1

Introduction to Modeling

Modeling in engineering and science transcends the glitz of fashion runways and the meticulous craftsmanship of hobbyists. It embodies a rigorous approach to understanding, analyzing, and optimizing systems, processes, and operations crucial to society's advancement.

Mathematical modeling, a cornerstone of applied mathematics, is more than a mere problem-solving tool – it is a language deeply rooted in mathematical physics. Engineers and scientists leverage mathematical models to conceptualize system behaviors, rigorously testing these hypotheses through meticulous validation and verification.

Engineering modeling entails:

- Crafting mathematical representations of physical reality, expressed through equations and algorithms.
- Employing simulations of these mathematical models as proxies for real-world experimentation.
- Elevating the value of models through rigorous validation processes.

While engineering modeling shares some traits with fashion and hobby modeling, its essence lies in its ability to distill complex phenomena into actionable insights. Like fashion modeling:

- Models serve as idealized depictions of reality, facilitating understanding and analysis.
- They establish relationships between different system states and parameters.

Resonating with hobby modeling:

- Careful construction is paramount to ensure fidelity to the real-world system.
- Models amalgamate empirical laws and constitutive relations to capture system dynamics.
- The distinction between system states and parameters is pivotal in model formulation.

Historically, engineering relied heavily on empirical methods, with designs forged through trial and error experimentation. However, the advent of mathematical modeling, coupled with modern computing tools, has revolutionized this paradigm. Mathematical models serve as powerful guides for equipment and process design, streamlining experimentation, and mitigating risks associated with empirical approaches.

Equation-based models, derived from fundamental principles like mass and energy conservation, thermodynamics, and kinetics, empower chemical engineers to address a myriad of questions, such as:

Chemical Engineering Analysis and Optimization Using MATLAB®, First Edition.
Weiguo Xie, Sam Toan, and Richard Davis.

Companion website: www.wiley.com/go/xie/chemicalengineering

- Will our concepts translate into practical solutions?
- What measurements are essential for validation?
- Which factors significantly influence system behavior?
- Is the proposed solution economically viable and safe?
- Can we effectively control and optimize the system?
- What scale can we feasibly implement?

Engineers and scientists navigate complex technical challenges with precision and confidence by strategically integrating mathematical modeling, propelling innovation, and societal progress. In navigating this terrain, we must balance simplicity and utility. Levenspiel's (2002) concept of "the US$10, US$100, and US$1000 models" serves as a reminder: while complex models may incur significant costs, they do not always offer proportional increases in understanding. Expensive experiments yield to relatively inexpensive computer simulations, offering a cost-effective alternative. Moreover, as depicted in Figure 1.1, the decreasing cost of computing contrasts with the rising costs of traditional experimentation over time.

Models vary in complexity based on their intended purposes. Simple models can often provide sufficient insight to guide decisions on whether to proceed with further development, pause for further investigation, or alter engineering directions. However, the demand for increased model detail grows as the need for accuracy and precision intensifies in design and decision-making processes.

The cost associated with developing a mathematical model hinges on the level of uncertainty deemed acceptable in the model predictions. As depicted in Figure 1.2, generally, the higher the tolerance for uncertainty in the calculated results, the lower the overall cost of the model.

The escalating costs of achieving greater certainty in model predictions often stem from the endeavor to solve the model equations. In preliminary engineering models, there is a propensity toward higher uncertainty – lower precision – particularly when efforts to mitigate uncertainty fail to justify the exponentially increasing modeling expenses.

Contrastingly, scientific models tend to gravitate toward minimizing uncertainty – higher precision – aiming for a comprehensive comprehension of natural phenomena. To navigate the uncertainty, engineers often incorporate design safety factors. For instance, a chemical engineer might design a distillation column by calculating the minimum number of ideal stages, then doubling this figure and adjusting the reflux to attain the desired separation (Walas 1990).

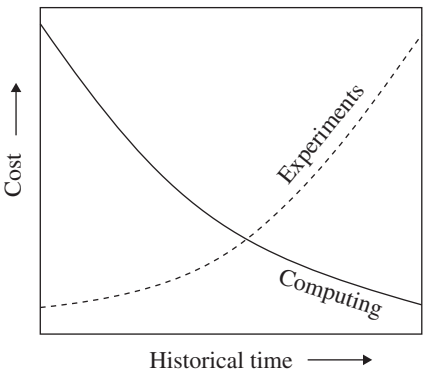


Figure 1.1 Computing costs are decreasing as the cost of experimentation increases over time.

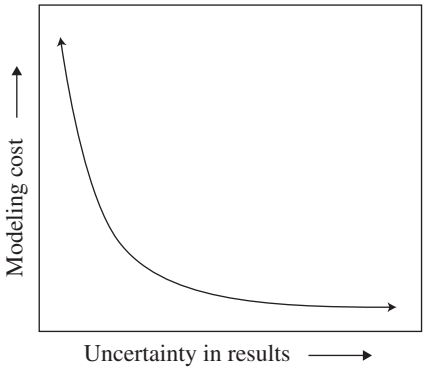


Figure 1.2 Cost of mathematical modeling as a function of the degree of uncertainty in the model predictions.

1.1 Numerical Methods

Numerical methods are computational techniques employed to approximate solutions to mathematical models that may prove inconvenient, challenging, or even impossible to solve through standard analytical methods. Analytical solutions yield symbolic representations, often involving explicit rearrangements of equations to isolate variables. However, when analytical solutions are unattainable, numerical methods offer practical approximations.

Consider calculating the molar volume of a gas using models that correlate molar volume with temperature and pressure. The ideal gas law and the Redlich–Kwong equation of state for nonideal behavior are two potential candidates. We can rearrange the ideal gas law to solve for the molar volume:

$$V = \frac{R_g T}{P} \tag{1.1}$$

where R_g is the ideal gas constant, T is the temperature, and P is the pressure. However, we cannot rearrange the following nonideal Redlich–Kwong model *explicitly* for either molar volume *or* temperature[1]:

$$P = \frac{R_g T}{V - b} - \frac{a}{V(V + b)\sqrt{T}} \tag{1.2}$$

The parameters a and b are functions of the critical temperature and gas pressure. There is no way to rearrange Eq. (1.2) with the molar volume only appearing on one side of the equation. Instead, we determine the Redlich–Kwong equation's molar volume from numerical approximation methods for specific cases of T and P.

Many numerical methods rely on iterative calculations, often facilitated by computers. Compared to analytical solutions, numerical approaches can be easier to implement, saving time and effort that can be directed toward addressing other challenges.

In chemical engineering, typical problems involve large systems of linear equations arising from applying fundamental principles like the conservation of momentum, mass, and energy. Additionally, nonlinear equations abound due to the inherent complexities of physical and chemical properties, transport phenomena, thermodynamics, and chemical reactions.

Some common examples of nonlinear functions encountered in chemical engineering include the temperature-dependent Antoine equation for vapor pressure, the modified

1 Explicit is defined as an expression containing only independent variables.

Arrhenius function for reaction rate constants, and the modified Henri function for enzyme reaction kinetics with substrate inhibition.

$$\text{Antoine} \quad \ln P_v(T) = A - \frac{B}{C+T} \tag{1.3}$$

$$\text{Modified Arrhenius} \quad k(T) = AT^2 \exp\left(-\frac{E_a}{R_g T}\right) \tag{1.4}$$

$$\text{Modified Henri} \quad v(S) = \frac{v_m S}{K_m + aS + bS^2} \tag{1.5}$$

While linear equations possess analytical solutions, determining the solution of a system larger than three or four equations can often prove tedious. Analytical solutions for such large systems entail extensive algebraic manipulations and symbolic bookkeeping. Moreover, many nonlinear equations lack analytical solutions altogether. Hence, methods for obtaining practical approximations of nonlinear functions and large systems of linear equations are essential.

To underscore the necessity of numerical methods, let us consider a simple model from chemical reaction engineering.

The following reaction mechanism is elementary, irreversible, and first order in species A:

$$A \rightarrow B \tag{1.6}$$

Consider the steady-state mole balance for product species B around a perfectly stirred chemical reactor, as illustrated in Figure 1.3, with feed and effluent concentrations of C_{A0} and unreacted C_A and product C_B, respectively.

We start with species conservation to mathematically model this reactor. With no B in the feed, the concentration of B leaving the reactor equals the product of the residence time, τ, with the rate of generation of B per unit volume in the reactor, r_B:

$$C_B = \tau\, r_B \tag{1.7}$$

For elementary, first-order reactions, the rate of production of B is a linear function of concentration:

$$r_B = k(C_{A0} - C_B) \tag{1.8}$$

where $(C_{A0} - C_B)$ is equivalent to the concentration of unreacted A in the reactor in terms of the initial concentration of reactant A, and k is the first-order reaction rate constant. We can rearrange Eqs. (1.7) and (1.8) explicitly for the concentration of product C_B exiting the reactor:

$$C_B = \frac{\tau k C_{A0}}{1 + \tau k} \tag{1.9}$$

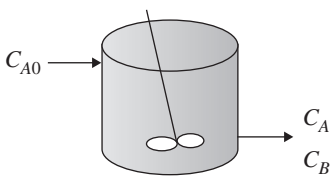


Figure 1.3 Well-mixed, steady-state reactor to produce B from A.

However, we typically describe the generation rate by a nonlinear concentration function for higher-order elementary or nonelementary reactions. Thus, for example, a catalyzed reaction may behave according to some form of Langmuir kinetics:

$$r_B = \frac{k(C_{A0} - C_B)}{\left[1 + K_A(C_{A0} - C_B) + \frac{K_B}{C_B}\right]^2} \tag{1.10}$$

where K_A and K_B are reaction equilibrium constants for reversible steps in the reaction mechanism. Substituting the rate law from Eq. (1.10) into the mole balance of Eq. (1.7) yields a nonlinear equation implicit[2] in C_B:

$$C_B = \frac{\tau k(C_{A0} - C_B)}{\left[1 + K_A(C_{A0} - C_B) + \frac{K_B}{C_B}\right]^2} \tag{1.11}$$

We cannot rearrange Eq. (1.11) explicitly for C_B. Instead, we calculate C_B iteratively by guessing the solution in a "trial and error" approach[3] until our C_B calculations converge on the solution.

The previous illustration demonstrates the need for numerical methods to solve nonlinear equations. Other problems encountered in chemical engineering may require solutions to nonlinear integrals,

$$I(x) = \int f(x)dx \tag{1.12}$$

or coupled systems of nonlinear, nonhomogeneous, "initial-value" first-order differential equations, with f and g representing nonlinear functions of some combination of the independent and dependent variables:

$$\frac{dy}{dx} = f(w, x, y) \text{ and } \frac{dw}{dx} = g(w, x, y) \tag{1.13}$$

Our models may involve nonhomogeneous, second-order "boundary-value"-type differential equations:

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \lambda r(C) = \frac{\rho}{\alpha}\frac{\partial C}{\partial t} \tag{1.14}$$

C is the concentration, $r(C)$ may be a nonlinear rate law, and α, ρ, and λ are model parameters that may be the functions of x, y, or t.

1.1.1 Linear vs. Nonlinear Equations

We can easily distinguish between linear and nonlinear functions using visual graphical analysis (Figure 1.4):

2 Implicit is defined as an expression in which the dependent variable and one or more independent variables are not separated on opposite sides of an equation. http://www.merriam-webster.com/dictionary/implicit.

3 Trial and error methods iterate on two steps: (1) Trial = guess the solution and evaluate the result; (2) Error = use an estimate of the error from the trial to upgrade the guess for a better solution.

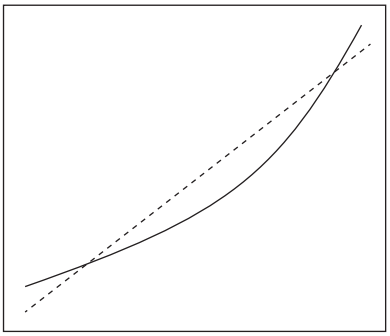


Figure 1.4 Graphical analysis to detect linear vs. nonlinear functions.

Without a graphical analysis, we recognize nonlinear equations from their terms involving exponents, transcendental functions, or combinations of dependent variables. To illustrate nonlinearity, consider the following second-order, ordinary differential equation:

$$\frac{d^2y}{dx^2} + \frac{x}{y}\frac{dy}{dx} + x^2y^2 = 0 \tag{1.15}$$

The first term in Eq. (1.15) is linear in the dependent variable, y. However, the second and third terms are nonlinear in y because they involve the product of two functions of y. Consequently, Eq. (1.15) is a nonlinear ordinary differential equation and may not have an analytical solution.

Of course, we may cite the well-known common exception – the second-order polynomial or quadratic equation:

$$ax^2 + bx + c = 0 \tag{1.16}$$

with two roots calculated by the standard quadratic formula[4]:

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \text{ and } x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{1.17}$$

Higher-order polynomials of order $m > 2$ have m roots.

$$\alpha x^m + \beta x^{m-1} + \cdots ax^2 + bx + c = 0 \tag{1.18}$$

In principle, an analytical solution for the m number of roots of a polynomial is possible. However, in practice, analytical solutions of polynomials with m greater than three are not trivial. When analytical solutions are difficult or impossible to achieve, one must resort to numerical approximations as a matter of practicality.

1.1.2 Standard Features of Numerical Methods

Most numerical methods share several standard features:

- They rely on local linear approximations of nonlinear functions to simplify calculations.
- Error approximation is utilized to refine the solution.
- They employ iteration and successive approximation to improve solutions.
- They are susceptible to computational round-off errors, which may lead to instability in the solution method.

4 See Section 1.1.3 to learn about computing problems with the standard quadratic formula.

Many numerical methods necessitate robust linear approximations to nonlinear equations at a local level. As discussed earlier, nonlinear equations can take various forms – algebraic, differential, or integral. Our solution approximation techniques are based on locally linearizing the nonlinear equations through truncated Taylor series expansions for each problem type. Novice practitioners of numerical methods soon discover that a surprisingly small set of numerical techniques, based on the Taylor series, can solve a wide range of nonlinear problems.

Linear approximations for nonlinear functions enable approximate analytical solutions to nonlinear problems. We gain access to all the analytical tools required to solve linear equations by employing good linear approximations of nonlinear functions. We utilize linear approximation error estimates iteratively to refine solutions until the error falls below our specified threshold. Each refinement step is considered one iteration. While a rudimentary approach to finding an approximate solution involves guessing and checking, we aim to systematically improve the approximation through numerical methods, removing guesswork from the process.

Once we grasp these fundamental numerical computation features, we can delve into the specifics of individual methods.

Problems involving solutions across a range of time or space necessitate a sequence of calculations. We successively approximate the solution at different points in time or space based on the solution at previous points. For instance, we might approximate solutions across spatial and temporal dimensions by discretizing the problem's time and space domains. Whether aiming for a precise solution at a single point or traversing through a function to obtain a range of solutions, iteration, and successive approximation principles remain relevant.

A common point of confusion arises when implementing iteration in a computer program, where we replace a variable's value with a mathematical operation involving that variable. For instance, from a mathematical standpoint, we might stipulate that the following equation is only valid when $\phi = 0$:

$$x = x + \phi \tag{1.19}$$

When we state that ϕ must equal zero, we imply that the direction of equality goes both ways (e.g. $x + \phi = x$). However, when setting up a sequence of iterative calculations using a computer, we mean that the current value of x is *replaced* by $x + \phi$, such that the flow of information is in one direction: $x \leftarrow x + \phi$. For example, suppose we need to calculate the sum of a series $\phi_1, \phi_2 \ldots \phi_n$. We start by initializing $x = 0$ and $i = 0$, then, iterate on Eq. (1.20) until $i = n$:

$$i \leftarrow i + 1 \quad x_i \leftarrow x_{i-1} + \phi_i \tag{1.20}$$

where the subscript i is an index variable for ϕ and the sum of the values in the series is the final value of x when $i = n$. We make extensive use of this type of operation in numerical methods. As an illustration, if $x_0 = 0$ and $\phi = (1, 2, 3, 4, 5\ldots)$, then $x_1 = 0 + 1 = 1, x_2 = 1 + 2 = 3$, $x_3 = 3 + 3 = 6, x_4 = 6 + 4 = 10, x_5 = 10 + 5 = 15 \ldots$.

1.1.3 Significant Figures and Computer Round-Off Error

In scientific and engineering practice, numerical results are typically reported with only the significant figures accounted for. Significance is attributed to digits deemed reliable within a number, with convention dictating that only the rightmost digit retains any significant

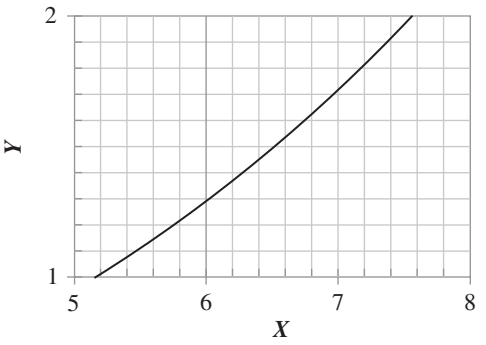


Figure 1.5 Reading values from a graph.

uncertainty. Computers store numbers in either single-precision format, allowing for seven significant digits, or double-precision format, accommodating 15 significant digits. When a single-precision number extends to eight significant digits, the computer rounds it to seven significant digits.

Consider reading a value from the Y-axis of the following graph, where the smallest division is 1/10 or 0.1. If a value at $X = 6.4$ correlates to $Y = 1.46$, only three significant figures are appropriate – the last digit, "6," being uncertain (Figure 1.5).

Consider the addition of two numbers in *the MATLAB* programming environment using single vs. double-precision values:

- *Single-precision*: 123456789 + 987654321 = 1111111168 (round to seven significant figures 1111111000, **incorrect**)
- *Double-precision*: 123456789 + 987654321 = 1111111110 (**correct**)

The single-precision result is accurate to just seven significant figures, whereas the double-precision result gives the correct answer. This example teaches us not to trust single-precision results involving more than seven significant figures. How far can we go with double precision?

$$123456789123456789 + 987654321987654321 = 1111111111111110000$$

The result has 15 significant figures! What do we conclude? Use double precision.

By default, MATLAB uses double precision. However, simply using double precision does not eliminate all round-off errors. Avoid adding or subtracting numbers with extreme differences in their orders of magnitude. For example, in double precision, $123456789 \pm 0.00000001 = 123456789$, with no change! Generally, we should sum up the small positive numbers first and sum the small negative numbers before combining them with large numbers.

Example 1.1 ***Computer Round-Off Errors in Derivative Approximations***

We use the following definition of the calculus derivative of a function to demonstrate computer round-off errors and the limits of numerical approximation:

$$\frac{df}{dx} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{1.21}$$

Our test function with its analytical result for the derivative evaluated at $\pi/4$:

$$f(x) = \sin(x) \qquad \frac{df}{dx} = \cos(x) \qquad \cos\left(\frac{\pi}{4}\right) = \frac{1}{\sqrt{2}} = 0.70711 \tag{1.22}$$

Substitute the test function into Eq. (1.21) for a first-order approximation:

$$\frac{df}{dx} \cong \frac{f\left(\frac{\pi}{4}+\Delta x\right)-f\left(\frac{\pi}{4}\right)}{\Delta x} = \frac{\sin\left(\frac{\pi}{4}+\Delta x\right)-\sin\left(\frac{\pi}{4}\right)}{\Delta x} \tag{1.23}$$

Solution:

```
%MATLAB code for Example 1.1
% Define the function to be plotted
%f = @(dx) (sin(pi/4 + dx) - sin(pi/4)) ./ dx;
syms dx
f = (sin(pi/4+dx)-sin(pi/4))/dx;
g = abs((cos(pi/4)-(sin(pi/4+dx)-sin(pi/4))/dx)/cos(pi/4));
% Plot f(dx)
figure
fplot(f, [1e-16, 1], 'LineWidth', 1)
set(gca, 'XScale', 'log', 'YScale', 'log') % Set both axes to
% logarithmic scale
xlabel('dx')
ylabel('df/dx')
grid on
% Plot g(dx)
figure
fplot(g, [1e-16, 1], 'LineWidth', 1)
set(gca, 'XScale', 'log', 'YScale', 'log') % Set both axes to
% logarithmic scale
xlabel('dx')
ylabel('|% Error|')
grid on
```

The first *semi*-log plot in Figure 1.6 shows the numerical derivative approximation of the test function using Eq. (1.23) approaching the true solution near 0.71 as Δx decreases before the error becomes noticeable at small Δx. Note that the plots show decreasing values for Δx moving from left to right for convenience.

The second log–log plot shows the absolute percent error in the numerical approximation of the test function's derivative function as $\Delta x \rightarrow 0$. The error in the derivative approximation decreases exponentially as expected until $\Delta x \approx 10^{-8}$, then starts to climb again to noticeable values because of round-off error. These results have implications for controlling computational round-off error when we use derivative approximations for several numerical methods to find roots of algebraic equations, optimization, and uncertainty analysis.

Note that the abscissa (Δx-axis) scale is reversed for values in decreasing magnitude.

Example 1.2 ***Avoid Round-Off Errors in the Quadratic Formula***

One of the two standard roots to a quadratic equation, defined by Eqs. (1.16) and (1.17), potentially loses precision due to round-off error through subtraction. To avoid a loss of precision, we keep the root involving addition (x^+) and find the second root (x^-) from the relationships:

$$x^+ \cdot x^- = \frac{c}{a} \text{ and } x^- = \frac{c}{a \cdot x^+} \tag{1.24}$$

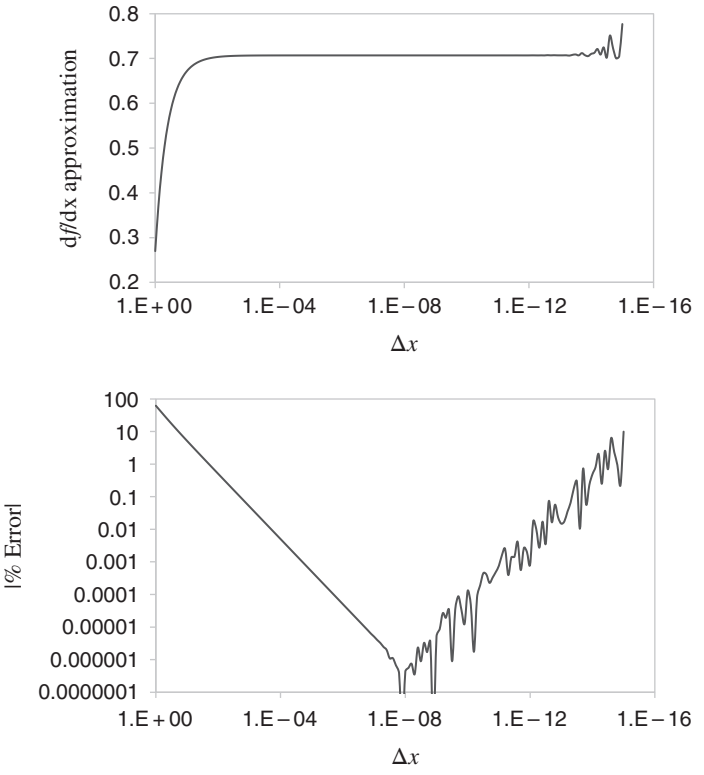


Figure 1.6 Semi-log plot of the numerical derivative approximation in Eq. (1.23) vs. Δx. The numerical approximation approaches the true solution before the round-off error becomes noticeable at $\Delta x = 10^{-13}$. Log–log plot displaying the percent error in the numerical derivative approximation of sin(x) evaluated at $\pi/4$. Round-off errors in the computer cause the approximation error to increase for small numbers.

Thompson's (1987) example of a quadratic root involves a buffer solution of 0.0010 M chloroacetic acid and 0.0100 M sodium chloroacetate. The chemical equilibrium relationship in terms of the hydrogen ion concentration x is

$$1.4 \times 10^{-3} = \frac{x(0.0100 + x)}{0.0010 - x}$$

which reduces to a quadratic function in x:

$$x^2 + 0.0114x - 1.4 \times 10^{-6} = 0$$

The standard roots using 10^{-4} precision are

$$x = \frac{-0.0114 \pm 0.0116}{2}$$

The first root involves a subtraction to give an incorrect answer $x^- = (0.0116 - 0.0114)/2 = 0.0001$. The second root involves an addition (of two terms with the same sign) to give a correct value of $x^+ = (-0.0114 - 0.0116)/2 = -0.0115$. Use x^- in Eq. (1.24) to get the other correct root:

$$x^- = \frac{-1.4 \times 10^{-6}}{-0.0115} = 1.22 \times 10^{-4}$$

Press et al. (1992) recommend the following alternative form of the quadratic rule that eliminates round-off error by avoiding subtraction between two potentially small terms:

$$q = -0.5\left[b + \text{sign}(b)\sqrt{b^2 - 4ac}\right] \tag{1.25}$$

where

$$x_1 = \frac{q}{a} \text{ and } x_2 = \frac{c}{q} \tag{1.26}$$

and

$$\text{sign}(b) = \begin{cases} -1 \text{ for } b < 0 \\ +1 \text{ for } b \geq 0 \end{cases} \tag{1.27}$$

To illustrate further, compare the calculation with the roots in the following equation:

$$f(x) = 0.321x^2 + 214x + 0.00123 = 0 \tag{1.28}$$

Solution:

```
%MATLAB code for Example 1.2
% Coefficients of the quadratic equation
a = 0.321; % Quadratic coefficient
b = 213;   % Linear coefficient
c = 1.23e-3; % Constant term
% Standard quadratic formula
x1_standard = (-b + sqrt(b^2 - 4*a*c)) / (2*a);
x2_standard = (-b - sqrt(b^2 - 4*a*c)) / (2*a);
% Alternative method for numerical stability
q = -0.5 * (b + sign(b) * sqrt(b^2 - 4*a*c));
x2_correct = q / a;
x1_correct = c / q;
% Function evaluations to check correctness
f1_standard = a*(x1_standard^2) + b*x1_standard + c; % Should be
% close to 0
f1_correct = a*(x1_correct^2) + b*x1_correct + c; % Should be
% close to 0
% Display results
fprintf('Standard method roots: x1 = %.16f, x2 = %.16f\n',...
 x1_standard, x2_standard);
fprintf('Alternative method roots: x1 = %.16f, x2 = %.16f\n',...
 x1_correct, x2_correct);
fprintf('Function values at standard roots: f(x1) = %.16e,...
 f(x2) = %.16e\n', f1_standard, a*(x2_standard^2) +...
 b*x2_standard + c);
fprintf('Function values at alternative roots: f(x1) = %.16e,...
 f(x2) = %.16e\n', f1_correct, a*(x2_correct^2) + b*x2_correct...
 + c);
```

Comparing the two solution methods for the root in Table 1.1 reveals the problem with the standard version of the quadratic formula. The bottom row shows the standard solution's

Table 1.1 Comparing the standard quadratic formula with an alternative to eliminate round-off errors using $a = 0.321$, $b = 213$, and $c = 1.23 \times 10^{-3}$.

Standard quadratic formula	Alternative quadratic formula
—	$q = -212.999998146338000000$
$x_1 = -0.00000577464792771470$	$x_1 = c/q = -0.00000577464793757862$
$x_2 = -663.551396094511000000$	$x_2 = q/a = -663.551396094511000000$
$f(x_1) = 0.00000000000210101511$	$f(x_1 = c/q) = 0.00000000000000000000$

inaccuracy when substituting the first root into Eq. (1.28). From this example, we learn to be careful with arithmetic involving the addition or subtraction of an extremely large number and an extremely small number.

1.2 Mathematical Model Building

Henri's equation is a classic example of empirical model building, which he discovered for describing the rate of enzyme-catalyzed reactions as a function of the reacting substrate concentration. A typical plot in Figure 1.7 of reaction rate, r, vs. substrate concentration, $[S]$ reveals the asymptotic nature observed in many catalysis reactive systems.

The curve plotted in Figure 1.7 represents a *graphical* model. Therefore, we may interpret the function's behavior directly from the plot. Henri observed that the reaction rate is proportional to the substrate concentration, $[S]$, for small substrate concentrations. For large $[S]$, the rate approaches an asymptotic maximum value, $r \to r_{max}$. He proposed the following mathematical equation, or model, to match the observations of the data:

$$r = \frac{a[S]}{b + [S]} \tag{1.29}$$

where a and b are the constants specific to the reaction mechanism. For a small substrate concentration, $[S] \ll b$, the rate expression approaches a linear function in $[S]$ with proportionality constant a/b. In the limit of large $[S] \gg b$, the reaction rate approaches a constant, maximum value, $a = r_{max}$:

$$r \cong \begin{cases} \dfrac{a}{b} & [S] \ll b \\ a & [S] \gg b \end{cases} \tag{1.30}$$

We may also determine the value of parameter b by recognizing that parameter a disappears from Eq. (1.29) when $r = a/2$:

$$\frac{a}{2} = \frac{a[S]}{b + [S]} \to \frac{1}{2} = \frac{[S]}{b + [S]} \to b = [S] \quad \text{at } r = \frac{a}{2} \tag{1.31}$$

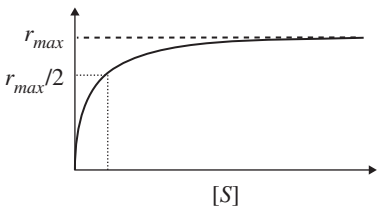


Figure 1.7 Henri plot of enzyme reaction rate, r, as a function of substrate concentration s.

Henri's model successfully fits enzyme reaction data for interpolation but does not explain *how* or *why* the reaction kinetics behave this way. Menten and Michaelis (1913) proposed a two-step reaction mechanism to explain how Henri's model works. Their model involves an initial, reversible step that forms a complex intermediate molecule of enzyme–substrate. Then, the enzyme–substrate complex reacts irreversibly to form the reaction product in the second step:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \tag{1.32}$$

$$ES \xrightarrow{k_2} E + P \tag{1.33}$$

E is the free enzyme, ES is the enzyme–substrate complex, and k_1, k_{-1}, and k_2 are the forward and reverse reaction rate constants for the two steps. If the second step is rate limiting, the reaction rate is proportional to the concentration of the complex $[ES]$:

$$r = k_2[ES] \tag{1.34}$$

while the first step reaches a pseudo-equilibrium state:

$$\frac{[ES]}{[S][E]} = \frac{k_1}{k_{-1}} \tag{1.35}$$

The conservation of enzyme catalyst in the reaction yields the following balance:

$$[E] = [E]_0 - [ES] \tag{1.36}$$

where $[E]_0$ is the initial enzyme concentration in the reactor. Combining Eqs. (1.34) through (1.36) to eliminate $[E]$ and $[ES]$ gives the well-known Michaelis–Menten expression for the reaction rate:

$$r = \frac{k_2[E]_0[S]}{\frac{k_{-1}}{k_1} + [S]} = \frac{r_{\max}[S]}{K_M + [S]} \tag{1.37}$$

where K_M is the Michaelis equilibrium constant. We should confirm that Eq. (1.37) has the same basic functionality of reaction rate with substrate concentration as the Henri equation (1.29) (i.e. what are a and b?) and comes with an explanatory mechanism in Eqs. (1.32) and (1.33). We encounter the alternative Briggs–Haldane mechanism in the next section.

Like Henri's equation, empirical models lack a foundation in the fundamental principles of momentum, energy, or mass conservation. Nevertheless, they can incorporate mathematical functionality that captures trends in the data. Polynomials and rational functions, such as Eq. (1.29), serve as convenient models for interpolating experimental or process data. However, they may struggle to elucidate the underlying mechanisms of the system.

We introduce various approaches to mathematical modeling, each based on underlying assumptions about the system. We commence with a review of the inviolable law of conservation, as our models are rendered ineffective if they fail to account for all mass and energy inputs. Subsequently, we explore pairs of opposing principles (Rasmuson et al. 2014), with underlying assumptions denoted in brackets: mechanistic (based on physics) vs. empirical, steady (time-independent) vs. unsteady-state, lumped (uniform or well-mixed) vs. distributed systems, and stochastic (incorporating uncertainty) vs. deterministic modeling.

1.2.1 Laws of Conservation

Mathematical models bridge the descriptive language of conservation equations and the symbolic language of mathematics. While not mandatory, expressing governing equations in symbolic mathematical form proves significantly more convenient and efficient. To illustrate, attempt to describe the derivative of the Henri equation's reaction rate (1.29) concerning substrate concentration without employing mathematical terms. It is undoubtedly challenging!

Constructing a mathematical model involves establishing conservation laws for momentum, mass, and energy. Initially, boundaries are defined around the system's control volume. Subsequently, rate balances are applied to the system and across the defined boundaries for each conservation law:

$$(\text{Rate of } \xi \text{ in}) - (\text{Rate of } \xi \text{ out}) + (\text{Net rate of } \xi \text{ generation}) = (\text{Rate of } \xi \text{ accumulation}) \tag{1.38}$$

where ξ may represent mass, energy, or momentum. Conservation laws often yield nonlinear algebraic, integral, or differential equations. Operations on these equations are straightforward, using standard analytical and numerical techniques. Mathematical models replace each term in Eq. (1.38) with its equivalent symbolic mathematical representation of measurable, quantifiable parameters. As an illustration, the volumetric flow rate product with density, $Q \times \rho$, may replace the mass (rate in). Check that the dimensions of each term in Eq. (1.38) are mutually consistent. The dimensions are usually in velocity units, mass (or moles), or energy per time. Thus, if we describe the (rate in) term by the flux (rate per unit area), multiply by the normal cross-sectional area to convert the flux into a rate. Table 1.2 contains typical constitutive rates for transport phenomena.

By definition, a batch reactor has neither inlet nor outlet streams, which leaves only the generation and accumulation terms in the mass balance:

$$0 - 0 + (\text{Product generation rate}) = (\text{Product accumulation rate}) \tag{1.39}$$

Steady-state systems have no accumulation terms. A steady-state batch process has reached an equilibrium where the net generation rate is zero. This leads to one form of the chemical equilibrium constant as a ratio of reaction rate constants for reversible reactions. For example, consider the first-order reversible reaction $A \rightleftharpoons B$. At steady-state for this reaction, Eq. (1.39)

Table 1.2 Constitutive expressions of transport rates.

	Rate	Diffusive	Advection	Convection
Newton's law of viscosity	Momentum transfer	$\tau = -\mu \dfrac{\partial u}{\partial x}$	$V\rho u$	—
Fourier's law of heat conduction	Heat transfer	$q = -A\kappa \dfrac{\partial T}{\partial y}$	$q = \dot{m}\left(H^0 + \displaystyle\int_{T^0}^{T} c_p dT\right)$	$q = hA\Delta T$
Fick's law of molecular diffusion	Mass transfer	$N = -AD \dfrac{\partial C}{\partial y}$	$\dot{m} = A\rho u$	$\dot{m} = kA\Delta C$

A = area, c_p = specific heat, h = heat transfer coefficient, H = enthalpy, k = mass transfer coefficient, m = mass flow rate, T = temperature, u = velocity, V = volume, x = length dimension, κ = thermal conductivity, ρ = density, and μ = dynamic viscosity.

reduces to the following algebraic expression:

$$-r = k_f C_A - k_r C_B = 0 \tag{1.40}$$

where k_f and k_r represent the forward and reverse reaction rate constants, respectively. Rearrange Eq. (1.40) for the reaction equilibrium constant:

$$K = \frac{C_B}{C_A} = \frac{k_f}{k_r} \tag{1.41}$$

Equation (1.35) is another example of an equilibrium constant defined in terms of rate constants.

The generation term in Eq. (1.39) requires a reaction rate law describing a molecular species' net generation (generation–consumption). Chemical reactions conserve mass, but not necessarily molecules. The Michaelis–Menten reaction rate law derived from Eq. (1.37) is an example of chemical species generation. Consider an alternative model of enzyme reactions proposed by Briggs and Haldane to develop further the steady-state, equilibrium, accumulation, and generation concepts. Instead of assuming an equilibrium step, as in the case of Michaelis and Menten, Briggs and Haldane proposed conservation equations for each species, including the enzyme–substrate complex:

Rate of generation = Rate of accumulation

$$k_1[S]\cdot[E] - k_{-1}[ES] - k_2[ES] = \frac{d[ES]}{dt} \text{ and } k_2[ES] = \frac{d[P]}{dt} \tag{1.42}$$

Briggs and Haldane assumed that the enzyme–substrate complex concentration conservation quickly reaches a pseudo-steady-state condition, $d[ES]/dt \cong 0$. Thus, their model combines Eqs. (1.36) and (1.42) to eliminate $[E]$ and $[ES]$ from the enzyme-catalyzed reaction rate expression:

$$r = \frac{k_2[E]_0[S]}{\frac{k_{-1} + k_2}{k_1} + [S]} \tag{1.43}$$

Equation (1.43) maintains the same functionality as the Henri and Michaelis–Menten equations but contains an alternative form of the Michaelis constant.

Another approach solves the conservation equations without assumptions of pseudo-equilibrium or steady state for the reaction steps. However, this approach does not yield a simple algebraic equation. Instead, we must solve a simultaneous system of differential and algebraic equations:

$$\frac{d[ES]}{dt} = k_1[S]\cdot[E] - k_{-1}[ES] - k_2[ES] \text{ and } \frac{d[P]}{dt} = k_2[ES] \tag{1.44}$$

$$[E]_0 = [E] + [ES] \text{ and } [S]_0 = [S] + [P] \tag{1.45}$$

The initial concentrations of the enzyme–substrate complex $[ES]$ and product $[P]$ are assumed to be zero.

Indeed, the heat-generation term in a thermal energy conservation equation becomes essential when there are endo- or exothermic chemical reactions or electrical heating within the system boundaries. It is crucial to note that the heat-generation term can be either negative or positive, depending on the nature of the processes involved.

For instance, a chemical reaction that consumes molecular species typically results in a negative heat-generation term. Conversely, an exothermic reaction, conventionally characterized by a negative heat of reaction, leads to a positive heat-generation term. Conversely, an endothermic reaction, where the reaction absorbs heat from its surroundings, results in a negative heat-generation term. This occurs when the rate of consumption of the reactant (r_A) is negative.

Understanding the sign convention for heat generation is vital for accurately modeling and analyzing thermal systems involving chemical reactions or other sources of heat generation. It allows engineers to effectively predict and control heat transfer processes, ensuring optimal performance and safety in various industrial applications.

$$\text{Chemical reaction energy: } q = V_s r_A \Delta H_{rxn,A} \tag{1.46}$$

where V_s is the system volume, r_A is the reaction rate of species A, and ΔH is the heat of the reaction.

Thermal energy generation may involve heat from an electrical resistor:

$$\text{Electrical energy: } q = \frac{I^2 R}{V} \tag{1.47}$$

where I, R, and V are the electrical current, resistance, and voltage, respectively.

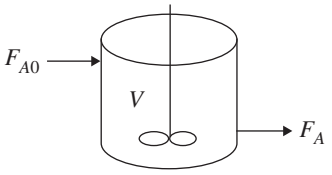
1.2.2 Lumped vs. Distributed Systems

Indeed, engineers rely on mathematical models for design, simulation, and gaining insights into processes. These models often necessitate simplifying assumptions to tackle complexities arising from the system's geometry, flow patterns, or other properties. While these assumptions and approximations can introduce errors and limit the model's applicability, they are often indispensable for obtaining tractable solutions to mathematical equations.

Let us explore six examples demonstrating how to relax simple modeling assumptions to enhance descriptive capabilities:

1. *Incorporating nonlinearities*: Initially, assuming linear relationships between variables may simplify the model, but relaxing this assumption to account for nonlinearities can provide a more accurate representation of real-world behavior.
2. *Considering three-dimensional geometry*: Many models assume two-dimensional geometry for simplicity. However, relaxing this assumption to include three-dimensional effects can offer a more comprehensive understanding of the system's behavior.
3. *Accounting for turbulence*: Simplified models often assume laminar flow, neglecting turbulence effects. Relaxing this assumption to incorporate turbulent flow phenomena can improve the accuracy of predictions, particularly in fluid dynamics applications.
4. *Introducing time dependency*: Some models assume steady-state conditions, disregarding temporal variations. Engineers can more effectively capture dynamic behavior and transient effects by relaxing this assumption and incorporating time dependency.
5. *Incorporating spatial variability*: Simplified models often assume spatial uniformity, overlooking variations across different regions of the system. Relaxing this assumption to account for spatial variability allows a more nuanced understanding of localized phenomena.
6. *Considering non-Newtonian behavior*: Many models assume Newtonian fluid behavior due to its simplicity. However, relaxing this assumption to accommodate non-Newtonian

Figure 1.8 Well-mixed flow reactor.



behavior can be essential for accurately representing the rheological properties of complex fluids.

By relaxing these simplifying assumptions, engineers can develop models that better capture the intricacies of real-world systems, leading to more reliable predictions and insights for design, simulation, and troubleshooting processes.

We begin with a well-mixed reactor's steady-state isothermal operation, pictured in Figure 1.8. F_{A0} and F_A are species A's inlet and effluent molar flow rates, respectively, and V is the reactor volume.

The conservation equation for a molecular species in a steady-state reactor is simply the difference between the molar flow rates in and out of the reactor vessel and generation due to reaction:

$$F_{A0} - F_A + Vr_A = 0 \tag{1.48}$$

where r_A is the reaction rate of formation of A per unit volume of the reactor. Define conversion in a flow reactor in terms of molar flow rates in and out of the reacting species:

$$X_A = \frac{F_{A0} - F_A}{F_{A0}} \tag{1.49}$$

We solve Eq. (1.49) for F_A, then substitute it into the conservation Eq. (1.48) and rearrange it for alternative forms of species conservation in terms of conversion of reactant A:

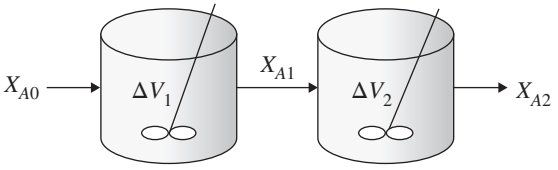
$$\frac{V}{F_{A0}} = \frac{X_A}{-r_A} \text{ or } \frac{X_A}{V} = \frac{-r_A}{F_{A0}} \tag{1.50}$$

Numerical methods for finding roots to nonlinear algebraic equations may be required to solve Eq. (1.50) for X_A when the reaction rate law for r_A is nonlinear.

The straightforward design Eq. (1.53) for a well-mixed reactor overlooks the influence of mixing on the conversion of a chemical reaction. This lumped model assumes that the reactor's composition is uniform throughout, mirroring the composition of the exit stream. We require a model that considers spatial or temporal variations to represent mixing effects accurately.

For example, we can devise a model that incorporates mixing effects by simulating a reactor behaving as if it consists of two mixing zones, akin to two stirred reactors-in-series, as illustrated in the flow diagram depicted in Figure 1.9. This approach allows us to account for the gradual mixing of reactants and products within the reactor, providing a more realistic representation of the chemical reaction's conversion. We can better understand and optimize

Figure 1.9 Two well-mixed tanks in series represent the reactor.



reactor performance in real-world applications by incorporating such spatial or temporal variations into our model.

The steady-state material balances for species A around each tank are:

$$\frac{X_{A1}}{\Delta V_1} = \frac{-r_{A1}}{F_{A0}} \text{ and } \frac{X_{A2} - X_{A1}}{\Delta V_2} = \frac{-r_{A2}}{F_{A0}} \tag{1.51}$$

where the sum of volumes for the mixing zones equals the total volume of the reactor:

$$V = \Delta V_1 + \Delta V_2 \tag{1.52}$$

Optimization tools may help determine the best split between the two volumes that match model predictions with experimental results for conversion.

Eventually, as the number of tanks-in-series increases to infinity, the volume of each tank becomes infinitely small, and we replace the ratio of conversion to volume with a derivative:

$$\lim_{\Delta V \to 0} \frac{X_{A,V+\Delta V} - X_{A,V}}{\Delta V} = \frac{dX_A}{dV} \tag{1.53}$$

In this case, the model approaches the distributed design equation for a continuous plug flow reactor (PFR), pictured in Figure 1.10:

$$\frac{dX_A}{dV} = \frac{-r_A}{F_{A0}} \tag{1.54}$$

The solution to Eq. (1.54) may require numerical methods to approximate solutions to integrals or ordinary differential equations when the reaction rate law, r_A, is nonlinear.

By adding increasing degrees of complexity, we transformed the simple algebraic design equation for a well-mixed reactor into the differential *PFR* design equation. The *PFR* equation accounts for changes in the axial direction of flow but ignores mixing effects in the radial direction (hence, "plug flow"). One accounting method for deviations from plug flow without a three-dimensional model introduces a dispersion term into the *PFR* balance. Dispersion accounts for uneven mixing in the radial direction by assuming that the gradient in the concentration along the flow axis drives the mixture away from a flat concentration profile in the radial direction, akin to Fick's law. The dispersion flux equation is

$$J_A = -S^2 E \frac{dC_A}{dV} \tag{1.55}$$

E is the dispersion coefficient, like a diffusion coefficient, and S is the cross-sectional area of the PFR. The molar flow in and out terms now takes the form.

$$\text{Molar flow rate in or out} = F_{A0} X_A + S J_A \tag{1.56}$$

Substitute the results from Eqs. (1.55) and (1.56) into the differential *PFR* design Eq. (1.54), divide each term by the differential volume, and take the limit as the differential volume goes to zero:

$$\lim_{\Delta V \to 0} \frac{\Delta \left[F_{A0} X_A + C_{A0} S^3 E \left(\frac{dX_A}{dV} \right) \right]}{\Delta V} = F_{A0} \frac{dX_A}{dV} + C_{A0} S^3 E \frac{d^2 X_A}{dV^2} = -r_A \tag{1.57}$$

The second-order differential Eq. (1.57) may require appropriate boundary conditions and a numerical method for approximating the solution to this boundary-value problem.

Figure 1.10 Plug flow reactor (PFR) with uniform conditions in the radial dimension.

Indeed, adding higher levels of rigor can transform the original algebraic model into a second-order differential equation, introducing increased complexity. Consequently, a simple reactor model utilizing the well-mixed design equation becomes more rigorous to accommodate imperfect mixing effects in a continuous flow reactor. However, if perfect mixing is a reasonable assumption, the simple algebraic well-mixed design equation yields the same result as more complex plug flow or dispersion models. This convergence occurs, for instance, as the dispersion coefficient approaches zero ($E \to 0$).

Generally, when constructing a mathematical model, making as many reasonable simplifying assumptions as possible is advisable. Only incorporate higher levels of rigor into the model as dictated by the problem's requirements. From a practical standpoint, there is no need to develop an expensive, highly rigorous mathematical model to describe a process if the solution to the model equations is intractable or if a more straightforward model provides satisfactory results (Figure 1.2). Often, simple models can offer sufficient information and insights to fulfill preliminary design and economic analysis needs.

The famous Bernoulli equation from fluid mechanics is another classic example of a lumped system that relates the fluid velocity to pressure in a pipe:

$$\frac{\Delta P}{\rho} + g\Delta z + \frac{\Delta u^2}{2} = 0 \tag{1.58}$$

where ΔP is the pressure difference between the fluid inlet and outlet of a system, g is the acceleration due to gravity, Δz is the vertical change in height between the fluid inlet and outlet points, u is the average fluid velocity, and ρ is the fluid density. When ignoring friction, the only information needed to determine the pipe flow is the conditions at the ends. The model requires neither the length nor the pipe geometry between the inlet and outlet, as displayed in Figure 1.11.

When we include a term in Eq. (1.58) to account for friction, we need to understand the complete geometry to calculate the friction term:

$$\frac{\Delta P}{\rho} + g\Delta z + \frac{\Delta u^2}{2} = F = \frac{2fLu^2}{d} \tag{1.59}$$

where f is the Fanning friction factor, and L and d are the tube's length and diameter, respectively. The nonlinear Colebrook equation gives the friction factor for flow in rough circular pipes as a function of the Reynolds number, $Re = \rho u D/\mu$, where ρ, u, D, and μ are the fluid density, velocity, pipe diameter, and viscosity, respectively. With this information, we can calculate the velocity or pressure at any position along the pipe's length using a numerical root-finding method.

Consider the mass and energy balances around a mixing tank shown in Figure 1.12. To simplify our analysis, we assume all the fluid mixing and heat transfer behaviors are lumped together. In this case, we cool a full tank of hot water with volume V and initial temperature T_0, with a stream of cold liquid with a constant volumetric flow rate Q and temperature T_{in}.

We derive an expression for calculating the water temperature as a function of time by assuming constant properties for water, a constant feed rate, and a well-mixed tank. Our model system is the water in the tank. Because the tank is full, the water exiting the tank must

Figure 1.11 Lumped equations ignore geometric considerations.

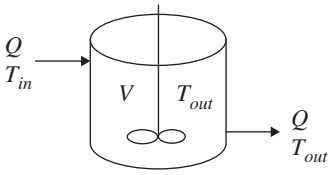


Figure 1.12 Mixed tank with volume V and constant inlet and outlet flow rate Q.

equal the entering rate. Thus, there is no mass accumulation or generation in the tank, i.e.:

$$\text{Rate of mass in} = \text{Rate of mass out} \tag{1.60}$$

For the case of constant density, the mass of water in the tank remains constant, such that:

$$\rho \frac{dV}{dt} = 0 \tag{1.61}$$

where ρ is the fluid density.

Next, we apply the energy conservation principle to the water in the tank. Because the system generates no energy within the water in the tank, the energy balance reduces to

$$(\text{Rate of energy in}) - (\text{Rate of energy out}) = (\text{Rate of energy accumulation}) \tag{1.62}$$

We replace the terms with their mathematical equivalents from thermodynamics:

$$Q\rho c_p(T_{in} - T) = V\rho c_p \frac{dT}{dt} \tag{1.63}$$

where c_p is the average specific heat of the water. We separate variables and integrate Eq. (1.63) assuming constant Q and V:

$$\frac{Q}{V}\int_0^t dt = \int_{T_0}^{T} \frac{dT}{(T_{in} - T)} \tag{1.64}$$

then solve Eq. (1.64) for the temperature in the tank at time t:

$$T = T_{in} - (T_{in} - T_0)\exp\left(-\frac{t}{\tau}\right) \tag{1.65}$$

where the residence time is $\tau = Q/V$. The function in Eq. (1.65) behaves asymptotically: $T \to T_{in}$ as $t \to \infty$.

Study heat convection from a hot aluminum sphere with diameter d suddenly plunged into an infinite volume of colder fluid (also known as quenching). Our model system is a solid sphere. If the diameter of the sphere is small, and the thermal conductivity of the material is large (true for aluminum) such that the temperature distribution in the solid is practically uniform, the energy balance gives:

$$0 - (\text{Rate of energy out}) = (\text{Rate of energy accumulation})$$

$$-h(T - T_\infty)\frac{\pi d^2}{4} = \frac{\pi d^3}{6}\rho c_p \frac{dT}{dt} \tag{1.66}$$

with the initial condition $T = T_0$ at $t = 0$.

We assume the temperature in Eq. (1.66) is independent of position in the sphere (lumped).

On the other hand, it may be necessary to consider variations in the system's physical space without uniform properties. We treat this type of problem at the local level of the system to distribute the properties of viscosity, temperature, or concentration. The modeling approach for a distributed system applies the conservation equations in a shell balance to a local finite volume element. We describe the local *in* and *out* terms of a finite volume element with

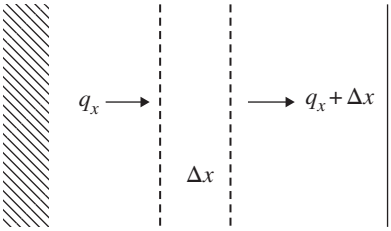


Figure 1.13 Energy balance around a finite wall element.

constitutive equations, such as Newton's law of viscosity, Fick's law of diffusion, or Fourier's law of heat conduction from Table 1.2. For example, consider the illustration of heat conduction through a wall in Figure 1.13. The left side of the wall is heated. The right side of the wall loses heat by convection to a neighboring fluid. Start with a differential element having a thickness Δx at any point in the wall along the x-direction, as illustrated in Figure 1.13.

An energy balance around the differential length Δx gives:

$$\text{(Rate of energy in)} - \text{(Rate of energy out)} = \text{(Rate of energy accumulation)}$$

$$-A\kappa \left.\frac{\partial T}{\partial x}\right|_x - A\left(-\kappa \left.\frac{\partial T}{\partial x}\right|_{x+\Delta x}\right) = A\Delta x \rho c_p \frac{\partial T}{\partial t} \tag{1.67}$$

where the terms on the left side represent heat conduction in and out by Fourier's law. The term on the right side accounts for the accumulation of thermal energy. Divide Eq. (1.67) by the differential volume of the wall represented by the product of the cross-sectional area and differential thickness, $A\Delta x$, and then take the limit as the differential length goes to zero:

$$\lim_{\Delta x \to 0} \frac{-A\kappa \left.\frac{\partial T}{\partial x}\right|_x - A\left(-\kappa \left.\frac{\partial T}{\partial x}\right|_{x+\Delta x}\right) = A\Delta x \rho c_p \frac{\partial T}{\partial t}}{A\Delta x} \rightarrow \frac{\partial}{\partial x}\left(\kappa \frac{\partial T}{\partial x}\right) = \rho c_p \frac{\partial T}{\partial t} \tag{1.68}$$

The result is a second-order partial differential equation subject to the boundary conditions at the left and right sides of the wall where $x = 0$ and $x = L$.

Apply energy conservation at the walls for the boundary conditions. On the left side, the thermal heating rate equals the conduction rate from the surface into the wall. The heat conduction into the surface at the right-hand side of the wall equals the rate of thermal convection to the fluid:

$$\text{Rate of energy in} = \text{Rate of energy out}$$

$$q = -\kappa \left.\frac{\partial T}{\partial x}\right|_{x=0} \text{ and } -\kappa \left.\frac{\partial T}{\partial x}\right|_{x=L} = h(T_L - T_\infty) \tag{1.69}$$

Thermodynamic conditions at system boundaries may require additional constitutive models, such as vapor–liquid equilibrium. We derive Fick's *second* law in one dimension from the local species balance around a finite volume element of length Δx, width Δz, and height Δy, as shown in Figure 1.14.

N represents the molar flux for the diffusing species. The rate balance in Figure 1.14 accounts for the flux in and out at positions x and $x + \Delta x$. Similar expressions apply to the y and z dimensions:

$$\text{(Rate in} - \text{out)}_x + \text{(Rate in} - \text{out)}_y + \text{(Rate in} - \text{out)}_z + \text{(Rate in} - \text{out)}$$
$$= \text{(Rate of accumulation)}$$

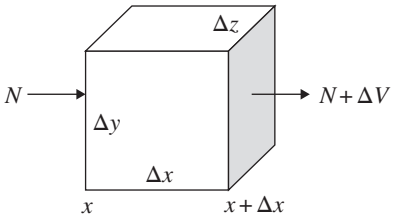


Figure 1.14 Finite volume element in a Cartesian coordinate system with dimensions Δx, Δy, and Δz.

$$\Delta y\Delta z(N_x - N_{x+\Delta x}) + \Delta x\Delta z(N_y - N_{y+\Delta y}) + \Delta x\Delta y(N_z - N_{z+\Delta z}) = \Delta x\Delta y\Delta z\frac{dC}{dt} \tag{1.70}$$

where the product of the lengths of two sides $\Delta y\Delta z$ is the area of flux normal to the x-direction, and so forth. The product of all three sides $\Delta x\Delta y\Delta z$ is the finite element volume. The element volume multiplied by the time derivative of concentration represents the accumulation term. The equation takes the form of a second-order *partial differential* equation (space and time dimensions) in the zero limits of the volume of the finite element $\Delta x\Delta y\Delta z \to 0$:

$$\lim_{\Delta x\to 0}\left\{\frac{\Delta y\Delta z(N_x - N_{x+\Delta x}) + \Delta x\Delta z(N_y - N_{y+\Delta y}) + \Delta x\Delta y(N_z - N_{z+\Delta z})}{\Delta x\Delta y\Delta z}\right\}$$
$$= -\left(\frac{\partial}{\partial x}N_x + \frac{\partial}{\partial y}N_y + \frac{\partial}{\partial z}N_z\right) = \frac{\partial C}{\partial t} \tag{1.71}$$

If we replace the molar flux terms in Eq. (1.71) by Fick's first law from Table 1.2, Eq. (1.71) becomes:

$$-\left[\frac{\partial}{\partial x}\left(-D\frac{\partial C}{\partial x}\right) + \frac{\partial}{\partial y}\left(-D\frac{\partial C}{\partial y}\right) + \frac{\partial}{\partial z}\left(-D\frac{\partial C}{\partial z}\right)\right] = \frac{\partial C}{\partial t} \tag{1.72}$$

subject to an initial condition: $C = C_0$ at $t = 0$ and two boundary conditions for each dimension. For example, the x-dimension boundary conditions are:

Rate in = Rate out

$$\beta_2(C - C_\infty) = \beta_1\frac{\partial C}{\partial x}\text{ at } x = x_0 \text{ and } \beta_3\frac{\partial C}{\partial x} = \beta_4(C - C_\infty)\text{ at } x = x_1 \tag{1.73}$$

where β_1 through β_4 are diffusion and convection parameters. The y and z dimensions require similar boundary conditions. The initial condition is the equation's solution at $t = 0$ and is the calculation's starting point. Boundary conditions impose the unique, physical nature of the system into the model. We derive boundary conditions from applying equilibrium thermodynamics or conservation equations to the boundary surfaces.

For constant diffusivity and in the limit $V \to 0$, Eq. (1.72) reduces to

$$D\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right) = \frac{\partial C}{\partial t} \tag{1.74}$$

To complicate the matter further, reaction rates other than first-order give nonlinear rate expressions, which give nonlinear differential equations when substituted for the generation terms. For example, consider the one-dimensional problem with a nonlinear reaction term for the generation part of the mole balance:

$$D\frac{\partial^2 C}{\partial x^2} - \frac{kC}{K + C} = \frac{\partial C}{\partial t} \tag{1.75}$$

We can approximate solutions to this partial differential equation using the numerical methods introduced in Chapter 14.

1.2.3 Degrees of Freedom

Understanding the relationship between the number of equations (E) and the number of variables (V) in a system is crucial for finding solutions. Here is a breakdown of the different scenarios and their implications:

$E = V$: In an independent system of equations, where the number of equations equals the number of variables, one unique solution exists. This is characteristic of linear, independent equations. However, nonlinear equations may have multiple roots.

$E \neq V$: An ill-conditioned system of equations has an unequal number of equations and unknowns, leading to challenges in finding solutions.

$E < V$: In this scenario, the solution has $F = V - E$ degrees of freedom. Specifying the values of F unknowns allows for solving equations for a unique solution. An underspecified problem may present an opportunity for optimization.

$E > V$: If there are more equations than unknowns, the system is over-specified and has no unique solution. In such cases, the set of solutions to the system of equations is infinite. However, a subset of V-independent equations may have a unique solution. Utilizing uncertainty in the additional equations can help find a solution that best satisfies the system of equations.

When dealing with an extensive system of equations, it is crucial to specify it correctly before attempting a solution. Simply matching the number of variables with the number of equations may not suffice, as some linear systems may be singular and lack a unique solution. For nonlinear problems, multiple solutions may exist, but identifying the correct one involves ensuring that the results align with the system's physics. For instance, solutions that yield negative mole fractions or concentrations greater than one should be eliminated, as absolute temperatures and concentrations must be positive by definition.

1.2.4 Dimensionless Equations

We may recast model equations in dimensionless terms to identify *significant* vs. *negligible* terms and minimize the effects of round-off error in numerical solutions. Follow Aris' (1993) rules when making equations dimensionless:

1. When we are interested in the effect of a certain quantity, that quantity should appear in only one dimensionless parameter, not in a dimensionless variable.
2. Dimensionless dependent variables should range from zero to one.
3. Dimensionless independent variables should range over finite intervals, e.g. (0, 1) or $(-p, +p)$.
4. Dimensionless parameters show the relative magnitudes of terms in the model equations. Consider ignoring relatively small terms.

Making the model equations dimensionless usually reduces the number of parameters by at least one. Nondimensionalization also reveals the minimum number of parameters needed to describe the physical nature of the system.

To illustrate, make Eq. (1.57) dimensionless following the Aris rules:

$$\frac{dX_A}{d\overline{V}} + \varepsilon \frac{d^2X_A}{d\overline{V}^2} = \phi \tag{1.76}$$

where the conversion X_A is already dimensionless, with a range of zero to one. Define the dimensionless independent variable and parameters as follows:

$$\bar{V} = \frac{V}{V_0} \quad \varepsilon = \frac{C_{A0}A^2E}{F_{A0}V_0} \quad \phi = \frac{-r_AV_0}{F_{A0}}$$

Note the reduction in the total number of parameters. The three parameter groups F_{A0}, $C_{A0}S^3E$, and $-r_A$ are combined into two parameters ε and ϕ. Aris' fourth rule of dimensionless numbers applies to the parameter ε. For small ε (for cases of large $F_{A0}V_0$ or small $C_{A0}S^3E$), Eq. (1.76) reduces to the PFR result in Eq. (1.54).

For some problems, there are advantages of scaling the independent variable to range between the integration limits of zero to one:

$$x^* = \frac{x - x_0}{x_n - x_0} \rightarrow x = x_0 + x^*(x_n - x_0) \tag{1.77}$$

where x_0 and x_n are the integration limits or initial and final values for x. The advantages include:

- Increased computational stability with smaller integration steps
- Improved accuracy with smaller integration steps
- Higher computational efficiency with fewer integration steps
- Ability to integrate both directions of x (positive from x_0 to x_n or negative from x_n to x_0)

Recall that x^* is a function of x according to Eq. (1.77). To apply variable scaling in our numerical methods, we multiply each function for the differential equations by the difference in the limits of integration $(x_n - x_0)$ and define x in terms of x^* in the expressions for f. As a result, the first-order differential equation becomes dimensionless in the independent variable:

$$\frac{dy}{dx} = \frac{1}{(x_n - x_0)}\frac{dy}{dx^*} = f(x^*, y) \quad \rightarrow \quad \frac{dy}{dx^*}$$
$$= (x_n - x_0)f(x^*, y)x^* = \frac{x - x_0}{x_n - x_0} \quad \rightarrow \quad x = x_0 + x^*(x_n - x_0) \tag{1.78}$$

1.2.5 Stochastic vs. Deterministic Modeling

Unlike deterministic models, which generate outcomes based on fixed numerical inputs, stochastic models take into account the inherent uncertainty of the system by producing a probability distribution of results. This distribution is derived from random inputs drawn from unique probability distributions of uncertainty. Stochastic models predict the expected outcome and provide confidence intervals, which serve as reliability measures. Smaller confidence intervals indicate higher reliability of the probable outcome. Given the unpredictable nature of forecasting, models in this domain often rely on stochastic inputs.

This book also delves into empirical and stochastic modeling, exploring techniques such as evolutionary optimization and uncertainty propagation through data analysis and least-squares regression. These approaches broaden the scope of modeling by incorporating real-world data and addressing the inherent uncertainty present in many systems.

1.2.6 Model Verification and Validation

The utility of mathematical models is indisputable, yet a prudent approach is essential. Models are invaluable tools for comprehending systems, identifying correlations among variables,

refining processes, and facilitating design. However, it is crucial to recognize that while mathematical modeling enhances our understanding, it cannot wholly replace empirical evidence.

Advancements in computing power, accuracy, and accessibility have transformed the landscape. In chemical property prediction, reliance on computer models instead of experimentation is becoming commonplace. Notably, esteemed journals such as the *ACS Journal of Chemical and Engineering Data* now welcome submissions based on computational experiments. In harsh conditions where conventional measurements are impractical, computer simulations serve as "soft sensors," providing unattainable insights through traditional means.

Despite these strides, the pinnacle of chemical process simulation still necessitates a blend of bench-scale experiments and pilot-scale studies before unequivocal trust is bestowed upon simulations. The modeling journey is iterative, involving continuous refinement:

1. Define the physical system and assess existing knowledge.
2. Formulate simplified models to render the system mathematically tractable.
3. Iteratively solve model equations across varying parameters to elucidate relationships.
4. Validate the model against empirical data, refining assumptions and understanding as needed.

Computers dazzle us with their computational and graphical prowess. However, it is imperative not to succumb to complacency by unquestioningly trusting computer-generated results. Rigorous verification and validation processes are paramount. Robust models should yield accurate solutions and demonstrate predictive efficacy and uncertainty resilience.

The adage "Garbage in, garbage out" underscores the importance of input quality in determining model output reliability. Computational tools like MATLAB offer immense potential when coupled with validated models, enabling rapid exploration of "what-if" scenarios without disrupting operations. The true value of modeling lies not merely in its creation but in its ongoing utilization to probe process dynamics and uncover latent opportunities. Failure to leverage models for such exploration squanders the effort invested in their development, solving, and validation.

1.3 Expert Problem-Solvers

Engineers derive satisfaction from solving problems and moving on to the next challenge. However, it is crucial to resist the temptation to skip the critical step in problem-solving: documenting and reflecting on our results. Our proven problem-solving methodology comprises four principal components: starting with basic equations, clearly stating all assumptions, performing the analysis, and checking the results for engineering consistency and common sense.

The legacy of the US NASA space shuttle program serves as a sobering reminder of the profound impact and consequences of both practical and flawed problem-solving processes, evident in the triumphs and tragic accidents of the Challenger in 1986 and Columbia in 2003 (Gehman 2003). Indeed, history underscores the adage that "there are no small mathematical errors in engineering work." (Kowalczyk 1963) The loss of NASA's Mars Orbiter due to overlooked unit conversions and the collapse of the Quebec Cantilevered Bridge due to failure to redesign for added length are poignant examples of costly engineering mistakes.

The accuracy of our results and the clarity of our presentation are matters of ethics and professionalism, as lives may depend on the reliability of our work. It is emphasized that a solution checked by an engineer significantly enhances credibility and trust, and this psychological gain must be matched with technical sufficiency. Checking calculations is an essential layer of safety protection, and verification and validation are not trivial parts of the solution process. The problem-solving format recommended here promotes engineering documentation, proper numerical precision, accuracy, sound engineering judgment, and critical evaluation of results.

Mathematical models empower engineers to explore parameter space, understand processes, uncover relationships, and identify improvement opportunities. These models may stem from first principles (laws of conservation) (Farhadi et al. 2009) or convenient functions such as polynomials that capture observed system behavior. It is advisable to start with simple expressions and add rigor only when necessary. In many cases, solving model equations requires numerical approximation techniques. The subsequent chapters introduce MATLAB as a platform for building models and provide a comprehensive suite of tools for solving common modeling problems.