

Numerical methods to biosystems and food engineering students: Hands-on practices and cross-disciplinary integration

Jose A. Rabi  | Fernando L. Caneppele

Department of Biosystems Engineering,
University of São Paulo – Pirassununga
Campus, Pirassununga, SP, Brazil

Correspondence

Jose A. Rabi, Department of Biosystems
Engineering, University of São Paulo –
Pirassununga Campus, Av. Duque de
Caxias Norte 225, Pirassununga, SP, 13635-
900, Brazil.
Email: jrabi@usp.br

Abstract

While virtualization has supported many engineering sectors, it remains underexplored in agroindustrial engineering. Accordingly, hands-on activities in numerical methods have been proposed to food and biosystems engineering students at a didactic computational laboratory. As those learning-by-doing tasks refer to agroindustrial problems, integration with forthcoming disciplines has been achieved and pedagogical ethos has indeed been promoted.

KEYWORDS

biosystems engineering, cross-disciplinary integration, food engineering, hands-on practice, numerical methods

1 | INTRODUCTION

Innovative design, scale-up and optimization can be rapidly achieved via modeling and simulation inasmuch as different design parameters and exploratory scenarios can be numerically tested towards viability [4]. Virtualization is helpful not only to save material, energy and human resources in research, development, and innovation (RD&I) but also when obtaining experimental or real-operation data becomes awkward or even unfeasible [3].

Agroindustrial engineering applications may integrate fluid flow and transport phenomena, whose model equations are relatively complex [21]. As analytical solutions are solely achievable in idealized scenarios, comprehensive knowledge of agri-food-bioprocesses invokes numerical methods [20]. Initially used for academic purposes, computational modeling of fluid flow, and transport phenomena (i.e., computational fluid dynamics) has experienced notable evolution and has become strategic to RD&I in food processing [18].

Yet, “the food industry (that represents more than 5% of the global GDP) is still lagging to utilize the wide spectrum potential offered by virtualization as an engineering design tool,” as quoted from [10]. Modeling and virtualization have

been identified as new challenges and opportunities for food engineers in the 21st century [16] and such a rationale can be expanded to the agroindustrial sector as a whole.

Due to unique inherent phenomena [2], all-inclusive computational modeling of food and bioprocesses may require tailor-made routines possibly absent in general-purpose software. For that reason, specific code lines must be externally programmed by (perhaps non-trained) end-users. Also, license and technical support costs of commercial-off-the-shelf software can be prohibitive for non-academic users. Aforesaid hurdles are prone to jeopardize virtualization-supported RD&I in agroindustrial engineering, particularly in developing countries.

Engineers must be suitably qualified and versatile to tackle multidisciplinary problems [15]. Hence, the point is not whether agroindustrial engineers should or should not come to terms with virtualization, but rather to what extent they should grasp virtualization concepts. In competitive scenarios, engineers (regardless of their specialization) must be not only aware of computational modeling but also minimally skilled with respect to its use to support RD&I.

This paper addresses hands-on practices in face-to-face lectures of numerical methods to undergraduate students in

food and biosystems engineering programs. By aiming at excellence in teaching and learning, those learning-by-doing activities have been proposed and accomplished at a didactic computational laboratory. Furthermore, they refer to problems found in agroindustrial engineering curricula, thus linking computational methods to real-world problems.

2 | NUMERICAL METHODS TO AGROINDUSTRIAL ENGINEERING STUDENTS: A CASE STUDY IN BRAZIL

Settled at an experimental farm 210 km from the main campus, Pirassununga campus of the University of São Paulo (USP, Brazil) hosts two undergraduate STEM programs, namely Food Engineering (evening shift since 2001 and day shift since 2004) and Biosystems Engineering (day shift since 2009). In those programs, authors teach Numerical Methods and Applied Computational Methods with the help of MS Power Point (besides the chalkboard) in face-to-face lectures. In both disciplines MS Excel is chosen as the computational environment as Matlab, Maple, and Mathematica are not widespread outside the academic environment in Brazil.

While Numerical Methods is a core compulsory course, Applied Computational Methods is an elective course in the Biosystems Engineering program (but Food Engineering students may enroll if there are vacancies). A further distinction between them refers to their goals. Numerical Methods introduces numerical analysis to solve rather simplified engineering problems whereas Applied Computational Methods advances students' expertise towards the numerical solution of more complex problems using more advanced MS Excel tools (e.g., macros and VBA codes).

In order to foster didactic ethos linking computational methods and engineering problems, learning-by-doing activities have been proposed to students in face-to-face lectures of aforesaid disciplines. The next section describes hands-on practices concerning Numerical Methods while those related to Applied Computational Methods are left for future publication.

At USP—Pirassununga campus, Numerical Methods is taught at a computational didactic laboratory with capacity for 60 students. A video projector is connected to lecturer's computer so that students can follow the implementation of the numerical method (besides lecture slides and chalkboard notes). At their computers, students can implement the numerical method themselves and check their results. Figure 1 shows a picture of the didactic computational laboratory.

Each face-to-face lecture begins with a problem to be numerically solved (hereafter referred to as opening problem), which recalls the agroindustrial vocation of the campus. Either previous or upcoming disciplines dealing with



FIGURE 1 Didactic computational laboratory for face-to-face lectures of Numerical Methods

the opening problem are then identified in the engineering curriculum. The idea is to motivate students to learn the numerical method being addressed by implementing it themselves as associated to a real-world problem in the agroindustrial scope.

After proposing the opening problem, the numerical solution method is first taught on general basis, for example, by numerically solving a mathematical problem in terms of functions f of variables x and y . This is when the theory of calculations and proof of methods are presented. Preference is given to problems with analytical solutions, which are initially omitted. Only after students have successfully implemented the numerical method, the analytical solution (if any) is provided for comparison purposes. Accordingly, students can check their numerical outcomes while acquiring confidence and/or clarifying eventual doubts. Students are then recalled that real-world problems may not be simple enough to allow analytical solutions being deduced.

At this stage of the lecture, students have learned the numerical solution method, being able to apply it to real-world problems. Attention then returns to the opening problem and students are challenged to numerically solve it at their computers, that is, the lecturer does not solve (i.e., display) it at his computer. However, students are encouraged to help each other, for example, by comparing their numerical answers, particularly when analytical solutions are unfeasible or not provided.

3 | NUMERICAL METHODS IN FOOD AND BIOSYSTEMS ENGINEERING PROGRAMS: HANDS-ON PRACTICES AND CROSS-DISCIPLINARY INTEGRATION

This section describes the opening problems proposed in face-to-face lectures of Numerical Methods, together with the related numerical solution method and its basic implementation in MS Excel cells. Detailed explanation of how each numerical method can be implemented is beyond the goals of

this paper and a compromise has been attempted between comprehensiveness and conciseness. As subsections are independent, they can be skipped in line with reader's interest.

Opening problems are listed in Table 1 according to their order as presented throughout the course. Table 1 equally lists the disciplines in the engineering program dealing with the opening problem (or a similar one). Most of them are forthcoming disciplines and they are identified in the lectures of Numerical Methods for cross-disciplinary integration purposes.

3.1 | Flow friction factor from Colebrook's equation

Darcy friction factor f is a dimensionless parameter to assess linear head losses in turbulent flows in pipes and it can be obtained from Colebrook's equation [5]:

$$\frac{1}{\sqrt{f}} = -2\log\left(\frac{\varepsilon_{\text{rel}}}{3.7} + \frac{2.51}{\text{Re}\sqrt{f}}\right) \quad (1)$$

where ε_{rel} is pipe relative roughness and Re is Reynolds number (both are dimensionless as well). Students are asked to numerically solve Equation (1) for $\varepsilon_{\text{rel}} = 0.0017$ and $\text{Re} = 105,000$, which is a scenario with no printed lines in the graphical representation of Equation (1) (i.e., Moody's diagram).

Students are free to implement either fixed-point or secant methods [9]. Newton-Raphson (i.e., tangent) method is a less suitable choice as it involves a cumbersome derivative

(students do agree!). The change of variables $x = 1/\sqrt{f}$ is suggested as a hint so that Equation (1) becomes:

$$x = -2\log\left(\frac{\varepsilon_{\text{rel}}}{3.7} + \frac{2.51}{\text{Re}}x\right) \quad (2)$$

Bearing in mind Equation (2), fixed-point method is quite straightforward:

$$x_{i+1} = g(x_i), \quad g(x) = -2\log\left(\frac{\varepsilon_{\text{rel}}}{3.7} + \frac{2.51}{\text{Re}}x\right) \quad (3)$$

whereas the secant method becomes:

$$x_{i+1} = x_i - \phi(x_i) \frac{x_i - x_{i-1}}{\phi(x_i) - \phi(x_{i-1})}, \quad \phi(x) = x + 2\log\left(\frac{\varepsilon_{\text{rel}}}{3.7} + \frac{2.51}{\text{Re}}x\right) \quad (4)$$

In previous equations, subscripts $i + 1$, i , and $i - 1$ refer to the iterative level.

With the help of renamed cells "ere137" and "Re_251," respectively containing $\varepsilon_{\text{rel}}/3.7$ and $2.51/\text{Re}$ values, Table 2 exemplifies key cells to implement those two numerical methods in MS Excel. Those formulas must be replicated in subsequent cells in rows below and initial guesses (i.e., x_0 and additionally x_1 in secant method) are not shown for brevity.

For the proposed pair $\varepsilon_{\text{rel}} = 0.0017$ and $\text{Re} = 105,000$, the friction factor results as $f = 0.0242$ regardless of the numerical solution method. The number of iterations depends on initial

TABLE 1 Opening problems proposed in face-to-face lectures of Numerical Methods, corresponding numerical solution methods, and disciplines dealing with them

| Opening problem | Numerical solution method(s) | Discipline in food engineering program | Discipline in biosystems engineering program |
|--|---|---|--|
| Flow friction factor from Colebrook's equation | Fixed-point, Newton-Raphson or secant methods | Transport Phenomena 1 | Hydraulics |
| Direct currents in an electric network | Jacobi or Gauss-Seidel methods for linear systems | Electro-technical Engineering | Electrical Circuits |
| Rheological parameters of a non-Newtonian fluid | Least squares method (weighted by uncertainties) | Unit Operations 1 | Transport Phenomena |
| Non-tabulated entries in a saturated steam table | Lagrange or Newton interpolating polynomials | Thermodynamics, Refrigeration & Cold Chain Operations | Thermodynamics, Technology of Grain Storage & Drying |
| Extraction yield in a continuous-flow process | Numerical integration | Unit Operations 3 | Biofuels |
| Temperature-time profile of food under cooling | Runge-Kutta method (1st order O.D.E.) | Transport Phenomena 2 | Transport Phenomena |
| Damped harmonic oscillator | Runge-Kutta-Nyström method (2nd order O.D.E.) | General Experimental Physics 2 | General Experimental Physics 2 |
| Steady-state 1-D heat conduction | Finite differences method | Transport Phenomena 2 | Transport Phenomena |

TABLE 2 Example of MS Excel cells to implement fixed-point and secant methods to obtain flow friction factor from Colebrook's equation

| Numerical method | Value | Cell | Expression in MS Excel (as seen in formula bar) |
|------------------|-------------|------|---|
| Fixed-point | $g(x_i)$ | B5 | = -2*LOG10(ere137 + Re_251*A5) |
| | x_{i+1} | A6 | = B5 |
| Secant | $\phi(x_i)$ | K6 | = J6 + 2*LOG10(ere137 + Re_251*J6) |
| | x_{i+1} | J7 | = J6 - K6*(J6 - J5)/(K6 - K5) |

guesses and desired accuracy. For instance, by starting from $x_0 = 10$ ($f = 0.0100$) in fixed-point method and from $x_0 = 10$ and $x_1 = 11$ ($f = 0.0083$) in secant method, converged solution ($x = 6.4250$) is achieved after four and two iterations, respectively.

3.2 | Direct currents in an electric network

In this problem, students must numerically assess the electric currents i_1 , i_2 , and i_3 in a DC network as shown in Figure 2. By applying Kirchhoff's current and voltage laws [7], one obtains the following system of linear algebraic equations:

$$\begin{aligned} i_1 - i_2 + i_3 &= 0 \\ R_1 i_1 - R_3 i_3 &= \varepsilon_1 \\ -R_2 i_2 - R_3 i_3 &= \varepsilon_2 \end{aligned} \quad (5)$$

Students are then asked to solve Equation (5) for $R_1 = 1 \text{ k}\Omega$, $R_2 = 2 \text{ k}\Omega$, $R_3 = 3 \text{ k}\Omega$, $\varepsilon_1 = 10 \text{ V}$ and $\varepsilon_2 = 13 \text{ V}$, being free to choose between Jacobi or Gauss-Seidel methods [9].

The coefficient matrix must be diagonally dominant so as to assure numerical convergence of either aforesaid method. In view of values assigned to R_1 , R_2 , R_3 , ε_1 , and ε_2 , Equation (5) must be rearranged so that Jacobi or Gauss-Seidel should be implemented respectively as:

$$\begin{aligned} i_1^{(j+1)} &= i_2^{(j)} - i_3^{(j)} \\ i_2^{(j+1)} &= (\varepsilon_2 + R_3 i_3^{(j)}) / (-R_2) \\ i_3^{(j+1)} &= (\varepsilon_1 - R_1 i_1^{(j)}) / (-R_3) \end{aligned}$$

or

$$\begin{aligned} i_1^{(j+1)} &= i_2^{(j)} - i_3^{(j)} \\ i_2^{(j+1)} &= (\varepsilon_2 + R_3 i_3^{(j)}) / (-R_2) \\ i_3^{(j+1)} &= (\varepsilon_1 - R_1 i_1^{(j+1)}) / (-R_3) \end{aligned} \quad (6)$$

where superscripts $(j+1)$; and (j) refer to iterative levels, that is, they are not powers.

By using renamed cells "R1.," "R2.," "R3.," "emf1.," and "emf2.," respectively for R_1 , R_2 , R_3 , ε_1 , and ε_2 , Table 3 exemplifies key cells to implement Jacobi and Gauss-Seidel methods in MS Excel. Those are formulas to be replicated in subsequent cells and initial guesses (i.e., $i_1^{(0)}$, $i_2^{(0)}$, $i_3^{(0)}$) are omitted for brevity. For proposed R_1 , R_2 , R_3 , ε_1 , and ε_2 values, converged values are $i_1 = 1.000 \text{ mA}$, $i_2 = -2.000 \text{ mA}$ and $i_3 = -3.000 \text{ mA}$, regardless of the numerical solution method.

Implementation of those methods differs only with respect to equation for current $i_3^{(j+1)}$ as one can use recently-assessed value $i_1^{(j+1)}$ in Gauss-Seidel method. This tiny difference drastically reduces the number of iterations for convergence from 70 in Jacobi method down to 30 iterations in Gauss-Seidel method, both starting from $i_1^{(0)} = i_2^{(0)} = i_3^{(0)} = 1.000 \text{ mA}$. Conversely, those two methods numerically diverge if Equation (5) is not rearranged as Equation (6). Divergence is indeed observed (by careless students!) when Equation (5) is rearranged and numerically implemented as:

$$\begin{aligned} i_1^{(j+1)} &= (\varepsilon_1 + R_3 i_3^{(j)}) / R_1 \\ i_2^{(j+1)} &= (\varepsilon_2 + R_3 i_3^{(j)}) / (-R_2) \\ i_3^{(j+1)} &= i_2^{(j)} - i_1^{(j)} \end{aligned}$$

or

$$\begin{aligned} i_1^{(j+1)} &= (\varepsilon_1 + R_3 i_3^{(j)}) / (R_1) \\ i_2^{(j+1)} &= (\varepsilon_2 + R_3 i_3^{(j)}) / (-R_2) \\ i_3^{(j+1)} &= i_2^{(j+1)} - i_1^{(j+1)} \end{aligned} \quad (7)$$

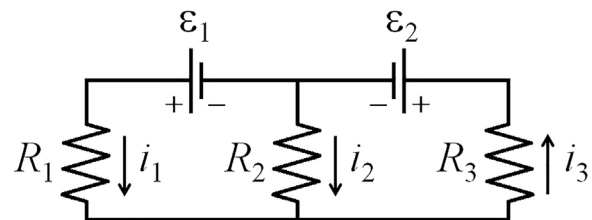
**FIGURE 2** DC network with electric currents i_1 , i_2 , and i_3 to be numerically assessed

TABLE 3 Example of MS Excel cells to implement Jacobi and Gauss-Seidel methods to assess electric currents in a given DC network

| Current | Cell | Expression for Jacobi method | Expression for Gauss-Seidel method |
|---------------|------|------------------------------|------------------------------------|
| $i_1^{(j+1)}$ | A6 | = B5 – C5 | = B5 – C5 |
| $i_2^{(j+1)}$ | B6 | = (fem2. + R3.*C5)/(-R2.) | = (fem2. + R3.*C5)/(-R2.) |
| $i_3^{(j+1)}$ | C6 | = (fem1. – R1.*A5)/(-R3.) | = (fem1. – R1.*A6)/(-R3.) |

3.3 | Rheological parameters of a non-Newtonian fluid

Many food fluids are non-Newtonian [17]. For instance, tomato paste is a Bingham plastic fluid so that the following rheological relation holds between shear stress τ and shear rate $\dot{\gamma}$:

$$\tau = \tau_0 + \mu_p \dot{\gamma} \quad (8)$$

where τ_0 and μ_p are respectively referred to as yield stress and plastic viscosity [17]. By applying least squares method [9], students are asked to assess τ_0 and μ_p from data comprising n triplets $(\dot{\gamma}_i, \tau_i, \sigma_i)$, where σ_i is the experimental uncertainty transferred to the dependent variable τ_i [8].

Parameters τ_0 and μ_p respectively refer to y-intercept and slope in Equation (8). In order to apply least squares method (weighted by the uncertainties), the following summations are needed:

$$S_1 = \sum_{i=1}^n \frac{1}{\sigma_i^2}, \quad S_x = \sum_{i=1}^n \frac{\dot{\gamma}_i}{\sigma_i^2}, \quad S_y = \sum_{i=1}^n \frac{\tau_i}{\sigma_i^2},$$

$$S_{xx} = \sum_{i=1}^n \frac{\dot{\gamma}_i^2}{\sigma_i^2}, \quad S_{xy} = \sum_{i=1}^n \frac{\dot{\gamma}_i \tau_i}{\sigma_i^2} \quad (9)$$

which are the elements of the following matrices:

$$\mathbf{A} = \begin{bmatrix} S_1 & S_x \\ S_x & S_{xx} \end{bmatrix}, \quad \mathbf{A}_0 = \begin{bmatrix} S_y & S_x \\ S_{xy} & S_{xx} \end{bmatrix}, \quad \mathbf{A}_1 = \begin{bmatrix} S_1 & S_y \\ S_x & S_{xy} \end{bmatrix} \quad (10)$$

Together with summations S_1 and S_{xx} , determinants of previous matrices are used to numerically assess rheological parameters τ_0 and μ_p as well as their uncertainties σ_{τ_0} and σ_{μ_p} according to:

$$\tau_0 = \frac{\det(\mathbf{A}_0)}{\det(\mathbf{A})}, \quad \sigma_{\tau_0} = \sqrt{\frac{S_{xx}}{\det(\mathbf{A})}}, \quad \mu_p = \frac{\det(\mathbf{A}_1)}{\det(\mathbf{A})},$$

$$\sigma_{\mu_p} = \sqrt{\frac{S_1}{\det(\mathbf{A})}} \quad (11)$$

Table 4 shows the data provided to students and Table 5 exemplifies key cells to implement least squares method in

MS Excel based on Equation (9–11). In this example, raw data $(\dot{\gamma}_i, \tau_i, \sigma_i)$ occupy columns A, B, and C whereas summands for Equation (9) are assessed in columns D to H. Rheological parameters are assessed as $\tau_0 = 41.1 \pm 0.3$ Pa and $\mu_p = 10.04 \pm 0.04$ Pa · s (MS Excel trend line parameters are $\tau_0 = 41.815$ Pa and $\mu_p = 9.932$ Pa · s).

Before tackling this problem, a general linear equation $y = a_0 + a_1 x$ is numerically fitted against triplets (x_i, y_i, σ_i) with different uncertainties σ_i . At this moment of the lecture, built-in functions SUM and MDETERM are explained so as to implement summations and determinants. It is also shown that y-intercept a_0 and slope a_1 are the very same equation parameters obtained via MS Excel trend line if (and only if) all uncertainties σ_i are rigorously the same in the data set.

3.4 | Non-tabulated entries in a saturated steam table

In this problem students must evaluate thermodynamic properties of saturated steam [19] for a non-tabulated saturation pressure entry. A steam table extract is provided (Table 6) and students may use Lagrange interpolation or, as

TABLE 4 Data concerning a Bingham plastic fluid whose rheological parameters must be determined

| $\dot{\gamma}_i$ (s ⁻¹) | τ_i (Pa) | σ_i (Pa) |
|-------------------------------------|---------------|-----------------|
| 0.20 | 42.8 | 0.4 |
| 1.45 | 52.4 | 0.4 |
| 2.10 | 61.9 | 0.4 |
| 3.00 | 71.4 | 0.4 |
| 3.90 | 80.9 | 0.4 |
| 5.10 | 100.0 | 0.4 |
| 6.60 | 102.5 | 0.4 |
| 7.80 | 123.8 | 0.2 |
| 9.10 | 128.6 | 0.2 |
| 10.70 | 151.9 | 0.2 |
| 12.20 | 164.2 | 0.2 |
| 14.00 | 178.5 | 0.2 |
| 15.90 | 201.1 | 0.1 |
| 17.90 | 221.3 | 0.1 |
| 19.90 | 235.6 | 0.1 |

TABLE 5 Example of MS Excel cells to implement least squares method to best-fit rheological parameters of a given Bingham plastic fluid

| Value | Cell(s) | Expression(s) in MS Excel (as seen in formula bar) |
|-----------------------------|--------------------|--|
| S_1 | S (D17 renamed) | =SUM(D2:D16) |
| S_x | Sx (E17 renamed) | =SUM(E2:E16) |
| S_y | Sy (F17 renamed) | =SUM(F2:F16) |
| S_{xx} | Sxx (G17 renamed) | =SUM(G2:G16) |
| S_{xy} | Sxy (H17 renamed) | =SUM(H2:H16) |
| A | A21; B21; A22; B22 | =S; =Sx; =Sx; =Sxx (respectively) |
| A₀ | D21; E21; D22; E22 | =Sy; =Sx; =Sxy; =Sxx (respectively) |
| A₁ | G21; H21; G22; H22 | =S; =Sy; =Sx; =Sxy (respectively) |
| det(A) | J1 | =MDETERM(A21:B22) |
| det(A₀) | J2 | =MDETERM(D21:E22) |
| det(A₁) | J3 | =MDETERM(G21:H22) |
| τ_0 | J5 | =J2/J1 |
| σ_{τ_0} | J6 | =SQRT(Sxx/J1) |
| μ_p | J8 | =J3/J1 |
| $\sigma_{\mu p}$ | J9 | =SQRT(S/J1) |

Table 6 is uniform (i.e., regularly spaced) with respect to pressure entries, Gregory-Newton interpolation [9]. Interpolation through Newton's divided differences is here omitted for brevity. Interpolating polynomials are limited to degree 3 because there are 4 entries in Table 6. Students are advised to use renamed cell "P" for the non-tabulated pressure entry P as well as renamed cells "P0.," "P1.," "P2.," and "P3." for pressure entries P_i in Table 6 (but only "P0." and "P1." will be required in Gregory-Newton interpolation).

In order to implement the 3rd degree Lagrange interpolating polynomial, a column (e.g., adjacent to the saturation pressure entries) should receive the following auxiliary polynomials:

$$L_0(P) = \frac{(P - P_1)(P - P_2)(P - P_3)}{(P_0 - P_1)(P_0 - P_2)(P_0 - P_3)},$$

$$L_1(P) = \frac{(P - P_0)(P - P_2)(P - P_3)}{(P_1 - P_0)(P_1 - P_2)(P_1 - P_3)}$$

$$L_2(P) = \frac{(P - P_0)(P - P_1)(P - P_3)}{(P_2 - P_0)(P_2 - P_1)(P_2 - P_3)},$$

$$L_3(P) = \frac{(P - P_0)(P - P_1)(P - P_2)}{(P_3 - P_0)(P_3 - P_1)(P_3 - P_2)} \quad (12)$$

A further column receives the tabulated entries of the thermodynamic property to be interpolated, here indicated as ϕ_i . Accordingly, 3rd degree Lagrange interpolating polynomial results as:

$$p_3(P) = \sum_{i=0}^3 L_i(P) \cdot \phi_i \Rightarrow p_3(P) = L_0(P) \cdot \phi_0 + L_1(P) \cdot \phi_1 + L_2(P) \cdot \phi_2 + L_3(P) \cdot \phi_3 \quad (13)$$

During the lecture, built-in function SUMPRODUCT is explained so as to implement Equation (13).

Gregory-Newton interpolation can be implemented via a table of forward differences. First, second, and third forward differences of thermodynamic property ϕ are defined respectively as:

$$\Delta\phi_i = \phi_{i+1} - \phi_i, \quad \Delta^2\phi_i = \Delta\phi_{i+1} - \Delta\phi_i,$$

$$\Delta^3\phi_i = \Delta^2\phi_{i+1} - \Delta^2\phi_i \quad (14)$$

TABLE 6 Some thermodynamic properties of saturated steam for interpolation purposes

| Saturation pressure, bar | Saturation temperature, °C | Specific volume – vapor, m ³ /kg | Latent heat of vaporization, kJ/kg | Specific heat, kJ/(kg · K) |
|--------------------------|----------------------------|---|------------------------------------|----------------------------|
| 7.0 | 165.0 | 0.273 | 2065 | 2.44 |
| 8.0 | 170.4 | 0.240 | 2046 | 2.50 |
| 9.0 | 175.4 | 0.215 | 2030 | 2.55 |
| 10.0 | 179.9 | 0.194 | 2013 | 2.59 |

Only the first forward difference $\Delta\phi_0 = \phi_1 - \phi_0$ must be typed in a given cell while all remaining differences in Equation (14) can be straightforwardly calculated via replication from $\Delta\phi_0$ to cells below and sideways (e.g., to the right). Students are advised to use a renamed cell “r.” for the parameter:

$$r(P) = \frac{P - P_0}{\Delta P} \Rightarrow r = \frac{P - P_0}{P_1 - P_0} \quad (15)$$

Auxiliary columns must be used to evaluate (via replication with proper adjustments) the required factorials as well as the following sequence of Gregory-Newton interpolating polynomials:

$$p_1(P) = \phi_0 + \frac{r}{1!} \Delta\phi_0, \quad p_2(P) = p_1(P) + \frac{r(r-1)}{2!} \Delta^2\phi_0,$$

$$p_3(P) = p_2(P) + \frac{r(r-1)(r-2)}{3!} \Delta^3\phi_0 \quad (16)$$

By sharing the same cells with saturated steam data (e.g., saturation pressures P_i in renamed cells “P0.” to “P3.” in column B and thermodynamic properties ϕ_i in cells C2 to C5), Table 7 exemplifies key cells to implement Lagrange and Gregory-Newton interpolations. Indices i and their factorials in Equation (16) occupy cells G2 to G5 and H2 to H5, respectively.

Equations (13) and (16) yield the same results, thus corroborating existence and uniqueness concepts [9] previously explained to students during the lecture. Based on values in Table 6, 3rd degree interpolation for $P = 9.2$ bar provides 175.4°C for saturation temperature, 0.211 m³/kg for specific volume, 2,027 kJ/kg for latent heat of vaporization, and 2.56 kJ/(kg · K) for specific heat.

3.5 | Extraction yield in a continuous-flow process

Continuous-flow extraction performance can be expressed as the extraction yield on mass basis, $Y = Y(t)$, defined as the compound mass extracted up to instant t divided by the total mass m_{total} of raw material within the extractor. If $c_{\text{exit}}(t)$ is the instantaneous compound concentration in the solvent (i.e., fluid phase) at the extractor exit, the following relation holds [14]:

$$Y(t) = \frac{1}{m_{\text{total}}} \int_0^t \dot{V}_{\text{solvent}} c_{\text{exit}}(t') dt' = \frac{\dot{V}_{\text{solvent}}}{m_{\text{total}}} \int_0^t c_{\text{exit}}(t') dt' \quad (17)$$

where the volumetric flow rate of solvent \dot{V}_{solvent} is allegedly constant over the extraction process. In this problem students must numerically assess the integral in

Equation (17) so as to evaluate a given extraction yield Y . Table 8 shows the data provided to students concerning the exit concentration c_{exit} for increasing extraction time t . The total mass of raw material and the volumetric flow rate of solvent are given as $m_{\text{total}} = 20$ g and $\dot{V}_{\text{solvent}} = 0.03$ L/min, respectively.

Students are instructed to use either trapezoidal or Simpson's rules of integration, which are similar to each other. Those numerical integration methods lead to the following expression [9]:

$$\int_0^t c_{\text{exit}}(t') dt' \cong \frac{\Delta t}{m} \sum_{i=0}^n w_i c_{\text{exit},i} = \frac{\Delta t}{m} (w_0 c_{\text{exit},0} + w_1 c_{\text{exit},1} + w_2 c_{\text{exit},2} + \dots + w_{n-1} c_{\text{exit},n-1} + w_n c_{\text{exit},n}) \quad (18)$$

where $c_{\text{exit},i}$ are the values of fluid-phase extract concentrations at extractor exit. Table 9 shows parameter m as well as weighting factors w_i in Equation (18) for each numerical integration method.

The numerical integrations can be implemented in MS Excel by sharing the same cells for $c_{\text{exit},i}$, for example, cells B2 to B22 (besides cells in column A for time t_i), with expressions exemplified in Table 10. Different columns are used for related weighting factors w_i , for example, cells C2 to C22 for trapezoidal rule and cells D2 to D22 for Simpson's rule, which can be evaluated via replication while adjusting w_n to 1. Renamed cells “dt,” “mtotal,” and “Vsolv” are used respectively for Δt (= 3 min), m_{total} and \dot{V}_{solvent} . The built-in function SUMPRODUCT is recalled during the lecture so as to assess Equation (18). Extraction yield results as $Y = 0.00126$ g-extract/g-solids via trapezoidal rule and $Y = 0.00127$ g-extract/g-solids through Simpson's rule, which are close to each other.

3.6 | Temperature-time profile of food under cooling

In thermal problems with low Biot numbers ($Bi < 0.1$), heat conduction is much faster than heat convection so that uniform temperature can be assumed within the given body and lumped-parameter analysis can be applied [1]. Inspired by the blast-cooling of catering meals examined in [12], students are asked to obtain temperature-time profiles $T(t)$ of food under cooling as ruled by the following 1st-order ordinary differential equation (i.e., Newton's law of cooling):

$$\frac{dT(t)}{dt} = \beta' [T_{\text{air}} - T(t)] \quad (19)$$

TABLE 7 Example of MS Excel cells to implement 3rd degree Lagrange and Gregory-Newton interpolations to obtain non-tabulated entries concerning a saturated steam table

| Interpolation | Values | Cells | Expressions in MS Excel (as seen in formula bar) |
|----------------|------------------|------------|---|
| Lagrange | $L_0(P)$ | A2 | $= (P - P1.)*(P - P2.)*(P - P3.)/((P0.-P1.)*(P0.-P2.)*(P0.-P3.))$ |
| | $L_1(P)$ | A3 | $= (P - P0.)*(P - P2.)*(P - P3.)/((P1.-P0.)*(P1.-P2.)*(P1.-P3.))$ |
| | $L_2(P)$ | A4 | $= (P - P0.)*(P - P1.)*(P - P3.)/((P2.-P0.)*(P2.-P1.)*(P2.-P3.))$ |
| | $L_3(P)$ | A5 | $= (P - P0.)*(P - P1.)*(P - P2.)/((P3.-P0.)*(P3.-P1.)*(P3.-P2.))$ |
| | $p_3(P)$ | C7 | $= \text{SUMPRODUCT}(A2:A5;C2:C5)$ |
| Gregory-Newton | $\Delta\phi_i$ | D2; D3; D4 | $= C3 - C2; = C4 - C3; = C5 - C4$ |
| | $\Delta^2\phi_i$ | E2; E3 | $= D3 - D2; = D4 - DC3$ |
| | $\Delta^3\phi_i$ | F2 | $= E3 - E2$ |
| | $p_0(P)$ | I2 | $= C2$ |
| | $p_1(P)$ | I3 | $= I2 + D2*r./H3$ |
| | $p_2(P)$ | I4 | $= I3 + E2*r.*(r. - 1)/H4$ |
| | $p_3(P)$ | I5 | $= I4 + F2*r.*(r. - 1)*(r. - 2)/H5$ |

In this initial value problem, lumped-parameter β' is constant and defined as [12]:

$$\beta' = \frac{1/(\rho c l_{1/2})}{1/h + l_{1/2}/k} \quad (20)$$

TABLE 8 Data concerning an extraction process whose yield should be evaluated

| t (min) | c_{exit} (g/L) |
|-----------|-------------------------|
| 0 | 0.00 |
| 3 | 0.42 |
| 6 | 0.56 |
| 9 | 0.45 |
| 12 | 0.34 |
| 15 | 0.26 |
| 18 | 0.18 |
| 21 | 0.13 |
| 24 | 0.10 |
| 27 | 0.08 |
| 30 | 0.06 |
| 33 | 0.05 |
| 36 | 0.04 |
| 39 | 0.03 |
| 42 | 0.03 |
| 45 | 0.02 |
| 48 | 0.02 |
| 51 | 0.01 |
| 54 | 0.01 |
| 57 | 0.00 |
| 60 | 0.00 |

Besides convective heat transfer coefficient h , meal parameters include density ρ , specific heat c , half-length $l_{1/2}$ (between core and horizontal edge) and thermal conductivity k . Initial condition is imposed as $T(0) = T_0$ and the cooling air temperature T_{air} is allegedly constant for didactic and comparison purposes against the following analytical solution:

$$T_{\text{exact}}(t) = T_{\text{air}} + (T_0 - T_{\text{air}})\exp(-\beta't) \quad (21)$$

In order to numerically solve Equation (19), students apply 4th-order Runge-Kutta method [9]. Renamed cells “h,” “rho,” “c.,” “lhalf,” “k,” “T0,” and “Tair” store the necessary model parameters, whose values are provided as in Table 11. Lumped-parameter is assessed as $\beta' = 0.000365 \text{ s}^{-1}$ in a renamed cell ‘beta’ while a renamed cell “dt” stores the advancing time step set as $\Delta t = 60 \text{ s}$.

Students are warned that implementing 4th-order Runge-Kutta method is relatively simpler in this case as the right-hand side (RHS) of Equation (19) lacks explicit dependence on the independent variable, that is, time t . The following numerical marching expressions should be implemented:

$$t_{i+1} = t_i + \Delta t \quad \text{and} \quad T_{i+1} = T_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \quad (22)$$

Auxiliary parameters k_i in Equation (22) are evaluated in a similar way by using the time step Δt and function $f(T) = \beta'(T_{\text{air}} - T)$, that is, RHS of Equation (19), with the following arguments:

$$k_1 = \Delta t f(T_i), \quad k_2 = \Delta t f\left(T_i + \frac{1}{2}k_1\right),$$

TABLE 9 Parameters to implement trapezoidal or Simpson's rules of numerical integration

| Parameter | m | w_0 | w_1 | w_2 | w_3 | w_4 | w_5 | ... | w_{n-5} | w_{n-4} | w_{n-3} | w_{n-2} | w_{n-1} | w_n |
|------------------|-----|-------|-------|-------|-------|-------|-------|-----|-----------|-----------|-----------|-----------|-----------|-------|
| Trapezoidal rule | 2 | 1 | 2 | 2 | 2 | 2 | 2 | ... | 2 | 2 | 2 | 2 | 2 | 1 |
| Simpson's rule | 3 | 1 | 4 | 2 | 4 | 2 | 4 | ... | 4 | 2 | 4 | 2 | 4 | 1 |

$$k_3 = \Delta t f(T_i + \frac{1}{2}k_2), k_4 = \Delta t f(T_i + k_3) \quad (23)$$

One may profit from this similarity to implement the numerical method in MS Excel via copy-and-paste of triplets (T, f, k) with proper amendments, as Table 12 exemplifies for a given iterative level (e.g., row 7). The numerical method starts from the initial condition, that is, $t_{i=0} = 0$ and $T_{i=0} = T_0$. Up to three decimal digits, numerically simulated temperatures T_i (e.g., in column C) are exactly the same as analytical counterparts T_{exact} (e.g., in column A) assessed via Equation (21).

3.7 | Damped harmonic oscillator

Harmonic oscillators occur in different natural phenomena such as pendulums, acoustical systems, waves, and RLC circuits [6]. In this problem, students must numerically simulate a damped harmonic oscillator modeled by the following 2nd-order ordinary differential equation:

$$m \frac{d^2x}{dt^2} + B \frac{dx}{dt} + kx = 0 \Leftrightarrow \frac{d^2x}{dt^2} = -\frac{1}{m} \left(kx + B \frac{dx}{dt} \right) \quad (24)$$

In this initial value problem, $x = x(t)$ is the position at time t , m is the mass, B is the damping coefficient, and k is the restoring force constant. Initial conditions are imposed as:

$$x(0) = x_0 \text{ and } v(0) = \left. \frac{dx}{dt} \right|_{t=0} = v_0 \quad (25)$$

where x_0 is the initial displacement and v_0 is the initial velocity.

Students must apply Runge-Kutta-Nyström method [9] so as to numerically solve Equation (24) subjected to Equation (25). Renamed cells “m,” “B,” “k,” “x0,” and “v0” store

model parameters, whose values are given in Table 13. Different values are suggested to coefficient B consistent with the damping type (i.e., under, critical or over-damping) to be numerically simulated. A renamed cell “dt” stores the advancing time step suitably set as $\Delta t = 0.05$ s to simulate some under-damping oscillations. Lower Δt can be used to slowly depict either over or critical damping.

Bearing in mind functions f in Equation (27) ahead, students are again advised that implementing Runge-Kutta-Nyström method to solve this problem is slightly simpler because Equation (24) does not explicitly depend on variable t . Numerical marching expressions should be implemented as:

$$\begin{aligned} t_{i+1} &= t_i + \Delta t, \quad x_{i+1} = x_i + \Delta t \left(v_i + \frac{k_1 + k_2 + k_3}{3} \right), \\ v_{i+1} &= v_i + \frac{k_1 + 2k_2 + 2k_3 + k_4}{3} \end{aligned} \quad (26)$$

where parameters k_i are calculated using time step Δt and function $f(x, v) = -(kx + Bv)/m$ as:

$$\begin{aligned} k_1 &= \frac{\Delta t}{2} f(x_i, v_i), \quad k_2 = \frac{\Delta t}{2} f(x_i + K, v_i + k_1), \\ k_3 &= \frac{\Delta t}{2} f(x_i + K, v_i + k_2), \quad k_4 = \frac{\Delta t}{2} f(x_i + L, v_i + 2k_3) \end{aligned} \quad (27)$$

Additional auxiliary parameters K and L are defined as follows:

$$K = \frac{\Delta t}{2} \left(v_i + \frac{k_1}{2} \right) \text{ and } L = \Delta t (v_i + k_3) \quad (28)$$

Profiting again from their resemblance, calculations can be implemented in MS Excel via copy-and-paste of quintets

TABLE 10 Example of MS Excel cells to implement trapezoidal and Simpson's integration rules to assess extraction yield in continuous-flow process

| Value ^a | Expression for trapezoidal rule | Expression for Simpson's rule |
|------------------------------------|----------------------------------|----------------------------------|
| $\int_0^t c_{\text{exit}}(t') dt'$ | = SUMPRODUCT(B2:B22;C2:C22)*dt/2 | = SUMPRODUCT(B2:B22;D2:D22)*dt/3 |
| $Y(t)$ | = G6*Vsolv/mtotal | = G9*Vsolv/mtotal |

^aNumerical integrals are stored in cells G6 and G9 respectively in trapezoidal and Simpson's rules.

TABLE 11 Model parameters to numerically simulate temperature-time profiles of food under cooling

| $h_c, \text{W}/(\text{m}^2 \cdot \text{K})$ | $\rho, \text{kg}/\text{m}^3$ | $c, \text{J}/(\text{kg} \cdot \text{K})$ | $k, \text{W}/(\text{m} \cdot \text{K})$ | $l_{1/2}, \text{m}$ | T_0, K | T_{air}, K |
|---|------------------------------|--|---|---------------------|-----------------|----------------------------|
| 50.0 | 1014.4 | 3545.5 | 0.488 | 0.015 | 300 | 260 |

(x, v, f, k, P) followed by proper adjustments, where P stands for either parameter K or L . Rigorously, one should only replicate and adjust the quartet (x, v, f, k) so as to evaluate k_4 . The numerical method starts from initial conditions $t_{i=0} = 0$, $x_{i=0} = x_0$, and $v_{i=0} = v_0$, and Table 14 exemplifies key cells implemented for a given iterative level (e.g., row 7).

For comparison purposes, the analytical solution for under-damping is deduced as [6]:

$$x_{\text{under}}(t) = A \exp(-\gamma t) \cos(\omega_1 t - \alpha) \quad (29)$$

where (by recalling that null initial velocity was assumed, $v_0 = 0$):

$$\begin{aligned} \omega_1 &= \sqrt{\omega_0^2 - \gamma^2}, \quad \omega_0 = \sqrt{\frac{k}{m}}, \quad \gamma = \frac{B}{2m}, \\ \alpha &= \arctan\left(\frac{1}{\omega_1}\right), \quad A = \frac{x_0}{\cos(\alpha)} \end{aligned} \quad (30)$$

As Table 14 shows, the analytical solution was implemented (e.g., in column A) with the help of renamed cells for parameters in Equations (29) and (30). Numerical positions x_i (e.g., in column C) are exactly the same as analytical counterparts $x_{\text{under},i}$ up to three decimal digits.

Analytical solutions for over and critical damping are omitted for brevity and they can be found in [6].

3.8 | Steady-state one-dimensional heat conduction

The blast-cooling of catering meals [12] is again referred in this lecture but it is discussed that temperature may vary inside foodstuff. This thermal food processing was studied in [13] via finite differences method (FDM), whose fundamentals [1] are taught to students together with different types of boundary conditions (e.g., Dirichlet, Neumann, Robin, radiative heat transfer). A steady-state 1-D thermal problem is put forward so that students must numerically simulate via FDM the temperature variation $T(x)$ inside a homogenous solid slab of thickness L . The problem is extended to include an internal energy source \dot{e} (i.e., energy released per unit volume) so that the governing ordinary differential equation becomes:

$$\frac{d^2 T}{dx^2} + \dot{e} = 0 \quad (31)$$

Boundary conditions comprise Dirichlet condition at $x = 0$ and Robin condition at $x = L$, namely:

TABLE 12 Example of MS Excel cells to implement 4th-order Runge-Kutta method to numerically simulate temperature-time profiles of food under cooling

| Value | Cell | Expression in MS Excel (as seen in formula bar for row 7) |
|--------------------|------|---|
| t_i | B7 | = B6 + dt |
| T_i | C7 | = C6 + (E6 + 2*H6 + 2*K6 + N6)/6 |
| $f(T_i)$ | D7 | = beta*(Tair - C7) |
| k_1 | E7 | = dt*D7 |
| $T_i + k_1/2$ | F7 | = \$C7 + E7/2 |
| $f(T_i + k_1/2)$ | G7 | = beta*(Tair - F7) |
| k_2 | H7 | = dt*G7 |
| $T_i + k_2/2$ | I7 | = \$C7 + H7/2 |
| $f(T_i + k_2/2)$ | J7 | = beta*(Tair - I7) |
| k_3 | K7 | = dt*J7 |
| $T_i + k_3$ | L7 | = \$C7 + K7 |
| $f(T_i + k_3)$ | M7 | = beta*(Tair - L7) |
| k_4 | N7 | = dt*M7 |
| T_{exact} | A7 | = Tair + (T0 - Tair)*EXP(-beta*B7) |

TABLE 13 Model parameters to numerically simulate damped harmonic oscillator

| <i>m</i> (kg) | <i>B</i> (kg/s) | | | <i>k</i> (N/m) | <i>x</i> ₀ (m) | <i>v</i> ₀ (m/s) |
|---------------|-----------------|------------------|--------------|----------------|---------------------------|-----------------------------|
| | Under-damping | Critical damping | Over-damping | | | |
| 5 | 10 | 90 | 150 | 405 | 0.15 | 0 |

$$T(0) = T_0 \text{ and } -k \frac{dT}{dx} \Big|_{x=L} = h[T_\infty - T(L)] \quad (32) \qquad \Rightarrow \quad T_4 = \frac{h\Delta x T_\infty + kT_3 + \frac{1}{2}\dot{e}(\Delta x)^2}{h\Delta x + k} \quad (33)$$

where temperature *T*₀ is kept constant at *x* = 0, *k* is thermal conductivity, *h* is convective heat transfer, and *T*_∞ = constant is surrounding air temperature (sufficiently away from *x* = *L*).

As Figure 3 shows, the solution domain is discretized into five nodal points, numbered from *i* = 0 to *i* = *m* = 4, so that Δ*x* = *L*/*m* is the distance between them. Bearing in mind that *T*_{*i*=0} = *T*₀ in all iterations, FDM applied to Equations (31) and (32) renders the following algebraic equations:

$$1 \leq i \leq 3 : \frac{T_{i-1} - 2T_i + T_{i+1}}{(\Delta x)^2} + \frac{\dot{e}}{k} = 0$$

$$\Rightarrow T_i = \frac{T_{i-1} + T_{i+1}}{2} + \frac{\dot{e}(\Delta x)^2}{2k}$$

$$i = 4 : h(T_\infty - T_4) + k \frac{T_3 - T_4}{\Delta x} + \frac{\dot{e}\Delta x}{2} = 0$$

Students may solve Equation (33) via either Jacobi or Gauss-Seidel method [9]. Besides “m” and “dx,” renamed cells “L,” “e,” “k,” “h,” “T0,” and “Tinf” store model parameters, whose values are in Table 15. Table 16 exemplifies key cells implemented via Gauss-Seidel method for a given iterative level. Expressions for *T*₂ and *T*₃ can be replicated from the expression for *T*₁ and numerical results can be compared with the following analytical solution:

$$T_{\text{exact}}(x) = T_0 + \frac{\dot{e}L + \dot{e}L^2h/(2k) + h(T_\infty - T_0)}{hL + k}x - \frac{\dot{e}}{2k}x^2 \quad (34)$$

By starting from the initial guess *T*₁⁽⁰⁾ = *T*₂⁽⁰⁾ = *T*₃⁽⁰⁾ = *T*₄⁽⁰⁾ = *T*₀, convergence of the numerical solution (against *T*_{exact}) with 1 decimal digit is observed after 46 iterations.

TABLE 14 Example of MS Excel cells to implement Runge-Kutta-Nyström method to numerically simulate damped harmonic oscillators

| Value | Cell | Expression in MS Excel (as seen in formula bar for row 7) |
|---|------|---|
| <i>t</i> _{<i>i</i>} | B7 | = B6 + dt |
| <i>x</i> _{<i>i</i>} | C7 | = C6 + dt*(D6 + (F6 + K6 + P6)/3) |
| <i>v</i> _{<i>i</i>} | D7 | = D6 + (F6 + 2*K6 + 2*P6 + U6)/3 |
| <i>f</i> (<i>x</i> _{<i>i</i>} , <i>v</i> _{<i>i</i>}) | E7 | = - (k*C7 + B*D7)/m |
| <i>k</i> ₁ | F7 | = 0.5*dt*E7 |
| <i>K</i> | G7 | = 0.5*dt*(D7 + \$F7/2) |
| <i>x</i> _{<i>i</i>} + <i>K</i> | H7 | = \$C7 + G7 |
| <i>v</i> _{<i>i</i>} + <i>k</i> ₁ | I7 | = \$D7 + F7 |
| <i>f</i> (<i>x</i> _{<i>i</i>} + <i>K</i> , <i>v</i> _{<i>i</i>} + <i>k</i> ₁) | J7 | = - (k*H7 + B*I7)/m |
| <i>k</i> ₂ | K7 | = 0.5*dt*J7 |
| <i>K</i> | L7 | = 0.5*dt*(D7 + \$F7/2) |
| <i>x</i> _{<i>i</i>} + <i>K</i> | M7 | = \$C7 + L7 |
| <i>v</i> _{<i>i</i>} + <i>k</i> ₂ | N7 | = \$D7 + K7 |
| <i>f</i> (<i>x</i> _{<i>i</i>} + <i>K</i> , <i>v</i> _{<i>i</i>} + <i>k</i> ₂) | O7 | = - (k*M7 + B*N7)/m |
| <i>k</i> ₃ | P7 | = 0.5*dt*O7 |
| <i>L</i> | Q7 | = dt*(D7 + P7) |
| <i>x</i> _{<i>i</i>} + <i>L</i> | R7 | = \$C7 + Q7 |
| <i>v</i> _{<i>i</i>} + 2 <i>k</i> ₃ | S7 | = \$D7 + 2*P7 |
| <i>f</i> (<i>x</i> _{<i>i</i>} + <i>L</i> , <i>v</i> _{<i>i</i>} + 2 <i>k</i> ₃) | T7 | = - (k*R7 + B*S7)/m |
| <i>k</i> ₄ | U7 | = 0.5*dt*T7 |
| <i>x</i> _{under} | A7 | = A*EXP(-gamma*B7)* COS(omega*1*B7 - alpha) |

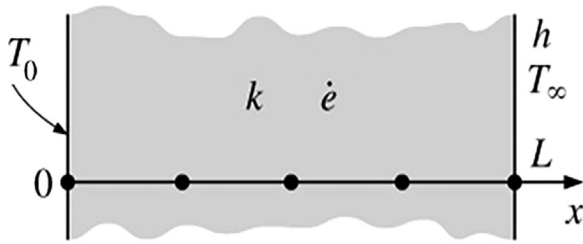


FIGURE 3 Discretization of solid slab for FDM simulation of steady-state 1-D heat conduction

4 | DISCUSSION

Industrial activities have progressively relied on computational tools as both hardware and software resources have become more affordable. Accordingly, educational opportunities to familiarize engineering students with numerical methods should be supported and explored. Computer-assisted didactic activities may complement and benefit not only theoretical concepts developed in lectures but also didactic experiments accomplished at physical laboratories.

As far as teaching numerical methods to agroindustrial engineering students is concerned, challenges are at least two-fold in Brazil, namely: (i) the holdup of the agroindustrial sector with respect to virtualization [10,16] allied to (ii) the feeble aptitude of Brazilian students towards mathematics as evidenced by international assessments [11]. In order to overcome those hurdles, Numerical Methods is a core discipline offered in the 2nd year (out of five) of biosystems and food engineering programs at USP Pirassununga campus. By introducing computational methods to students while providing the related training at a didactic computational laboratory, this discipline aims at fostering cross-link to forthcoming disciplines in aforementioned engineering programs.

Yet, Pirassununga campus is *sui generis* as it hosts no academic unit dedicated to modeling and simulation to some extent and where research is strongly biased to experimental methods. Such unfavorable scenario ratifies the challenges cited in the previous paragraph bearing in mind not only students' interest in computational methods but also faculty staff's. For instance, despite Numerical Methods is a compulsory discipline to food engineering students, it is not requisite to any subsequent discipline (whether compulsory or elective) in their curriculum.

While one may claim that cross-disciplinary integration herein described is (say) unilateral, ineffective cross-link

would be evident had Numerical Method been taught by solely relying on, for example, functions f of variables x , y , and z completely unrelated to other disciplines. This is not the case of the discipline reported in the present paper as it addresses the numerical solution of problems concerning several engineering disciplines (Table 1), in line with authors' effort to enhance students' expertise in computational methods.

Feedback from both food and biosystems engineering students has been encouraging. In recent assessments of Numerical Methods, students have confirmed that accomplishing hands-on activities at individual computers do promote pedagogical ethos, which is definitely reinforced by the consent to help each other (and students are indeed observed to do so in lectures). They have equally endorsed that learning-by-doing activities at a didactic computational laboratory intensify their commitment towards the discipline in the short term.

In the long term, an increasing number of students have become interested in taking up Applied Computational Methods (in the 4th year) after having attended Numerical Methods (in the 2nd year). In fact, authors launched this elective discipline in 2016 in response to students' demand to complement their expertise. As "advertised" to students, Numerical Methods uses MS Excel to present the numerical solution of agroindustrial problems while Applied Computational Methods uses agroindustrial problems to present advanced MS Excel tools (e.g., VBA codes).

In terms of connecting case studies with other disciplines, students have been able to compare the numerical solution of harmonic oscillators (section 3.7) with analytical counterparts in core physics disciplines. As far as calculus courses are concerned, they have been able to identify links with definite integrals (section 3.5) and ordinary differential equations (sections 3.6 and 3.8). While further disciplines listed in Table 1 are upcoming ones, it is worth citing that in Biosystems Engineering program Numerical Methods is requisite to Applied Computational Methods (elective discipline), Artificial Intelligence (compulsory discipline), and Operations Research & Optimization of Agricultural Systems (elective discipline).

Numerical Methods has also helped students in the Supervised Internship discipline, which is the closing one in both food and biosystems engineering curricula. According to testimonies of senior undergraduate students in this discipline, internship recruitments have regularly pointed to MS Excel (or similar environment) as most wanted skills. A

TABLE 15 Model parameters to numerically simulate steady-state 1-D heat conduction within a solid slab

| L , m | \dot{e} , W/m ³ | k , W/(m · K) | h , W/(m ² · K) | T_0 , K | T_{inf} , K |
|---------|------------------------------|-----------------|------------------------------|-----------|---------------|
| 0.04 | 5.0×10^6 | 28.0 | 45.0 | 273 | 303 |

TABLE 16 Example of MS Excel cells to implement Gauss-Seidel method to numerically simulate via finite differences method the steady-state 1-D heat conduction within a solid slab

| Value | Cells | Expressions implemented in MS Excel (as seen in formula bar) |
|--------------------------|----------|--|
| $T_0^{(j+1)}$ | B7 | = T0 |
| $T_1^{(j+1)}$ | C7 | = (B7 + D6)/2 + e.*dx*dx/(2*k) |
| $T_2^{(j+1)}$ | D7 | = (C7 + E6)/2 + e.*dx*dx/(2*k) |
| $T_3^{(j+1)}$ | E7 | = (D7 + F6)/2 + e.*dx*dx/(2*k) |
| $T_4^{(j+1)}$ | F7 | = (h*dx*Tinf + k*E7 + e.*dx*dx/2)/(h*dx + k) |
| x_i | B3 to F3 | = 0; = B3 + dx; = C3 + dx; = D3 + dx; = E3 + dx |
| $T_{\text{exact}}^{(*)}$ | B4 to F4 | = T0 + B3*(e.*L.+e.*L.*L.*h/(2*k) + h*(Tinf - T0))/(h*L.+k) - B3*B3*e./(2*k) |

(*) Expression to evaluate analytical solution $T_{\text{exact}}(x)$ at $x = x_0$. Counterparts for remaining x_i can be obtained via replication.

similar feedback is given from our graduate students with respect to job recruitments in Brazilian agroindustrial sector.

The cross-disciplinary integration attempted in Numerical Methods is mutually profitable. Given that RD&I has increasingly combined analytical, experimental, and numerical methods, forthcoming disciplines in food and biosystems engineering programs can restore a helpful tool. Once implemented by relying on comprehensive models, agroindustrial engineers can confidently apply numerical solvers and simulators to solve real-world problems.

In their turn, food and biosystems engineering students become familiar with virtualization by numerically solving problems while testing different model parameters and assumptions. Computational modeling entails forward and backward movements where each assumption must be re-examined in view of others. Care should be exercised to prevent over-complicating or over-simplifying a certain approach to the detriment of another. This task is non-transferable to numerical simulators (as it is up to the computational modeler!) and such a skill must be trained among agroindustrial engineering students.

Last but not least, all course material (e.g., MS Power Point slides from lectures, examples discussed in lectures, and list of exercises) becomes available online after face-to-face lectures (at so-called TIDIA-AE depository system). Provided that MS Excel is typically installed in home PCs and notebooks, students have also profited from distance learning environment.

5 | CONCLUSION

In view of prospective challenges and opportunities for virtualization in agroindustrial engineering, knowledge, and skills in computational modeling must be continuously developed. At USP Pirassununga campus, Numerical Methods is a core undergraduate discipline in food and biosystems engineering curricula, which has initiated computational modeling education and training at a didactic

computational laboratory. By putting forward hands-on activities linking computational methods with engineering problems, this discipline has indeed increased students' commitment and interest toward numerical methods. Such synergy between scientific computing and agroindustrial engineering should be fostered and supported in subsequent disciplines as exemplified in this paper, on pain of letting modeling and virtualization remain underexplored in agri-bio-food engineering.

ORCID

Jose A. Rabi  <http://orcid.org/0000-0001-5041-4680>

REFERENCES

1. Y. A. Çengel, *Heat and mass transfer—A practical approach*, 3rd ed., McGraw-Hill, New York, 2007.
2. A. K. Datta and S. S. Sablani, Mathematical modeling techniques in food and bioprocess: an overview. In: *Handbook of food and bioprocess modeling techniques*, (S. S. Sablani, M. S. Rahman, A. K. Datta, and A. R. Mujumdar, eds.), CRC Press, Boca Raton, 2007, 1–11.
3. M. L. de Souza-Santos, *Solid fuels combustion and gasification: Modeling, simulation, and equipment operation*, 2nd ed., CRC Press, Boca Raton, 2010.
4. M. S. Diaz and E. A. Brignole, *Modeling and optimization of supercritical fluid processes*, *J. Supercrit. Fluids*, **47** (2009), 611–618.
5. R. W. Fox and A. T. McDonald, *Introduction to fluid mechanics*, 5th ed., John Wiley & Sons, New York, 1999.
6. D. Halliday, R. Resnick, J. Walker, *Fundamentals of physics*, 6th ed., John Wiley & Sons, New York, 2001.
7. A. R. Hambley, *Electrical engineering: Principles and applications*, 5th ed., Pearson Prentice Hall, Upper Saddle River, 2011.
8. J. P. Holman, *Experimental methods for engineers*, 6th ed., McGraw-Hill, New York, 1994.
9. E. Kreyszig, *Advanced engineering mathematics*, 7th ed., Wiley, New York, 1993.
10. F. Marra, *Editorial: Virtualization of processes in food engineering*, *J. Food Eng.* **176** (2016), 1.

11. OECD—Organization for Economic Cooperation and Development (2015). *PISA 2015 results—Results by country*. <http://www.compareyourcountry.org/pisa/country/BRA?lg=en> (Accessed Dec 6, 2017).
12. J. A. Rabi et al., *Blast-cooling of beef-in-sauce catering meals: Numerical results based on a dynamic zero-order model*, *Int. J. Food Studies*, **3** (2014), 213–227.
13. J. A. Rabi et al., *Catering meals under blast-cooling: Comparison between zero and first-order modeling with respect to spatial dependence of temperature*, *Int. Rev. Mech. Eng.* **6** (2012), 218–227.
14. R. H. Rosa et al., *Lattice Boltzmann simulation of cafestol and kahweol extraction from green coffee beans in high-pressure system*, *J. Food Eng.* **176** (2016), 88–96.
15. A. Rugarcia et al., *The future of engineering education I. A vision for a new century*, *Chem. Eng. Educ.* **34** (2000), 16–25.
16. I. S. Saguy, *Challenges and opportunities in food engineering: Modeling, virtualization, open innovation and social responsibility*, *J. Food Eng.* **176** (2016), 2–8.
17. J. F. Steffe, *Rheological methods in food process engineering*, 2nd ed., Freeman Press, 1996.
18. D.-W. Sun, *Computational fluid dynamics in food processing*. CRC Press, Boca Raton, FL, 2007.
19. G. J. Van Wylen, R. E. Sonntag, and C. Borgnakke, *Fundamentals of classical thermodynamics*, 4th ed., Wiley, New York, 1994.
20. L. Wang, D.-W. Sun, *Recent developments in numerical modelling of heating and cooling processes in the food industry—a review*, *Trends Food Sci. Technol.* **14** (2003), 408–423.
21. J. Welte-Chanes, J. F. Vélez-Ruiz, G. V. Barbosa-Cánovas, *Transport phenomena in food processing*, CRC Press, Boca Raton, FL 2002.



J. A. RABI holds BSc (Hons.) in Applied Physics & Instrumentation from the University of Sao Paulo (Brazil, 1995) and both MSc and PhD in Mechanical Engineering, respectively, from the Technological Institute of Aeronautics (Brazil, 1998) and the University of Campinas (Brazil,

2002). He accomplished a postdoctoral fellowship in transport phenomena in porous media at the University of Calgary (Canada, 2003–2004) as well as a research fellowship in modeling thermal food processing at CEMAGREF (currently IRSTEA, France, 2009–2010). He is presently an associate professor at the Department of Biosystems Engineering, University of Sao Paulo—

Pirassununga campus. His research interests include computational modeling of bioprocesses and biosystems via lattice Boltzmann method (LBM). His current teaching duties entail transport phenomena, numerical methods, and applied computational methods at undergraduate level, in addition to mathematical methods for differential equations and pedagogical preparation at postgraduate level.



F. L. CANEPELE holds a degree in Electrical Engineering from the University Center of FEI - Faculty of Industrial Engineering (1999), degree in Agronomy—Energy in Agriculture by the Faculty of Agronomic Sciences—UNESP/FCA (2007), for Bachelor, Technologists and Mathematics by the University Center Nove de Julho—UNINOVE (2008), PhD in Agronomy—Energy in Agriculture by the Faculty of Agronomic Sciences—UNESP/FCA (2011) and postdoctoral degree in Agribusiness and Development, Faculty of Sciences and Engineering—UNESP/FCE (2018). He has an experience in the area of electrical engineering and works in energy efficiency, alternative and renewable energy sources, energy matrix studies, energy generation and environment, industrial automation, industrial electrical installations and fuzzy logic. He is currently a PhD professor at the University of São Paulo (USP/FZEA), a Pirassununga-SP-Brazil campus and a post-graduate advisor (master's and doctorate) in Agronomy—Energy in Agriculture and Agronomy—Irrigation and Drainage programs at Universidade Estadual Paulista—UNESP/FCA, Botucatu-SP-Brazil campus.

How to cite this article: Rabi JA, Caneppele FL. Numerical methods to biosystems and food engineering students: Hands-on practices and cross-disciplinary integration. *Comput Appl Eng Educ.* 2018;1–14. <https://doi.org/10.1002/cae.21933>