

Introducing Partial Differential Equations and Their Numeric Solution Prior to Transport Courses

Dr. Jason C. Ganley, Colorado School of Mines

Dr. Ganley is a Teaching Professor in the Department of Chemical and Biological Engineering at the Colorado School of Mines, where he has served since 2012. His previous faculty appointments have been as an Associate Professor at Tuskegee University in Tuskegee, AL and Howard University in Washington, DC. His first professorial appointment was in 2004, following earning his doctoral degree in Chemical Engineering from the University of Illinois at Urbana-Champaign. His undergraduate studies were in Chemical Engineering at the University of Missouri at Rolla. His research interests include experiential learning and the production of alternative fuels from renewable energy.

Introducing Partial Differential Equations and Their Numeric Solution Prior to Transport Courses

Jason C. Ganley

Colorado School of Mines, Department of Chemical and Biological Engineering

Abstract

The field of chemical engineering is replete with systems that exhibit multidimensional dependence. Common developments of these include position-dependent velocity distributions in viscous flow, transient thermal conduction, and diffusive mass transfer within multiphase or reacting conditions. These types of systems are typically introduced to undergraduate students in junior-level core courses – these are the “transfer” (or transport) lecture classes: fluid mechanics (also known as momentum transfer), heat transfer, and mass transfer. Undergraduate programs in the field of chemical engineering typically require the completion of a differential equations course by the end of the sophomore year. These courses focus on the delivery of classical techniques for the analytical solution of singular (or systems of) ordinary differential equations (ODEs), but do not provide more than the general definition of partial differential equations (PDEs). In this paper, the methods for introducing sophomore-level students to PDEs and their computer-aided solution are described with respect to learning objectives and a detailed example of a student exercise. The impact of the instruction is presented in the context of student pre- and post-instruction self-evaluation, and performance on junior-level laboratory work that involves the analysis of a transient system.

Keywords

Partial differential equations, numeric methods, chemical engineering transport.

1. Introduction

Individual and combined transport courses make up the instructional core of most Chemical Engineering B.S. programs in the United States today¹⁻³, a pedagogical focus within the discipline that remains unchanged since at least the middle of the 20th century⁴⁻⁶. These courses deliver the principles of unit operations theory, the laboratory practices of which are usually employed as a program capstone⁷. While a foundational Material and Energy Balance (MEB) course is typically the first to introduce students in these programs to the techniques of applying transient matter and energy balances, those systems are most commonly treated as spatially homogeneous^{8,9}. Limiting system analysis to a strictly temporal variation ensures that the only a single ordinary differential equation (ODE), or a system thereof, will arise.

The transport courses following MEB focus on fluid mechanics, heat transfer, and mass transfer/separations. The material delivered in transport instruction presents additional complexity in the form of spatial system variations, which may occur in either steady-state or transient conditions¹⁰⁻¹². The mathematic description of system aspects such as a velocity, temperature, or concentration gradients with position in more than one direction (or in unsteady conditions) will necessarily give rise to partial differential equations (PDEs). The student is provided with a more complete and realistic explanation of a chemical process system versus

simpler, homogeneous descriptions, but she or he is simultaneously presented with a significant challenge for the solution or analysis of the system^{13,14}. Chemical process systems that give rise to PDEs with complete analytical solutions may be unrealistic or impractical, and those PDEs that adequately describe real-world systems usually have no analytical solution readily within the grasp of a typical undergraduate student. This is particularly true if a student is provided with only a very cursory exposure to PDEs in the mathematics core of the curriculum¹⁵.

The complexity of PDEs and their solution has therefore long provided an excellent opportunity to provide chemical engineering students with an introduction to their numeric solution and the techniques associated therewith. The following presentation is an illustration of efforts by the Department of Chemical and Biological Engineering (CBE) at the Colorado School of Mines (CSM) to both increase student understanding of multivariant chemical processes, as well as to provide them with a flexible computational tool for their analysis.

2. Institutional Setting and Program Illustration

Following a 2016 curriculum revision, the CBE Department at CSM has provided a new required sophomore-level course: Computational Methods in Chemical Engineering (CMCE), which is offered simultaneously with MEB as a corequisite. The main instructional goals of the CMCE course are to provide students with exposure to the computational tools used throughout the later curriculum in course projects and assignment calculations, as well as to provide limited instruction on programming techniques (flowsheet generation and coding), a dedicated course for which had been absent in the curriculum since 2002. An introduction to process simulation software (ASPEN Plus, Aspen Technology Inc.) is also included, making the course rather unique among those computation-based courses offered by our peer institutions.

The primary mathematical modeling and analysis software package used in CMCE is MATLAB (MathWorks, Inc.), which is provided to all students at no cost via a University-funded site license. The mid-semester MEB modules which focus on time-dependent mass and energy balances coincide with the introduction of methods for the numeric solution of ODEs in the CMCE course. In CMCE, such examples are examined following discussions of the temporal ODE systems which arise from chemical reaction kinetics, such as those encountered by students in general chemistry lecture and laboratory courses in the freshman year.

The solution of ODE systems is used as a natural starting point for the recasting of PDEs as systems of ODEs within MATLAB using the Numeric Method of Lines^{16,17}. Following this, its built-in PDE solver (**pdepe**) is used to provide students with direct instruction on dimensional discretization and solution of parabolic or elliptic PDEs. Parabolic PDEs make up the plurality of those arising from transport problems (e.g., Navier-Stokes, one-dimensional transient diffusion, unsteady heat conduction), although simpler elliptic equations are also encountered (e.g., steady-state heat conduction or diffusion). In the CMCE course, students construct a customizable script (MATLAB m-file) for the solution of parabolic or elliptic PDEs in various geometries. The program output includes a solution matrix for the supplied dimensions, as well as a detailed surface plot covering the discretized coordinate ranges.

While the creation of the script is carried out cooperatively during a specific segment of lecture instruction, the particular nature of each section of the script is explained in detail as it is created,

so that the student may immediately know which sections of it must be modified from one application to another. The script itself is divided into distinct functional segments:

- Specification of the coordinate system of interest (slab/rectangular, cylindrical, or spherical)
- Assignment of spatial or temporal coordinates, their overall ranges, and sectioning/discretization to create a solution mesh
- Customization of the solution surface plot
- Sub-function 1 (**pdefun**): Parameters for the PDE in standard form
- Sub-function 2 (**icfun**): Initial condition specification
- Sub-function 3 (**bcfun**): Parameters for boundary conditions in standard form

When the script file is correctly outfitted with specifications for each of these segments and executed, a numeric solution and accompanying surface plot is generated. The task of identifying and supplying the necessary input to the script requires only a thorough inspection of the problem statement and appropriate organization of the information given.

For example, in the Spring semester of 2019, students were presented with the following homework problem:

A tube with circular cross section is fashioned from compacted dark chocolate powder (inner wall radius $R = 2$ cm, total length $L = 20$ cm). Warm milk is flowing slowly ($v = 6$ cm s^{-1}) in laminar fashion (no chaotic mixing) through it.

The milk entering the tube is chocolate-free (0.0 g cm^{-3}). Chocolate dissolving in the milk is carried downstream by convection and is also spreading radially inward by diffusion. The diffusion coefficient of the chocolate in the milk is $D = 0.35$ $cm^2 s^{-1}$.

You realize that the system may be described with the following PDE (z is the length down the tube, r is the radius, c is the local concentration of chocolate in the milk):

$$v \left(1 - \frac{r^2}{R^2} \right) \frac{\partial c}{\partial z} = \frac{D}{r} \frac{\partial}{\partial r} \left(r \frac{\partial c}{\partial r} \right)$$

You also identify the following boundary conditions:

(BC #1) At the center of the tube, the concentration of chocolate is continuous and is at its minimum.

$$\frac{\partial c}{\partial r} = 0$$

(BC #2) At the tube wall, the chocolate is always at a saturation concentration in the milk.

$$c(r = R) = 0.2 \text{ g cm}^{-3}$$

Use an m-file function to invoke **pdepe** in MATLAB: solve for the concentration distribution of chocolate in milk within the tube over the entire radius and length of the tube.

(a) Plot this concentration distribution against z and r in a properly labeled surface plot.

(b) What are the following chocolate concentration values, to four significant digits?

- Midpoint of the tube length, and at $r = 1 \text{ cm}$
- End of the tube length ($z = 20 \text{ cm}$), and at center of tube

(c) What is the average concentration of chocolate in the milk exiting the tube?

Each of the questions (a) through (c) are answered by solution of the PDE provided, and according to the initial and boundary conditions described. The first steps of specifying problem geometry and dimensional assignment/spacing are straightforward (Fig. 1).

```
m = 1; % Set for coordinate system:
      % 0=rectangular, 1=cylindrical, 2=spherical
ri = 0; % Minimum radius (r) value, cm
rf = 2; % Maximum radius (r) value, cm
rmesh = 20; % Divide radius into this number of points
zi = 0; % Minimum length (z) value, cm
zf = 20; % Maximum length (z) value, cm
zmesh = 50; % Divide length into this number of points
% Set up the radial and length vectors.
r = linspace(ri,rf,rmesh); % Vector for radial coordinate
z = linspace(zi,zf,zmesh); % Vector for length coordinate
```

Figure 1: Excerpt from **pdepe** script illustrating geometry selection, dimension definition, and mesh resolution.

This segment of code is followed with a standard call of the **pdepe** function (and sub-functions), the generated solution of which is used to generate a customizable surface plot (Fig. 2).

```

% Run the PDEPE function!
solution = pdepe(m,@pdefun, @icfun, @bcfun,r,z);
% We save the first solution component as u.
u = solution(:,:,1);
% Create a surface plot: u against r and z.
surf(r,z,u)
title(['Numeric Solution Computed with ',int2str(rmesh),
'Mesh Points'])
xlabel('Radius (r, cm)')
ylabel('Length (z, cm)')
zlabel('Chocolate Concentration (c, g/cc)')

```

Figure 2: Excerpt from **pdepe** script illustrating the **pdepe** function call, solution matrix generation, and surface plot details.

Most of the effort required to solve the problem lies ahead. First, the student must write the PDE in the proper form, so that input parameters for the script may be determined and entered. The **pdepe** program requires the form

$$c \frac{\partial u}{\partial t} = x^{-m} \frac{\partial}{\partial x} (x^m f) + s \quad (1)$$

Here, the value of the coefficient **m** is determined by the geometry selected (0 for slab/rectangular, 1 for cylindrical, and 2 for spherical). The student must still determine values for **c**, **f**, and **s**. Each of these three values may be functions of the dimensions **t** or **x**, the partial derivative of **u** with respect to **x**, or of the function **u** itself.

In this example, the provided PDE is:

$$v \left(1 - \frac{r^2}{R^2} \right) \frac{\partial c}{\partial z} = \frac{D}{r} \frac{\partial}{\partial r} \left(r \frac{\partial c}{\partial r} \right) \quad (2)$$

After making the problem-specific dimensional and function substitutions of **t** → **z**, **x** → **r**, **c** → **u**, and with **m** = 1 for a cylindrical geometry, inspection and comparison of (1) with (2) allows the student to assign:

$$c = \frac{v}{D} \left(1 - \frac{r^2}{R^2} \right) \quad (3)$$

$$f = \frac{\partial u}{\partial r} \quad (4)$$

$$s = 0 \quad (5)$$

Additionally, the physical parameters for diffusion, convection, and tube radius are supplied (Fig. 3).

```

function [c,f,s]=pdefun(r,z,u,dudr)
D = 0.35; % sqcm/s
v = 6; % cm/s
R = 2; % cm
c = v/D*(1-r^2/R^2);
f = dudr;
s = 0;
end

```

Figure 3: Excerpt from **pdepe** script illustrating **pdefun** sub-function call with physical and function parameter designations.

The sub-function **icfun** is usually applied as an initial condition, that is, when the partial derivative on the left side of the standard PDE form refers to variation in time. In this example, the usual time-based partial derivative is replaced with a spatial derivative with respect to distance down the tube (**z**). Hence, the condition described by **icfun** in this instance is the chocolate concentration at **z = 0** for all radii (Fig.4).

```

function u0=icfun(r)
u0 = 0;
end

```

Figure 4: Excerpt from **pdepe** script illustrating **icfun** sub-function call.

The final and perhaps most time-intensive task for completing modifications to the script requires the arrangement of the “left” and “right” boundary conditions into the proper form. The **pdepe** function and its sub-function **bcfun** require boundary conditions to be in the form of:

$$p + qf = 0 \tag{6}$$

Here, **f** is obtained first from the examination of the PDE as detailed above (Eqn. 4). The first boundary condition given above (the “left” side condition), with substitution **c** → **u** is:

$$\frac{\partial u}{\partial r} = 0 \tag{7}$$

With **f** defined as in (4), inspection of (7) by the form of (6) yields:

$$p_{\text{left}} = 0 \tag{8a}$$

$$q_{\text{left}} = 1 \tag{8b}$$

The saturation condition of chocolate at the tube’s inner wall (the “right” side condition), again with **c** → **u** is given as:

$$u_{\text{right}} = 0.2 \quad (9)$$

Which, in the form of (6), becomes:

$$u_{\text{right}} - 0.2 = 0 \quad (9')$$

Inspection gives:

$$p_{\text{right}} = u_{\text{right}} - 0.2 \quad (10a)$$

$$q_{\text{right}} = 0 \quad (10b)$$

These values are shown as incorporated into the **bcfun** call below (Fig. 5).

```
function [pl,q1,pr,qr]=bcfun(xl,ul,xr,ur,t)
pl = 0;
q1 = 1;
pr = ur-0.2;
qr = 0;
end
```

Figure 5: Excerpt from **pdepe** script illustrating **bcfun** sub-function call.

Executing the script immediately generates the correct response to question (a). The plot provides students with visual feedback about the quality of the solution and therefore the setup of the script – illustrating the saturated condition at the wall, the absence of chocolate at the entry, and the increasing centerline concentration as the end of the tube is approached (Fig. 6).

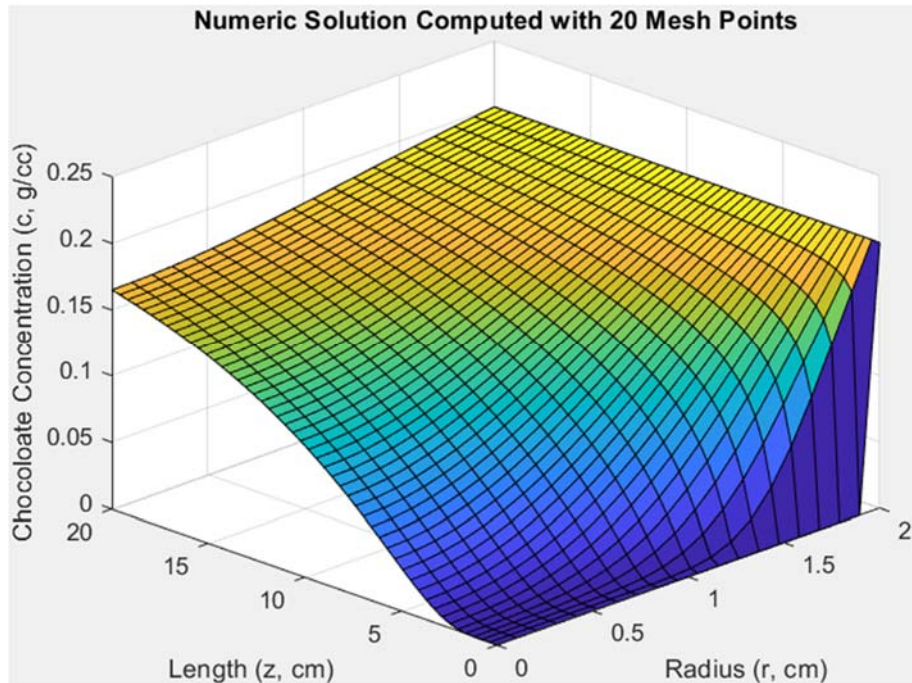


Figure 6: Surface plot from problem solution. Here, the reported mesh density refers to the radial spacing.

Question (b) is resolved by inspection of the resulting solution (\mathbf{u}) matrix, and reporting the value of \mathbf{u} at the \mathbf{z} , \mathbf{r} points of (10 cm, 1 cm) and (20 cm, 0 cm) as scaled to the corresponding mesh locations:

$$u(25,10) = 0.1327 \text{ g cm}^{-3} \quad (11a)$$

$$u(50,1) = 0.1651 \text{ g cm}^{-3} \quad (11b)$$

Question (c) requires the selection of all \mathbf{u} values at $z = 20$ and evaluating the arithmetic mean:

$$\text{mean}(u(50,:)) = 0.1800 \text{ g cm}^{-3} \quad (12)$$

The plot may be examined once more to verify that the correct matrix values have been obtained.

3. Application by Students to Future Coursework

The instructional focus of the CMCE course is to introduce students to certain computational tools and techniques to solve chemical engineering problems, including those described by PDEs. The students solve problems such as the one illustrated in the previous section prior to their completion of any of the required junior-level transport courses; they are directly provided with the appropriate equation to solve, rather than being required to develop the equation in following with theory as introduced in those succeeding courses. However, it is evident that students recall and take advantage of **pdepe** in future coursework. The best example of this is the modeling and simulation work carried out by students in the junior-level Unit Operations Laboratory (UOL)¹⁸ on an experiment involving unsteady heat transfer.

This experiment, titled “Transient Thermal Conduction in Solids” (TTCS), was constructed and first deployed to the UOL in the Summer of 2017. The experiment is of simple design; students are provided with a large heated water bath, into which cylinders (both metallic and nonmetallic) of various dimensions may be immersed. The centerline temperature is measured and recorded over time using an embedded thermocouple sensor and a computer-based data acquisition system. Students are instructed to use a cylinder of known composition (alloy C110, 99.9% copper), for which the intrinsic thermal conductivity (\mathbf{k}) is extremely high, to determine ranges of heat transfer coefficients (\mathbf{h}). These estimated \mathbf{h} values required for estimating the unknown thermal conductivities of all other materials (aluminum, titanium, and brass alloys, stainless and carbon steels, and various plastics). The cylinders provided have a sufficiently high length-to-diameter ratios such that heat transfer is considered to be one-dimensional. For a material of constant thermal diffusivity (α), the unsteady radial temperature distribution within the cylinder reduces to:

$$\frac{\partial T}{\partial t} = \frac{\alpha}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) \quad (13)$$

In the Heat and Mass Transfer course, a UOL prerequisite, transient problems such as these are approached through the application of dimensionless groups, specifically the Fourier and Biot numbers alongside tabulated Bessel function solutions^{19,20} for the scenario above. Students have access to plots or tables of the various dimensionless groups, the use of which requires an iterative/guess-and-check solution approach for the unknown thermal conductivities. In contrast, Eqn. 13 is easily recognized by those students experienced with the translation of PDEs into MATLAB for numeric solution with **pdepe**.

The inaugural offering of the CMCE course was the Spring semester of 2018; juniors having formal training in MATLAB and the **pdepe** utility did not appear in the UOL until the Summer of 2019. At the time of the present communication, two UOL student cohorts (2017, 2018) had received no formal PDE modeling instruction. These were followed by two others (2019, 2020) comprised of students which had practiced both the Method of Lines and **pdepe** in the CMCE course as sophomores. A summary evaluation of the primary calculation approach used in all student reports, which represent the 3-student team report submissions with the TTCS laboratory module, appears in Table 1 below.

Table 1: Characteristics of Primary Solution Techniques Appearing in Experimental Reports, TTCS Laboratory Module (2017 – 2020).

Summer UOL Session	Graphical or Tabulated Solution Approach	Numeric Method of Lines*	MATLAB pdepe Solution
2017 (n = 30)	96.7 %	3.3 %	0.0 %
2018 (n = 33)	100.0 %	0.0 %	0.0 %
2019 (n = 30)	36.7 %	10.0 %	53.3 %
2020 (n = 31)	32.3 %	3.2 %	64.5 %

* Count includes all software-based approaches, MATLAB or otherwise

The impact introducing MATLAB for the numeric solution of PDEs is both clear and striking. Students enrolled in the UOL course in the years prior to those receiving instruction in CMCE

did not explore the use of non-tabular/non-graphical approaches at all, with one exception – a team that included a student working towards a second major in Computational and Applied Mathematics. For those students having completed the CMCE course, selection rate of **pdepe** was very strong. For the Summer of 2020, the author of the present communication was the sole supervisor of the TTCS experiment. Conversations with student groups (following reporting) regarding the choice of solution/calculation approach may be generally categorized as follows.

- Those selecting the graphical/tabulated solution approach did not enjoy using MATLAB and felt that iterative solution using charts was not overly problematic.
- Those choosing the Numeric Method of Lines felt comfortable with setting up and solving systems of ODEs in either MATLAB or Polymath (Polymath Software), also choosing to avoid iterative solution.
- Those favoring the use of **pdepe** in MATLAB found incorporating Eqn. 13 to a student's saved copy of the m-file script (generated for the CMCE course) to require little effort.

As students do not know the identities of the alloys or plastic compositions of the samples tested, it is worth noting that none of the three methods outlined above appears to provide a predictably superior identification rate of the unknown materials, nor of their thermal properties. The quality of laboratory data acquired by the student teams, which depends heavily on the experimental design and its execution, was the dominant factor with respect to the accuracy of reported results.

3. Student Self-evaluations

PDEs and the systems described by them are often quite complex. Sophomores, even those enrolled in technical programs, will have limited or no exposure to them or to their solution. This is naturally a cause for concern for an instructor of a course such as CMCE, especially when considered in the context of introducing the students to unfamiliar software packages such as MATLAB. In order to properly focus instructional techniques and topic priorities, a voluntary, informal pair of surveys were administered to students from 2019 – 2021 (three course offerings). The survey includes a visual example of a simple PDE and requests a first and last name so that personal responses to only one half of the survey and duplicate responses may be screened out. Responses were organized on a 5-point Likert scale (excellent, good, fair, poor, and none). A condensed result summary appears below in Table 2.

Table 2: Student Self-assessment of Evolving Familiarity with PDEs and Their Numeric Solution, Before and After PDE Instruction (2019 – 2021).

Rate your own ability to...	Spring CMCE Session		
	2019 (n = 73)	2020 (n = 95)	2021 (n = 119)
(a) recognize a system that may be described with a PDE.			
Good or better, before instruction	25 %	29 %	33 %
Good or better, after instruction	97 %	96 %	97 %
(b) explain to a peer how a PDE may describe a system.			
Good or better, before instruction	12 %	9 %	15 %
Good or better, after instruction	85 %	81 %	88 %
(c) qualitatively diagram the solution of a PDE.			
Good or better, before instruction	6 %	5 %	9 %
Good or better, after instruction	58 %	63 %	71 %
(d) solve a PDE numerically using the Method of Lines.			
Good or better, before instruction	6 %	4 %	4 %
Good or better, after instruction	82 %	88 %	85 %
(e) solve a PDE numerically using specialized software, e.g., MATLAB.			
Good or better, before instruction	2 %	2 %	4 %
Good or better, after instruction	79 %	85 %	88 %

The results displayed in Table 2 appear to indicate that students have at least received a cursory introduction to PDEs in previous mathematics classes. Unsurprisingly, almost none of the students appear to have any familiarity with qualitative characteristics of multivariate systems, nor with numeric PDE solution theory or techniques. The increase in self-confidence regarding these techniques following instruction in CMCE implies at least perceived mastery of the material, although student confidence in qualitative diagramming appears to be less improved. This suggests a possible disconnect between using the computational software and understanding the overall significance of the calculated results within the context of a given system.

5. Conclusions and Recommendations

The prevalence of systems which may be described with PDEs in chemical engineering courses at the junior level and above provides an impetus for introducing students to the characteristics and solution of PDEs in advance. Chemical engineering, even at the undergraduate level, is distinct from other engineering disciplines in that its programmatic core not only requires the consideration of systems with spatial and/or temporal variation, but also their modeling and simulation in order to foster a more complete understanding of how they work. However, students will typically receive no more than a passing mention of them in advanced calculus or

introductory differential equations courses. It is reasonable to expect that this instructional deficit should be resolved directly via the chemical engineering program curriculum.

In the discussion presented here, the provided example is of a standalone course offered at the sophomore level, alongside the near-universal and foundational course of MEB. This course, while operating with the broad goals of introducing students to software, general calculation strategies, and structured programming, may provide students with a series of opportunities to practice and become familiar with certain aspects of PDEs through the creation and use of a customizable script (**pdepe**). This is accomplished alongside building student proficiency with the powerful and flexible MATLAB computational toolset overall.

Should the creation of a similar course not be feasible for a given chemical engineering program, it is suggested to include end-of-semester ODE- or PDE-centric modules to many of the courses that are mainstays for most departments in the United States. Examples include:

- The analysis of unsteady material and energy balances to create and solve systems of ODEs (MEB)
- The descriptions of unsteady flow (ODE) and multidirectional shear (PDE) in a laminar flow system (Fluid Mechanics)
- Heat generating/consuming systems (ODE) and unsteady conduction (PDE) in one dimension (Heat Transfer)
- The conservation of mass (ODE) or unsteady diffusion (PDE) in one dimension (Mass Transfer)

Additionally, with the aid of data acquisition systems, a department may achieve a great deal of success in implementing experimental platforms or modules which exhibit transient behavior - allowing students to study systems at conditions other than steady state. In any of these cases, modern computational tools such as MATLAB and its component **pdepe** lend themselves well to enriching the student experience - building confidence and expanding familiarity with complex, yet realistic, chemical process systems.

References

- 1 R. S. Voronov, S. Basuray, G. Obuskovic, L. Simon, R. B. Barat, E. Bilgili, "Statistical Analysis of Undergraduate Chemical Engineering Curricula of United States of America Universities: Trends and Observations," *Education for Chemical Engineers*, vol. 20, 2017, pp. 1-10.
- 2 R. C. Armstrong, "A Vision of the Chemical Engineering Curriculum of the Future," *Chemical Engineering Education*, vol. 40, 2006, pp. 104-109
- 3 D. Kauffman, R. D. Busch, A. K. Datye, T. L. Ward, "The Minimum Core for Chemical Engineering: How Much Bathwater Can We Throw Out?" *Proceedings of the ASEE Annual Conference and Exposition*, 2005, pp. 10413-10427.
- 4 J. C. Forman, "The Chemical Engineering Education Situation in the U.S.A.," *Proceedings of the 2nd International Conference on Chemical Engineering Education*, 1987, pp. 37-53.
- 5 D. H. Barker, "Undergraduate Curricula 1976," *Chemical Engineering Education*, vol. Y, 1977, pp. 60-96.
- 6 A. X. Schmidt, "What is the Current B.Ch.E. Curriculum?" *Journal of Engineering Education*, vol. 50 (1), 1959, pp. 65-74.

- 7 Y. H. Elawady, A. S. Tolba, "Educational Objectives of Laboratory Types: A Comparative Study,"
International Journal of Computer Science and Information Security, vol. 6 (2), 2009, pp. 89-96.
- 8 R. M. Felder, R. W. Rousseau, L. G. Bullard, in Elementary Principles of Chemical Processes, 4th ed., John
Wiley & Sons, 2016, pp. 570-590.
- 9 J. Newman, "Numerical solution of coupled, ordinary differential equations," Industrial & Engineering
Chemistry Fundamentals, vol. 7 (3), 1968, pp. 514-517.
- 10 D. Adair, Z. Bakenov, M. Jaeger, "Building on a Traditional Chemical Engineering Curriculum Using
Computational Fluid Dynamics," Education for Chemical Engineers, vol. 9 (4), 2014, pp. e85-e93.
- 11 A. G. Dixon, D. DiBiasio, "Integrating COMSOL into a Mathematical Modeling Course for Chemical
Engineers," Proceedings of the COMSOL Conference (Boston), 2008.
- 12 M. B. Cutlip, M. Shacham, "Efficient Integration of Numerical Problem Solving Throughout the Chemical
Engineering Curriculum," Economic Studies, vol. 23, 2008, pp. 68-78.
- 13 S. Mejri, H. Binous, K. Mahgoub, A. Bellagi, "A Model Parameter Estimation Method Using Mathematica
Applied to Transient Chemical Engineering Processes." Computer Applications in Engineering Education,
vol. 26 (5), 2018, pp. 1405-1421.
- 14 J. N. Harb, A. Jones, R. L. Rowley, W. V. Wilding, "Use of Computational Tools in Engineering
Education," Chemical Engineering Education, vol. 31 (3), 1997, pp. 180-187.
- 15 K. E. Thompson, "Teaching PDE-Based Modeling to ChE Undergraduates: Overcoming Conceptual and
Computational Barriers." Chemical Engineering Education, vol. 34 (2), 2000, pp. 146-152.
- 16 V. R. Subramanian, R. E. White. "Semianalytical Method of Lines for Solving Elliptic Partial Differential
Equations.," Chemical Engineering Science, vol. 59 (4), 2004, pp. 781-788.
- 17 M. B. Cutlip, M. Shacham, "The Numerical Method of Lines for Partial Differential Equations," CACHE
News, vol. 47, 1998, pp. 18-21.
- 18 J. C. Ganley, "Enhanced Experiential Learning in the Unit Operations Laboratory," Proceedings of the 2016
ASEE Rocky Mountain Section Conference, 2016, pp. 114-123.
- 19 M. P. Heisler, "Temperature Charts for Induction and Constant Temperature Heating," Transactions of the
American Society of Mechanical Engineers, vol. 69, 1947, pp. 227-236.
- 20 H. P. Gurney, J. Lurie. "Charts for Estimating Temperature Distributions in Heating or Cooling Solid
Shapes." Industrial & Engineering Chemistry, vol. 15 (11), 1923, pp. 1170-1172.

Jason C. Ganley

Dr. Ganley is a Teaching Professor in the Department of Chemical and Biological Engineering at the Colorado School of Mines, where he has served since 2012. His previous faculty appointments have been as an Associate Professor at Tuskegee University in Tuskegee, AL and Howard University in Washington, DC. His first professorial appointment was in 2004 following earning his doctoral degree in Chemical Engineering from the University of Illinois at Urbana-Champaign. His undergraduate studies were in Chemical Engineering at the University of Missouri at Rolla. His research interests include experiential learning and alternative fuels production from renewable energy.