use in training undergraduate students in the introductory chemical engineering material and energy balances course and in an undergraduate research experience on the topic of biomedical applications of systems engineering tools. Each interactive coding template provides background information about the topic, the equations or a diagram defining the technique, an example problem with worked solution, and fully functional code that can solve the example problem and can be extended to new problems that use the same types of numerical methods.

Keywords: MATLAB live script, numerical methods, undergraduate education, undergraduate research.

1. Introduction

Undergraduate students in core chemical engineering courses spend a significant amount of time solving problems. Often assignments and examinations focus on analytical solutions to simplified problems. More complicated realistic problems require the use of computers to determine numerical solutions. For early courses such as material and energy balances (MEB), students often have not yet taken advanced mathematics, numerical methods, or programming courses; learning to program is typically not an explicit learning objective of the core chemical engineering courses. We aim to enable students to solve realistic problems early in the curriculum with instructor-provided interactive coding templates built in the MATLAB live script format. Interactive coding templates typically provide background information about the topic, the equations or a diagram defining the technique, an example problem with a worked solution, fully functional code that can solve the example problem, and a clear pattern or instructional notes for interacting with or editing the code to solve new problems of the same type. Several previous resources have provided interactive coding templates for use in engineering courses, including a conference paper by our team that

focused on providing and surveying MATLAB and Python-based interactive coding templates for courses across the chemical engineering curriculum and training faculty to use these materials (Johns et al. 2023). Readers are encouraged to see the references and detailed table surveying the literature in this previous publication (Johns et al. 2023) and the associated repository of open-access materials that we developed featuring nine interactive coding templates demonstrating numerical methods through examples from MEB, fluid mechanics, heat transfer, separations, thermodynamics, and reaction engineering (Ford Versypt et al. 2022). We also have previously developed graphical user interfaces for use in engineering education (Eastep et al. 2019; Bara et al. 2020); however, a disadvantage of these tools is that users cannot easily modify the original problem statement to adapt to new problems, which is very straightforward using interactive coding templates. Distinct from the previously published materials, our emphasis here is on early undergraduates (first- and second-year students) and providing templates that students can use to solve a variety of problems encountered in their first core chemical engineering course, MEB. Beyond coursework, undergraduates interested in joining research teams in computer-aided process engineering are often discouraged from doing so until they have completed a suite of advanced mathematics courses. To onboard students with limited mathematics and programming backgrounds into research that involves describing dynamic processes through systems of ordinary differential equations (ODEs), we provide the same type of interactive coding templates to reduce the barrier to entry into mathematical systems engineering research.

We developed a set of MATLAB live scripts for training undergraduate students in the introductory chemical engineering MEB course and in an undergraduate research experience on the topic of biomedical applications of systems engineering tools. The MATLAB live script files combine executable MATLAB code, formatted explanatory text and equations, images, and code output directly in a single file. Students are instructed on the interactive coding template and then are assigned various problems to work on their own, starting from the interactive coding template rather than from a blank MATLAB file. The pedagogical emphasis is on the covered topic rather than on learning to program or the details of the numerical methods. For the MEB course, our interactive coding templates focus on solving linear systems of equations. For undergraduate research, the topic is applying conservation balances to populations of cells and amounts of chemical species in living organisms and solving these dynamic problems with systems of ODEs. Undergraduates with a wide range of mathematics and programming backgrounds have successfully used interactive coding templates as they study realistic applications with linear systems of equations and systems of ODEs.

2. MATLAB live scripts for material and energy balances course

Classically in the MEB course systems of linear equations are solved algebraically by hand or via spreadsheets as in three popular textbooks (Felder, Rousseau, and Bullard 2016; Liberatore 2019; Murphy 2023). The material developed here supplements the section in the Liberatore (2019) textbook titled Systems of linear equations, specifically as a MATLAB Live Script-based alternative to the subsection titled “Solving systems of linear equations in a spreadsheet.”

In the MEBLinearSystems repository (Johns and Ford Versypt 2022a), we have three MATLAB live script files for use directly in MEB courses or for introducing systems of linear equations more broadly. The first file is Systems_of_Linear_Equations.mlx (Figure 1), which includes explanatory text and instructions, a worked example, and an

interactive example. The second file contains the corresponding solution to the interactive example and is named Systems_of_Linear_Equations_sol.mlx. We have created an explanatory video (Johns and Ford Versypt 2022b) narrating the solution. Systems_of_Linear_Equations.mlx includes an interactive example section (Figure 1) that can also be used as a template for solving other systems of linear equations, such as those encountered in homework assignments in MEB courses.

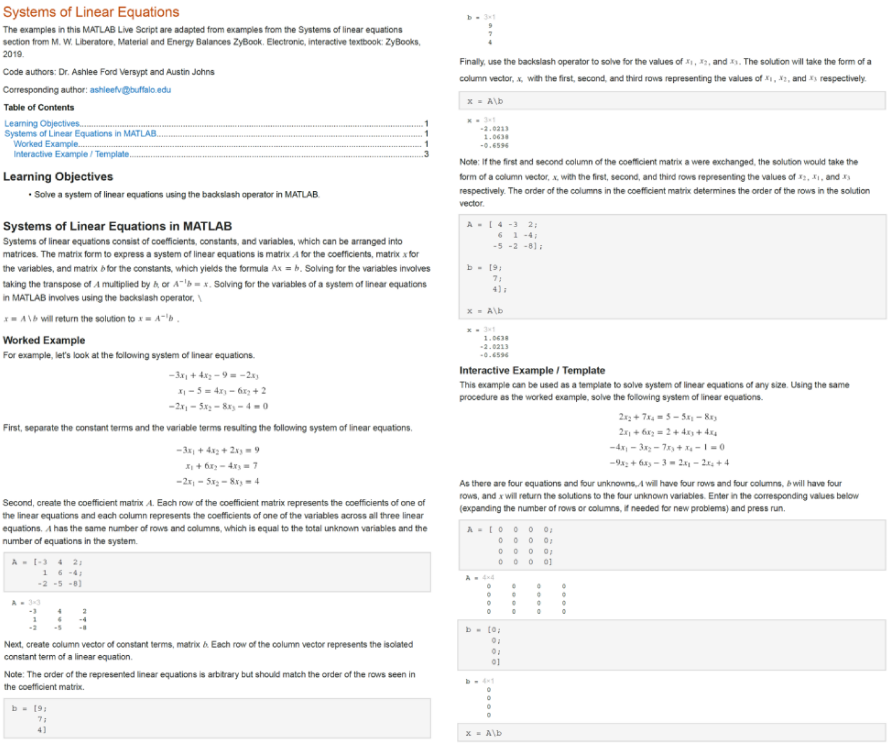

Figure 1. Screenshots of the MATLAB live script Systems_of_Linear_Equations.mlx: explanatory text and instructions, a worked example, and an interactive example. Content is split into two columns to fit within this manuscript.

The third file Lecture_With_Examples.mlx is a lecture to introduce the topic, even before students learn about material balances. The lecture includes the explanatory text and worked examples from Systems_of_Linear_Equations.mlx and five additional examples from three MEB textbooks: (Liberatore 2019; Felder, Rousseau, and Bullard 2016; Murphy 2023). The GitHub repository includes a README.md file for summarizing the information above and two additional files for viewing on GitHub or static sharing: Lecture_With_Examples.pdf and Systems_of_Linear_Equations_sol.pdf. The .mlx files can be run interactively via MATLAB after downloading from GitHub.

3. MATLAB live scripts for onboarding undergraduate researchers

We have had 51 undergraduate research scholars in the Ford Versypt Lab since it started in fall of 2014. Since spring of 2022, 14 new undergraduate researchers were trained using the MATLAB live scripts and onboarding materials described in this section. Prior to that four 1st year students helped write the solution codes, and one graduate

student drafted the instructional portions of the interactive coding templates. Collectively, these 18 undergraduate students studied chemical engineering, biomedical engineering, and nursing, with a wide range of mathematics and programming backgrounds. Generally, these students had no prior knowledge of MATLAB or ODEs.

The Ford Versypt Lab uses applied mathematics and process systems engineering methods to model tissues, treatments, and toxicology. We introduce new undergraduate students to a suite of techniques for these topics in mathematical systems biology. We start with training on applying conservation balances to populations of cells and amounts of chemical species in living organisms and solving these problems with systems of ODEs. During the training period, students meet weekly with Dr. Ford Versypt as a group. They learn from each other's issues and receive feedback about their progress. Then we provide guidance for an open exploration period for the students to investigate topics of their interest that use the techniques and templates.

The first training assignment is to use the Write_system_of_ODEs.mlx (Figure 2) interactive coding template, available from our UGResearch repository (Ford Versypt 2023). We introduce material balances through the acronym IOGA for In – Out + Generation = Accumulation. We provide a worked example of a system of ODEs for tracking the concentrations of three chemical species (Figure 2). Students are tasked with using the interactive coding template to solve numerous ODEs from chemical engineering to practice adapting the template for new problems. We use Chapter 10 of the Felder, Rousseau, and Bullard (2016) textbook, which provides several examples with worked solutions to ODEs from dynamic chemical engineering applications.

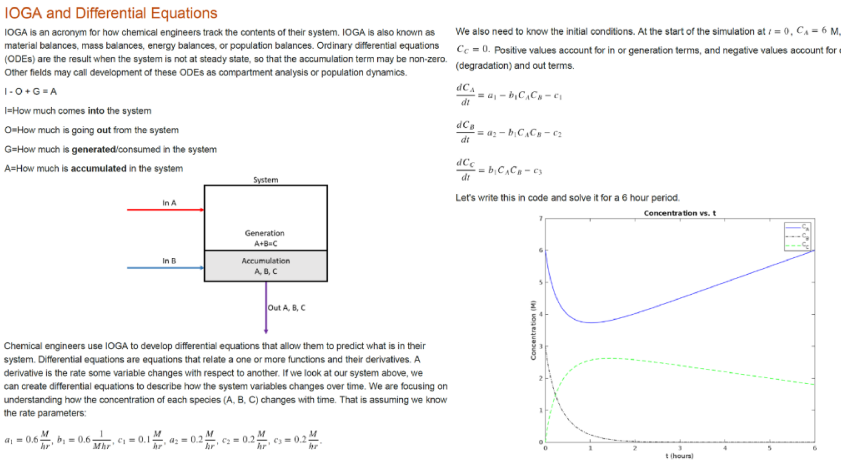

Figure 2. Screenshots of the MATLAB live script Write_system_of_ODEs.mlx: problem statement and output figure.

The second assignment in the training period introduces the use of ODEs for systems biology applications. Students are assigned to read Peskov et al. (2019), which reviews the process of building mechanistic mathematical models for systems biology for applications to cancer treatment via the immune system, so called "immuno-oncology". Figure 1 in Peskov et al. (2019) is particularly useful for students as it illustrates the types of biological interactions in complex cancer systems that can be modeled with ODEs to track the cell populations and chemical interactions relevant to cancer

treatments. We ask students to explain how the population balance principles introduced in assignment one apply to the illustrations for the cancer system (Peskov et al. 2019).

The third assignment tasks students with reading de Pillis, Gu, and Radunskaya (2006), which is an in-depth published example for systems dynamics of cancer chemotherapy and immunotherapy tracking the populations of four types of cells and the concentrations of two types of treatments. We prompt them to connect the IOGA concepts and the illustrations from Peskov et al. (2019) to the system of ODEs defined in de Pillis, Gu, and Radunskaya (2006). The repository includes a file titled Intro_to_Bio_ODEs.mlx, which reproduces the results of Figure 7 of de Pillis, Gu, and Radunskaya (2006). Students explore this file to see how a larger system of ODEs can be solved in much the same way as the examples in the first assignment.

The fourth and final training assignment is for students to adapt a partial solution available in our file dePillisSoln.mlx for the de Pillis, Gu, and Radunskaya (2006) model into a full solution for various scenarios to match the results in Figures 6 – 14 of the publication. The purposes of this exercise are for students to encounter issues with reproducible research computing and to gain practice with editing interactive coding templates to simulate other equations, parameter values, or conditions.

After the one-month training period, students enter the exploration period for the remainder of the academic term. Here is the prompt to the students:

> Using what you've learned so far about mathematical biology, MATLAB, and using ODEs and IOGA to describe biological problems from a chemical engineering perspective, you'll spend the rest of the term working on a topic area of your choice.
>
> 1. Choose a biomedical topic (note: this is likely an iterative process)
> 2. Search the literature to find two mathematical biology papers that involve ODE models for your selected biomedical topics. These two papers should have different equations. Each paper should list the full equations, parameters, and show some output plots.
> 3. Your task by the end of the term is to use MATLAB to replicate the two models that you find. You are encouraged to use the interactive coding template Write_system_of_ODEs.mlx available online (Ford Versypt 2023) to solve the systems of ODEs for each model. In your final presentation and report, you'll discuss the pros/cons of each model for the biomedical topic and any issues you encountered in reproducing them. You'll think about future directions that could involve merging the two models or otherwise expanding them to address new aspects of the biomedical topic.

Weekly feedback on topic selection and candidate papers is provided by Dr. Ford Versypt along with additional instruction in oral and poster presentations, searching the literature, and writing technical reports. After students select their papers, they meet weekly with graduate mentors to report on progress and to troubleshoot technical issues. At the end of the term, they deliver written and oral reports on the topic background and their progress in using MATLAB live scripts to explore published models and their completed codes for the project and plans for future extensions. By using MATLAB live scripts, students focus on the learning goals related to the research concepts instead of being hindered by programming or analytical mathematics proficiency. Senior students also appreciate that the templates enable them to quickly make progress towards using advanced techniques.

4. Conclusions

MATLAB live script resources to enable MEB students to solve linear systems of equations and for undergraduate researchers to solve systems of ODEs were developed by the Ford Versypt Lab and have been shared in two GitHub repositories with links provided in the References section (Johns and Ford Versypt 2022a; Ford Versypt 2023), respectively. We acknowledge that many are shifting from MATLAB to alternative software (Johns et al. 2023). We used MATLAB because of our institution's adoption of MATLAB. Transferring MATLAB live script content to Jupyter Notebooks is a relatively straightforward process (Johns et al. 2023). The emphasis on the materials presented here is on training students without advanced mathematics or programming skills to use interactive coding templates for classes of problems routinely encountered in first chemical engineering courses or in undergraduate research focused on dynamic systems. Similar types of materials could be developed for onboarding new researchers or industry professionals studying and solving realistic problems in contexts that do not have programming or advanced mathematics as a learning objective or prerequisite. Others are encouraged to reuse or adapt these materials as needed.

Acknowledgements

The authors acknowledge support of U.S. National Science Foundation grant 2133411 and the University at Buffalo. Ford Versypt Lab members are acknowledged for participating in or mentoring for undergraduate research.

References

J. E. Bara, A. N. Ford Versypt, R. B. Getman, C. A. Kieslich, and R. S. Voronov. 2020. Apps for chemical engineering education: off-the shelf and do-it-yourself development options. *Chem Eng Ed* 54 (3):137-42.

L. G. de Pillis, W. Gu, and A. E. Radunskaya. 2006. Mixed immunotherapy and chemotherapy of tumors: modeling, applications and biological interpretations. *J Theoretical Biology* 238 (4):841-62. doi: 10.1016/j.jtbi.2005.06.037.

C. V. Eastep, G. K. Harrell, A. N. McPeak, and A. N. Ford Versypt. 2019. A MATLAB app to introduce chemical engineering design concepts to engineering freshmen through a pharmaceutical dosing case study. *Chem Eng Ed* 53 (2):85-90.

R. M. Felder, R. W. Rousseau, and L. G. Bullard. 2016. *Elementary Principles of Chemical Processes*. 4th ed. Hoboken, NJ: John Wiley & Sons, Inc.

A. N. Ford Versypt. "UGResearch." https://github.com/ashleefv/UGResearch. doi: 10.5281/zenodo.10157864.

A. N. Ford Versypt, R. P. Hesketh, A. N. Johns, and M. D. Stuber. "ChESS2022." https://github.com/ashleefv/ChESS2022. doi: 10.5281/zenodo.7477475.

A. N. Johns, and A. N. Ford Versypt. "MEBLinearSystems." https://github.com/ashleefv/MEBLinearSystems. doi: 10.5281/zenodo.10157856.

A. N. Johns, and A. N. Ford Versypt. "[YouTube Video] Solving Systems of Linear Equations Using a MATLAB Live Script." https://youtu.be/78n8XaCBt9w.

A. N. Johns, R. P. Hesketh, M. D. Stuber, and A. N. Ford Versypt. 2023. Numerical Problem Solving across the Curriculum with Python and MATLAB Using Interactive Coding Templates: A Workshop for Chemical Engineering Faculty. Proceedings of the ASEE Annual Conference, Baltimore, MD. https://peer.asee.org/43749.

M. W. Liberatore. 2019. *Material and Energy Balances ZyBook*: Electronic textbook: ZyBooks.

R. M. Murphy. 2023. *Introduction to Chemical Processes: Principles, Analysis, Synthesis*. 2nd ed. New York, NY: McGraw Hill.

K. Peskov, I. Azarov, L. Chu, V. Voronova, Y. Kosinsky, and G. Helmlinger. 2019. Quantitative Mechanistic Modeling in Support of Pharmacological Therapeutics Development in Immuno-Oncology. *Frontiers in Immunology* 10:924. doi: 10.3389/fimmu.2019.00924.