Chapter 1

Linear Equation Methods

1.1 Matrix Multiplication

Models frequently are described by sets of coupled, linear equations written in the form

\[ Ax = b. \]  

(1.1)

The product \( Ax \) requires us to define the matrix multiplication operation. For systems where \( x \) is a column vector, we can write

\[ b_i = \sum_{j=1}^{J} A_{i,j} x_j \quad i = 1, \ldots, I \]

where the first index of \( A_{i,j} \) refers to the row and the second to the column of matrix \( A \). This operation fixes the sizes of the arrays as

\[ A^{I \times J} x^{J \times 1} = b^{I \times 1}. \]

In general, we can premultiply the array \( B \) by array \( A \) if the maximum column index of \( A \) is identical to the maximum row index of \( B \):

\[ A^{I \times J} B^{J \times K} = C^{I \times K} \]

where

\[ C_{i,k} = \sum_{j=1}^{J} A_{i,j} B_{j,k} \quad ,i = 1, \ldots, I, \quad k = 1, \ldots, K. \]

It is not difficult to see that in general

\[ AB \neq BA. \]

1.1.1 \( n \)-dimensional array operations

Numerous modeling and numerical solution procedures result in variables with more than two indices (i.e., variables in the form \( a_{i,j,k} \)); we will see that direct methods for computing the steady-state solution of these systems subsequently generate Jacobian arrays with six-dimensions. While one can re-index these variable arrays to obtain conventional one- and two-dimensional arrays, we may find that implementation
of the computational solution procedure is clearer if the variables and equations were left in their original form. This motivated developing computational methods to generalize matrix multiplication to these higher-dimensional systems.

Consider the product operation

\[ A^{I \times J} B^{J \times K} = C^{I \times K} \]

\[ C_{i,k} = \sum_{j=1}^{J} A_{i,j} B_{j,k}, \quad i = 1, \ldots, I, \quad k = 1, \ldots, K. \]

We see that this motivates the development of the generalization of matrix multiplication to these higher-dimensional systems. For example, if

\[ A^{(L_1 \times L_2 \times \ldots \times L_p)} \times (M_1 \times M_2 \times \ldots \times M_q) = A^{L \times M} \]

\[ B^{(M_1 \times M_2 \times \ldots \times M_q)} \times (N_1 \times N_2 \times \ldots \times N_r) = B^{M \times N} \]

we can define the generalized matrix multiplication operation

\[ A^{L \times M} B^{M \times N} = C^{L \times N} \]

where

\[ C_{l_1, l_2, \ldots, l_p, n_1, n_2, \ldots, n_r} = \sum_{m_1=1}^{M_1} \ldots \sum_{m_q=1}^{M_q} A_{l_1, l_2, \ldots, l_p, m_1, m_2, \ldots, m_q} B_{m_1, m_2, \ldots, m_q, n_1, n_2, \ldots, n_r}. \]

Given this definition, we can define a square array as

\[ A^{L \times M} \] such that \( L_1 = M_1, L_2 = M_2, \ldots L_q = M_q \), and \( p = q \)

the identity array as a square array \( A \) with all zero elements except

\[ I^{L \times M} : \quad I_{l_1, l_2, \ldots, l_p, l_1, l_2, \ldots, l_q} = 1 \]

and the matrix inverse as

\[ [A^{L \times M}]^{-1} A^{L \times M} = I^{L \times M} \]

We will now consider three Chemical Engineering systems to illustrate modeling problems involving linear equations and to review how linear models are written in matrix form.

### 1.2 The Nonadiabatic Flash Drum Example

As an example to be used for many of the linear algebra examples to follow, consider the nonadiabatic flash separation system shown in Fig. 1.1. It is assumed the system is operated in such a way that at equilibrium, two phases (liquid and vapor) exist, and that the drum temperature and pressure can be maintained at some specified set of values. Under these conditions, component M (methane) has a boiling point of \(-162^\circ\text{C}\) and component O (n-octane) \(126^\circ\text{C}\); we should expect that virtually all of the methane leaves through the vapor steam and that there is a distribution of octane in both product streams.
By the definition of mole fractions

\[ x_M + x_O = 1 \] (liquid mole fractions)

\[ y_M + y_O = 1 \] (vapor mole fractions)

Phase equilibrium is achieved when the chemical potential \( \mu_i^\alpha \) of component \( i \) is equal in each of the phases \( \alpha \). The methane/octane mixture can be accurately approximated by properties of ideas mixtures, so

\[
\begin{align*}
\mu_M^L(x_M, x_O, T_l, P_l) - \mu_M^V(y_M, y_O, T_v, P_v) &= 0 \implies \frac{y_M}{x_M} = K_M(T, P) = 220 \\
\mu_O^L(x_M, x_O, T_l, P_l) - \mu_O^V(y_M, y_O, T_v, P_v) &= 0 \implies \frac{y_O}{x_O} = K_O(T, P) = 0.5
\end{align*}
\]

The \( K \) values are taken from [20]. The material balances and definition of total methane feed rate give the four equations

\[
\begin{align*}
\text{Mass in} &= \text{Mass out} \\
F &= V + L \\
F z_M &= V y_M + L x_M \\
F z_O &= V y_O + L x_O \\
F z_M &= 10
\end{align*}
\]

At this point we define the vector of state variables

\[
x = \begin{bmatrix} x_M \\ x_O \\ y_M \\ y_O \\ V \\ L \\ F \end{bmatrix}
\]

and so it appears there are 7 variables and 8 equations, some of which are nonlinear due to product terms such as \( V y_O \).

If we rearrange the equations,

\[
\begin{align*}
x_M + x_O &= 1 \\
y_M - 220x_M &= 0 \\
y_O - 0.5x_O &= 0 \\
y_M + y_O &= 1 \implies \text{Intensive}
\end{align*}
\]
\[ V + L - F = 0 \]  
\[ y_M V + x_M L - z_M F = 0 \]  
\[ z_M F = 10 \]  
\[ y_O V + x_O L - z_O F = 0 \]

the full set of modeling equations can be split into two sets (corresponding to intensive and extensive variables) which, if solved consecutively, can be solved independently:

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
-220 & 0 & 1 & 0 \\
0 & -0.5 & 0 & 1 \\
0 & 0 & 1 & 1 \\
y_M & x_M & -z_M \\
y_O & x_O & -z_O
\end{bmatrix}
\begin{bmatrix}
x_M \\
x_O \\
y_M \\
y_O
\end{bmatrix} =
\begin{bmatrix}
1 \\
0 \\
0 \\
1
\end{bmatrix}
\text{ or } Ay = b
\]

\[
\begin{bmatrix}
1 & 1 & -1 \\
y_M & x_M & -z_M \\
y_O & x_O & -z_O
\end{bmatrix}
\begin{bmatrix}
V \\
L \\
F
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
10 \\
0
\end{bmatrix}
\text{ or } Bz = c.
\]

1.3 Isothermal Chemical Vapor Deposition Reactor Model

Microelectronic devices are manufactured in a sequence of processing steps that include numerous thin-film deposition and etching operations. As one example, consider the reaction process where thin films of polycrystalline silicon are deposited using a Chemical Vapor Deposition (CVD) system [12].

This process consists of a feed gas of silane \((\text{SiH}_4)\) fed to a vacuum chamber containing the wafer(s) to be processed. The reaction process consists of the gas phase decomposition of silane to silylene

\[ \text{SiH}_4 \rightarrow \text{SiH}_2 + \text{H}_2 \]

where the gas phase reaction rate \(R_g\) is given by

\[ R_g = k_g C_y \text{SiH}_4 \]

has units \(\text{mol}/(m^3 \text{s})\). The actual Si film deposition is governed by the surface reaction:

\[ \text{SiH}_2 \rightarrow \text{Si} + \text{H}_2 \]

where

\[ R_s = k_s C_y \text{SiH}_2 \]

and has units \(\text{mol}/(m^2 \text{s})\). We note that the gas phase reaction produces two moles of gas for each consumed while the surface reaction produces one mole of gas for each consumed.

If we approximate the gas phase behavior of this system as an isothermal, well-mixed reactor with pure silane feed \((y_{\text{SiH}_4} = 1)\) and for now assume the conversion rate of silane is low in this reactor (we will
check the validity of this assumption later), a material balance on each gas phase chemical species gives the following nonlinear equations

\[
CV \frac{dy_{SiH_4}}{dt} = QC - (QC + k_g CV y_{SiH_4})y_{SiH_4} - k_g CV y_{SiH_4}
\]

\[
CV \frac{dy_{SiH_2}}{dt} = -(QC + k_g CV y_{SiH_4})y_{SiH_2} - k_s CW_a y_{SiH_2} + k_g CV y_{SiH_4}
\]

\[
CV \frac{dy_{H_2}}{dt} = -(QC + k_g CV y_{SiH_4})y_{H_2} + k_s CW_a y_{SiH_2} + k_g CV y_{SiH_4}
\]

Under the assumption of low silane conversion in our system \((y_{SiH_4} \to 1 \text{ and } y_{SiH_2}, y_{H_2} \to 0)\), we can replace the nonlinear terms in the modeling equations with the following linear approximations (the topic of linearization will be discussed later in this text):

\[
CV \frac{dy_{SiH_4}}{dt} = QC - QC y_{SiH_4} - k_g CV - 2k_g CV(y_{SiH_4} - 1) - k_g CV y_{SiH_4}
\]

\[
= -QC y_{SiH_4} - 3k_g CV y_{SiH_4} + QC + k_g CV \quad (1.3)
\]

\[
CV \frac{dy_{SiH_2}}{dt} = -QC y_{SiH_2} - k_g CV y_{SiH_2} - k_s CW_a y_{SiH_2} + k_g CV y_{SiH_4}\quad (1.4)
\]

\[
CV \frac{dy_{H_2}}{dt} = -QC y_{H_2} - k_g CV y_{H_2} + k_s CW_a y_{SiH_2} + k_g CV y_{SiH_4}\quad (1.5)
\]

For operating conditions typical for low-pressure CVD of poly-Si (1 torr total pressure and 900K wafer temperature) the reaction rate coefficients [12] and gas total concentration \(C\) computed using the ideal gas law are

\[
k_g = 0.016 \text{ s}^{-1}
\]

\[
k_s = 190 \text{ m/s}
\]

\[
C = 0.0178 \text{ mol/m}^3
\]

If we consider a reactor loaded with 20, 200mm wafers, and that the reactor is a tube 1m in length and 0.15m in radius, and that the feed gas flow to the reactor is pure silane at 100 sccm, the remaining model parameters can be computed:

\[
Q = 0.0042 \text{ m}^3/s
\]

\[
V = 0.0707 \text{ m}^3
\]

\[
W_a = 2 \times 20 \times 0.0314 \text{m}^2 = 1.2566 \text{ m}^2
\]

The final form of the reactor model can be written in matrix form as

\[
\frac{dy}{dt} = Ay - b \quad \text{with} \quad y = \begin{bmatrix} y_{SiH_4} \\ y_{SiH_2} \\ y_{H_2} \end{bmatrix}
\]

with

\[
A = \begin{bmatrix}
-0.1071 & 0 & 0 \\
0.016 & -3378 & 0 \\
0.016 & 3378 & -0.0751
\end{bmatrix} \quad b = \begin{bmatrix}
-0.0751 \\
0 \\
0
\end{bmatrix}.
\]

All coefficients have units \(s^{-1}\).
We derive the CVD reactor model as an ODE model because in some cases, their steady-state generates a linear algebra problem. The steady state solution is computed by setting the time-derivatives to zero and solving the resulting linear system in the form of (1.1). A numerical solution will be discussed later in this chapter.

1.4 Staged Multicomponent Absorption Process

We consider the problem of computing the concentration profiles of a gas mixture contacting water in a countercurrent staged absorption process. A single stage of the process is shown in Fig. 1.2.

![Figure 1.2: Stage j of the gas/water absorption cascade.](image)

We denote the gas phase and liquid phase mode fractions of species $i$ in stage $j$ as $y_{i,j}$ and $x_{i,j}$, respectively; the state of the system is described by the two, two-dimensional arrays $x$ and $y$ where

$$
x_{i,j} = \begin{bmatrix}
x_{N_2,j} \\
x_{H_2S,j} \\
x_{H_2,j} \\
x_{CO_2,j}
\end{bmatrix} \quad \text{and} \quad y_{i,j} = \begin{bmatrix}
y_{N_2,j} \\
y_{H_2S,j} \\
y_{H_2,j} \\
y_{CO_2,j}
\end{bmatrix} \quad j = 1, \ldots, J.
$$

The gas and liquid phases may contain other chemical species (e.g., water in the gas phase) which are not accounted for in the mole fraction definitions above, so we should not expect $x_{i,j}$ and $y_{i,j}$ to sum to 1.

If each stage is at thermodynamic equilibrium, the gas/liquid phase distribution of the species can be approximated using Henry’s Law:

$$y_{i,j} = H_i x_{i,j}.$$ 

It is assumed the gas fed to the column is saturated with water and has the following properties

<table>
<thead>
<tr>
<th>Gas species</th>
<th>Feed mole fraction</th>
<th>Henry’s Law constant (293 K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_2$</td>
<td>$y_{1,j+1} = 0.2$</td>
<td>$H_1 = 8.04 \times 10^4$</td>
</tr>
<tr>
<td>$H_2S$</td>
<td>$y_{2,j+1} = 0.2$</td>
<td>$H_2 = 0.0483 \times 10^4$</td>
</tr>
<tr>
<td>$H_2$</td>
<td>$y_{3,j+1} = 0.1$</td>
<td>$H_3 = 6.83 \times 10^4$</td>
</tr>
<tr>
<td>$CO_2$</td>
<td>$y_{4,j+1} = 0.1$</td>
<td>$H_4 = 0.142 \times 10^4$</td>
</tr>
</tbody>
</table>

The feed water is pure, so $x_{i,0} = 0$, $i = 1, \ldots, 4$; because the indices $j = 0$ and $j = J + 1$ are outside the range used for the variables, $x_{i,0}$ and $y_{i,J+1}$ are model parameters.

We assume the gas molar flow rate $G$ ($mol/s$) and molar liquid flow rate $L$ ($mol/s$) in each stage are constant; likewise, we assume the volume $V$ ($m^3$) of each stage and the liquid void fraction $\epsilon$ are constant.
1.4. STAGED MULTICOMPONENT ABSORPTION PROCESS

and identical for each stage. Under these assumptions, the material balance for each component and stage can be written as

\[ C_g V \frac{dy_{i,j}}{dt} = G(y_{i,j+1} - y_{i,j}) - KV(y_{i,j} - H_i x_{i,j}) \] (1.6)

\[ C_l (1 - \epsilon) \frac{dx_{i,j}}{dt} = L(x_{i,j-1} - x_{i,j}) + KV(y_{i,j} - H_i x_{i,j}) \] (1.7)

where \( K \) acts as a mass-transfer rate constant. We consider a laboratory-scale version of this absorption column operating at room temperature and pressure with the following parameters:

\[ C_g = 40.9 \text{ mol/m}^3 \]
\[ C_l = 5.56 \times 10^4 \text{ mol/m}^3 \]
\[ G = 0.0204 \text{ mol/s (500 cm}^3/\text{s)} \]
\[ L = 5.56 \text{ mol/s (100 cm}^3/\text{s)} \]
\[ V = 2 \times 10^{-4} \text{ m}^3 (200 cm}^3) \]
\[ \epsilon = 0.5 \]
\[ K = 50 \text{ mol/(m}^3 \text{s)} \]

To write the modeling equations in matrix form, we “stack” the variable arrays \( y \) and \( x \) to form the new array \( z \), thus

\[ \frac{dz}{dt} = Az - b \quad \text{with} \quad z = \begin{bmatrix} y \\ x \end{bmatrix} \]

To illustrate how the 4-dimensional array \( A \) is constructed, we list explicitly the array elements below.

For the gas phase and \( i = 1, \ldots, I \),

\[ \frac{d_{z_{i,j}}}{dt} = \frac{d_{y_{i,j}}}{dt} = \frac{G}{C_g V \epsilon} (y_{i,j+1} - y_{i,j}) - \frac{K}{C_g \epsilon} (y_{i,j} - H_i x_{i,j}) \]

and for \( j = 1, \ldots, J - 1 \):

\[ A_{i,j,i,j} = -\frac{G}{C_g V \epsilon} - \frac{K}{C_g \epsilon} \]
\[ A_{i,j,i,j+1} = \frac{G}{C_g V \epsilon} \]
\[ A_{i,j,i+1,j} = \frac{K H_i}{C_g \epsilon} \]

and \( j = J \):

\[ A_{i,i,i,J} = -\frac{G}{C_g V \epsilon} - \frac{K}{C_g \epsilon} \]
\[ A_{i,i+1,J} = \frac{K H_i}{C_g \epsilon} \]
\[ b_{i,J} = -\frac{G}{C_g V \epsilon} y_{i,J+1} \]
CHAPTER 1. LINEAR EQUATION METHODS

For the liquid phase and \( i = 1, \ldots, I \),
\[
\frac{dz_{i+1,j}}{dt} = \frac{dx_{i,j}}{dt} = \frac{L}{C_l V (1 - \epsilon)} (x_{i,j-1} - x_{i,j}) + \frac{K}{C_l (1 - \epsilon)} (y_{i,j} - H_i x_{i,j})
\]
and for \( j = 2, \ldots, J \):
\[
A_{i+1,j,i+1,j} = -\frac{L}{C_l V (1 - \epsilon)} - \frac{K H_i}{C_l (1 - \epsilon)}
\]
\[
A_{i+1,j,i+1,j-1} = \frac{L}{C_l V (1 - \epsilon)}
\]
\[
A_{i+1,j,i,j} = \frac{K}{C_l (1 - \epsilon)}
\]
and \( j = 1 \):
\[
A_{i+1,1,i+1,1} = -\frac{L}{C_l V (1 - \epsilon)} - \frac{K H_i}{C_l (1 - \epsilon)}
\]
\[
A_{i+1,1,i,1} = \frac{K}{C_l (1 - \epsilon)}
\]

We note that in defining \( A \), the first pair of indices denotes equation number and the second pair denote variable index number.

1.5 Solving Ax=b

Our goal is to solve linear algebra problems of the form
\[
A^{I \times J} x^{J \times 1} = b^{I \times 1}
\]  
(1.8)
using numerical methods that are computationally reliable and give the most information about the structure of the modeling equations themselves. Algebraically, it is possible to write the solution to (1.8) as
\[
x = A^{-1} b
\]
if \( A^{-1} \) exists, but computing \( A^{-1} \) directly satisfies neither of the stated goals. Therefore, we wish to consider numerical methods that transform the hard problem to solve (1.8) into the easier problem
\[
U x = d
\]  
(1.9)
where \( U \) has a structure that makes the solution to (1.9) simpler to find and gives insight into the the problem we are solving, such as whether there are a sufficient number of equations to determine a unique solution.

1.5.1 Factorization by Gaussian Elimination

There are an infinite number of such transformations; we will discuss Gaussian elimination with partial pivoting, a numerical technique which reduces computational inaccuracies. Gaussian elimination applied to (1.8) consists of a \( J \)-step forward elimination procedure to obtain the system (1.9) followed by back substitution to compute the values \( x \). The forward elimination procedure at each step \( j = 1, \ldots, J \) has three main operations:
1. In the $j$th step of the elimination procedure, the pivot is chosen as the largest (in absolute value) element $A_{k,j}$ of those found in the subcolumn below (and including) the pivot point $A_{j,j}$;

2. The pivot row $A_{k,l} = 1, \ldots, J$, $b_k$ then is interchanged with the $j$th row (this leaves the arrays unchanged if the pivot point is the largest-magnitude element);

3. For each nonzero $A_{m,j}$, $m = j + 1, \ldots, I$ in the rearranged system, a multiplier is computed as $-A_{m,j}/A_{j,j}$ and the product of the multiplier and the pivot row is added to row $m$; this procedure is used to remove all nonzero terms below the pivot. Generally, large multipliers (small pivots) contribute to numerical errors: in general it is a poor idea to multiply the pivot row with a large number and then add it to another, because the first might wash out any contribution of the second through round off error;

The main idea behind partial pivoting is that it tends to make the solution more resistant to small numerical errors.

The array permutations and multiplications can all be recorded as array operations. By factoring the coefficient array of (1.8) into two or more separate arrays

$$A = PLU$$

where $L$ and $U$ have a lower and upper triangular structure such that the solution of (1.9) can be found by simple back substitution. $P$ contains row permutation (exchange) information; the product $PL$ may not have a lower triangular structure. The elements of $d$ are also computed from the back substitution problem

$$PLd = b$$

or $$Ld = Pb$$ because $P^{-1} = P$.

It is easy to show that $P^{-1} = P$: because $P$ interchanges rows, an additional product operation will return the array to its original form. Therefore, $PP = I$ and $P^{-1} = P$ follows.

1.5.2 An example of a solution computed using partial pivoting

Returning to the four modeling equations corresponding to the intensive variable of the flash drum example (1.2), consider a solution obtained with pivoting, starting with interchanging the first two rows:

$$\begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
0 \\
-220 \\
1 \\
0 \\
\end{bmatrix}$$

$$\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4 \\
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}$$

$$A = PLU$$

where $L$ and $U$ have a lower and upper triangular structure such that the solution of (1.9) can be found by simple back substitution. $P$ contains row permutation (exchange) information; the product $PL$ may not have a lower triangular structure. The elements of $d$ are also computed from the back substitution problem

$$PLd = b$$

or $$Ld = Pb$$ because $P^{-1} = P$.

It is easy to show that $P^{-1} = P$: because $P$ interchanges rows, an additional product operation will return the array to its original form. Therefore, $PP = I$ and $P^{-1} = P$ follows.
so if we multiply row 1 by $1/220$ (a small multiplier) and add the product to row 2 (do not forget the RHS!):

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1/220 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
-220 & 0 & 1 & 0 \\
0 & -0.5 & 0 & 1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 & 0 \\
1/220 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
0 \\
1 \\
0 \\
0
\end{bmatrix}
$$

Our multiplier is now $1/2$, which gives

$$
\begin{bmatrix}
-220 & 0 & 1 & 0 \\
0 & 1 & 1/220 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
1 \\
0.5
\end{bmatrix}
$$

so interchanging rows 3 and 4,

$$
\begin{bmatrix}
-220 & 0 & 1 & 0 \\
0 & 1 & 1/220 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
1 \\
1
\end{bmatrix}
$$

and using $-1/440$ as the multiplier gives

$$
\begin{bmatrix}
-220 & 0 & 1 & 0 \\
0 & 1 & 1/220 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
1 \\
1
\end{bmatrix}
$$

The solution is found by back substitution. We now notice a very important point: If we do all of our calculations ignoring the small terms such as $1/220$ and $1/440$,

$$
y_O \approx 0.5 \\
y_M \approx 1 - 0.5 = 0.5 \\
x_O \approx 1 \\
x_M \approx 0.5/220 \approx 0.0
$$

The same results would have been found if we neglected these terms during the elimination steps.

### 1.5.3 The solution without pivoting

We now compare the results obtained without pivoting:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
-220 & 0 & 1 & 0 \\
0 & -0.5 & 0 & 1 \\
0 & 0 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}
= 
\begin{bmatrix}
1 \\
0 \\
0 \\
1
\end{bmatrix}
$$
which means we find by back substitution and sloppy arithmetic an incorrect result:

\[ y_O \approx 0.5 \]
\[ y_M \approx 440(0.5 - 0.5) = 0 \]
\[ x_O \approx 1 \]
\[ x_M \approx 1 - 1 = 0 \]

### 1.5.4 Computational example: Isothermal flash

It is simple to use the Gaussian elimination technique in MATLAB because factorization and back substitution are combined in the \ operation. Factorization alone is done with the `lu` routine; examples of both are presented below.

```matlab
>> A = [1 1 0 0; -220 0 1 0; 0 -0.5 0 1; 0 0 1 1];
>> b = [1; 0; 0; 1];
>> y = A\b; y'
ans =
    0.0023 0.9977 0.5011 0.4989

>> [L,U,P] = lu(A);
```

The factorization routine `lu` returns three arrays

\[ L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.0045 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -0.5 & 0.0023 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} -220 & 0 & 1 & 0 \\ 0 & 1 & 0.0045 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0.9977 \end{bmatrix}, \quad P = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \]

The array `U` is the upper triangular array we obtained from the Gaussian elimination procedure and `P` contains the row changes. The benefits of decomposing the problem into matrix factorization followed by back substitution are realized in large, iterative problems where the coefficient matrix `A` is constant and the RHS `b` changes (e.g., finite difference discretization of PDEs). If try the following computational
experiment and measure the number of floating point operations (1 + 2 requires one flops),

\[
A = \text{randn}(200,200);
\]
\[
b = \text{randn}(200,1);
\]
\[
[L,U,P] = \text{lufact}(A); \quad 5.31 \text{ million flops}
\]
\[
c = A\backslash b; \quad 5.59 \text{ million flops}
\]
\[
d = L\backslash (P\ast b); \quad y=U\backslash d; \quad 161,000 \text{ flops}
\]

Notice how the back substitution accounts for only about 3% of the total computational cost of the linear equation solution procedure.

### 1.5.5 Computational example: PolySi CVD deposition rate

Returning to the chemical vapor deposition reactor model discussed in Section(1.3), using the constants listed, we can set up the coefficient array and vector of nonhomogeneous terms by

\[
A = \text{zeros}(3,3); \quad b = \text{zeros}(3,1);
\]
\[
A(1,1) = -Q/V - 3\ast kg;
\]
\[
b(1,1) = -Q/V - kg
\]
\[
A(2,1) = kg;
\]
\[
A(2,2) = -Q/V - kg - ks\ast Wa/V;
\]
\[
A(3,1) = kg;
\]
\[
A(3,2) = ks\ast Wa/V;
\]
\[
A(3,3) = -Q/V - kg;
\]

and compute the steady-state solution using

\[
>> \text{format short e}
\]
\[
>> y = A\backslash b
\]
\[
y =
\]
\[
7.0115e-01
\]
\[
3.3211e-06
\]
\[
2.9885e-01
\]

We can determine the deposition rate using this solution by computing the rate of the deposition reaction by solid silicon’s molar density \( \rho = 82920 \text{ mol/m}^3 \):

\[
>>\text{DepRate} = ks\ast C\ast y(2)/82920 \ast 1e6 \ast 3600 \text{ deposition rate in microns/hr}
\]
\[
\text{DepRate} =
\]
\[
0.4881
\]

which indicates that this wafer would have to be processed for over one hour to achieve a 0.5\( \mu \text{m} \) film, which is a typical length scale for microelectronic devices.
1.7 Number of Equations ≠ Number of Unknowns

To this point, the linear equation sets contained equal numbers of equations and unknowns, giving the square systems
\[ A^{I \times J} x^{J \times 1} = b^{I \times 1} \]
where \( I = J \). For the case of \( I = J = 4 \), the Gaussian elimination procedure resulted in a system in upper-triangular form (where the *'s denote potentially non-zero elements):
\[
\begin{bmatrix}
* & * & * & * \\
0 & * & * & * \\
0 & 0 & * & * \\
0 & 0 & 0 & * \\
\end{bmatrix}
\]
\( x = d \) with \( d = \)
\[
\begin{bmatrix}
* \\
* \\
* \\
* \\
\end{bmatrix}
\]
and \( d \neq b \) in general.

In this Section we mainly consider the cases of having more equations than unknowns – the opposite case normally can only be solved satisfactorily by returning to the modeling stage and finding the missing modeling equations. A more formal discussion of this situation will take place later when the null space is defined.

Cases of \( I > J \) can be split into two distinctly different classes of computational problems

1. **Redundant** equations, where one or more equations is not linearly independent from the rest;
2. **Overspecified** problems, where all of the equations are linearly independent and no unique solution exists.

The distinction between these cases is important for understanding the meaning of the modeling equations and for selecting the proper numerical solution procedure.

The basic matrix operations used in the Gaussian elimination procedure are valid for the case where more equations exist than unknowns and are the key to a rigorous test for distinguishing between these cases. Consider, for example, if we use our correct approximate solution \( y = [x_M x_O y_M y_O]^T \approx [0 \ 1 \ 0.5 \ 0.5]^T \) to compute the coefficients of \( B \),

\[
Bz = \begin{bmatrix}
1 & 1 & -1 \\
0.5 & 0 & -0.1 \\
0 & 0 & 0.1 \\
0.5 & 1 & -0.9 \\
\end{bmatrix}
\begin{bmatrix}
V \\
L \\
F \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
10 \\
0 \\
\end{bmatrix}
= c
\]

Proceeding with the elimination procedure as before, we find

\[
\begin{bmatrix}
1 & 1 & -1 \\
0 & -0.5 & 0.4 \\
0 & 0 & 0.1 \\
0 & 0.5 & -0.4 \\
\end{bmatrix}
\begin{bmatrix}
z \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
10 \\
0 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 1 & -1 \\
0 & -0.5 & 0.4 \\
0 & 0 & 0.1 \\
0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
z \\
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
10 \\
0 \\
\end{bmatrix}
\]  \hspace{1cm} (1.10)

which is easily solved to find

\[
F = 100\text{moles/hr} \quad L = 80\text{moles/hr} \quad V = 100 - 80 = 20\text{moles/hr}.
\]
To summarize,

1. Because $Q$ is orthonormal, the projection operation simply is an array product operation and so $Q^T = Q^{-1}$;

2. If $Q$ is generated from $A$, the columns of $A$ span the space defined by the orthonormal basis $Q$.

### 1.9 The QR Factorization

Put in a more abstract way, we can factor our coefficient matrix $A^I \times J$ into the product

$$A^I \times J = Q^I \times I R^I \times J$$

so that

$$QRx = b$$

$$Rx = Q^T b$$

$$Rx = d$$

where for $J = 4$

$$Rx = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \\ \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} * \\ * \\ * \\ 0 \text{ or } * \\ \vdots \end{bmatrix}$$

where the elements in the fifth and subsequent rows are zero when there are four independent equations, and are possibly nonzero when the number of independent equations exceeds the number of variables.

### 1.10 Least Squares

Perhaps the most important numerical techniques used for solving modeling equations consists of the following two steps:

1. Define a residual $R(x)$;

2. Minimize this residual by choosing appropriate values for the variables $x_i$ of the residual.

We define the residual of a set of linear equations $Ax = b$ as

$$r = Ax - b$$
noting that the residual is a vector with the same dimensions as \( b \) (an \( N \times 1 \) column vector). Minimizing a vector means minimizing its length, which is defined by the Euclidean norm:

\[
\text{length of } r = \|r\| = \sqrt{r \cdot r}
\]

Having defined the residual, we minimize the square of the norm (to make computations easier) by determining the values of \( x \) which minimize

\[
\min_x \|r\|^2 = \min_x \|Ax - b\|^2.
\]

Now,

\[
\|Ax - b\|^2 = (Ax - b)^T (Ax - b) = (Ax)^T (Ax) - 2b^T (Ax) + b^T b
\]

The derivatives with respect to the \( x_i \) will vanish at the extrema of the norm of the residual:

\[
\frac{\partial}{\partial x_i} \|Ax - b\|^2 = \frac{\partial}{\partial x_i} [(Ax)^T (Ax) - 2b^T (Ax)] = 0 \quad \text{for } i = 1, \ldots, N
\]

It is a little hard to see how this differentiation works; if we write the problem out explicitly for a \( 2 \times 2 \) system and compute the derivative w.r.t. \( x_1 \) and \( x_2 \),

\[
\frac{\partial}{\partial x_1} \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix}^T \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix} = 0
\]

so

\[
\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}^T \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix} + \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix}^T \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = 2 \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}^T \begin{bmatrix} A_{11}x_1 + A_{12}x_2 \\ A_{21}x_1 + A_{22}x_2 \end{bmatrix} - 2 \begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}^T \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = 0
\]

Computing the derivative with respect to \( x_2 \) reveals a similar structure, the only difference being in the transposed row; this means we can write the results of the minimization simply as determining the solution to

\[
A^T Ax = A^T b
\]

We note that the product of \( A^T A \) gives a square, symmetric array of the same length as \( x \).

Returning to our over-determined problem, we find

\[
B^T B = \begin{bmatrix} 1.5 & 1.5 & -1.5 \\ 1.5 & 2 & -1.9 \\ -1.5 & -1.9 & 1.83 \end{bmatrix} \quad B^T c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}
\]

It is left as an exercise for the reader to show that this system is equivalent to (1.10).
1.10.1 Computational example: Fitting a linear equation to experimental data

It has been observed that the temperature of a lamp-heated wafer in a chemical vapor deposition reactor can be a strong function of gas temperature when the lamp power and all other factors are kept constant [48]. In the cited tungsten CVD reactor system study, the following dependency was measured:

<table>
<thead>
<tr>
<th>H$_2$ mole fraction (x)</th>
<th>1</th>
<th>0.8</th>
<th>0.6</th>
<th>0.4</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>N$_2$ mole fraction</td>
<td>0</td>
<td>0.2</td>
<td>0.4</td>
<td>0.6</td>
<td>1</td>
</tr>
<tr>
<td>Wafer temperature K (T)</td>
<td>553.3</td>
<td>577.3</td>
<td>602.3</td>
<td>626.7</td>
<td>672.7</td>
</tr>
</tbody>
</table>

We wish to fit these data to the following linear equation:

$$T = a_1 + a_2 x$$

using a least squares procedure.

```matlab
x = [1; 0.8; 0.6; 0.4; 0];
A = [ones(5,1) x];
T = [553.3; 577.3; 602.3; 626.7; 672.7];

a = (A'*A) \ (A'*b)  % use direct definition of least squares
a = A \ b            % produces the same result
```

The results are plotted in Fig. 1.4, illustrating the good fit achieved using the linear interpolation. The large temperature variation with gas composition is attributed to the much higher thermal conductivity of H$_2$ relative to N$_2$.

![Figure 1.4: Experimental data plotted with the linear interpolation.](image-url)
Chapter 2

Linear ODE Methods

Recall that several of the models derived in the previous Chapter had the form
\[
\frac{dx}{dt} = Ax - b.
\]
It is possible to solve linear, time-invariant vector ODE problems such as these to find a general solution; additional information in the form of a boundary (initial or otherwise) is needed to find a particular solution. For the CVD reactor example discussed in the previous Chapter,
\[
x(t = 0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\]
is a valid initial condition and corresponds to the chamber condition prior to introducing reactant gases. For the multicomponent absorber, \(x(t = 0) = y(t = 0) = 0\) at startup.

We now develop several additional ODE models because the steady-state problem either does not exist in this context, or requires solution of the ODE model to be found.

2.1 Quenching Dynamics

Consider a partially insulated tank with \(V_w\) m\(^3\) of water into which a steel sphere of volume \(V_s\) is dropped. The tank walls are in contact with a coolant maintained at 0\(^\circ\) C. If the heat transfer coefficient between the sphere and water is \(h\) J/(m\(^2\) s K), surface area of the sphere \(A\) m\(^2\), heat capacity of steel and water is \(C_{p_s}\) and \(C_{p_w}\), respectively, and density of each is \(\rho_s\) and \(\rho_w\), heat transfer coefficient between the tank and coolant \(h_c\) and heat transfer area \(A_c\), and defining
\[
\alpha_s = \frac{hA}{\rho_s V_s C_{p_s}} \quad \alpha_w = \frac{hA}{\rho_w V_w C_{p_w}} \quad \alpha_c = \frac{h_c A_c}{\rho_w V_w C_{p_w}}
\]
we can write
\[
\begin{align*}
\frac{dT_s}{dt} &= \alpha_s (T_w - T_s) \quad (2.1) \\
\frac{dT_w}{dt} &= \alpha_w (T_s - T_w) - \alpha_c T_w \quad (2.2)
\end{align*}
\]
subject to the initial conditions

\[ T_s(t = 0) = 80^\circ C \quad T_w(t = 0) = 20^\circ C. \]  

(2.3)

In matrix form,

\[ \frac{d}{dt}x = \frac{d}{dt} \begin{bmatrix} T_s \\ T_w \end{bmatrix} \begin{bmatrix} -\alpha_s & \alpha_s \\ \alpha_w & -\alpha_w - \alpha_c \end{bmatrix} x \]

Figure 2.1: Water tank and steel sphere of the quenching dynamics problem.

2.2 Tubular Reactor Boundary-Value Problem

Consider a system where a reacting species travels down a long tube (of length \( L \) m), transported by a combination of diffusion and convection due to the bulk fluid velocity \( v \) (in m/sec). Assuming the reactant concentration \( C \) (in moles/m\(^3\)) is a function of only the axial position and time, we can write the reactant material balance over a differential element as

\[ A(J|_z - J|_{z+\Delta z}) = A\Delta z R_c \]

with the reactant flux \( J \) (in mole/m\(^2\) sec) written as

\[ J = -D \frac{dC}{dz} + vC \]

and the reactant consumption rate per unit volume (moles/(sec m\(^3\))) term as proportional only to reactant concentration

\[ R_c = k_c C. \]

This model is based on the assumptions of a homogeneous, first-order, isothermal reaction (the rate constant \( k_c \) has units sec\(^{-1}\)). By taking \( \Delta z \to 0 \) and combining terms, we find

\[ 0 = \frac{d}{dz} (-J) - k_c C \]

\[ = D \frac{d^2 C}{dz^2} - v \frac{dC}{dz} - k_c C \]

The boundary conditions for both ends of the reactor are derived from the continuity of reactant flux across the boundaries. At the reactor inlet \( z = 0 \), the total flux must equal the reactant feed rate, so

\[ vC_{in} = J|_{z=0} \]

\[ vC_{in} = -D \frac{dC}{dz} \bigg|_{z=0} + vC|_{z=0} \]
2.2. TUBULAR REACTOR BOUNDARY-VALUE PROBLEM

so

\[ -D \frac{dC}{dz} = v(C_{in} - C) \quad \text{at } z = 0. \]

Defining the flux as constant at the reactor outlet,

\[ D \frac{dC}{dz} \bigg|_{z=L-e} + vC \bigg|_{z=L-e} = D \frac{dC}{dz} \bigg|_{z=L+e} + vC \bigg|_{z=L+e} \]

Without any other information on reactant transport beyond the reactor exit, it is impossible to simplify this equation. Under the assumption that there are no concentration gradients for \( z > 1 \),

\[ \frac{dC}{dz} = 0 \quad \text{at } z = 1 \]

because physically unrealistic concentration differences result when the derivative is not zero.

The modeling equations can be put in dimensionless form with the following definitions

\[ \xi = \frac{z}{L}, \quad x = \frac{C_{in} - C}{C_{in}}, \quad P_e = \frac{Lv}{D}, \quad K = \frac{k_c L}{v} \]

to find the nonhomogeneous ODE model

\[ 0 = 1 \frac{d^2x}{P_e d\xi^2} - \frac{dx}{d\xi} + K(1 - x) \]

subject to the homogeneous boundary conditions

\[ \frac{1}{P_e} \frac{dx}{d\xi} = x \quad \text{at } \xi = 0 \]
\[ \frac{dx}{d\xi} = 0 \quad \text{at } \xi = 1 \]

If we split the second-order differential equation into the following two first-order ODEs

\[ \frac{dx}{d\xi} = y \]
\[ \frac{dy}{d\xi} = P_e (y + Kx - K) \]

we can write the problem in matrix notation

\[ \frac{dw}{d\xi} = \begin{bmatrix} 0 & 1 \\ KP_e & P_e \end{bmatrix} w - \begin{bmatrix} 0 \\ KP_e \end{bmatrix} = Aw - b. \]

subject to boundary conditions

\[ y = P_e x \quad \text{at } \xi = 0 \]
\[ y = 0 \quad \xi = 1 \]
2.3 Tank Cascade Control

Consider the two well-stirred mixing tanks arranged in the sequential cascade in Fig. 2.2.

\[
\begin{align*}
\frac{dx_1}{dt} &= \beta_1(x_f - x_1) \\
\frac{dx_2}{dt} &= \beta_2(x_1 - x_2) \\
\frac{dx_f}{dt} &= -k_p\beta_2(x_1 - x_2) + k_1(x_S - x_2)
\end{align*}
\]

These modeling equations can be written in matrix form,

\[
\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_f \end{bmatrix} = \begin{bmatrix} -\beta_1 & 0 & \beta_1 \\ \beta_2 & -\beta_2 & 0 \\ -k_p\beta_2 & k_p\beta_2 - k_1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_f \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ -k_1x_S \end{bmatrix}
\]

subject to initial conditions

\[
x(t = 0) = \begin{bmatrix} 1 \\ 0 \\ k_p x_S \end{bmatrix}.
\]

It can be shown that the steady-state solution must be \( x_1 = x_2 = x_f = x_S \) for \( k_1 > 0 \).

2.4 Steady State Solution of the Quenching Problem

It is straightforward to prove the only steady-state solution to (2.1)-(2.2), written as,

\[
\begin{bmatrix} -\alpha_s & \alpha_s \\ \alpha_w & -\alpha_w - \alpha_c \end{bmatrix} \begin{bmatrix} T_s \\ T_w \end{bmatrix} = 0
\]

is \( T_s = T_w = 0 \) for \( \alpha_c > 0 \); however, for \( \alpha_c = 0 \) it appears there is no unique solution because of the linear dependence of each. This means a particular, steady-state solution for this model depends entirely on the initial conditions of the problem, a characteristic of closed systems. Equilibrium solutions to open systems such as the flash drum and CVD CSTR, on the other hand, normally will consist only
of a finite number of solutions, and may depend on initial conditions in the case of multistability for nonlinear systems.

One piece of information we have not explicitly made use of is found by adding the two differential equations:

\[ \frac{d}{dt} \left[ \frac{T_s}{\alpha_s} + \frac{T_w}{\alpha_w} \right] = 0 \]
\[ \alpha_w T_s + \alpha_s T_w = \delta \quad \text{(a constant)} \]
\[ \alpha_w T_s + \alpha_s T_w = \frac{\alpha_w T_s(t = 0) + \alpha_s T_w(t = 0)}{\alpha_s} \]

Substituting (2.5) into (2.1) gives

\[ \frac{dT_s}{dt} = \alpha_w T_s(t = 0) + \alpha_s T_w(t = 0) - \alpha_w T_s - \alpha_s T_s \]
\[ \frac{d\gamma T_s + \delta}{dt} = \gamma T_s + \delta \quad (2.6) \]

where \( \gamma = -\alpha_w - \alpha_s \) and so is always negative and has a magnitude that increases with decreasing steel sphere total heat capacity \( (\rho_s V_s C_p_s) \).

### 2.5 Scalar Ordinary Differential Equations

The analysis leading to (2.6) indicates that (2.1)-(2.2) only has a single dynamic degree of freedom. This linear, scalar, time-invariant ODE is solved easily by first transforming the nonhomogeneous problem (2.6) to homogeneous form by using the variable transformation

\[ y = \gamma T_s + \delta \]
\[ \frac{dy}{dt} = \gamma \frac{dT_s}{dt} = \gamma [\gamma T_s + \delta] = \gamma y \]

We will solve this problem with the three-step procedure:

1. Guess that the form of a solution consists of the product of a time-dependent function and a constant;
2. Substitute this guess into the differential equation and determine a general solution by minimizing the residual;
3. Compute a particular solution which satisfies the initial conditions.

While this may first appear to turn a simple problem into a more elaborate procedure, we will find this procedure to be applicable to all differential equation models to follow, including the numerical procedures for finding approximate solutions to nonlinear partial differential equation models.
Thus, as step 1, we try
\[ y = e^{\lambda t}A \]  
(2.7)
as a solution. Step 2 requires substituting this into (2.6):
\[
\begin{align*}
\frac{d}{dt}(e^{\lambda t}A) &= \gamma e^{\lambda t}A \\
\lambda e^{\lambda t}A &= \gamma e^{\lambda t}A \\
\lambda &= \gamma.
\end{align*}
\]

The parameter \( \lambda \) is called an eigenvalue. Thus, the general solution satisfies the ODE of (2.6), and can be written as
\[ y = e^{\gamma t}A. \]
The particular solution satisfies both the ODE and the initial condition:
\[ y_0 = e^{\gamma \times 0}A \quad \implies \quad A = y_0 \]
so the particular solution is
\[ y = y_0 e^{\gamma t}. \]
\[ \gamma T_s(t) + \delta = e^{\gamma t} \begin{bmatrix} \gamma T_s(t = 0) + \delta \end{bmatrix} \\
T_s(t) = -\frac{\delta}{\gamma} + \begin{bmatrix} T_s(t = 0) + \delta \end{bmatrix} e^{\gamma t} \]
(2.8)
This solution makes physical sense, because at \( t = 0 \) \( T_s = T_s(t = 0) \) and as \( t \to \infty \) \( \alpha_w T_s + \alpha_s T_s = \alpha_w T_s(t = 0) + \alpha_s T_s(t = 0) \) which implies \( T_s = T_w \) and that energy is conserved.

### 2.6 Spectral Decomposition

If we now consider the problem of determining the solution to the full set of ODEs (2.1)-(2.2) subject to the initial conditions (2.3), we can pose a guess for the solution as
\[ z = e^{\Lambda t}a \]  
(2.9)
which is similar in form to (2.7) except for the more complicated exponential function in time. Thinking back to our array factorization techniques, if we can find an array \( U \) such that
\[ A = U \Lambda U^{-1} = U \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} U^{-1} \quad \text{so} \quad \Lambda = U^{-1} A U, \]
we could rewrite (2.9) using the following definition for the array exponential function:
\[ z = U e^{\Lambda t} U^{-1} a. \]  
(2.10)

**Proof** follows [13]: we first define a new set of state variables which are a combination of the original states \( z \)
\[ q = U^{-1} z \]
and substitute it into (2.1)-(2.2) and (2.3):

\[\dot{U}q = AUq, \quad q_0 = U^{-1}z_0\]

\[\dot{q}_1 = U^{-1}AUq\]

\[= \Lambda q\]

The last equation is so important, we write it out explicitly and will return to it in a later discussion:

\[\frac{dq_1}{dt} = \lambda_1 q_1\]  \hspace{1cm} (2.11)

\[\frac{dq_2}{dt} = \lambda_2 q_2\]  \hspace{1cm} (2.12)

We have solved this problem before (2.6); the solution is similar to (2.8). Thus, we can write

\[q = \begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{bmatrix} q_0\]

\[U^{-1}z = e^{\Lambda t} U^{-1}z_0\]

\[z = U e^{\Lambda t} U^{-1}z_0\]

so ultimately we find

\[a = z_0.\]

Substituting this guess into the set of differential equations (2.1)-(2.2),

\[\frac{d}{dt} \left( U e^{\Lambda t} U^{-1} a \right) = AU e^{\Lambda t} U^{-1} a\]

\[U \Lambda e^{\Lambda t} U^{-1} a = AU e^{\Lambda t} U^{-1} a\]

\[U \Lambda = AU\]

or

\[Au_1 = \lambda_1 u_1, \quad Au_2 = \lambda_2 u_2\]

where \(u_i\) are the column vectors of \(U\). This concludes the proof.

### 2.6.1 Sample eigenvalue/vector calculations

As part of our procedure for computing a general solution to the set of ODEs, we must solve the eigenvalue problem

\[Au = \lambda u \quad \text{or} \quad (A - \lambda I)u = 0.\]

Writing this out for our specific problem,

\[\begin{bmatrix}
-\alpha_s - \lambda & \alpha_s \\
\alpha_w & -\alpha_w - \alpha_c - \lambda
\end{bmatrix} u = 0 \quad (2.13)
\]

\[\begin{bmatrix}
-\alpha_s - \lambda & \alpha_s \\
0 & -\alpha_w - \alpha_c - \lambda + \alpha_s \alpha_w / (\alpha_s + \lambda)
\end{bmatrix} u = 0.\]
These equations are linearly dependent when

\[
0 = -\alpha_w - \alpha_c - \lambda + \frac{\alpha_s \alpha_w}{\alpha_s + \lambda} \\
= -(\alpha_s + \alpha_c + \lambda)(\alpha_w + \lambda) + \alpha_s \alpha_w \\
= \lambda^2 + (\alpha_s + \alpha_w + \alpha_c)\lambda + \alpha_s \alpha_w
\]

so

\[
\lambda = \frac{-(\alpha_s + \alpha_w + \alpha_c) \pm \sqrt{(\alpha_s + \alpha_w + \alpha_c)^2 - 4\alpha_c \alpha_w}}{2}
\]

**Case** \(\alpha_c = 0\): When \(\alpha_c = 0\)

\[\lambda_{1,2} = 0, -\alpha_s - \alpha_w.\]

Substituting the eigenvalues \(\lambda_i\) into (2.13), we can determine the corresponding eigenvectors \(u_i\):

\[
\begin{bmatrix}
-\alpha_s & \alpha_s \\
\alpha_w & -\alpha_w
\end{bmatrix}
\begin{bmatrix}
1 \\
1
\end{bmatrix}
= 0
\]

\[
u_1 = c_1 \begin{bmatrix}
1 \\
1
\end{bmatrix}
\text{ corresponding to } \lambda_1 = 0
\]

\[
\begin{bmatrix}
\alpha_w & \alpha_s \\
\alpha_w & \alpha_s
\end{bmatrix}
\begin{bmatrix}
1 \\
1
\end{bmatrix}
= 0
\]

\[
u_2 = c_2 \begin{bmatrix}
1 \\
-\alpha_w/\alpha_s
\end{bmatrix}
\text{ corresponding to } \lambda_2 = -\alpha_s - \alpha_w
\]

The constants \(c_1, c_2\) can be anything including zero: we can write the results in array form:

\[
V = \begin{bmatrix}
1 & 1 \\
1 & -\alpha_w/\alpha_s
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2
\end{bmatrix}
= UC.
\]

and substituting these eigenvectors into the solution using \(q = V^{-1}z\),

\[
q = e^{\Lambda t} q_0,
\]

\[
z = UC e^{\Lambda t} V^{-1}z_0
\]

\[
= UC e^{\Lambda t} C^{-1} U^{-1} z_0
\]

\[
= U e^{\Lambda t} U^{-1} z_0 \quad \text{because } C \text{ and } e^{\Lambda t} \text{ are diagonal arrays.}
\]

Therefore, the solution does not depend on the values \(c_1\) and \(c_2\).

### 2.6.2 Physical interpretation of eigenvectors

We continue our practice of avoiding matrix inversion, and so actually solve the system

\[
\begin{bmatrix}
1 & 1 \\
1 & -\alpha_w/\alpha_s \\
1 & 1 \\
0 & -\alpha_w/\alpha_s - 1
\end{bmatrix}d = \begin{bmatrix}
T_s(0) \\
T_w(0) \\
T_s(0) \\
T_w(0) - T_s(0)
\end{bmatrix}
\]
2.6. SPECTRAL DECOMPOSITION

\[ d = \begin{bmatrix} \frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} + T_s(0) \\ -\frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} \end{bmatrix} \]

This gives a clear view into the modal structure of the particular solution:

1. At \( t = 0 \), we decompose the initial condition column vector into some linear combination of the (column) eigenvectors (i.e., we project the initial condition vector onto the eigenvectors):
   \[ z_0 = d_1 u_1 + d_2 u_2; \]

2. The solution then evolves as the lengths of the individual vectors \( d_j u_j \) shrink or grow, independently of one-another (so we see that if our solution falls directly on one of the eigenvectors, it stays on that eigenvector for all time):
   \[ z = d_1 e^{\lambda_1 t} u_1 + d_2 e^{\lambda_2 t} u_2. \]

This geometric interpretation is shown in Fig. 2.3. For our system, we write the solution out explicitly as

\[
\begin{bmatrix} T_s \\ T_w \end{bmatrix} = \begin{bmatrix} \frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} + T_s(0) \\ -\frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -\frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} \\ \frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} \end{bmatrix} e^{-(\alpha_s + \alpha_w)t} \begin{bmatrix} 1 \\ -\alpha_w/\alpha_s \end{bmatrix}
\]

Notice how the zero eigenvalue means that solutions evolve parallel to the other (“nonzero”) eigenvector and ultimately come to fall on the eigenvector associated with the zero eigenvalues. Inspection of the coefficients of the ODE solution shows that

\[ T_s, T_w \to \begin{bmatrix} \frac{\alpha_s(T_w(0) - T_s(0))}{\alpha_s + \alpha_w} + T_s(0) \end{bmatrix} \text{ as } t \to \infty. \]

These results corroborate with our previous analysis of the steady-state, suggesting the vector \( d_1 u_1 \) can be thought of as the total energy of the system.

Case \( \alpha_c > 0 \):

2.6.3 Symmetric systems

We notice that if

\[ \alpha_s = \alpha_w \]

the coefficient array \( A \) becomes symmetric. This is an important special case because if we compute the eigenvectors we find (for \( \alpha_c = 0 \))

\[ U = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

and if we compute the inner product of the two eigenvectors

\[ u_1^T u_2 = 1 - 1 = 0 \implies u_1 \text{ and } u_2 \text{ are orthogonal.} \]

This is, in fact, one of the general properties of symmetric systems. Proof follows [9], if we consider the two eigenvalue problems where \( \lambda_1 \) and \( \lambda_2 \) are distinct eigenvalues of \( A = A^T \):

\[ A u_1 = \lambda_1 u_1 \quad A u_2 = \lambda_2 u_2 \]
so

\begin{align*}
(Au_1)^T &= \lambda_1 u_1^T \\
u_1^T A &= \lambda_1 u_1^T \\
u_1^T Au_2 &= \lambda_1 u_1^T u_2
\end{align*}

where the last equation was obtained from postmultiplying the second by $u_2$. Premultiplying the second equation of (2.14) by $u_1^T$ we find

\[u_1^T Au_2 = \lambda_2 u_1^T u_2\]

and subtracting this from (2.15) gives

\[0 = (\lambda_1 - \lambda_2)u_1^T u_2\]

which implies that $u_1$ and $u_2$ are orthogonal for $\lambda_1 \neq \lambda_2$:

\[u_1 \cdot u_2 = 0.\]

Another important property of the eigenvectors generated from symmetric coefficient arrays $A$ is demonstrated by first writing the array of column eigenvectors as

\[U = \begin{bmatrix} u_1 & u_2 & \ldots & u_N \end{bmatrix}\]

and so

\[U^T U = \begin{bmatrix}
    u_1 \cdot u_1 & u_1 \cdot u_2 & u_1 \cdot u_3 & \ldots & u_1 \cdot u_N \\
    u_2 \cdot u_1 & u_2 \cdot u_2 & u_2 \cdot u_3 & \ldots & u_2 \cdot u_N \\
    u_3 \cdot u_1 & u_3 \cdot u_2 & u_3 \cdot u_3 & \ldots & u_3 \cdot u_N \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    u_N \cdot u_1 & u_N \cdot u_2 & u_N \cdot u_3 & \ldots & u_N \cdot u_N
\end{bmatrix}\]
2.6. SPECTRAL DECOMPOSITION

\[
\begin{bmatrix}
u_1 \cdot u_1 & 0 & 0 \\
0 & u_2 \cdot u_2 & 0 \\
& & \ddots \\
& & & u_N \cdot u_N
\end{bmatrix}
\]

If \(u_j \cdot u_j = 1\) for \(j = 1, \ldots, N\),

\[U^T U = I.\]

If the last condition \((u_j \cdot u_j = 1)\) holds, the eigenvectors form an orthonormal set of vectors and

\[U^T = U^{-1}.\]

This simplifies evaluating the vector of coefficients \(d\) because

\[d = U^T z_0\]

and so the initial conditions are projected directly onto the eigenvectors – in other words, we can calculate the \(d_j\) as being the lengths along the \(u_j\) directly.

2.6.4 Unsymmetric systems

In cases where the matrix \(A\) is unsymmetric \((A \neq A^T)\), if we compute the eigenvalues of \(A\) and \(A^T\), we find that the eigenvalues are identical. If the eigenvalues \(\lambda_1\) and \(\lambda_2\) are distinct,

\[Au_1 = \lambda_1 u_1 \quad A^Tv_2 = \lambda_2 v_2\]  (2.16)

so

\[(Au_1)^T = \lambda_1 u_1^T\]

\[u_1^T A^T = \lambda_1 u_1^T\]

\[u_1^T A^T v_2 = \lambda_1 u_1^T v_2\]  (2.17)

where the last equation was obtained from postmultiplying the second by \(v_2\). Premultiplying the second equation of (2.16) by \(u_1^T\) we find

\[u_1^T A^T v_2 = \lambda_2 u_1^T v_2\]

and subtracting this from (2.17) gives

\[0 = (\lambda_1 - \lambda_2) u_1^T v_2\]

which implies that \(u_1\) and \(v_2\) are orthogonal for \(\lambda_1 \neq \lambda_2:\)

\[u_1 \cdot v_2 = 0.\]

This means the factorization which diagonalizes \(A\) is

\[U^{-1} AU = V^T AU = \Lambda.\]
2.6.5 Complex eigenvalues

For cases of complex eigenvalues \( \lambda_j = \alpha + i\beta, \lambda_{j+1} = \alpha - i\beta \) and the associated eigenvectors \( u_j = u_R^j + iu_I^j, u_{j+1} = u_R^j - iu_I^j \), the solution can be written in the form

\[
z = \ldots + d_j e^{\alpha t} \left[ u_R^j \cos(\beta t) - u_I^j \sin(\beta t) \right] + d_{j+1} e^{\alpha t} \left[ u_R^{j+1} \cos(\beta t) + u_I^{j+1} \sin(\beta t) \right] + \ldots
\]

and so the solution has an oscillatory component with frequency \( \beta \) and an amplitude growth rate governed by \( \alpha \); the eigenspace associated with this oscillatory mode is defined by \( \text{span}\{u_R^j, u_I^j\} \). See [8] for more information.

2.6.6 Summary of solutions to sets of linear ODEs

Given the general set of \( N \) independent linear equations

\[
\frac{dz}{dt} = Az
\]

subject to initial conditions \( z(t = 0) = z_0 \), we can write the solution as

\[
z(t) = U e^{A t} U^{-1} z_0 \quad \text{for any } A
\]

\[
z(t) = U e^{A t} U^T z_0 \quad \text{for symmetric } A
\]

where \( A \) is a diagonal array of eigenvalues and \( U \) is the array of associated eigenvectors.

Solutions to the nonhomogeneous system

\[
\frac{dz}{dt} = Az - b
\]

where \( b \) is a vector of constants the same size as \( z \), subject to initial conditions \( z(t = 0) = z_0 \), can be computed using a change of variables. Defining \( s = Az - b \) and substituting it into (2.18), we find a homogeneous set of ordinary differential equations subject to the transformed initial conditions \( s_0 \). Solving this problem and transforming back to \( z \), the solution, after several simplifying steps, can be written as

\[
z(t) = U e^{A t} U^{-1} z_0 - A^{-1} \left[ U e^{A t} U^{-1} - I \right] b \quad \text{for any } A
\]

\[
z(t) = U e^{A t} U^T z_0 - A^{-1} \left[ U e^{A t} U^T - I \right] b \quad \text{for symmetric } A
\]

where \( I \) is the identity matrix. Note that the steady state solution can be computed as \( z_{\text{stst}} = A^{-1} b \).

2.6.7 Computational example: CVD reactor dynamics

Using the coefficient array \( A \) and nonhomogeneous array \( b \) used for computing the steady-state solution of the CVD problem, we can compute the solution to the dynamic problem using the spectral factorization technique of the lodeSolve.m method:

\[
t = [0:1:60];
y0 = [0; 0; 1];
y = lodeSolve(A, t, y0, b);
\]
2.7. MIXED ODE AND AE SYSTEMS

Figure 2.4: Dynamic behavior of the CVD reactor during the first minute of start-up operation.

2.6.8 Computational example: Complex eigenvalues

\[
\beta_1 = 1; \\
\beta_2 = 1; \\
k_P = 0.1; \\
k_I = 1; \\
x_S = 0.5; \ % \text{setpoint}
\]

\[
\% \text{Initial condition based on } x_1=x_2=0 \text{ at } t=0 \\
x_f = k_P \cdot (x_S-0); \\
x_0 = [1; 0; x_f];
\]

\[
A = [-\beta_1 \ 0 \ \beta_1 ; \ \beta_2 \ -\beta_2 \ 0 ; \ -k_P\beta_2 \ k_P\beta_2-k_I \ 0] \\
b = [0; 0; -k_I \times x_S]
\]

\[
t = [0:0.1:50];
\]

\[
x = \text{lodeSolve}(A, t, x_0, b);
\]

2.7 Mixed ODE and AE systems

2.8 Boundary Value Problems

Returning to the tubular reactor boundary-value problem, choosing parameter values \( P_e = 4 \) and \( K = 5/4 \)
gives the nonhomogeneous set of ordinary differential equations

\[
\frac{dw}{d\xi} = \begin{bmatrix} 0 & 1 \\ 5 & 4 \end{bmatrix} w - \begin{bmatrix} 0 \\ 5 \end{bmatrix} = Aw - b.
\]
subject to boundary conditions

\[ y = 4x \text{ at } \xi = 0 \quad \text{and} \quad y = 0 \text{ at } \xi = 1. \]

The eigenvalues of \( A \) can be computed to find \( \lambda_{1,2} = 5, -1 \) and their associated eigenvectors

\[ u_1 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \]

This means the solution takes the form

\[ w = d_1 e^{5 \xi} \begin{bmatrix} 1 \\ 5 \end{bmatrix} + d_2 e^{-\xi} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \]

To determine \( d \), we solve for the coefficients using the boundary conditions. At \( \xi = 0 \) we find \( 5d_1 - d_2 = 4d_1 + 4d_2 + 4 \) and at \( \xi = 1 \) we find \( 5d_1 e^5 - d_2 e^{-1} = 0 \); putting this in matrix form

\[ \begin{bmatrix} 1 \\ 5e^5 \\ -5 \\ -e^{-1} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}. \]

It is interesting to solve this system by hand using Gaussian Elimination because it shows the importance of pivoting. Exchanging the two rows, performing the single elimination step, and eliminating small terms gives

\[ \begin{bmatrix} 5e^5 \\ 0 \\ -e^{-1} \\ -5 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}. \]

which gives

\[ d_2 = -\frac{4}{5}, \quad d_1 = -\frac{4}{25} e^{-6}. \]

from back-substitution. We now compare this solution to that which is obtained without the pivoting operation; carrying out the elimination procedure and neglecting relatively small terms gives

\[ \begin{bmatrix} 1 \\ 0 \\ -5 \\ 25e^5 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 4 \\ -20e^5 \end{bmatrix}. \]
which gives

\[ d_2 = -\frac{4}{5}, \quad d_1 = 0. \]

While this solution is numerically close to the solution obtained with pivoting, they are qualitatively different in that the second does not satisfy the outlet boundary condition.

We now write the exact solution

\[ \begin{bmatrix} x \\ y \end{bmatrix} = -\frac{4e^{-6}}{25} e^{\xi} \begin{bmatrix} 1 \\ 5 \end{bmatrix} - \frac{4}{5} e^{-1} \xi \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \]

The solution is plotted in Fig. 2.6.

### 2.8.1 The shooting technique

As one potential numerical technique, which can be extended to nonlinear and other, more complicated systems, we can integrate both equations starting from \( \xi = 0 \) by guessing a value for \( x(\xi = 0) = x_0 \) and computing \( y(\xi = 0) = y_0 \) from

\[ y_0 = P_e(x_0 - 1). \]

If, after integrating to the other boundary at \( \xi = 1 \) we find that \( y(\xi = 1) = y_1 = 0 \), our first guess for \( x_0 \) was correct and we have the solution – we will most likely not be so lucky. We can, however, refine our guess by defining and linearizing the residual

\[ R = 0 = y_1 - 0 \]

\[ \approx y^n_1 - 0 + \frac{\partial y^n_1}{\partial x_0}(x^{n+1}_0 - x^n_0) + \frac{\partial y^n_1}{\partial y_0}(y^{n+1}_0 - y^n_0) \]

\[ = y^n_1 + \frac{\partial y^n_1}{\partial x_0}(x^{n+1}_0 - x^n_0) + \frac{\partial y^n_1}{\partial y_0}[P_e(x^{n+1}_0 - 1) - P_e(x^n_0 - 1)]. \]

Setting the residual to zero and solving for the next guess for \( x_0 \) gives the Newton iteration procedure

\[ x^{n+1}_0 = x^n_0 - \frac{y^n_1}{\frac{\partial y^n_1}{\partial x_0} + P_e \frac{\partial y^n_1}{\partial y_0}}, \quad (2.19) \]

The last element of the iteration scheme is a method for computing the sensitivity of the states \( x \) and \( y \) with respect to the initial conditions \( x_0 \) and \( y_0 \). These derivatives, evaluated at \( \xi = 1 \), are used in (2.19). If we consider the general, possibly nonlinear, case of

\[ \frac{dx}{d\xi} = f(x, y) \]

\[ \frac{dy}{d\xi} = g(x, y) \]

with initial conditions \( x_0 \) and \( y_0 \), we can compute the sensitivity of state \( x \) with respect to its initial value at point \( \xi \) as the solution to the differential equation

\[ \frac{\partial}{\partial x_0} \frac{dx(\xi)}{d\xi} = \frac{\partial f}{\partial x} \frac{dx(\xi)}{d\xi} + \frac{\partial f}{\partial y} \frac{dy(\xi)}{d\xi}. \]
Computing the derivative of $x$ with respect to $y_0$ and computing the derivative of $y$ with respect to $x_0$ and $y_0$ will give a total of four equations. The equations can be compactly written in matrix notation

$$\frac{\partial}{\partial \xi} \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial x_0} & \frac{\partial x}{\partial y_0} \\ \frac{\partial y}{\partial x_0} & \frac{\partial y}{\partial y_0} \end{bmatrix}$$

or

$$\frac{dV(\xi)}{d\xi} = JV. \quad (2.20)$$

Because the variables $x$ and $y$ are independent variables, evaluating $V$ at $\xi = 0$ means terms such as $\frac{\partial x_0}{\partial y_0}$ will be zero, and $\frac{\partial x_0}{\partial x_0} = 1$. Thus, the derivatives required for the shooting algorithm are computed by integrating the equations (2.20) from $\xi = 0$ to $\xi = 1$ together with the two modeling ODEs, using the identity matrix $I$ as the initial condition.

![Figure 2.6: Comparison of exact solution to results computed from the shooting algorithm.](image)

### 2.9 Review Problems

1. Consider the CSTR modeling equations describing the constant-volume reactor in which the reaction $A \rightleftharpoons B \rightleftharpoons C$ takes place:

   $$\frac{dx_A}{dt} = (0.1 - x_A) - 4x_A + x_B$$
   $$\frac{dx_B}{dt} = (0 - x_B) + 4x_A - x_B - x_B$$

   (a) Write the equations in matrix form;
   (b) Compute the steady state solution(s), if any;
   (c) Compute the eigenvalues and associated eigenvectors of the coefficient array using hand-calculation methods that can be extended to higher-dimensional systems;
   (d) Normalize the eigenvectors;
   (e) Write the solution for $x_A(0) = 1$, $x_B(0) = 0$ and plot the results.