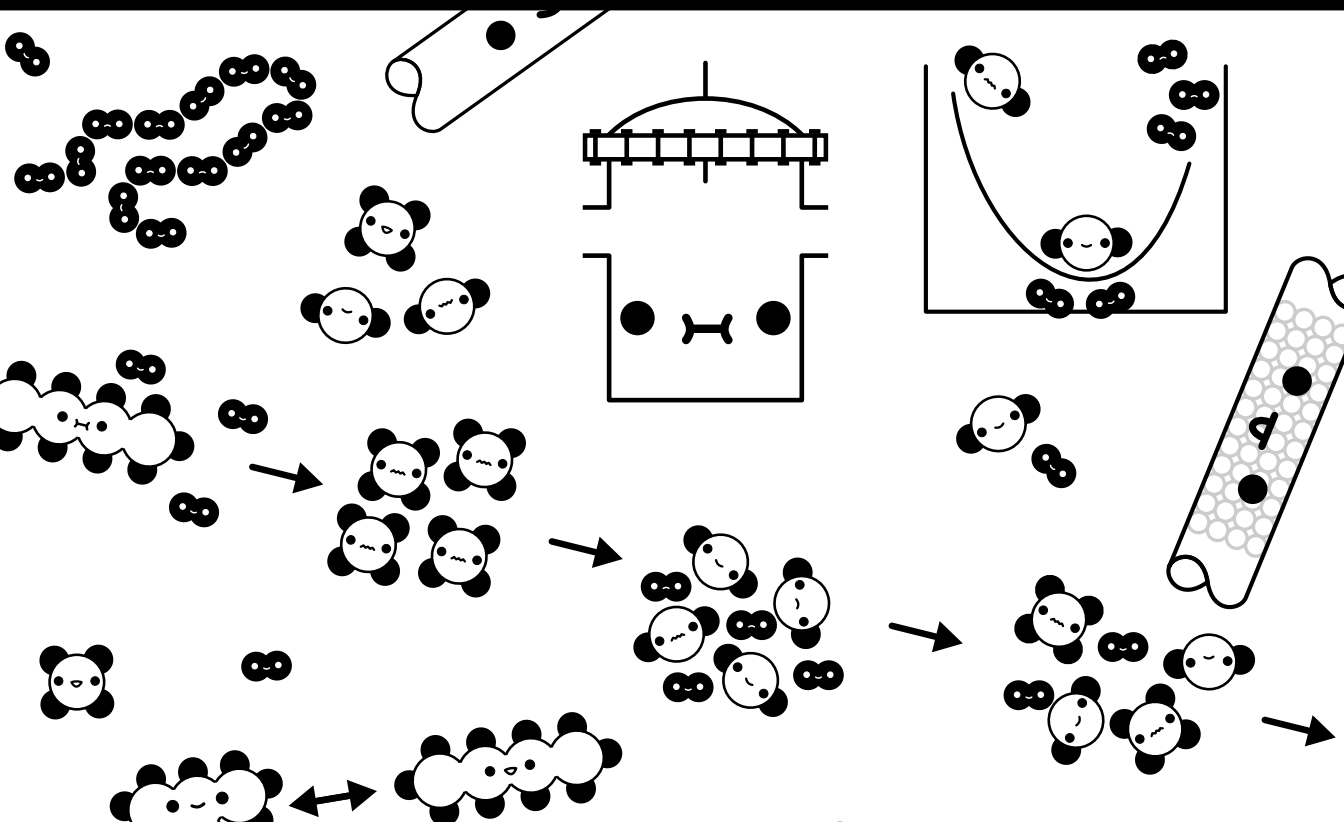


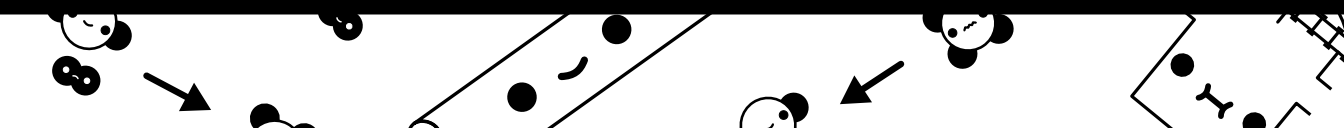
# An Introduction to Chemical Reaction Engineering with MATLAB

Version 1.1

Aaron Drews



This book can be downloaded for free,  
without purchase, at [ceng113.eng.ucsd.edu](http://ceng113.eng.ucsd.edu)





MATLAB I: Review	1
The Equilibrium Reactor	2
Rate Laws	3
The Continuously Stirred Tank Reactor, or CSTR	4
Special Cases of the CSTR	5
MATLAB II: Ordinary Differential Equations	6
The Plug Flow Reactor, or PFR	7
Special Cases of the PFR	8
The Packed Bed Reactor, or PBR	9
Special Cases of the PBR	10
Multireactor Systems	11
MATLAB III: Regression Analysis	12
Experimental Determination of Rate Laws	13
Useful Math Formulae	A
Vector Formulations	B
The Gibbs Reactor	C
Physical Properties	D



# Contents

<b>1</b>	<b>MATLAB I: Review</b>	<b>13</b>
1.1	The most common numerical arrays are scalars, vectors, and matrices . . . . .	13
1.2	Plots and subplots can be used to visualize data . . . . .	15
	<i>Example 1.1: Creating a basic plot</i> . . . . .	15
1.3	Local functions within scripts are convenient ways to organize problems . . .	18
	<i>Example 1.2: Using a local function</i> . . . . .	19
1.4	The for loop is enormously useful . . . . .	21
	<i>Example 1.3: Using a for loop</i> . . . . .	22
1.5	Root finding solves equations for unknowns . . . . .	24
	<i>Example 1.4: Finding a simple root with fsolve</i> . . . . .	26
1.6	Multiple roots must be found by trial-and-error . . . . .	27
	<i>Example 1.5: Finding multiple roots with fsolve</i> . . . . .	28
1.7	The fsolve function can accept additional input arguments . . . . .	29
	<i>Example 1.6: Passing additional arguments to fsolve</i> . . . . .	30
	<i>Example 1.7: Using fsolve within a for loop</i> . . . . .	30
1.8	The fsolve function can solve more than one equation . . . . .	31
	<i>Example 1.8: Solving multiple equations with fsolve</i> . . . . .	33
<b>2</b>	<b>The Equilibrium Reactor</b>	<b>35</b>
2.1	Reaction equilibrium is a dynamic process . . . . .	35
2.2	The Gibbs energy $G$ is minimized at equilibrium . . . . .	38
2.3	A stoichiometric table is a great organizational tool . . . . .	40
2.4	$\partial G / \partial \xi = 0$ at equilibrium . . . . .	43
2.5	The equilibrium constant $K_a$ is a function of temperature . . . . .	45
	<i>Example 2.1: Using the shortcut van't Hoff equation</i> . . . . .	47
2.6	Chemical activities are related to composition . . . . .	49
	<i>Example 2.2: Relating activity to concentration</i> . . . . .	52

2.7	The Equilibrium Reactor converts a feed to its equilibrium composition . . . .	52
	<i>Example 2.3: Finding the outlet of an Equilibrium Reactor</i> . . . . .	53
	<i>Example 2.4: Using fsolve to solve an Equilibrium Reactor</i> . . . . .	54
2.8	Equilibrium conversion is the conversion at equilibrium . . . . .	58
	<i>Example 2.5: Calculating the equilibrium conversion using <math>\xi</math></i> . . . . .	58
	<i>Example 2.6: Calculating equilibrium conversion using <math>X_A</math></i> . . . . .	59
<b>3</b>	<b>Rate Laws</b>	<b>63</b>
3.1	Rate laws must be determined experimentally . . . . .	63
3.2	The power-law form is a common rate law . . . . .	65
3.3	The units of rate laws can be tricky . . . . .	66
	<i>Example 3.1: Finding units in a rate law</i> . . . . .	68
3.4	The rate constant $k$ follows an Arrhenius form . . . . .	68
	<i>Example 3.2: Using the Arrhenius equation</i> . . . . .	69
3.5	Reactions and rate laws are sometimes called “elementary” . . . . .	71
	<i>Example 3.3: Writing an elementary rate law</i> . . . . .	73
3.6	Individual rates are summed to give net rates . . . . .	74
	<i>Example 3.4: Writing net rates</i> . . . . .	76
3.7	Rates of reversible reactions can be related to $K_a$ . . . . .	77
	<i>Example 3.5: Writing a reversible rate</i> . . . . .	80
3.7.1	For pure liquids $K_C = K_a C_T^\delta$ . . . . .	81
3.7.2	For ideal gases $K_C = K_a (P^\circ/RT)^\delta$ . . . . .	81
3.8	Reversible rates can be written in terms of partial pressures . . . . .	82
<b>4</b>	<b>The Continuously Stirred Tank Reactor, or CSTR</b>	<b>85</b>
4.1	The CSTR material balances are algebraic equations . . . . .	86
4.2	The CSTR energy balance is an algebraic equation . . . . .	89
4.3	Shaft work is negligible and heat is transferred by a heat exchanger . . . . .	94
	<i>Example 4.1: Estimating the overall heat transfer coefficient</i> . . . . .	96
4.4	CSTR problems can be solved by hand calculations . . . . .	97
	<i>Example 4.2: Isothermal CSTR sizing</i> . . . . .	97
	<i>Example 4.3: Isomerization in a CSTR</i> . . . . .	99
	<i>Example 4.4: Measuring activation energy in a CSTR</i> . . . . .	104
4.5	CSTR problems can be solved in MATLAB . . . . .	105
	<i>Example 4.5: Isothermal CSTR sizing with MATLAB</i> . . . . .	106
	<i>Example 4.6: Isomerization in a CSTR using MATLAB</i> . . . . .	108

<i>Example 4.7: Optimizing selectivity in a CSTR</i> . . . . .	110
<i>Example 4.8: Liquid-phase isomerization of xylenes</i> . . . . .	113
4.6 Solutions from fsolve should be checked for stability . . . . .	117
<i>Example 4.9: Biodiesel from soybean oil</i> . . . . .	118
<b>5 Special Cases of the CSTR</b>	<b>123</b>
5.1 A fluidized bed reactor can be modeled as a gas-phase CSTR . . . . .	123
<i>Example 5.1: Catalytic cracking of naphthalene in a fluidized bed</i> . . . . .	126
5.2 Rewriting equations in terms of conversion can be useful . . . . .	129
5.2.1 $C_i(X_A)$ for liquids . . . . .	130
5.2.2 $C_i(X_A)$ for gases . . . . .	131
<i>Example 5.2: Isomerization in a CSTR, revisited</i> . . . . .	132
<i>Example 5.3: Naphthalene cracking, revisited</i> . . . . .	133
5.3 Multiple steady states can exist in CSTRs . . . . .	135
<i>Example 5.4: Multiple steady states in a jacketed CSTR</i> . . . . .	136
5.4 Steady states can be stable, unstable, or metastable . . . . .	139
5.5 Multiple steady states lead to ignition and extinction behavior . . . . .	142
5.6 Multiple steady states occur in the laboratory and in nature . . . . .	145
5.7 Cooling ability can be dangerously overestimated . . . . .	146
<b>6 MATLAB II: Ordinary Differential Equations</b>	<b>149</b>
6.1 A difference equation can approximate a differential equation . . . . .	149
<i>Example 6.1: The finite difference approximation</i> . . . . .	150
6.2 MATLAB's ode45 function can solve differential equations . . . . .	153
<i>Example 6.2: Basic usage of ode45</i> . . . . .	154
6.3 Output points from ode45 can be made different from solution points . . . . .	155
6.3.1 Spans containing more than two values provide outputs at those values	156
6.3.2 A solution object can be returned instead of vectors . . . . .	156
6.4 The ode45 function can accept additional input parameters . . . . .	157
<i>Example 6.3: Passing an extra input parameter to ode45</i> . . . . .	158
6.5 The local ODE function can return multiple outputs . . . . .	159
6.6 The ode45 function can solve systems of ODEs . . . . .	160
<i>Example 6.4: Solving a system of coupled ODEs with ode45</i> . . . . .	163
<b>7 The Plug Flow Reactor, or PFR</b>	<b>165</b>
7.1 The PFR material balances are ordinary differential equations . . . . .	166
7.2 The PFR energy balance is an ordinary differential equation . . . . .	167

7.3	Shaft work is negligible and heat is transferred by a heat exchanger . . . . .	169
7.4	PFR problems can be solved by hand calculations . . . . .	172
	<i>Example 7.1: Isothermal PFR sizing . . . . .</i>	173
	<i>Example 7.2: Isothermal shell-and-tube PFR . . . . .</i>	175
	<i>Example 7.3: Isomerization in a PFR . . . . .</i>	177
7.5	PFR problems can be solved in MATLAB . . . . .	181
	<i>Example 7.4: Isomerization in a PFR with MATLAB . . . . .</i>	182
	<i>Example 7.5: Enzyme kinetics in a PFR . . . . .</i>	186
	<i>Example 7.6: Thermal cracking in a PFR . . . . .</i>	190
<b>8</b>	<b>Special Cases of the PFR</b>	<b>197</b>
8.1	Rewriting equations in terms of conversion can be useful . . . . .	197
	<i>Example 8.1: Isothermal PFR sizing, revisited . . . . .</i>	198
	<i>Example 8.2: Adiabatic decomposition in a PFR . . . . .</i>	200
8.2	Pressure drop occurs in PFRs . . . . .	203
	<i>Example 8.3: Pressure drop in a PFR . . . . .</i>	206
8.3	PFR jackets can be co-flow or counter-flow . . . . .	209
8.3.1	Co-flow heat exchangers can be solved like any other PFR . . . . .	210
	<i>Example 8.4: Co-flow heat exchange in a PFR . . . . .</i>	211
8.3.2	Counter-flow heat exchangers require an iterative approach . . . . .	215
	<i>Example 8.5: Counter-flow heat exchange in a PFR . . . . .</i>	217
8.4	Membrane reactors are PFRs with permeable walls . . . . .	219
	<i>Example 8.6: Dehydrogenation in a membrane reactor . . . . .</i>	221
<b>9</b>	<b>The Packed Bed Reactor, or PBR</b>	<b>225</b>
9.1	The PBR material balances are ordinary differential equations . . . . .	226
9.2	The PBR energy balance is an ordinary differential equation . . . . .	229
9.3	Shaft work is negligible and heat is transferred by a heat exchanger . . . . .	232
9.4	PBR problems can be solved by hand calculations . . . . .	233
	<i>Example 9.1: Isothermal PBR sizing . . . . .</i>	233
	<i>Example 9.2: Removal of methyl blue from wastewater . . . . .</i>	235
	<i>Example 9.3: Scale-up of a packed bed reactor . . . . .</i>	237
9.5	PBR problems can be solved in MATLAB . . . . .	240
	<i>Example 9.4: Removal of methyl blue, revisited . . . . .</i>	240
	<i>Example 9.5: Industrial production of ammonia . . . . .</i>	243
	<i>Example 9.6: Methylation of alkenes with methanol . . . . .</i>	246



<b>10 Special Cases of the PBR</b>	<b>251</b>
10.1 Rewriting equations in terms of conversion can be useful . . . . .	251
<i>Example 10.1: Isothermal PBR sizing, revisited</i> . . . . .	252
<i>Example 10.2: Catalytic cracking of used palm oil</i> . . . . .	253
10.2 Significant pressure drop can occur in PBRs . . . . .	257
10.2.1 Absolute pressure is measured relative to vacuum . . . . .	261
<i>Example 10.3: Pressure drop in a packed bed</i> . . . . .	262
<i>Example 10.4: Industrial production of ammonia, revisited</i> . . . . .	265
<b>11 Multireactor Systems</b>	<b>271</b>
11.1 Overall conversion is the conversion achieved by the entire system . . . . .	272
<i>Example 11.1: Relating single-pass to overall conversion for serial reactors</i> . . . . .	274
11.2 Under limited conditions the adiabatic energy balances simplify to one expres- sion . . . . .	275
11.3 Interstage heat exchangers can increase conversion of equilibrium-limited pro- cesses . . . . .	278
<i>Example 11.2: Water-gas shift reaction</i> . . . . .	279
<i>Example 11.3: Water-gas shift reaction, improved</i> . . . . .	284
11.3.1 Derivation of Equation (11.15) . . . . .	287
11.4 Levenspiel plots can estimate the type, arrangement, and size of multireactor systems . . . . .	289
<i>Example 11.4: Reactor selection for saponification</i> . . . . .	290
<i>Example 11.5: Reactor optimization for reversible, exothermic reactions</i> . . . . .	294
<b>12 MATLAB III: Regression Analysis</b>	<b>299</b>
12.1 A model can be fitted to data by minimizing the sum-of-squares . . . . .	299
<i>Example 12.1: Calculating the SSE</i> . . . . .	301
12.2 Linear models can be developed using the fitlm function . . . . .	302
<i>Example 12.2: Constructing the predictor matrix</i> . . . . .	304
<i>Example 12.3: Linear regression to a first-order polynomial</i> . . . . .	305
<i>Example 12.4: Linear regression without intercept</i> . . . . .	306
12.3 Several statistics should be evaluating when deciding model quality . . . . .	308
<i>Example 12.5: Choosing the best model</i> . . . . .	309
12.4 Nonlinear models can be developed using the fitnlm function . . . . .	311
<i>Example 12.6: Basic usage of fitnlm</i> . . . . .	311
<i>Example 12.7: Fitting a nonlinear model</i> . . . . .	313

<b>13 Experimental Determination of Rate Laws</b>	<b>317</b>
13.1 The batch reactor material balances are differential equations . . . . .	317
13.2 The integrated material balance can be used to fit rate expressions . . . . .	319
13.3 Classical linearization methods should be avoided . . . . .	320
13.4 $C_{A0}$ can be treated as an adjustable parameter even if its value is known . . . .	322
13.5 Determination of experimental rate laws . . . . .	323
<i>Example 13.1: Methyl orange degradation</i> . . . . .	323
<i>Example 13.2: Methyl blue degradation</i> . . . . .	327
<i>Example 13.3: Saponification of ethyl acetate</i> . . . . .	331
<b>A Useful Math Formulae</b>	<b>337</b>
<b>B Vector Formulations</b>	<b>339</b>
B.1 Variables can be expressed as vectors and matrices . . . . .	340
B.2 Vector and matrix operations greatly simplify many common calculations . .	340
B.3 CSTR equations can be written quickly and easily using vectors and matrices .	343
B.4 PFR and PBR equations can also be written using vectors and matrices . . . .	345
<b>C The Gibbs Reactor</b>	<b>347</b>
C.1 $G$ can be minimized directly for ideal gases (Gibbs Reactor) . . . . .	348
<i>Example C.1: Plotting the Gibbs free energy</i> . . . . .	350
<i>Example C.2: A simple Gibbs Reactor</i> . . . . .	352
C.2 A general Gibbs Reactor uses the fmincon function to minimize $G$ . . . . .	353
<i>Example C.3: Writing Aeq and beq</i> . . . . .	355
<i>Example C.4: Using a full Gibbs Reactor</i> . . . . .	356
C.3 The full van't Hoff expression can adjust $\Delta G_{f,i}^\circ$ for temperature . . . . .	357
<i>Example C.5: <math>\Delta G_{f,i}^\circ</math> for methane at 600 K</i> . . . . .	358
C.4 The Gibbs energy at 298.15 K can be estimated from $\Delta H_f^\circ$ and $S_f^\circ$ . . . . .	361
<b>D Physical Properties</b>	<b>363</b>
<b>Index</b>	<b>371</b>

# To my students

## Why I wrote this book

According to at least one website<sup>1</sup> the average annual in-state college tuition in California was just shy of \$14,000, and at UC San Diego the cost was a little less at about \$11,500. The cost of textbooks in particular has risen four times faster than the rate of inflation,<sup>2</sup> with students spending on average more than \$1,200 per year for books and materials. Particularly obscene is the rise of bundled “access codes” which prevent students from buying or renting used textbooks, or even accessing text materials in the future.

These ever-rising costs for materials, textbooks, and tuition have real impacts on my students, and students everywhere. The University of California<sup>3</sup> found the following in an annual survey of more than 55,000 students:

- 47% reported low or very low food security
- 43% couldn’t afford to eat balanced meals (sometimes or often true)
- 32% were hungry but didn’t eat
- Over 1,000 students were homeless at some point during the Fall-Spring academic year.

As a student I was fortunate to avoid these hardships but as an instructor I’ve watched my students struggle against them, and one of my guiding principles has been to design my courses to address these issues: I’ve created fully online courses to reduce transportation costs and free up time for family and work, I’ve reminded countless committees that lab fees can mean some students don’t eat, and I’ve worked with our bookstore to order used, international editions of quality textbooks that are a fraction of the cost of new, U.S. editions.

Unfortunately, international versions of my preferred reaction engineering textbook have become more difficult to locate, some students have bumped into academic integrity violations

---

<sup>1</sup>College Costs in California. [collegecalc.org/colleges/california](http://collegecalc.org/colleges/california). Accessed 17 Dec 2019.

<sup>2</sup>What’s behind the soaring cost of college textbooks. <https://www.cbsnews.com/news/whats-behind-the-soaring-cost-of-college-textbooks/>. Published 26 Jan 2019, accessed 17 Dec 2019.

<sup>3</sup>University of California Undergraduate Experience Survey, 2018. [universityofcalifornia.edu/infocenter/ucues-data-tables-2018](http://universityofcalifornia.edu/infocenter/ucues-data-tables-2018). Accessed 17 Dec 2019.

through use of the widely available solutions manual, and I've become increasingly less satisfied in the text itself. The text you're reading now is my attempt to address rising textbook costs in a long-term, sustainable manner and gives my students an excellent resource both during and after my course.

The physical copy of this textbook was created to be both inexpensive to produce and convenient to use. At the time of this writing its sale cost was estimated to be less than \$25, and I receive no royalties for its publication. The spiral binding small page size was chosen for both its low cost and utility: the pages lay flat, or can be turned around so that just a single page is visible—helpful on a small classroom desk or a tiny dormitory. As noted clearly on the front cover, a free PDF copy of this text is available at [ceng113.eng.ucsd.edu](http://ceng113.eng.ucsd.edu)—no purchase necessary, no access codes required, no expiration date once downloaded.

My choice of organization and nomenclature is intended to present reaction engineering as a straightforward application of prior coursework, namely material and energy balances and thermodynamics. The book begins with a review of equilibrium as a reminder that we can't beat the second law, briefly reviews rate laws, and then introduces reactors according to their mathematical familiarity: tank reactors first for their algebraic design equations, followed by tubular reactors for their differential design equations.

The very first equations in the introductory chapter for each reactor are the general design equations, the material and energy balances. To highlight their applicability, most example problems begin from these equations even when simpler formulations may be possible. Special cases of the general design equations—such as those written in terms of conversion—are explicitly noted as just that, “special cases,” because I've found students to be overwhelmed with the number of “new” equations.

Woven throughout each chapter are sections devoted to reactor design with MATLAB, supported by several chapters devoted solely to MATLAB computational techniques as well as an appendix of advanced techniques which fully utilize MATLAB's matrix capabilities. The MATLAB language was chosen for its utility and simplicity, and to help show the similarities between applications some examples are more verbose than strictly necessary.

The development of this text is ongoing; it will undoubtedly continue to grow to meet the needs of my students. All feedback and corrections are welcome on its website, [ceng113.eng.ucsd.edu](http://ceng113.eng.ucsd.edu), and I hope that my enthusiasm for the topic is made apparent as you read. I wish you the best in your studies, and welcome to chemical reaction engineering!

# Chapter 1

## MATLAB I: Review

“Your previous instruction in this subject has been, disturbingly, uneven. But you will be pleased to know, from now on, you will be following a carefully structured, Ministry-approved course of defensive magic.”

Dolores Umbridge, *Harry Potter and the Order of the Phoenix* (2003)

Most of this book assumes that you’ve already had at least a cursory introduction to MATLAB, so you should know basics such as opening the program, accessing files, creating variables, plotting functions, and other things like that. On the other hand, it’s possible that you may not have practiced these skills in quite some time, or perhaps you were never introduced to some of the functions that we’ll need regularly in the upcoming chapters of this book. For these reasons we’ll spend a little bit of time reviewing how to get MATLAB to do useful stuff, starting with a review of data types and then moving on to plots, `for` loops, functions, and finally root finding.

### 1.1 The most common numerical arrays are scalars, vectors, and matrices

The name “MATLAB” is an abbreviation of the phrase “matrix laboratory,” which reflects MATLAB’s origin as specialized software for working with entire matrices and vectors all at once, as opposed to working with individual numbers one at a time. All data in MATLAB is an *array* of one kind of data or another, and when we’re dealing with numbers the most common arrays we’ll encounter are

**scalars**, which are arrays of size 1-by-1 (*i.e.*, a single number),

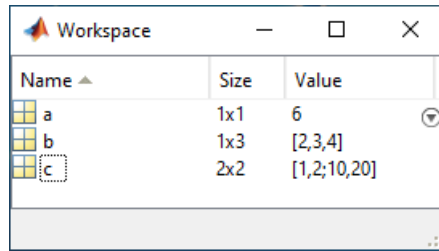


Figure 1.1: The workspace window in MATLAB. You can change what information is shown by clicking on the downward-facing triangle in the top-left corner and selecting “Choose columns.”

**vectors**, which are arrays of size  $n$ -by-1 or 1-by- $n$  (i.e., single rows or single columns of numbers), and

**matrices**, which are arrays of size  $m$ -by- $n$  (i.e., an array with more than one column and more than one row).

Suppose we wish to let  $a$  be the number 6;  $b$  be a row vector of the numbers 2, 3, and 4; and  $c$  be a matrix with 1 and 2 in the first row and 10 and 20 in the second row. To create these variables from the command prompt in MATLAB we’d type

```
>> a = 6;
>> b = [2,3,4]; % spaces or commas ok
>> c = [1,2;10,20]; % semicolons are required
```

The variables  $a$ ,  $b$ , and  $c$  should now exist in your *workspace*, which is the space in your computer’s memory that holds this information. The workspace window is a great troubleshooting aid because it shows you information about the variables you’ve created or their value after a calculation is performed. After running the commands above your workspace should look like the one shown in Figure 1.1.

It’s also important to remember how to access values within vectors and matrices by indexing. MATLAB’s indexing starts at 1: the first element of the vector  $b$  in the example above is  $b(1)$ , the second element is  $b(2)$ , and so on. For vectors you need only supply a single index and MATLAB will know what you mean, but if you’ve got a matrix then the suggested format is row-column indexing. Row-column indexing requires providing two indices where the first index corresponds to the row and the second index corresponds to the column. For example, to get the element in the bottom-right corner of  $c$  in the example above you’d use  $c(2,2)$ .

There are two other indexing tools that can be useful, the `end` keyword and the colon (`:`) operator. The `end` keyword gives you the highest index in the position it’s supplied, so  $b(\text{end})$  returns 4,  $c(\text{end},1)$  returns 10, and so on. The colon operator allows you to specify index

ranges so that you can return more than one value, so `b(2:end)` returns the vector `[3,4]`, `c(end, :)` returns the entire last row of `c`, and so on.

Finally, a quick way to create a vector of equally-spaced data is the `linspace(x,y)` function. The default behavior of `linspace(x,y)` is to create a vector of 100 equally-spaced data points beginning at `x` and ending at `y`, but if you provide an additional input as `linspace(x,y,n)` then you'll generate `n` equally-spaced data points. This function appears most often when we're either plotting something or using a `for` loop.

## 1.2 Plots and subplots can be used to visualize data

The most common plots we'll use in this book require two input vectors, one for the  $x$ -location values and one for the  $y$ -location values of all the points we wish to plot. These two vectors are passed as the first two arguments of the `plot` function as `plot(x,y)` and this generates a simple line plot. At minimum you should always annotate your plot with at least an  $x$ - and  $y$ -axis label; these commands and several others are summarized in Table 1.1.

Often we'll find ourselves wanting to compare to curves, say  $f(x)$  and  $g(x)$ , either directly with one data series overlaid atop the other or side-by-side with one data series in plot and the other data series in a plot adjacent to the first. The two commands we use to accomplish this behavior are the `hold` command and the `subplot` command.

To overlay one data series atop another, first create a plot of one of the data series using the `plot` function and then provide the command `hold on` to tell MATLAB to add all subsequent plots to the same figure. Plot all the other data series you need, then provide the command `hold off` so that subsequent plots override the existing plots.

To place plots next to each other but on separate axes, first use the command `subplot(r,c,loc)` where `r`, `c`, and `loc` are all integers. This creates a subplot figure with space for `r` rows and `c` columns of plots, and tells MATLAB that the next plot should go in the location specific by `loc`.

### Example 1.1: Creating a basic plot

Create a subplot figure of three trigonometric functions. On the left, plot  $\sin(x)$  from 0 to  $2\pi$  and overlay  $\cos(x)$  for the same range. On the right, plot  $\tan(x)$  from 0 to  $2\pi$ .

### Solution

Table 1.1: Helpful plotting commands.

Command	Behavior
<code>plot(x,y)</code>	Plot $y$ versus $x$
<code>plot(x,y,ls)</code>	Plot $y$ versus $x$ with line specification <code>ls</code> , where <code>ls</code> is a string for the color (' <code>k</code> ', ' <code>b</code> ', ' <code>r</code> ', etc), point style (' <code>o</code> ', ' <code>.</code> ', ' <code>x</code> ', etc), and/or line style (' <code>-</code> ', ' <code>--</code> ', ' <code>-.</code> ', etc). For example, ' <code>ro-</code> ' creates a plot with a red line and red circles as markers.
<code>axis square</code>	Set aspect ratio to square
<code>xlim([l,u])</code>	Set $x$ -axis limits to $l$ (lower) and $u$ (upper)
<code>ylim([l,u])</code>	Set $y$ -axis limits to $l$ (lower) and $u$ (upper)
<code>hold on</code>	Make the next plot command add its data series to the existing plot
<code>hold off</code>	Make the next plot command overwrite the existing figure
<code>subplot(r,c,loc)</code>	Make a subplot figure with $r$ rows, $c$ columns, and places the next plot at location <code>loc</code> .
<code>grid on</code>	Show a grid
<code>xlabel('str')</code>	Add a label ' <code>str</code> ' to the current plot's $x$ -axis
<code>ylabel('str')</code>	Add a label ' <code>str</code> ' to the current plot's $y$ -axis
<code>title('str')</code>	Add a title ' <code>str</code> ' to the current plot
<code>legend('s1', 's2')</code>	Adds a legend with entries ' <code>s1</code> ', ' <code>s2</code> '.

We first use `subplot(1,2,1)` to establish a 1-by-2 array of plots and indicate that the first plot goes in position 1 (*i.e.*, the left plot). We'll then use `hold on` to overlay the  $\sin(x)$  and  $\cos(x)$  plots. We'll then use `subplot(1,2,2)` to send the next plot to position 2, and finish with the plot of  $\tan(x)$ .

As noted in the text, only the basic plot commands are provided in most of this book. This example is one of the few where every command for every annotation is shown.

```
1 % Create the data set to be plotted
2 x = linspace(0,2*pi);
3 f = sin(x);
4 g = cos(x);
5 h = tan(x);
6
7 % Create the left subplot
```

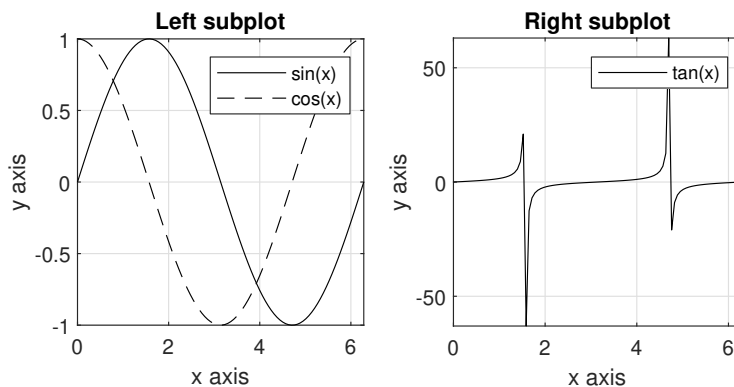


```

8 subplot(1,2,1)
9 plot(x, f, 'k')
10 hold on
11 plot(x, g, 'k--')
12 hold off
13 xlabel('x axis')
14 ylabel('y axis')
15 legend('sin(x)', 'cos(x)')
16 axis square
17 grid on
18 title('Left subplot')
19
20 % Create the right subplot
21 subplot(1,2,2)
22 plot(x, h, 'k')
23 xlabel('x axis')
24 ylabel('y axis')
25 legend('tan(x)')
26 axis square
27 grid on
28 title('Right subplot')

```

The figure generated by this script is shown below. There are two things to note about this example, the space needed for calculations and the space needed for plotting. The former is often fairly short: in this case, only four lines are needed to generate all the necessary data. The latter is often lengthy but doesn't fundamentally change the solution.



To save space and hopefully add clarity, only the basic plot commands will be shown in most of the examples in this book. Commands like `axis square` and `grid on` will often be used but not shown explicitly in the code because they take up a fair amount of space but don't add much to your understanding of how to solve the problem.

### 1.3 Local functions within scripts are convenient ways to organize problems

The code generated in Example 1.1 can be saved exactly as presented and it will become a *script*. Running a script is identical to entering each line of the script one at a time at the command prompt, and after you run the script every variable created by that script will appear in your base workspace. Nearly every solution you code for the problems or examples in this book should be coded as a script!

There are several useful tips to keep in mind when working with scripts:

**Quickly run entire scripts:** Press the F5 key on your keyboard while your cursor is inside the script you want to run and the entire script will run. This is equivalent to pressing the green “Run” arrow in the ribbon or typing the name of your script at the command prompt.

**Run only the highlighted code:** Highlight any portion of your code and press the F9 key on your keyboard to run *only* that portion of your code. This can be useful for debugging purposes.

**Enable the debugger:** Click on one of the horizontal lines next to any line number in the script; click the resulting red circle to disable the debugger. The debugger really shines when more complicated scripts are used, such as those that include `for` loops or those that make calls to other functions.

A *user-defined function* can be written in almost exactly the same way as a script except that the first line contains a function definition such as

```
function [out1, out2] = myFunc(in1, in2)
```

where the number of in and out arguments can be anything you like (even zero). To use this function save it to your working directory as a file with the same name as the function name (in this case, `myFunc.m`) and then call it from either the command line or a function or script as

```
>> [a, b] = myFunctionName(c, d);
```

where `c` and `d` are inputs and `a` and `b` are outputs.

One disadvantage of the custom function format is that it creates a new file named `myFunctionName`, and believe it or not it can get tedious to come up with new, meaningful file names (file names such as `file1`, `file2`, etc quickly become useless because they all look the same). To

get around this problem you can define functions within a script; functions defined in this way are called *local functions*.

To create a local function, first create a script file, add any amount of code at the top of the script, then write your function at the bottom of the script. This generally organizes your script into a *script portion* at the top of the file and a *local function portion* at the bottom of your file:

```

1 % Script portion
2 a = 1;
3 b = myLocalFunction(a);
4
5 % Local function portion
6 function out = myLocalFunction(in)
7 out = 2*in;
8 end

```

There are a few important things to keep in mind when using local functions:

**Always use the end keyword:** The `end` keyword should always be included at the end of the local function definition, which is Line 8 in the example above.

**Write as many local functions as you want:** You can add an unlimited number of local functions at the bottom of your script. Each one needs its own `end` keyword.

**Reuse names:** A local function in one script can be named exactly the same as a local function in another script because they'll never "see" each other (*i.e.*, their scope is different).

**No functions in functions:** You can put a function within a script and things work normally, but if you put a function within another function then things start to get weird. The function-within-a-function is called a *nested function* and it's strongly discouraged because it becomes tricky to keep track of each variable's scope.

The local function structure provides a neat, compact way to organize your code for a given problem, and because it allows you to reuse the same general structure you'll find it a convenient way to draw parallels between problems.

### Example 1.2: Using a local function

Define a local function to calculate

$$k = 3 \exp \left[ \frac{-2000}{T} \right]$$

where  $T$  is the temperature in kelvin and  $k$  has units of  $\text{s}^{-1}$ . Evaluate  $k(650\text{ K})$ .

### Solution

```

1 % myFirstLocalFunction.m
2 % Script portion
3 T = 650; % K
4 k_ans = k(T); % 1/s
5 fprintf('k(%.0f K) = %.2f 1/s\n', T, k_ans);
6
7 % Local function portion
8 function output = k(input)
9 T = input; % K
10 output = 3*exp(-2000./T); % 1/s
11 end

```

Running this script by either pressing the green “Run” arrow, pressing F5, or entering the file name at the command prompt provides the following output in the command window:

```

>> myFirstLocalFunction
k(650 K) = 0.14 1/s

```

Although initially it may seem a little odd, Example 1.2 utilizes a simple three-part organization scheme that you’ll find very useful:

**Set up and solve the problem.** On Line 3 we define a temperature of 650 K and on Line 4 we use this temperature to “solve” the problem (*i.e.*, to calculate  $k$ ).

**Analyze the results.** On Line 5 we output the result to the command window so that we can see what value was calculated. Often the “analysis” will be to identify a number or create a plot.

**Write local functions.** On Lines 8-11 we define a local function to help us solve the problem.

It’s that last part—writing local functions—that can seem a little odd because it appears to come *after* you’ve already solved the problem. The reason for this organization is simply because that’s the syntax that MATLAB uses for local functions: they have to be listed at the bottom of the script.

## 1.4 The for loop is enormously useful

One of MATLAB's major advantages is its ability to process *vectorized* code. Vectorized code is code that's design to accept vectors (or matrices) as input, and it typically outputs vectors as well. For example, way back in Example 1.1 we had

```
2 x = linspace(0,2*pi);
3 f = sin(x);
```

In this code, Line 3 is an example of vectorized code: the sine function `sin()` accepted a vector input `x` and produced a vector output `f`. By some unknown process, MATLAB went through the entire vector `x`, calculated  $\sin(x)$  at each value of `x`, and stored the results in a new vector called `f`. Fortunately, a great number of common functions in MATLAB are already vectorized, and as long as you use the elementwise operators `.*` and `./` in your custom functions then your code will also generally be vectorized.

However, there are times when either (i) your code cannot be vectorized, or (ii) you're not sure how to vectorize your code. In these cases you can take a "brute force" approach by using the `for` loop to perform a calculation one element at a time. This approach is enormously useful because it changes your approach from needing to vectorize a complicated analysis to needing to only perform the analysis *once* and then automating the remaining calculations by means of a `for` loop.

As an example of when we may want to use such an approach, let's say you've got a material balance of the form

$$0 = X - 3 \exp\left[\frac{-2000}{T}\right] \left(1 - \frac{X}{T}\right) \quad (1.1)$$

and you'd like to solve Equation (1.1) for  $X$  at various values of  $T$ . Fundamentally such a thing should be possible: if a value of  $T$  is guessed then Equation (1.1) has only one unknown ( $X$ ), so it's solvable. It's also possible to rearrange Equation (1.1) to isolate  $X$  as a function of the guessed value of  $T$ , but let's assume that either you didn't notice such a thing was possible or such rearrangement was impossible.<sup>1</sup>

Herein lies the great utility of the `for` loop: rather than attempt to vectorize an already complicated problem, we reduce the problem to a single solvable instance and let the `for` loop repeat the solution many times. If we can figure out a way to solve Equation (1.1) just *once* for a single value of  $T$  then we can use a `for` loop to repeat this solution procedure for any number of  $T$  values.

The general structure of a `for` loop is as follows:

```
1 x = linspace(0, 100); % generate x vector
```

<sup>1</sup>It will be impossible to perform such rearrangement on some material and energy balance equations.

```

2 y = zeros(size(x)); % preallocate y vector
3 for i = 1:length(x) % start loop
4     y(i) = anyFunc(x(i)); % perform calculation
5 end % finish loop

```

On Line 1 a vector of  $x$  values is generated, and in many of the examples we'll see in this book the `linspace` function will be used because it'll give us a vector of equally-spaced points at which we can evaluate something else. On Line 2 we perform an optional step called *preallocation* which produces a vector of all zeros to act as placeholders for data we expect to generate soon. It's optional because often MATLAB can run the code whether or not you've preallocated a variable, but the advantages of preallocation are (i) it speeds up your code and (ii) it forces you to think about what kind and amount of data you expect your code to produce. The `size` function ensures that we'll also have exactly as many empty spots in  $y$  as we had points in  $x$ .

On Line 3 we start the `for` loop and use the `length()` function to determine how many elements are in the vector  $x$  (in this case `length(x)` is 100 because we used the default `linspace` function on Line 1). You could equally use the `numel` function to achieve the same result, or you could replace either of those functions with the number of steps you know you need to take, such as `for i = 1:100`. All three give the same result when working with vectors but you should review the MATLAB documentation for each function if you expect to work with matrices or non-numerical data types.

On Line 5 we use indexing to pass the  $i$ -th element of  $x$  as  $x(i)$  to `anyFunc()`. We don't care at this point what `anyFunc()` does, but we assume that it takes a scalar value as input and returns a scalar value as output. The output from this function is assigned to the  $i$ -th element of  $y$  as  $y(i)$ , and then the loop repeats until all  $i$  counts up to its final value. On Line 6 the loop terminates with the `end` statement.

The usage of this basic structure is illustrated below with our same  $k$  equation from Example 1.2. Keep in mind that the local function for  $k$  in that example is vectorized such that the `for` loop is not actually necessary to do the calculation we're using it for; this is just to illustrate the process. We'll see in Section 1.7 a more complicated example of code that cannot be vectorized and therefore requires use of a `for` loop.

### Example 1.3: Using a for loop

Use a `for` loop to calculate  $k$  according to Example 1.2 from 450 K to 700 K using at least 250 points. Plot the result.

## Solution

Let's assume that the local function from Example 1.2 will be written in exactly the same way, so all we need to worry about is the `for` loop part. First, define as `T` the range of temperature values at which we'd like to evaluate  $k$ . In most examples this is done with a call to the `linspace` function as

```
1 T = linspace(450, 700, 250); % K
```

Next, generate a zero vector to hold our soon-to-be-calculated values of  $k$ . Make sure it's the same size as `T` by using the `size()` function:

```
2 k_ans = zeros(size(T)); % 1/s
```

Finally, use a `for` loop over all 250 elements of `T` by using `length(T)`. Within the loop, get the  $i$ -th element of `T` as `T(i)`, pass it to the `k` function as `k(T(i))`, and store the result as the  $i$ -th element of `k_ans` as `k_ans(i)`:

```
3 for i = 1:length(T)
4     k_ans(i) = k(T(i)); % 1/s
5 end
```

We're now done with the "calculation" portion of the problem, so all that remains is to plot the result:

```
6 plot(T, k_ans)
```

Remember, in this book the plot commands are minimal in order to save space but you should be using additional plot commands to make your plot neat and informative (see Example 1.1).

Assembling all these steps into a script and running it produces a plot like the one below.

```
1 % myForLoop.m
2 % Script portion
3 T = linspace(450, 700, 250); % K
4 k_ans = zeros(size(T)); % 1/s
5 for i = 1:250
6     k_ans(i) = k(T(i)); % 1/s
7 end
8 plot(T, k_ans)
9
10 % Local function portion
11 function output = k(input)
```

```
12| T = input; % K
13| output = 3*exp(-2000./T); % 1/s
14| end
```

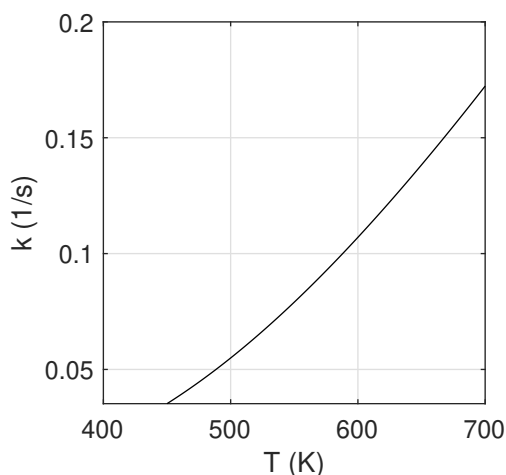


Figure 1.2: Plot output from myForLoop.m

## 1.5 Root finding solves equations for unknowns

Whenever you're asked to solve a problem "by hand" you can anticipate that the solution requires some algebraic rearrangement and perhaps integration of a simple differential equation. Such problems are limited in scope to only the simplest physical situations, most commonly systems that contain only one chemical reaction and are isothermal. As you can surely imagine, there are plenty of systems in which more than one chemical reaction occurs, and at the time of this book's writing the heat death of the universe was expected to be at least a few weeks in the future, making non-isothermal systems a must as well.

For those cases where hand calculations are not required you'll most likely be faced with the task of determining the numerical value of one or more variables which satisfy one or more equations. We usually refer to this process informally as "solving" the equations but more formally the process of finding these numerical values is called *root finding*. Since the equations we'll deal with are usually non-linear we can't rely on linear algebra to solve them; instead we'll need something that can handle more general cases, namely the `fsolve` function which is part of the **Optimization Toolbox** in MATLAB.

The steps to solve such equations (*i.e.*, to find the roots of such equations) are as follows:



1. Rearrange the equations to root-finding form and call this equation  $f(x)$ .
2. Guess initial values for each unknown.
3. Use `fsolve` to solve the equations.

Step 1 is easy: simply move all the terms in your equation to one side of the equal sign, call it  $f(x)$ , and you've got an equation in root-finding form. For example, the expression

$$x = \cos x$$

can be transformed to root-finding form as either

$$f(x) = x - \cos x = 0 \quad \text{or} \quad 0 = \cos x - x = f(x).$$

Either form is acceptable and will yield the same numerical value of  $x$  when solved, but some of the intermediate values during the solution procedure itself will be different. Notice that the value of  $x$  which satisfies  $x = \cos x$  is the same value that satisfies  $f(x) = 0$ .

Step 2 is also fairly easy, although it can be frustrating in some limited cases: guess the initial values for each unknown. Generally your guesses need only have the correct sign and be within a few orders-of-magnitude of the actual answer. Fortunately for us, most root finding problems in reaction engineering have some physical system as a basis and from that system we can usually guess a reasonable value (e.g., the initial temperature or the initial moles). If you have an arbitrary expression for which there is no corresponding physical system then it can be challenging to guess an appropriate initial value because you're essentially pulling values from a hat.

If you're attempting to solve just a single equation then a good way to get an initial guess is to plot the equation in root-finding form and look for places where the curves intersect zero. Those points of intersection are the roots and simply reading off the  $x$ -value of the root provides a good enough starting point that `fsolve` can take it from there.

Step 3 is straightforward but usually takes the most amount of time: use `fsolve` to solve the equations. The general syntax of `fsolve` is as follows:

```
x_sol = fsolve(@fun, x0)
```

where `fun` is the function of  $x$  (in root-finding form) you want to solve, `x0` is your initial guess, and `x_sol` is the value of  $x$  which satisfies the equation (*i.e.*, it's the solution).

The function `fun` corresponds directly to the function  $f(x)$  defined above, and therefore its input should be a value of  $x$  and its output should be the root finding equation evaluated at

$x$ . The @ symbol in @fun indicates a *function handle* and tells MATLAB to look for a function named fun and to use it as the root finding function.

The fsolve function will essentially guess values of  $x$ , evaluate  $f(x)$  at that value of  $x$ , and determine if  $f(x)$  is close enough to zero to consider  $x$  a solution. The phrase “close enough” is deliberately vague: by default, fsolve looks for values of  $x$  which produce  $f(x) \leq 10^{-6}$  but you can change this criteria by following the guidelines in the MATLAB documentation.

#### Example 1.4: Finding a simple root with fsolve

Find the root of  $x = \cos x$ .

#### Solution

A root finding equation for  $x = \cos x$  is  $f(x) = x - \cos x$ . A decent initial guess should be  $x = 0$  since  $\cos x$  will be small in that region. The following code uses a local function to define  $f(x)$  as fun and finds the root with fsolve:

```
1 % simpleRootFinding.m
2 % Script portion
3 x0 = 0;
4 x_sol = fsolve(@fun, x0);
5 fprintf('x_sol = %.5f\n', x_sol);
6
7 % Local functions
8 function out = fun(x)
9 out = x - cos(x); % root-finding form
10 end
```

Running this script produces the following output:

```
>> simpleRootFinding
Equation solved.

fsolve completed because the vector of function values is near zero as
measured by the default value of the function tolerance, and the
problem appears regular as measured by the gradient.

<stopping criteria details>

x_sol = 0.73909
>>
```

The numerical value we’re interested in is at the bottom:  $x = 0.73909$ . The extra output is from fsolve and essentially tells us that the problem has been solved. The

key phrase to look for is at the top, “Equation solved.” If you see anything other than “Equation solved” then you should carefully check your answer (e.g., by plotting it) to ensure that you’ve found the correct solution. You can also refer to MATLAB’s “When the Solver Fails” documentation to see other options you can try if the solver can’t solve your equations.

If you’re certain that `fsolve` will find the right value and you’re not worried about messages then you can turn off those messages by defining an options variable just before calling `fsolve`, and within those options setting the `'Display'` to `'off'`:

```
3 x0 = 0;
4 options = optimoptions(@fsolve, 'Display', 'off');
5 x_sol = fsolve(@fun, x0, options);
```

If you’d like to see how `fsolve` is going about its business of guessing  $x$  values (iterating) then you can change `'Display'` to `'iter'`:

```
3 x0 = 0;
4 options = optimoptions(@fsolve, 'Display', 'iter');
5 x_sol = fsolve(@fun, x0, options);
```

which produces a table of the number of iterations, number of times the function was evaluated, and the value of  $f(x)$  at each iteration. For Example 1.4, the number of iterations was four, the total number of evaluations of  $f(x)$  was ten, and the value of  $f(x)$  at the solution value was  $8.11137 \times 10^{-20}$ .

## 1.6 Multiple roots must be found by trial-and-error

If your function has more than one root then you must vary your starting point until you find the root you’re interested in.<sup>2</sup> For example, the function

$$f(x) = 0.125x - \sin(x - 3)$$

has three roots near  $-4$ ,  $0$ , and  $+4$  (Figure 1.3).

To find these roots we must provide various initial guesses until we get the roots of interest. The best way to ensure that you’ve found the relevant roots is to plot both the function and the roots.

<sup>2</sup>There are more systematic approaches that can be taken for one equation with `fzero` or for multiple equations with `fmincon`, but the approach here is sufficient for most of our problems.

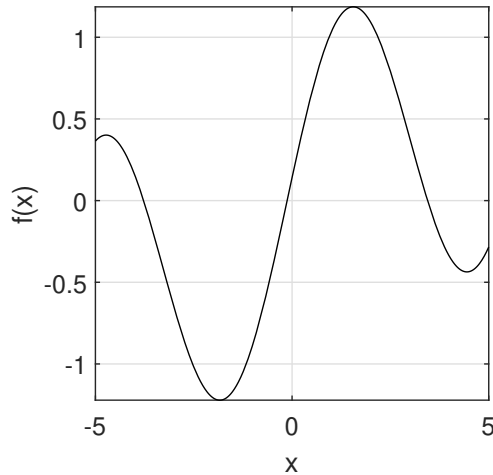


Figure 1.3: The roots of  $f(x) = 0.125x - \sin(x - 3)$  are near  $-4$ ,  $0$ , and  $+4$ .

### Example 1.5: Finding multiple roots with `fsolve`

Find the roots of  $f(x) = 0.125x - \sin(x - 3)$  between  $-5$  and  $+5$ .

#### Solution

A plot of  $f(x)$  is shown in Figure 1.3 and as already noted the roots are near  $-4$ ,  $0$ , and  $+4$ . We'll use these as our starting points for three different calls to `fsolve` and hopefully those will identify all the roots:

```

1 % multiRootFinding.m
2 % Script portion
3 x1 = fsolve(@fun, -4);
4 x2 = fsolve(@fun, 0);
5 x3 = fsolve(@fun, 4);
6 fprintf('x_sol = [%.2f, %.2f, %.2f]\n', x1, x2, x3);
7
8 % Local functions
9 function out = fun(x)
10 out = 0.125*x - sin(x-3);
11 end

```

Running this script produces the following output:

```

>> multiRootFinding
x_sol = [-3.77, -0.13, 3.45]

```

Those roots look fairly close to our initial guesses, but just to be sure these roots were overlaid as points on the plot of  $f(x)$  as shown below.

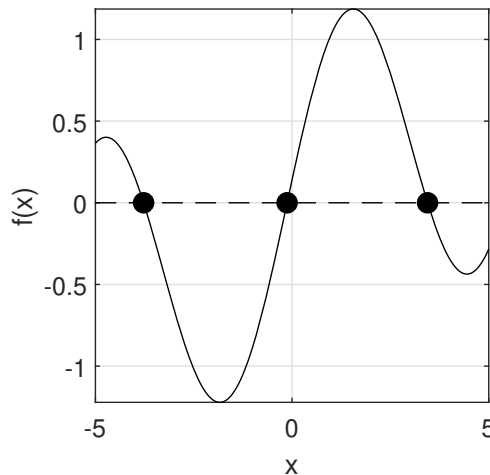


Figure 1.4: The roots of  $f(x)$  are at  $-3.77$ ,  $-0.13$ , and  $3.45$ .

## 1.7 The fsolve function can accept additional input arguments

Suppose we wish to solve an equation at several different values of some parameter within that equation. For example, recall the material balance we mentioned in Section 1.4,

$$0 = X - 3 \exp\left[\frac{-2000}{T}\right] \left(1 - \frac{X}{T}\right),$$

and suppose we wish to create a plot of  $X$  versus  $T$ . To do so we'll need to solve this equation for  $X$  at many values of  $T$ , and it can be tedious to manually change the value of  $T$ , run `fsolve`, change  $T$  again, run `fsolve` again, and repeat until enough values are accumulated to make a decent plot.

Instead, we can pass  $T$  as an additional parameter to `fsolve` by including it as an additional input for our local function and explicitly defining  $X$  as the independent variable for `fsolve`. To include  $T$  as an additional input parameter, our local function definition changes from

```
function out = fun(X)
```

to

```
function out = fun(X,T)
```

The variable  $X$  can be explicitly identified as the independent variable for `fsolve` by using `@(X)fun(X,T)` instead of `@fun` as the first argument to `fsolve`, as in

```
X_sol = fsolve(@(X)fun(X,T), X0);
```

### Example 1.6: Passing additional arguments to `fsolve`

Solve

$$0 = X - 3 \exp\left[\frac{-2000}{T}\right] \left(1 - \frac{X}{T}\right)$$

once for  $T = 450$  by passing  $T$  as an additional input argument to `fsolve`.

### Solution

```
1 % fsolveAdditionalParam.m
2 % Script portion
3 T = 450;
4 X_sol = fsolve(@(X)fun(X,T), 0.5);
5 fprintf('X_sol = %.2f\n', X_sol);
6
7 % Local functions
8 function out = fun(X, T)
9 out = X - 3*exp(-2000./T).*(1-X./T);
10 end
```

Running this script produces the following output:

```
>> fsolveAdditionalParam
X_sol = 0.04
```

Now that we have the basic outline of a script to solve for  $X$  at a *single* value of  $T$  we can apply the concepts from Section 1.4 to solve for  $X$  at *any number* of  $T$  values by using a `for` loop. We'll initialize a vector of lots of  $T$  values, place our call to `fsolve` within the `for` loop, pass the  $i$ -th element of  $T$  to `fsolve`, and store the result at the  $i$ -th location of a solution vector.

### Example 1.7: Using `fsolve` within a `for` loop

Plot  $X$  versus  $T$  according to

$$0 = X - 3 \exp\left[\frac{-2000}{T}\right] \left(1 - \frac{X}{T}\right)$$

from  $T = 450$  to  $T = 750$  using at least 100 data points.

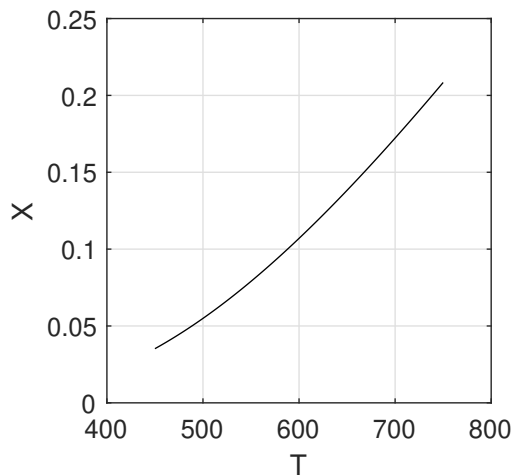
### Solution

```

1 % fsolveForLoop.m
2 % Script portion
3 T = linspace(450, 750);
4 X_sol = zeros(size(T));
5 for i = 1:length(T)
6     X_sol(i) = fsolve(@(X)fun(X,T(i)), 0.5);
7 end
8 plot(T, X_sol)
9
10 % Local functions
11 function out = fun(X, T)
12 out = X - 3*exp(-2000./T).*(1-X./T);
13 end

```

Since we're calling `fsolve` many times in this script, this is a good time to turn off the display output as mentioned in Section 1.5, but that's not strictly a requirement. The plot produced by this script is shown below.



## 1.8 The fsolve function can solve more than one equation

We've seen that `fsolve` can solve a variety of problems where there's one equation for one unknown but it's also capable of solving multiple equations with multiple unknowns. Although capable of solving thousands of equations simultaneously, `fsolve` will most commonly be

used to find the solution of several material balances (usually less than five) and one energy balance. Always keep in mind that `fsolve` is limited to problems where the number of unknowns is equal to the number of equations (the degrees of freedom are zero).

Recall from Example 1.4 that `fsolve` requires a function handle as its first input, and that the function referred to by this handle has two requirements: the input should be the unknown variable and the output should be the value of the root-finding function when evaluated at any value of the unknown variable. We follow exactly the same pattern to solve multiple equations but instead of a single root-finding equation and a single unknown we now have a *column vector* of root-finding equations and a *column vector* of unknowns.

To illustrate, suppose we have the equations

$$\begin{aligned}y &= \sqrt{x} \\ x + y &= 4\end{aligned}$$

which we want to solve for  $x$  and  $y$ . We first rewrite *both* equations in root-finding form as

$$\begin{aligned}f(x, y) &= 0 = \sqrt{x} - y \\ g(x, y) &= 0 = 4 - x - y\end{aligned}$$

where  $f(x, y)$  and  $g(x, y)$  will both equal zero when the correct values of  $x$  and  $y$  are identified. According to these two equations the column vector of root-finding equations is

$$\text{out} = \begin{bmatrix} f(x, y) \\ g(x, y) \end{bmatrix} = \begin{bmatrix} \sqrt{x} - y \\ 4 - x - y \end{bmatrix} = \begin{bmatrix} \text{sqrt}(x) - y \\ 4 - x - y \end{bmatrix}$$

We've named this column vector `out` in anticipation of using it as the output of our local function, and notice again that this is a *column vector*: `fsolve` won't work if you provide it a *row vector*.

The input to our local function will need to be a column vector of the unknown variables. To be consistent with some of the syntax we'll introduce later for solving differential equations, let's call this column vector  $Y$  such that

$$Y = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

and individual variables can be extracted with indexing as  $x=Y(1)$  and  $y=Y(2)$ . The corre-



sponding vector of initial guesses for  $x$  and  $y$  is

$$Y0 = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x0 \\ y0 \end{bmatrix}.$$

We're now all set to solve this system of equations with `fsolve`.

### Example 1.8: Solving multiple equations with `fsolve`

Solve the system of equations  $y = \sqrt{x}$  and  $x + y = 4$  for  $x$  and  $y$ .

#### Solution

All necessary formulations were derived in Section 1.8, and the only thing left to do is pick an initial guess for  $x$  and  $y$ , then code it up. Since we know nothing about a physical system represented by this system of equations we'll have to pick an initial guess somewhat randomly. Since  $x = 0$  and  $y = 0$  satisfies the first equation, those two values should be a decent starting point.

```

1 % multiEqnFsolve.m
2 % Script portion
3 Y0 = [0 0]; % [x0 y0]
4 Y = fsolve(@myFunc, Y0);
5 fprintf('x = %.2f, y = %.2f\n', Y(1), Y(2));
6
7 % Local function
8 function out = myFunc(Y)
9 x = Y(1);
10 y = Y(2);
11 out = [sqrt(x) - y; % f(x,y)
12       4 - x - y]; % g(x,y)
13 end

```

Running this script produces the following output:

```

>> multiEqnFsolve
x = 2.44, y = 1.56

```



## Chapter 2

# The Equilibrium Reactor

“If someone points out to you that your pet theory of the universe is in disagreement with Maxwell’s equations—then so much the worse for Maxwell’s equations. If it is found to be contradicted by observation—well, these experimentalists do bungle things sometimes. But if your theory is found to be against the second law of thermodynamics I can give you no hope; there is nothing for it but to collapse in deepest humiliation.”

Arthur Eddington, *The Nature of the Physical World* (1928)

### 2.1 Reaction equilibrium is a dynamic process

To aid our discussion of equilibrium we’re first going to perform a simple molecular simulation. Suppose in a system initially consisting of 81 particles of type A there are two reactions possible,



where the indicated probability is the chance that the reaction occurs whenever a collision between the reactants occurs. These two reactions roughly approximate the single, reversible reaction



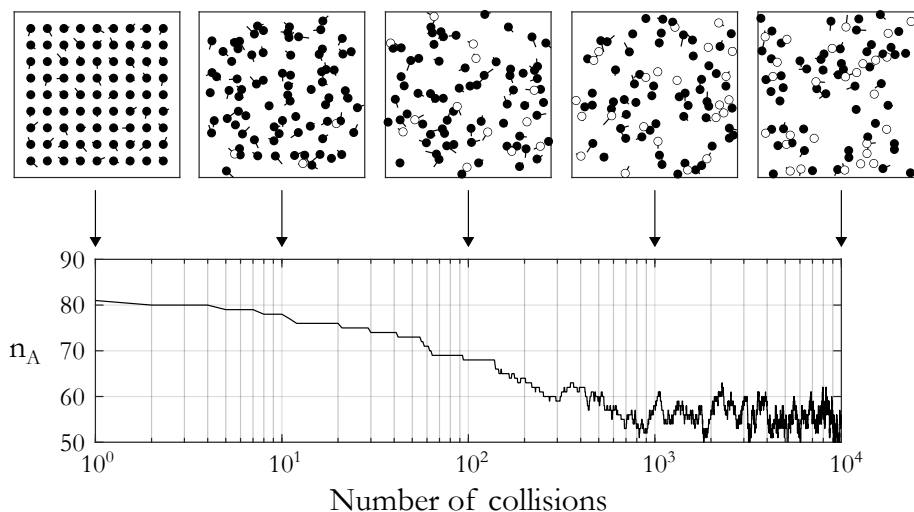


Figure 2.1: Development of dynamic equilibrium in a closed system. Particle A is denoted by a closed circle ( $\bullet$ ); particle B is denoted by an open circle ( $\circ$ ).

but the requirement for a collision to occur as a prerequisite for chemical reaction has been made explicit by separating the forward and reverse reactions. In fact the collisions could have been written as between particle A or B and anything else—a wall, another particle of A, or another particle of B—but this formulation will serve us equally well for the qualitative discussion which follows.

To perform our simulation, further imagine that the particles are initially placed on a uniform grid, each particle is assigned a random initial velocity, and during the simulation all collisions are perfectly elastic and all particles behave as rigid spheres. The number of A particles ( $n_A$ ) for the first 10,000 collisions in our simulation is shown in Figure 2.1.

The general behavior of particles in this simulation is a good picture for you to keep in mind when trying to visualize what's happening in a reacting system: molecules are always jostling about, colliding with one another, and every so often a collision results in a chemical reaction. There's a probability associated with that chemical reaction such that not every collision results in a reaction, and although there are methods to estimate such probabilities from first principles we'll assume that they're always either given to us or measured experimentally.

What's particularly interesting about this simulation is what happens to the number of A particles as time progresses. Initially, up until about  $10^3$  s, the number of A particles is gradually declining from initially pure A (100 mol% A) to somewhere between 50 and 60 particles of A (about 60-70 mol% A). From  $10^3$  s to  $10^4$  s—about 90% of the total simulated time—the number of A particles doesn't fluctuate much beyond that range of 60-70 mol%, and any fluctuations which do occur don't last long.

The reason for the stabilization is a balance between two driving forces, one for the consumption of A and generation of B according to Equation (2.1) and another for the consumption of B and generation of A according to Equation (2.2). Despite the 100% probability of Equation (2.1) the frequency of Equation (2.1) declines as time goes on because the number of A particles decreases and therefore so too does the number of collisions between two A particles. Had no mechanism for the production of A been in effect then the number of A particles would eventually fall to just one lonely A particle, forever colliding with B particles and the walls but never reacting.

However, there is in fact a mechanism for the production of A (and the destruction of B) according to Equation (2.2), and it's due to this opposing driving force that the number of A particles stabilizes between 50 and 60 rather than approaches 1. As the number of B particles increases, so too does the number of collisions between two B particles and therefore the frequency of Equation (2.2) also increases.

What we're seeing here is an example of *dynamic* equilibrium, a type of equilibrium in which ceaseless, microscopic motion produces small fluctuations around a stable equilibrium condition. Dynamic equilibrium is unlike *static* equilibrium wherein all motion has stopped, such as when a bridge is stable under its load. Reaction equilibrium is a form of dynamic equilibrium and has two important features:

**Forward and reverse reactions never stop** but the system achieves a state where the forward and reverse reaction rates are equal.

**Fluctuations beyond equilibrium are possible** but they don't last long because one of the driving forces—the forward or reverse reaction rates—will briefly increase while the other will briefly decrease to drive the system back to equilibrium.

One caveat to this simulation is that the number of particles is so small that the fluctuations are unusually large: a change of just one particle of A to a particle of B changes the mole fraction of A by  $1/81 \approx 1\%$ . A typical reacting system contains on the order of  $10^{23}$  molecules, so fluctuations of billions or even trillions of molecules away from equilibrium have a negligible effect on the overall system.

What we'll learn about in this chapter is how to describe the state of chemical equilibrium for isothermal, isobaric systems in which chemical reactions occur. This chapter draws heavily on your understanding of chemical engineering thermodynamics so some equations will be produced without an accompanying derivation or lengthy description. Also keep in mind that while our focus will be on the macroscopic scale—not the microscopic scale of this simulation—the behavior of particles on the *microscopic* scale can help us to understand what the equations tell us is happening at the *macroscopic* scale.

## 2.2 The Gibbs energy $G$ is minimized at equilibrium

A good place to start any discussion about energy is the First Law of Thermodynamics,

$$dU = \delta q + \delta w, \quad (2.4)$$

where  $dU$  is the change in total internal energy of the system,  $\delta q$  is energy transferred to or from the system by a temperature difference (*i.e.*, heat), and  $\delta w$  is energy transferred to or from the system by work. Internal energy is a state function so its differential is exact ( $dU$ ), whereas  $q$  and  $w$  are path functions so their differentials are inexact ( $\delta q$  and  $\delta w$ ). In this representation we've taken the system's center-of-mass as the frame of reference so there are no kinetic or potential energy terms to worry about, and we've also ignored any surface forces or internal gradients.

The Second Law of Thermodynamics establishes another state function called entropy ( $S$ ) which never decreases for any change in the system, and the changes in entropy can be calculated as

$$dS \geq \frac{\delta q}{T}, \quad (2.5)$$

where  $T$  is absolute temperature. Another way to express Equation (2.5) is

$$T dS \geq \delta q,$$

such that substituting the Second Law into the First Law yields

$$dU \leq T dS + \delta w.$$

In the absence of shaft work we can also write  $\delta w = -P dV$  which can be substituted into the First Law to yield

$$dU \leq T dS - P dV. \quad (2.6)$$

Should you ever invent a device that can directly measure changes in entropy or internal energy, please disregard the remainder of this book and apply for your Nobel Prize. For everyone else, please continue with your studies and note that the First Law written as Equation (2.6) is not particularly useful because we cannot measure  $S$  and  $U$  directly, nor their changes  $dS$  and  $dU$ . A better form would be one that involves temperature and pressure because those are two readily measured quantities.

Let's restrict our attention going forward to systems at constant temperature and pressure. Initially this can seem like an overly burdensome restriction, but do not despair: there

are plenty of interesting and useful applications even with such a restriction. Under these conditions we can move  $T$  and  $P$  inside the differentials of Equation (2.6) such that  $T dS = d(TS)$  and  $P dV = d(PV)$ , which lets us rearrange Equation (2.6) to yield

$$d(U - TS + PV) \leq 0.$$

The term in parentheses just happens to be *defined* as the Gibbs energy,

$$G \equiv U - TS + PV,$$

which allows us to finish our transformation of Equation (2.6) to yield

$$dG \leq 0 \quad \text{at constant } T \text{ and } P. \quad (2.7)$$

This is one of our jewels of thermodynamics: according to Equation (2.7) any change in a system at constant temperature and pressure must result in a Gibbs energy smaller than it was before the change. Eventually the system reaches a state where the Gibbs energy cannot get any smaller, and at this state no further changes are possible. We call this state *equilibrium* and give it a formal definition based on the Gibbs energy in this way: the Gibbs energy  $G$  is minimized at equilibrium.

Equation (2.7) is a general statement about chemical equilibrium, whether or not that equilibrium allows chemical reactions to occur. Equation (2.7) was the basis of the vapor-liquid equilibrium (VLE) calculations you studied in thermodynamics and it's the basis of the reaction equilibrium concepts you'll learn and use throughout this book. To simplify our discussion we'll further limit ourselves to single-phase systems but that's only for convenience; the central ideas would still hold even for multi-phase systems.<sup>1</sup>

Since the temperature, pressure, and phase of our system are all constant the only thing that can change is its composition. Using Equation (2.7) we can imagine some process by which we calculate the Gibbs energy of the system for various compositions to produce a plot of  $G$  versus composition, and whatever composition yields the smallest  $G$  is the equilibrium composition. Exactly this kind of analysis can be performed for ideal gases and is the basis of the *Gibbs Reactor*, the topic Appendix C.

More commonly we use concepts of calculus to identify the minimum of  $G$  directly, and this is the basis of the *Equilibrium Reactor*. Unlike the Equilibrium Reactor, the Gibbs Reactor does not need to know anything about exactly which chemical reactions occur, a distinct

---

<sup>1</sup>For a good introduction to simultaneous reaction and phase equilibrium, see Ch. 14 in Elliott and Lira, *Introductory Chemical Engineering Thermodynamics* (1998).

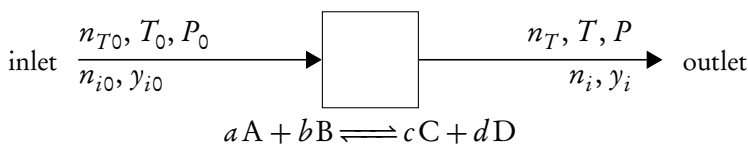


Figure 2.2: Stream notation used throughout this textbook. Subscript 0 denotes a parameter of the inlet stream; the same parameter without a subscript 0 denotes a parameter of the outlet stream. Other variables follow the same notation.

advantage when first approaching a new system. On the other hand, when designing reactors we often do know (or have a good idea of) what chemical reactions occur in a system, so the concepts introduced for the Equilibrium Reactor will find more regular application in the remainder of this book.

## 2.3 A stoichiometric table is a great organizational tool

Different sources tend to use slightly different notation when dealing with chemical reactions, so to avoid any confusion with nomenclature let's briefly review the notation used throughout this text and then we'll dig into the Equilibrium Reactor. Our notation is shown in Figure 2.2 and perhaps the most important feature to note is that parameters specific to the inlet stream are denoted with a subscript 0 (e.g.,  $T_0$  is the inlet stream temperature) and parameters specific to the outlet stream are denoted without the same subscript (e.g.,  $T$  is the outlet stream temperature). We adopt the usual notation of  $n$  for molar flow rates from which mole fractions  $x_i$  and  $y_i$  are defined as

$$x_i = \frac{n_i}{n_T} \quad (\text{liquid})$$

$$y_i = \frac{n_i}{n_T} \quad (\text{gas})$$

where  $n_T = \sum n_i$  is the total outlet molar flow rate. We also define a generic chemical reaction as



where the *stoichiometric coefficients* are  $\nu_A = -a$ ,  $\nu_B = -b$ ,  $\nu_C = c$ , and  $\nu_D = d$ . The stoichiometric coefficients for reactants are negative and for products are positive, which should be familiar by now.

We also define the extent of reaction  $\xi$  (sometimes called the reaction coordinate) for any



reaction as

$$\xi \equiv \frac{n_i - n_{i0}}{\nu_i} \quad (2.9)$$

which is sometimes rewritten to more clearly illustrate the relationship between inlet and outlet moles as

$$n_i = n_{i0} + \nu_i \xi. \quad (2.10)$$

A key feature of  $\xi$  is that any individual reaction has only *one* value for  $\xi$  and it's the same value regardless of which species was used for its calculation. If there's more than one reaction then each one gets its own  $\xi_j$  and Equation (2.10) becomes

$$n_i = n_{i0} + \sum_{j=1}^q \nu_{ij} \xi_j, \quad (2.11)$$

where  $q$  is the number of reactions,  $\nu_{ij}$  is the stoichiometric coefficient of species  $i$  in reaction  $j$ , and  $\xi_j$  is the extent of reaction for reaction  $j$ . For most of this chapter we'll focus on systems in which only one reaction occurs but at the end of the chapter we'll see an example of how to deal with multireaction systems.

If we substitute Equation (2.10) into the general definition for  $n_T$  we'll get

$$n_T = \sum_{i=A}^Z n_i = \sum_{i=A}^Z (n_{i0} + \nu_i \xi) = n_{T0} + \xi \sum_{i=A}^Z \nu_i, \quad (2.12)$$

where  $n_{T0}$  is the total *inlet* molar flow rate (the subscript 0 indicates a variable associated with the inlet stream). Sums over species are from A to Z in this text instead of the usual 1 to  $N$  because species are most commonly abbreviated with letters when dealing with about a dozen or fewer species. The sum of stoichiometric coefficients  $\sum \nu_i$  is often denoted by the variable  $\delta$  such that

$$\delta \equiv \sum_{i=A}^Z \nu_i, \quad (2.13)$$

and therefore the total outlet molar flow rate  $n_T$  can be written as

$$n_T = n_{T0} + \delta \xi. \quad (2.14)$$

Combining these expressions for  $n_i$  and  $n_T$  allows us to write any of the mole fractions in

Table 2.1: A basic stoichiometric or “stoich” table for the generic reaction  $aA + bB \rightleftharpoons cC + dD$ . Below the bottom rule are several quantities which can be calculated as the sum of the column above. Throughout this text we’ll modify the columns to different problems and reactors.

Sp.	$\nu_i$	$n_{i0}$ (mol · s <sup>-1</sup> )	$n_i = n_{i0} + \nu_i \xi$ (mol · s <sup>-1</sup> )	$y_i = n_i / n_T$
A	$-a$	$n_{A0}$	$n_A = n_{A0} - a\xi$	$y_A = n_A / n_T$
B	$-b$	$n_{B0}$	$n_B = n_{B0} - b\xi$	$y_B = n_B / n_T$
C	$c$	$n_{C0}$	$n_C = n_{C0} + c\xi$	$y_C = n_C / n_T$
D	$d$	$n_{D0}$	$n_D = n_{D0} + d\xi$	$y_D = n_D / n_T$
$\delta = \sum \nu_i \quad n_{T0} = \sum n_{i0} \quad n_T = \sum n_i = n_{T0} + \delta \xi \quad 1 = \sum y_i$				

terms of  $\xi$ . For example, the gas-phase mole fraction of species  $i$  can be written as

$$y_i = \frac{n_i}{n_T} = \frac{n_{i0} + \nu_i \xi}{n_{T0} + \delta \xi}. \quad (2.15)$$

Understandably there’s quite a bit to keep track of here, and we haven’t even gotten to the thermodynamics yet. Fortunately there’s a clever way of organizing this in tabular form that we’ll continue to use throughout this text because of its amazing utility. Such a table goes by several names like “magic table” or “table of wonder,” both of which are wonderful names that you’re encouraged to use whenever you like. We’ll formally refer to this table as a *stoichiometric table*, or sometimes just the abbreviated “stoich” table.

The basic components of a stoich table are shown in Table 2.1. The first two columns—species and stoichiometric coefficients—are included in nearly all stoichiometric tables, but any columns beyond these two are subject to modification based on the problem at hand. Column headers sometimes include expressions to remind us of what’s being calculated in that column but other times only the variable itself is included. Also, additional quantities such as  $n_{T0}$  or  $n_T$  can be included for convenience below the bottom rule.

The decision of what information to include or exclude from a stoich table is based on personal preference and experience. Throughout this book we’ll modify the stoich table’s columns to better summarize data for different problem specifications or reactor types, and hopefully along the way the habit of making a stoich table at the beginning of a problem will be so deeply ingrained that it becomes automatic.

## 2.4 $\partial G/\partial \xi = 0$ at equilibrium

With these basics now in mind, let's apply some of what we learned in calculus to the problem of minimizing Gibbs energy. Recall from your thermodynamics course that the total derivative of Gibbs energy is

$$dG = V dP - S dT + \sum_{i=A}^Z \mu_i dn_i, \quad (2.16)$$

where  $\mu_i$  is the chemical potential of species  $i$ . For constant  $n_{i0}$  the derivative of  $\xi$  according to Equation (2.9) is

$$dn_i = \nu_i d\xi. \quad (2.17)$$

and substituting Equation (2.17) into Equation (2.16) yields

$$dG = V dP - S dT + \sum_{i=A}^Z \nu_i \mu_i d\xi$$

The partial derivative of this expression with respect to  $\xi$  at constant temperature and pressure is

$$\left( \frac{\partial G}{\partial \xi} \right)_{T,P} = \sum_{i=A}^Z \nu_i \mu_i.$$

and since calculus tells us that the derivative of  $G$  is zero at its minimum we can set this expression equal to zero as

$$\sum_{i=A}^Z \nu_i \mu_i = 0 \quad (2.18)$$

which is a neatly compact way of stating the requirement for equilibrium. What we're going to do next is work on transforming this expression to get it in a form that contains things we can easily calculate, but it's going to look like we're making things worse because we're going to end up with quite a few more terms than we've got here.

A generally more useful term than chemical potential is the fugacity  $f$ , defined for species  $i$  as

$$d\mu_i \equiv RT d \ln f_i,$$

which admittedly doesn't look like we've done ourselves any favors because it seems that we've only further complicated things by introducing derivatives and logarithms. However, inte-

grating from the pure-component standard state to the actual (mixture) state as

$$\int_{G_i^\circ}^{\mu_i} d\mu_i = RT \int_{f_i^\circ}^{\hat{f}_i} d\ln f_i$$

yields

$$\mu_i = G_i^\circ + RT \ln \frac{\hat{f}_i}{f_i^\circ}, \quad (2.19)$$

where  $G_i^\circ$  is the pure-component Gibbs energy of species  $i$  in its standard state,  $f_i^\circ$  is the pure-component fugacity of species  $i$  in its standard state, and  $\hat{f}_i$  is the fugacity of species  $i$  in the actual (mixture) state. Substituting Equation (2.19) into Equation (2.18) yields

$$\sum_{i=A}^Z \nu_i (G_i^\circ + RT \ln a_i) = 0, \quad (2.20)$$

where for simplicity we've replaced the ratio of fugacities  $\hat{f}_i/f_i^\circ$  for species  $i$  with the activity  $a_i$  for species  $i$ . We can further rearrange to

$$\sum_{i=A}^Z \ln a_i^{\nu_i} = \frac{-\sum_{i=A}^Z \nu_i G_i^\circ}{RT}. \quad (2.21)$$

The general form of the sum on the right-hand side of Equation (2.21) is going to show up so many times that it's helpful to come up with a shorthand notation for it as

$$\Delta M \equiv \sum_{i=A}^Z \nu_i M_i \quad (2.22)$$

where  $M_i$  is any property of species  $i$  such as heat capacity, enthalpy, or Gibbs energy. It's often convenient to include columns for each  $M_i$  in the stoich table and to calculate  $\Delta M$  for each property  $M$  beneath the bottom rule as in Table 2.2. Applying the notation of Equation (2.22) to Equation (2.21) and noting that the sum of logs is equal to the log of the products we arrive at

$$\prod_{i=A}^Z a_i^{\nu_i} = \exp \left[ \frac{-\Delta G^\circ}{RT} \right]. \quad (2.23)$$

Equation (2.23) is boxed because this is the equation we'll come back to over and over when it's time to calculate equilibrium compositions. A reactor whose composition is dictated solely

Table 2.2: The general stoich table of Table 2.1 modified to include species properties  $M_i$ . The row beneath the bottom rule is used to calculate  $\Delta M$  quantities but be careful not to confuse the simple sums of quantities like  $n_{T0}$  or  $n_T$  with the  $\Delta M$  sum-product expression in Equation (2.22).

Sp.	$\nu_i$	$n_{i0}$	$n_i$	$y_i$	$M_i$
A	$-a$	$n_{A0}$	$n_A$	$y_A$	$M_A$
B	$-b$	$n_{B0}$	$n_B$	$y_B$	$M_B$
C	$c$	$n_{C0}$	$n_C$	$y_C$	$M_C$
D	$d$	$n_{D0}$	$n_D$	$y_D$	$M_D$
$\delta$ $n_{T0}$ $n_T$					$\Delta M$

by Equation (2.23) is called an *equilibrium reactor*.

As noted early, our journey from Equation (2.18) to Equation (2.23) appears to have made life more complicated because we've got more terms to deal with, but the advantage of this new form is that the right-hand side is a function of temperature only; the left-hand side is a function of temperature, pressure, and composition; and all of these quantities are readily measured. We're still a little ways off, however, and the remaining journey is long enough that it's helpful to split it into two parts: first we'll look at the right-hand side of Equation (2.23), and then the left-hand side.

## 2.5 The equilibrium constant $K_a$ is a function of temperature

The term on the right-hand side of Equation (2.23) is called the *equilibrium constant* and is denoted by the symbol  $K_a$ , or

$$K_a \equiv \exp \left[ \frac{-\Delta G^\circ}{RT} \right], \quad (2.24)$$

where  $\Delta G^\circ$  is the Gibbs energy of reaction at any *absolute* temperature  $T$ ,  $R$  is the gas constant, and the superscript  $^\circ$  means that a quantity is evaluated at the standard state (a specified pressure such as 101.325 kPa or 1 bar, a specified state of aggregation such as liquid or gas, *and* a specified temperature  $T$ ).

The name “equilibrium constant” is somewhat misleading because it's obviously a function of temperature:  $T$  clearly appears in the denominator, and somewhat less obviously  $\Delta G^\circ$  is evaluated  $T$ . This one temperature—the  $T$  in the denominator of Equation (2.24) and the temperature at which  $\Delta G^\circ$  is evaluated—is *the temperature at which the reactor operates*.

To actually calculate a number for  $K_a$  we'll follow the notation introduced by Smith and

Van Ness<sup>2</sup>, who elegantly split  $K_a$  into three contributions as

$$K_a = K_0 K_1 K_2, \quad (2.25)$$

where

$$K_0 = \exp \left[ \frac{-\Delta G_{T_R}^\circ}{RT_R} \right] \quad (2.26)$$

$$K_1 = \exp \left[ \frac{\Delta H_{T_R}^\circ}{R} \left( \frac{1}{T_R} - \frac{1}{T} \right) \right] \quad (2.27)$$

$$K_2 = \exp \left[ -\frac{1}{T} \int_{T_R}^T \frac{\Delta C_P^\circ}{R} dT + \int_{T_R}^T \frac{\Delta C_P^\circ}{R} \frac{dT}{T} \right] \quad (2.28)$$

In this text Equation (2.25) will be referred to as the *full van't Hoff equation* because it was developed using the van't Hoff equation. In Equation (2.26) through Equation (2.28) the reference temperature  $T_R$  is nearly always 298.15 K because everyone sat down one day and picked room temperature (25 °C) to be the value at which we'd tabulate values of the Gibbs energies and enthalpies of formation from which we calculate  $\Delta G_{T_R}^\circ$  and  $\Delta H_{T_R}^\circ$ . For simplicity you can almost always use  $T_R = 298$  K and not sacrifice significant accuracy.

Each of the  $K_0$ ,  $K_1$ , and  $K_2$  terms is associated with some step of estimating  $K_a$ :

- $K_0$  gets us started and is nearly always evaluated using the Gibbs energies of formation as

$$\Delta G_{T_R}^\circ = \sum_{i=A}^Z \nu_i \Delta G_{f,i}^\circ,$$

where the Gibbs energies of the elements in their naturally occurring states at  $T_R$  was arbitrarily chosen as a reference state for  $\Delta G_{f,i}^\circ$ .

- $K_1$  is the first and numerically the larger of the two adjustments of  $K_0$  to a temperature  $T$ .  $K_1$  is nearly always evaluated using the enthalpies of formation as

$$\Delta H_{T_R}^\circ = \sum_{i=A}^Z \nu_i \Delta H_{f,i}^\circ$$

In evaluating  $K_1$  it's assumed that  $\Delta H_{T_R}^\circ$  is independent of temperature.

- $K_2$  is the second and numerically the smaller of the two adjustments of  $K_0$  to a temperature  $T$ , and it's a correction to the assumption that  $\Delta H_{T_R}^\circ$  is temperature-independent.

<sup>2</sup>"Introduction to Chemical Engineering Thermodynamics" 6th ed. McGraw-Hill, New York, NY (2001).

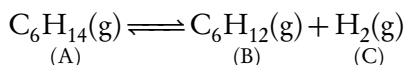
For plenty of common reactions we can ignore the correction of  $K_2$  by setting it equal to unity, which is equivalent to assuming that the enthalpy of reaction is independent of temperature. Doing so simplifies Equation (2.25) to

$$K_a \approx K_0 K_1 \equiv K_{sc} \quad (2.29)$$

which we'll refer to as the *shortcut van't Hoff equation* because it's faster to use Equation (2.29) than Equation (2.25) when doing hand calculations. Whenever we need to calculate  $K_a$  in this textbook we'll *always* use Equation (2.29), but sometimes we'll call the result  $K_a$  and sometimes we'll call the result  $K_{sc}$  (the subscript *sc* stands for "shortcut").

### Example 2.1: Using the shortcut van't Hoff equation

Use the shortcut van't Hoff equation to find  $K_a$  at 650 K for the gas-phase dehydrogenation of *n*-hexane ( $\text{C}_6\text{H}_{14}$ ) to 1-hexene ( $\text{C}_6\text{H}_{12}$ ) and hydrogen ( $\text{H}_2$ ) according to the reaction



### Solution

The enthalpies and Gibbs energies of formation of each species can be found in Appendix D and are summarized in the table below.

Abbr.	$\nu_i$	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )	$\Delta G_f^\circ$ (kJ · mol <sup>-1</sup> )
A	-1	-166.9	-0.1
B	1	-41.7	87.0
C	1	0.0	0.0
		$\Delta M = 125.2$	87.1

The two  $\Delta$  values under the last table row are the enthalpy of reaction ( $\Delta H_{T_R}^\circ$ ),

$$\begin{aligned}\Delta H_{T_R}^\circ &= \sum_{i=A}^C \nu_i \Delta H_{f,i}^\circ = \nu_A \Delta H_{f,A}^\circ + \nu_B \Delta H_{f,B}^\circ + \nu_C \Delta H_{f,C}^\circ \\ &= (-1)(-166.9) + (1)(-41.7) + (1)(0) \\ &= 125.2 \text{ kJ} \cdot \text{mol}^{-1}.\end{aligned}$$

and the Gibbs energy of reaction ( $\Delta G_{T_R}^\circ$ ),

$$\begin{aligned}\Delta G_{T_R}^\circ &= \sum_{i=A}^C \nu_i \Delta G_{f,i}^\circ = \nu_A \Delta G_{f,A}^\circ + \nu_B \Delta G_{f,B}^\circ + \nu_C \Delta G_{f,C}^\circ \\ &= (-1)(-0.1) + (1)(87) + (1)(0) \\ &= 87.1 \text{ kJ} \cdot \text{mol}^{-1},\end{aligned}$$

We now substitute these into Equation (2.26) to get  $K_0$  as

$$\begin{aligned}K_0 &= \exp \left[ \frac{-\Delta G_{T_R}^\circ}{RT_R} \right] = \exp \left[ \frac{-87100}{(8.314)(298.15)} \right] \\ &= 5.494 \times 10^{-16},\end{aligned}$$

where the gas constant  $R = 8.314 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$  was used and the units of  $\Delta G_{T_R}^\circ$  were adjusted appropriately. Similarly, we can use Equation (2.27) to get  $K_1$  as

$$\begin{aligned}K_1 &= \exp \left[ \frac{\Delta H_{T_R}^\circ}{R} \left( \frac{1}{T_R} - \frac{1}{T} \right) \right] \\ &= \exp \left[ \frac{125,200}{8.314} \left( \frac{1}{298.15} - \frac{1}{650} \right) \right] \\ &= 2.477 \times 10^{11}.\end{aligned}$$

Finally, we use Equation (2.29) to approximate  $K_a$  as

$$\begin{aligned}K_a &\approx K_{sc} = K_0 K_1 \\ &= (5.494 \times 10^{-16})(2.477 \times 10^{11}) \\ &= 4.108 \times 10^{-4}.\end{aligned}$$

The shortcut van't Hoff equation can also be manipulated to provide a means of relating



$K_{sc}$  at one temperature to  $K_{sc}$  at any other temperature. The ratio of any two  $K_{sc}$  evaluated at any two temperatures  $T_1$  and  $T_2$  is, according to Equation (2.29),

$$\frac{K_{sc}(T_2)}{K_{sc}(T_1)} = \frac{K_0(T_2)K_1(T_2)}{K_0(T_1)K_1(T_1)}. \quad (2.30)$$

From Equation (2.26) we note that  $K_0$  is not a function of temperature and therefore  $K_0(T_1) = K_0(T_2)$ . From Equation (2.27) we note that both enthalpies  $\Delta H$  in  $K_1$  are measured from the same reference temperature  $T_R$ , and therefore with a little algebraic manipulation we can simplify Equation (2.30) to

$$\frac{K_{sc}(T_2)}{K_{sc}(T_1)} = \exp \left[ \frac{\Delta H}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right], \quad (2.31)$$

which will prove useful whenever we're given one value of  $K_{sc}$  and need to calculate another  $K_{sc}$  at some new temperature.

## 2.6 Chemical activities are related to composition

The reason we chose to use the subscript  $a$  on  $K_a$  as opposed to some other randomly selected symbol is because the left-hand side of Equation (2.23) involves the product of activities  $a_i$  raised to stoichiometric coefficients  $\nu_i$ , which we now see as related to  $K_a$  as

$$\prod_{i=A}^Z a_i^{\nu_i} = \exp \left[ \frac{-\Delta G^\circ}{RT} \right] = K_a \approx K_0 K_1 = K_{sc}, \quad (2.32)$$

which is essentially a “revised” version of Equation (2.23) because we've included our definition of  $K_a$  from Equation (2.24) and the shortcut approximation of  $K_{sc}$ . The subscript  $a$  in  $K_a$  is meant to remind of us the  $\prod a_i^{\nu_i}$  term in Equation (2.23), and we'll use this kind of subscripting—a letter to remind of us a variable—several more times. As with Equation (2.23), any reactor whose outlet composition is determined solely by Equation (2.32) is called an *equilibrium reactor*. Also keep in mind that since all of our derivations in Section 2.4 assumed constant temperature and pressure, any equilibrium reactor is necessarily operated at constant temperature and pressure.<sup>3</sup>

There are many different expressions for  $a_i$  depending on the state of aggregation and

<sup>3</sup>In practical terms, this means the inlet temperature and pressure are the same as the outlet temperature and pressure.

Table 2.3: Species activities  $a_i$  for use in Equation (2.24) for low to moderate pressures.  $x_i$  and  $y_i$  are mole fractions,  $P_i$  is partial pressure,  $M_i$  is molality ( $\text{mol} \cdot \text{kg}^{-1}$ ), and  $M_i^\circ$  is a hypothetical, 1 molal ideal solution.

Mixture state	$a_i$	Comment
Ideal gas	$\frac{P_i}{P^\circ} = \frac{y_i P}{P^\circ} = \frac{y_i P}{101.325 \text{ kPa}}$	$P^\circ$ in this text is 101.325 kPa but is also commonly 1 bar.
Aqueous solution	$\frac{M_i}{M_i^\circ} = \frac{M_i}{1 \text{ molal}}$	1 molal = $1 \text{ mol} \cdot \text{kg}^{-1}$ . In dilute aqueous solutions, 1 molal is approximately 1 molar ( $\text{mol} \cdot \text{L}^{-1}$ ).
Pure liquid	$x_i$	
Solid	1	

your choice of reference state,<sup>4</sup> but the only four relevant to anything we'll be doing in the rest of the book are summarized in Table 2.3. There's a fair amount we're overlooking in this "discussion" of activity so go dig up your favorite thermodynamics textbook if you need a more in-depth refresher of this material.

Briefly, any time we need an activity of a species in solution we start off with the general relationship

$$a_i = \frac{\hat{f}_i}{f_i^\circ},$$

where (as in Equation (2.19)) the  $\hat{f}_i$  is the fugacity of species  $i$  in the actual (mixture) state and  $f_i^\circ$  is the pure-component fugacity of species  $i$  in its standard state. For example, for the ideal gas mixture the standard state is the ideal, pure gas at the temperature of interest and 101.325 kPa (or 1 bar, depending on your preference) such that

$$f_i^\circ = (101.325 \text{ kPa})\phi,$$

where  $\phi$  is the pure-component fugacity coefficient and

$$\hat{f}_i = y_i P \hat{\phi}_i,$$

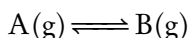
where  $\hat{\phi}_i$  is the fugacity coefficient of species  $i$  in solution. In an ideal gas mixture  $\hat{\phi}_i = \phi = 1$ ,

<sup>4</sup>See Table 13.1-2 in S.I. Sandler. Chemical, Biochemical, and Engineering Thermodynamics, 4th edition. Wiley, NJ (2006).

so we have

$$a_i = \frac{\hat{f}_i}{f_i^\circ} = \frac{y_i P \hat{\phi}_i}{(101.325 \text{ kPa}) \phi} = \frac{y_i P}{101.325 \text{ kPa}},$$

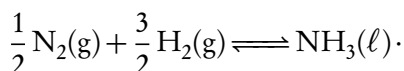
which is what appears in Table 2.3. The choice of which activity to use is dictated by how you write your chemical reaction: whichever phase you define in the reaction is the phase you should use when picking activities from Table 2.3. As a simple example, the reaction



implies the activities of both A and B should be those of ideal gases such that the equilibrium constant is

$$K_a = \prod_{i=\text{A}}^{\text{Z}} a_i^{v_i} = \frac{a_{\text{B}}}{a_{\text{A}}} = \frac{\frac{y_{\text{B}} P}{(101.325 \text{ kPa})}}{\frac{y_{\text{A}} P}{(101.325 \text{ kPa})}} = \frac{y_{\text{B}}}{y_{\text{A}}},$$

which should look familiar from your previous experience with equilibrium constants. Ever so conveniently the pressure  $P$  and standard state pressure  $P^\circ$  cancel out but that's not always the case. As a more complicated example, the production of liquid ammonia from nitrogen and hydrogen can be written as



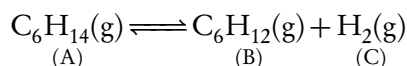
In this reaction,  $\text{N}_2(\text{g})$  and  $\text{H}_2(\text{g})$  have been specified as gas-phase, so we use  $a_i = y_i P / (101.325 \text{ kPa})$  for both of these components (which implicitly assumes that they're behaving as ideal gases, as all gases do in this text). Ammonia has been specified as a liquid so we use  $a_i = x_i$  for that one, which means our equilibrium constant  $K_a$  is

$$K_a = \prod_{i=\text{A}}^{\text{Z}} a_i^{v_i} = \frac{a_{\text{NH}_3(\ell)}}{a_{\text{N}_2(\text{g})}^{1/2} a_{\text{H}_2(\text{g})}^{3/2}} = \frac{x_{\text{NH}_3(\ell)}}{y_{\text{N}_2(\text{g})}^{1/2} y_{\text{H}_2(\text{g})}^{3/2}} \left( \frac{101.325 \text{ kPa}}{P} \right)^2.$$

Here we see the pressure obviously didn't cancel out so we'll have to be careful to include it in units of kPa whenever working with this expression. Since (by definition)  $K_a$  is only a function of temperature, we can see from this example that changes in pressure must be "balanced" by changes in composition in order to keep  $K_a$  unchanged. This is the origin of what you probably learned in an early chemistry course as "LeChatelier's Principle."

**Example 2.2: Relating activity to concentration**

Use Table 2.3 to expand  $\prod a_i^{\nu_i}$  in Equation (2.23) for the reaction given in Example 2.1,

**Solution**

Each of the species are in the gas phase so all activities for this example will be

$$a_i = \frac{y_i P}{P^\circ},$$

where  $P^\circ = 101.325 \text{ kPa}$ . Applying this to the product term in Equation (2.23) yields

$$K_a = \prod_{i=A}^C a_i^{\nu_i} = \frac{a_B a_C}{a_A} = \frac{y_B y_C}{y_A} \frac{P}{P^\circ}.$$

## 2.7 The Equilibrium Reactor converts a feed to its equilibrium composition

Given a temperature and pressure we can use Equation (2.32) to solve for an equilibrium composition by expressing each of the activities in terms of a single unknown.<sup>5</sup> Whenever you use the forms of Table 2.3 you'll inevitably be left with some combination of mole fractions,  $x_i$  or  $y_i$ , so our goal is to relate the mole fractions of all species to a single variable, then solve for that variable using Equation (2.32).

In terms of mathematical convenience the best choice is usually the extent of reaction  $\xi$ , defined previously as

$$\xi = \frac{n_i - n_{i0}}{\nu_i}. \quad (2.9)$$

The key feature of  $\xi$  is that for a given reaction it's the same regardless of which species is denoted by  $i$ , so we can relate all outlet molar flow rates  $n_i$  to the same  $\xi$ , substitute those into Equation (2.32), then solve for  $\xi$ . The corresponding outlet molar flow rates and mole fractions will then represent the *equilibrium* composition for the given feed, temperature, and

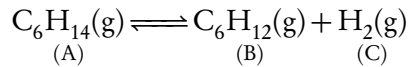
<sup>5</sup>Remember, it's only one equation so it can be used to solve for at most one variable.

pressure, and a reactor which achieves such composition is called an *Equilibrium reactor*.

It's important to keep in mind that the equilibrium reactor is, in many cases, a “best case scenario” in that a real reactor is subject to limitations not included in the development of Equation (2.32). We introduce these limitations in the next chapter but always keep in mind that the Equilibrium reactor is a great place to start because it can give a quick estimate of the operational limits of a real reactor. Let's finish up our running example and then look at a few more examples of the Equilibrium reactor.

### Example 2.3: Finding the outlet of an Equilibrium Reactor

Suppose  $1.5 \text{ mol} \cdot \text{s}^{-1}$  pure *n*-hexane is provided to an equilibrium reactor operating 650 K and 0.5 MPa, and the only reaction that occurs is the dehydrogenation of *n*-hexane to 1-hexene according to the reaction



Determine the outlet mole fractions of all species.

### Solution

This is the same system as described in Example 2.1 and Example 2.2 so we'll be able to re-use many of the calculations. Let's start off with the stoich table to organize the available information:

Table 2.4: Initial stoichiometric table for Example 2.3, which includes information provided in Example 2.1 and 2.2.  $\Delta H_f^\circ$  and  $\Delta G_f^\circ$  have units of  $\text{kJ} \cdot \text{mol}^{-1}$ ;  $n_{i0}$  and  $n_i$  have units of  $\text{mol} \cdot \text{s}^{-1}$ . Remember that entries below the bottom rule are calculated differently.

Sp.	$\nu_i$	$n_{i0}$	$n_i = n_{i0} + \nu_i \xi$	$\Delta H_f^\circ$	$\Delta G_f^\circ$
A	-1	1.5	$1.5 - \xi$	-166.9	-0.1
B	1	0	$\xi$	-41.7	87.0
C	1	0	$\xi$	0.0	0.0
<hr/>					
$\delta = 1$	$n_{T0} = 1.5$	$n_T = 1.5 + \xi$	$\Delta H_{T_R}^\circ = 125.2$	$\Delta G_{T_R}^\circ = 87.1$	

In the previous two examples we saw that Equation (2.32) can be written as

$$\prod_{i=A}^C a_i^{\nu_i} = K_a$$

$$\frac{y_B y_C}{y_A} \frac{P}{P^\circ} = 4.108 \times 10^{-4}.$$

We now replace each mole fraction  $y_i$  with  $n_i/n_T$  according to Equation (2.15) to get

$$\frac{n_B n_C}{n_A} \frac{1}{n_T} \frac{P}{P^\circ} = 4.108 \times 10^{-4}.$$

The stoich table allows us to quickly substitute each  $n_i$  as functions of  $\xi$  to get

$$\frac{(\xi)(\xi)}{(1.5 - \xi)(1.5 + \xi)} \frac{1}{P^\circ} \frac{P}{P^\circ} = 4.108 \times 10^{-4},$$

or after a little bit of simplification,

$$\frac{\xi^2}{(1.5 - \xi)(1.5 + \xi)} \frac{500 \text{ kPa}}{101.325 \text{ kPa}} = 4.108 \times 10^{-4}.$$

Using our favorite calculator or computer program we find that  $\xi = 0.0138$ . Substituting this value back into our expressions for each  $y_i$  yields  $y_A = 0.9819$  and  $y_B = y_C = 0.0091$ . Similarly the outlet molar flow rates for A, B, and C are  $1.4862 \text{ mol} \cdot \text{s}^{-1}$ ,  $0.0138 \text{ mol} \cdot \text{s}^{-1}$ , and  $0.0138 \text{ mol} \cdot \text{s}^{-1}$ .

The approach used in Example 2.1 through Example 2.3 is the approach you should generally follow whenever you need to solve an equilibrium reactor “by hand,” such as on an exam. On the other hand we’ve got an exceptionally powerful tool in the form of MATLAB that we can use to help check our work. A great advantage of coding in MATLAB is that you can avoid doing quite a bit of algebra, and by extension you can avoid making quite a few algebraic mistakes. The next example demonstrates how to use MATLAB to do the same calculation as in Example 2.3 but in such a way that the code more closely resembles the equations, which is quite useful when it comes to finding and correcting errors in your code or logic.

#### Example 2.4: Using `fsolve` to solve an Equilibrium Reactor

Repeat Example 2.3 but set up and solve the problem using MATLAB’s `fsolve` func-

tion.

## Solution

**Approach 1:** *Solve the equation from Example 2.3.*

The equation that we developed at the end of Example 2.3 was

$$\frac{\xi^2}{(1.5 - \xi)(1.5 + \xi)} \frac{500 \text{ kPa}}{101.325 \text{ kPa}} = 4.108 \times 10^{-4}.$$

which we can solve in MATLAB by arranging it in root-finding form as an anonymous function, then passing it to `fsolve` as we did in Section 1.5. This takes only two lines of code so it's convenient to do this from the command line as

```
% Script segment
xiSol = fsolve(@myFun, 0.5);

% Local function
function out = myFun(xi)
out = xi.^2./((1.5-xi).*(1.5+xi))*500/101.325 - 4.108e-4;
end
```

Running this script produces `xiSol=0.0138` which is the same as we found in Example 2.3. However, this approach has a significant disadvantage in that it's difficult to check our work: what if there was an algebraic mistake when we substituted  $n_i/n_T$  for  $y_i$ ? What if our pressure terms were incorrectly flipped upside down? What if the temperature used for  $K_a$  was incorrect? It's possible to check each of these but it's also something of a headache to do since we've got to go back to each of our hand calculations for each one. Let's see how we can use MATLAB to do better.

**Approach 2:** *Solve the entire problem within MATLAB.*

The first thing to do is to figure out which equation we're trying to solve and rewrite it in root-finding form. For this problem that's Equation (2.32), which in root-finding form is

$$0 = K_a - \prod_{i=A}^Z a_i^{v_i}.$$

We're essentially going to guess values of  $\xi$  until this equation is satisfied, but we're going to let `fsolve` do all the guessing. The general syntax of a script to solve this problem will look like this:

```

1 % Eq. dehydrogenation of n-hexane
2 % Script portion
3 xi_guess = 0.5; % guess a solution value
4 xiSol = fsolve(@myFun, xi_guess) % solve
5
6 % Local function
7 function output = myFun(input)
8 xi = input; % current value of xi
9
10 % Constants
11
12 % Algebraic equations
13
14 % Root-finding equation
15 output = Ka - aB*aC/aA;
16 end

```

Notice that this is the same general format that we introduced in Section 1.5: a script portion on top that calls `fsolve` followed by a local function that contains the root-finding form of the equation we’re trying to solve and the variable we’re trying to solve for as an input to the function.

This is now the key step where a MATLAB approach splits from an “algebraic” (or “by-hand”) approach: instead of replacing the output variable with the algebraic expression we derived in Example 2.3, let’s start at the bottom of the local function `myFun` and work backwards, defining each variable and constant in its most general form. For example, each activity can be rewritten as  $a_i = y_i P / P^\circ$ , so let’s add these equations to the code under the “Algebraic equations” portion:

```

12 % Algebraic equations
13 aA = yA*P/Pcirc;
14 aB = yB*P/Pcirc;
15 aC = yC*P/Pcirc;
16
17 % Root-finding equation
18 output = Ka - aB*aC/aA;
19 end

```

Let’s keep going: each mole fraction  $y_i$  can be expressed generally as  $n_i/n_T$ , so our growing code becomes

```

12 % Algebraic equations
13 yA = nA/nT;
14 yB = nB/nT;
15 yC = nC/nT;

```



```

16 aA = yA*P/Pcirc;
17 aB = yB*P/Pcirc;
18 aC = yC*P/Pcirc;
19
20 % Root-finding equation
21 output = Ka - aB*aC/aA;
22 end

```

Notice that we put our equations of  $y_i$  above the equations for  $a_i$  because MATLAB will need values of  $y_i$  before it can calculate  $a_i$ . The finished code is as follows:

```

1 % Eq. dehydrogenation of n-hexane
2 xi_guess = 0.5;
3 xi = fsolve(@myFun, xi_guess);
4 fprintf('xi = %.3f\n', xi);
5
6 function out = myFun(input)
7 xi = input; % mol/s
8 % Constants
9 nA0 = 1.5; % mol/s
10 nB0 = 0; % mol/s
11 nC0 = 0; % mol/s
12 nuA = -1;
13 nuB = 1;
14 nuC = 1;
15 P = 500; % kPa
16 Pcirc = 101.325; % kPa
17 Ka = 4.108e-4;
18
19 % Algebraic equations
20 nA = nA0+nuA*xi; % mol/s
21 nB = nB0+nuB*xi; % mol/s
22 nC = nC0+nuC*xi; % mol/s
23 nT = nA+nB+nC; % mol/s
24 yA = nA/nT;
25 yB = nB/nT;
26 yC = nC/nT;
27 aA = yA*P/Pcirc;
28 aB = yB*P/Pcirc;
29 aC = yC*P/Pcirc;
30
31 % Root-finding equation
32 out = Ka - aB*aC/aA;
33 end

```

Running this code gives  $\xi = 0.0138$ , the same value we found in Example 2.3. It's also possible to use MATLAB's vector math to greatly simplify the above code so that far

fewer lines have to be written, something which is described in Appendix B.

Why do all the extra coding of Example 2.4 if we can do the algebra in a few steps as in Example 2.3? Because *it's easy to catch and correct mistakes*. The format of Example 2.4 clearly lays out each step of the calculation in a way that's easy for us to check *only that calculation*, and if we find a mistake it'll be easy for us to correct it because we need correct only that line.

## 2.8 Equilibrium conversion is the conversion at equilibrium

Although the extent of reaction  $\xi$  is the mathematically more convenient variable to use with equilibrium reactors, we'll see that for real reactors a more convenient choice is *conversion*,  $X$ , which unlike  $\xi$  is always defined for a *species i* instead of a *reaction*. We typically calculate conversions based on the limiting reactant and by convention we label this as species A, and based on these conventions the conversion of species A is the same as it was in your material and energy balances course:

$$X_A = \frac{A \text{ in} - A \text{ out}}{A \text{ in}} = \frac{n_{A0} - n_A}{n_{A0}} = 1 - \frac{n_A}{n_{A0}}. \quad (2.33)$$

Again, keep in mind that the subscript A was chosen merely by convention; you can always calculate the conversion of any species by changing the subscripts everywhere in Equation (2.33).

If we calculate  $X_A$  for an equilibrium reactor then we refer to this conversion as the *equilibrium conversion* and denote it as  $X_A^{\text{eq}}$ , where the *eq* superscript stands for “equilibrium.”

### Example 2.5: Calculating the equilibrium conversion using $\xi$

Find the equilibrium conversion of *n*-hexane (species A) using the results from Example 2.3.

#### Solution

From Example 2.3 the inlet and outlet molar flow rates of *n*-hexane are  $1.5000 \text{ mol} \cdot \text{s}^{-1}$  and  $1.4862 \text{ mol} \cdot \text{s}^{-1}$ . From Equation (2.33) the equilibrium conversion of A is

$$X_A^{\text{eq}} = 1 - \frac{n_A}{n_{A0}} = 1 - \frac{1.4862}{1.5} = 0.92\%$$

Another alternative to first solving for  $\xi$  and then solving for  $X_A^{\text{eq}}$  is to substitute Equa-

tion (2.33) into Equation (2.10), rearrange to express all  $n_i$  in terms of  $X_A$  instead of  $\xi$ , and then proceed with the general approach of Section 2.7 but solving for  $X_A$  instead of  $\xi$ . In particular, Equation (2.10) becomes

$$n_i = n_{i0} - n_{A0} \frac{\nu_i}{\nu_A} X_A, \quad (2.34)$$

and if we factor out the  $n_{A0}$  term we can get

$$n_i = n_{A0} \left( \Theta_i - \frac{\nu_i}{\nu_A} X_A \right), \quad (2.35)$$

where

$$\Theta_i \equiv \frac{n_{i0}}{n_{A0}} \quad (2.36)$$

which was defined just to make Equation (2.35) look a little neater. Summing up all the  $n_i$  yields

$$n_T = \sum_{i=A}^Z n_i = n_{T0} - n_{A0} \frac{\delta}{\nu_A} X_A, \quad (2.37)$$

and so the mole fractions become

$$y_i = \frac{n_i}{n_T} = \frac{n_{A0} \left( \Theta_i - \frac{\nu_i}{\nu_A} X_A \right)}{n_{T0} - n_{A0} \frac{\delta}{\nu_A} X_A}. \quad (2.38)$$

Perhaps now it's clear why  $\xi$  was previously mentioned as the mathematically most convenient variable: compared to Equation (2.15) in terms of  $\xi$ , Equation (2.38) in terms of  $X_A$  looks bulkier and less user-friendly because there are more terms floating around. The reason for writing it at all is that we're going to use  $X_A$  so much in the rest of this book that Equation (2.38) will become convenient because we'll already have most of the terms written or calculated for other parts of the problem. Using either form will always give the same result when using Equation (2.32).<sup>6</sup>

### Example 2.6: Calculating equilibrium conversion using $X_A$

Reformulate Equation (2.32) from Example 2.3 in terms of conversion instead of extent of reaction. Solve for the equilibrium conversion and compare your answer to what was found in Example 2.5.

<sup>6</sup>Assuming your math is correct, of course.

**Solution**

Once again Equation (2.32) starts life as

$$\prod_{i=1}^3 a_i^{\nu_i} = K_a$$

$$\frac{y_B y_C}{y_A} \frac{P}{P^\circ} = 4.108 \times 10^{-4}.$$

and we again replace each mole fraction with  $n_i/n_T$  to get

$$\frac{n_B n_C}{n_A} \frac{1}{n_T} \frac{P}{P^\circ} = 4.108 \times 10^{-4}.$$

Let's now use Equation (2.35) to express each  $n_i$  in terms of  $X_A$ . According to Equation (2.36) we have  $\Theta_A = 1$  and  $\Theta_B = \Theta_C = 0$ ; using this and evaluating each stoichiometric ratio yields

$$\begin{aligned} n_A &= n_{A0}(1 - X_A) \\ n_B &= n_{A0}(0 + X_A) \\ n_C &= n_{A0}(0 + X_A) \\ n_T &= n_{T0} + n_{A0}X_A = n_{A0}(1 + X_A). \end{aligned}$$

where we noticed in  $n_T$  that for this problem we happen to have  $n_{T0} = n_{A0}$ . We can also take a moment to notice that outlet molar flow rates in terms of  $X_A$  can be used in stoichiometric tables in much the same way as  $\xi$ :

Sp.	$\nu_i$	$n_{i0}$	$n_i$	$\Delta H_f^\circ$	$\Delta G_f^\circ$
A	-1	1.5	$n_{A0}(1 - X_A)$	-166.9	-0.1
B	1	0	$n_{A0}X_A$	-41.7	87.0
C	1	0	$n_{A0}X_A$	0.0	0.0
<hr/>					
$\delta = 1$		$n_{T0} = 1.5$	$n_{A0}(1 + X_A)$	$\Delta H_{T_R}^\circ = 125.2$	$\Delta G_{T_R}^\circ = 87.1$

Substituting each of these into the previous equation yields

$$\frac{X_A^2}{(1 - X_A)(1 + X_A)} \frac{P}{P^\circ} = 4.108 \times 10^{-4},$$

which we can solve to get  $X_A = 0.0091$ . Not surprisingly, this is almost identical to what was found in Example 2.5 and the only reason for discrepancy is that the numbers were rounded differently throughout the two calculations. Retaining more digits of precision throughout both calculations would improve the agreement.



# Chapter 3

## Rate Laws

“You’re just not thinking fourth dimensionally!”

Dr. Emmett “Doc” Brown, *Back to the Future Part III* (1990)

We’re nearly to the point where we can start to introduce material that’s truly new, not just a review or revision of stuff you learned in your material and energy balances course or your thermodynamics course. Up to now the one element we haven’t considered is *time*—how long does it take for a chemical reaction to occur? This is directly related to several other important questions: What kind of reactor should we use? How big should the reactor be? How much will it cost?

Our discussion of equilibrium from Chapter 2 cannot give us an answer to these questions because all of the analysis we did in that chapter simply assumed that the system was already at equilibrium when we found it, or it treated the reactor as a “black box” that somehow magically achieved equilibrium. This chapter will broadly review some concepts of rate laws that you’re probably familiar with from general chemistry, and in subsequent chapters we’ll apply this information to answer these—and several other—questions.

### 3.1 Rate laws must be determined experimentally

The simple observation that all rate laws must be determined experimentally is so central, so crucial to our understanding of rate laws that it bears repeating yet a third time, and this time in a box so that you know we’re serious:

Rate laws must be determined experimentally!
--

Why is that the case? Why is it that with all our computing power, with all our knowledge of quantum mechanics and statistical mechanics and all that jazz, that we're still stuck making crude experimental measurements? Because this stuff is really complicated, that's why,<sup>1</sup> and at the moment the best we can do is push back the bounds of our ignorance. We have a pretty good idea of what's necessary in a chemical reaction but the process of actually predicting the rate at which those reactions occur almost inevitably requires us to measure *something* about the reaction itself.

Any mathematical form of a rate law tries to embody the three requirements generally interpreted as necessary for a chemical reaction to occur:

1. A molecule must collide with something, usually another reacting molecule or a catalyst surface.
2. Collisions must have sufficient energy to overcome repulsion forces.
3. Molecules must collide in the correct orientation.

Two of the most common physical models we use to capture these requirements are “collision theory” and “transition state theory,” but further details of these models are left to your chemistry courses.<sup>2</sup> Nearly any model that attempts to predict a rate law generally starts off with an idea or conceptual understanding of how a chemical reaction should proceed, applies as much of our understanding of physics and chemistry as possible to create predictions based on that model, and ends up with—among many other things—an expression containing one or more coefficients which must be determined experimentally.

Somewhat related to models—in the sense that they have unknown coefficients and that we're not going to look at them in this textbook—are the development of reaction *mechanisms*. It's quite rare for a reaction to proceed from start to finish in a single step such as  $A \longrightarrow B$ ; instead there are often several intermediate steps which, when enumerated, constitute a reaction mechanism. A reaction mechanism builds on a reaction model by listing the many steps hypothesized for a molecule of A to become a molecule of B, using the model to develop a rate law from these steps that can be measured experimentally, and then someone tries to measure it experimentally to see if it's accurate or not.

As with many other aspects of chemistry and physics, in this textbook we're not particularly concerned with how such models and mechanisms are developed, although we'll have a little bit of a discussion at the end of the book about how to determine some of the unknown

---

<sup>1</sup>Don't ask so many questions, Nosey McNoseypants.

<sup>2</sup>Or check out the excellent textbook by D.A. McQuarrie; J.D. Simon, *Physical Chemistry: A Molecular Approach*. (1997) University Science Books, CA.



coefficients previously mentioned. Instead, we're concerned with how these models impact material and energy balances written for reacting systems. For simplicity we'll restrict our attention almost entirely to a single kind of rate law—the power-law form—because the way in which we incorporate the rate laws into our balance equations doesn't change regardless of the rate law's complexity or form. The power-law form is quite general so if you're feeling like you're missing out on some kind of deeper or more powerful form of rate laws, you're really not: with the exception of biological reactions, the power-law form is the heart of a vast swath of reaction rate law theory.<sup>3</sup>

## 3.2 The power-law form is a common rate law

A power-law form of a rate law  $r$  for some hypothetical reaction has the general feature of a temperature-dependent “constant” multiplied by some combination of concentrations raised to powers, or

$$r = k(T)[A]^\alpha[B]^\beta, \quad (3.1)$$

where  $k(T)$  is the temperature-dependent constant,  $[A]$  and  $[B]$  are concentrations of A and B, and  $\alpha$  and  $\beta$  are constants (and they're actually constants in the sense that they're not temperature-dependent like  $k$ ). There's no requirement or restriction on the number of terms that can be involved in a rate like Equation (3.1), so something like

$$r = k_f[A][B] - k_r[C][D]$$

is also a power-law form. There's also no requirement or restriction that the exponents  $\alpha$  and  $\beta$  be integers, so something like

$$r = k[A]^{0.5}[B]^{1.5} \quad (3.2)$$

is also a power-law form. The powers  $\alpha$  and  $\beta$  indicate the *order* of the reaction in each species, so Equation (3.2) is order 0.5 in A and order 1.5 in B. The *overall order* of the reaction is simply the sum of the species orders and is usually denoted by the variable  $m$  as

$$\text{overall reaction order} = m = \alpha + \beta. \quad (3.3)$$

For example, the overall order of the rate given in Equation (3.2) is  $m = 0.5 + 1.5 = 2$ , or second order.

For whatever reason it's common in chemistry to see the concentration terms written us-

---

<sup>3</sup>Those biologists and bioengineers have some really neat expressions for biological rate laws; you're encouraged to check them out some time.

ing square brackets as  $[A]$  or  $[B]$  but within chemical engineering this notation is rare. Instead, you're more likely to come across one of these three notations:

**Mole fractions:** In older publications you'll see mole fractions like  $x_i$  or  $y_i$  used directly in the rate law, so you could see things like

$$r = k x_A x_B$$

where  $x_A$  and  $x_B$  are the mole fractions of A and B. Most modern texts don't use mole fractions but since the rates we need were sometimes measured back in the '50s or '60s it's helpful to at least be aware that this is a possibility.

**Partial pressures:** When dealing with reactions that occur in the gas-phase, particularly those involving gas phase species interacting with solid catalysts, a widely adopted convention is to use partial pressures ( $P_i = y_i P$ ) as the "concentration" terms in the rate law, like

$$r = k P_A P_B$$

where  $P_A$  and  $P_B$  are the partial pressures of A and B. Common units of partial pressure include bar, kPa, and psia. We'll bump into this notation again when we investigate tubular reactors starting in Chapter 7.

**Molar concentrations:** This is the most common expression for concentrations in rate laws, at least at present.<sup>4</sup> Molar concentrations are denoted by  $C_i$  and work for gas or liquid phase species, so you'll usually see things like

$$r = k C_A C_B$$

where  $C_A$  and  $C_B$  are the molar concentrations of A and B. The most common units of  $C_i$  are ones like  $\text{mol} \cdot \text{L}^{-1}$  and  $\text{kmol} \cdot \text{m}^{-3}$ , although any combination of  $[\text{amount}][\text{volume}]^{-1}$  is acceptable. **Going forward, we'll assume molar concentrations for all rate laws unless explicitly noted otherwise.**

### 3.3 The units of rate laws can be tricky

The discussion above should make it clear that units can get pretty wonky in rate laws, especially when dealing with multiple phases or non-integer reaction orders. The key idea to keep

---

<sup>4</sup>One day this notation will undoubtedly look just as outdated as the mole fractions do now.

in mind is that the rate  $r$  always has the same general set of units, namely

$$r \doteq \frac{\text{mol reacted}}{\text{volume} \times \text{time}}, \quad (3.4)$$

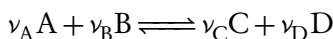
where the symbol  $\doteq$  means “has units of...”, and the unit “mol reacted” is the same—admittedly unusual—unit that we use for the extent of reaction,  $\xi$ . A “mol reacted” means that a reaction has proceeded once (or a mol times, or  $10^{23}$  times, if you prefer either of these interpretations), so if the reaction is



then a “mol reacted” for this reaction means 2 mol A have been consumed, 3 mol B have been consumed, and 1 mol C has been generated. We usually leave off the word “reacted” so you’ll commonly see the units of Equation (3.4) written as

$$r \doteq \frac{\text{mol}}{\text{volume} \times \text{time}}. \quad (3.5)$$

One trick that will help us write material and energy balances is that given the generic reaction



we can use the stoichiometric coefficients ( $\nu_i$ ) to relate  $r$  to the rates of individual species as

$$r = \frac{r_{\text{A}}}{\nu_{\text{A}}} = \frac{r_{\text{B}}}{\nu_{\text{B}}} = \frac{r_{\text{C}}}{\nu_{\text{C}}} = \frac{r_{\text{D}}}{\nu_{\text{D}}} = \frac{r_i}{\nu_i}, \quad (3.6)$$

where  $r_i$  is the rate of reaction *for species  $i$*  and has units of

$$r_i \doteq \frac{\text{mol } i \text{ generated or consumed}}{\text{volume} \times \text{time}}. \quad (3.7)$$

Notice that in Equation (3.6) the value of  $r$  will *always* be positive when the reaction proceeds (as written) in the forward direction, but the rates of *reactants* (A and B) will be *negative* and the rates of *products* (C and D) will be *positive*.

Most references will give you the units on two of the three quantities in a rate law— $r$ ,  $k$ , or  $C_i$ —and leave it up to you to figure out any remaining units. With more complex rates you’ll usually get the units of  $r$  and  $C_i$  because the units of  $k$  can get really strange really fast. In general you should avoid trying to convert units of  $k$  to keep from making unnecessarily complex unit conversions, which are naturally more prone to error.

**Example 3.1: Finding units in a rate law**

Find the units of  $k$  for the rate law

$$r = kP_A^2$$

if partial pressure is measured in bar and  $r$  has units of  $\text{mol} \cdot \text{L}^{-1} \cdot \text{s}^{-1}$ .

**Solution**

Rearranging the rate to isolate  $k$  yields

$$k = \frac{r}{P_A^2}$$

so  $k$  has units of

$$k \doteq \frac{\text{mol}}{\text{L} \cdot \text{s} \cdot \text{bar}^2}.$$

There are of course exceptions to the general concept of Equation (3.4): the biologists and bioengineers in particular like to use mass units because it's sometimes difficult to estimate molar masses of things like cells, so they might write Equation (3.4) as

$$r \doteq \frac{\text{mass reacted}}{\text{volume} \times \text{time}}. \quad (3.8)$$

If you happen to have the molar mass of whatever it is that's reacting then you can convert between the two units, but we'll stick mostly to the molar units of Equation (3.4) within this book.

**3.4 The rate constant  $k$  follows an Arrhenius form**

The rate constant  $k(T)$  that appears in Equation (3.1) is commonly expressed as the product of a constant and an exponential term,

$$k(T) = A \exp \left[ \frac{-E_a}{RT} \right], \quad (3.9)$$

where the constant  $A$  is called the pre-exponential factor<sup>5</sup>,  $E_a$  is the activation energy for the reaction,  $R$  is the gas constant, and  $T$  is the absolute temperature. The notation  $k(T)$  is to remind you that  $k$  is a function of  $T$  but often you'll see the rate constant written simply as  $k$ .

The Arrhenius form of the rate constant should look familiar from one of your chemistry courses and its interpretation is the same as it was there, so if you'd like to review the whole "activation energy is like an energy barrier" discussion there are lots of good resources to look through. Roughly speaking,  $A$  is a factor which corresponds to the probability that two molecules collide with the correct orientation, and the exponential term represents the probability (or fraction) of a collision with sufficient energy to allow the reaction to proceed.

Regarding units, Equation (3.9) implies that the units of  $A$  are the same as those of  $k$ , and that the quantity  $E_a/RT$  must be dimensionless.<sup>6</sup> As described in Section 3.3 the units of  $k$  can get wonky, so you should expect the units of  $A$  to be equally wonky. The most common units of  $E_a$  are  $\text{kJ} \cdot \text{mol}^{-1}$  or a similar SI combination, so you should expect to use quantities of  $R$  given in Appendix D and temperature in kelvin.

Given  $A$  and  $E_a$  it's a simple matter to use Equation (3.9) to calculate  $k$  at any temperature.

### Example 3.2: Using the Arrhenius equation

Plot  $k(T)$  from 25 °C to 200 °C if  $A = 0.002 \text{ s}^{-1}$  and  $E_a = 21 \text{ kJ} \cdot \text{mol}^{-1}$ .

#### Solution

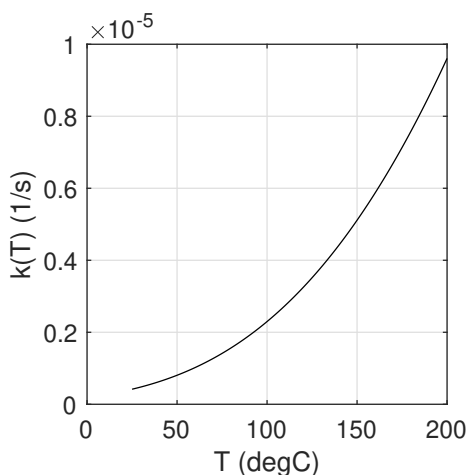
A simple MATLAB script will get the job done here:

```
1 T = linspace(25, 200) + 273.15; % K
2 Ea = 21e3; % J/mol
3 R = 8.314; % J/mol-K
4 A = 0.002; % 1/s
5 k = A.*exp(-Ea./(R.*T)); % 1/s
6 plot(T-273.15, k)
```

Notice that the temperature was converted to kelvin to calculate  $k$ , then converted back to °C for plotting. The resulting figure is shown below.

<sup>5</sup>We're not exactly clever with some of our naming conventions in this field.

<sup>6</sup>Recall that the argument of any transcendental function-like  $\exp[\ ]$ —must be dimensionless. Any time you can legitimately use the word "transcendental" you're encouraged to so.



Example 3.2 illustrates an important feature of the rate constant  $k$ : it will *always* increase with temperature because the arguments of the exponential— $E_a$ ,  $R$ , and  $T$ —are always positive. This means that if a reaction is irreversible then you'll always get a large rate  $r$  with a temperature increase. We'll soon see that the situation is somewhat more complicated for a reversible reaction because the rate depends on the equilibrium constant  $K_a$ , and  $K_a$  can increase or decrease with temperature depending on whether the reaction is endothermic or exothermic.

Equation (3.9) can also be used to adjust  $k$  from one temperature to another provided you know  $E_a$  and at least one value of  $k$  at some temperature  $T_1$ . If we take the ratio of two  $k$  values at different temperatures we get

$$\frac{k(T_2)}{k(T_1)} = \frac{A \exp\left[\frac{-E_a}{RT_2}\right]}{A \exp\left[\frac{-E_a}{RT_1}\right]} = \exp\left[\frac{E_a}{R} \left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]. \quad (3.10)$$

Not surprisingly, the reason for writing  $k(T)$  in this way is exactly for the reason already mentioned: occasionally you'll know  $k$  at some temperature  $T_1$  along with  $E_a$  (a constant), which means you can use Equation (3.10) to calculate  $k$  at any other temperature without having to know  $A$ . The alternative procedure is to use Equation (3.9) twice, first to get  $A$  using  $k(T_1)$  and  $E_a$ , then again with the calculated value of  $A$  to get  $k(T_2)$ . The latter process takes one extra calculation step compared to the former but provided both approaches are done correctly you'll end up with the same number for  $k(T_2)$ .

The reason we wrote Equation (3.10) the way we did is because it's structurally quite sim-

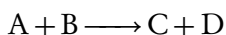
ilar to the way we adjusted  $K_{sc}$  from  $T_1$  to  $T_2$  in Chapter 2 as

$$\frac{K_{sc}(T_2)}{K_{sc}(T_1)} = \exp \left[ \frac{\Delta H}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right]. \quad (2.31)$$

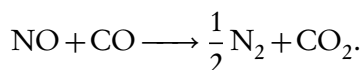
Since we’ll make regular use of these two equations it’s useful to have them in the same general form so that we have to remember a little bit less.

### 3.5 Reactions and rate laws are sometimes called “elementary”

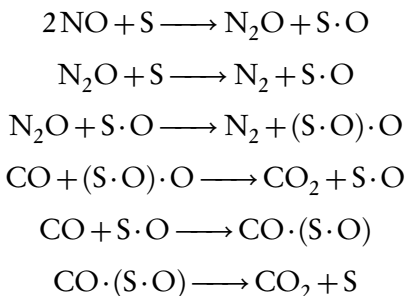
There are surprisingly few reactions which actually occur the way we write them! It’s easy to fall into the trap of interpreting a chemical reaction written as



to mean that when one A molecule collides with one B molecule, there’s a chance that the result is their immediate transformation to one C molecule and one D molecule. For example, consider the reduction of nitrogen monoxide (NO) by carbon monoxide (CO) in the presence of an iron oxide catalyst,

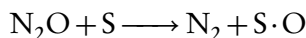


We call this reaction the *overall reaction* because it’s the one that shows the overall stoichiometry of products and reactants, and the apparent or observed rate at which this reaction occurs is called the *observed rate law*. However, we know the reaction can’t actually occur in a single step *exactly* as written here because we know there’s no such thing as half a molecule of  $\text{N}_2$ , but even if you meddle with the stoichiometry—say, by multiplying all the stoichiometric coefficients by 2—a one-step interpretation still won’t be accurate! The reason is because the actual reaction occurs in (at least) six steps:<sup>7</sup>



<sup>7</sup>H. Randall, R. Doepper, A. Renken. Model discrimination by unsteady-state operation: application to the reduction of NO with CO on iron oxide. Can. J. Chem. Eng. 74 (1996) 586.

In these reactions, S is a “site,” which is a location on a catalyst’s surface at which a molecule can physically adsorb, and combinations like  $S \cdot O$  mean that a molecule like O is adsorbed onto a site. The collection of reactions which describe the actual pathway by which the product is made is called a *reaction mechanism*, and each reaction within a mechanism is called a *reaction step*. The key difference between a reaction step and an overall reaction is that it’s assumed that a reaction step will occur *exactly* as written, so the reaction step

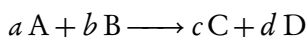


implies that whenever a  $N_2O$  molecule collides with a site S, the result will be one molecule of  $N_2$  and one atom of oxygen adsorbed to a surface site as  $S \cdot O$ .

Whenever we come across (or propose) a reaction which occurs in exactly one step—that is, it occurs exactly as written—we call that reaction an *elementary reaction*. Even more importantly, we always write the rate of an elementary reaction to be equal to the product of the reactant concentrations raised to their respective stoichiometric coefficients, multiplied by a temperature-dependent rate constant  $k$ . This rate we call an *elementary rate law*, and for our previous reaction step example we’d write the elementary rate law as

$$r = k C_{N_2O} C_S.$$

More generally, for any reaction written as



we write the elementary rate law as

$$r = k C_A^a C_B^b.$$

The connection between the elementary rate laws of mechanism steps and the observed rate law for the overall reaction is not easy to determine and must be found experimentally. Often several mechanisms are proposed, a series of experiments performed, and the mechanism which most accurately represents the data is deemed “the” mechanism, whether or not that mechanism can provide a neat and easily written overall rate law.

The approach we take in this book is to simplify the whole situation as follows: **you should assume the rate law is elementary unless otherwise indicated!** If the rate law isn’t elementary then we call the rate law *non-elementary*<sup>8</sup> and the rate law will be given to you.

---

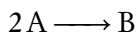
<sup>8</sup>Clever, yes?



We will often use phrasing such as, “Assuming the rate law is elementary...” or “Given the elementary reaction...” both of which tell you to write the corresponding rate as an elementary rate.

### Example 3.3: Writing an elementary rate law

Assume the reaction



follows an elementary rate law in terms of molar concentration ( $C$ ). Write the rate law for the reaction as well as for A and B.

### Solution

The elementary rate law for the reaction is

$$r = kC_{\text{A}}^2,$$

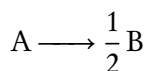
the rate of reaction for A is

$$r_{\text{A}} = \nu_{\text{A}} r = -2kC_{\text{A}}^2,$$

and the rate of reaction for B is

$$r_{\text{B}} = \nu_{\text{B}} r = kC_{\text{A}}^2.$$

Notice that if the stoichiometry in Example 3.3 were changed to



then the rate law also changes to

$$r = kC_{\text{A}},$$

which is fundamentally different from the rate as written in Example 3.3 because the exponent on  $C_{\text{A}}$  has changed from 2 to 1. This highlights just how imperative it is that you **do not change the stoichiometry before you write the rate!** You can mess around with stoichiometric coefficients to your heart’s content but you have to wait until *after* you write the reaction rate to do so.

### 3.6 Individual rates are summed to give net rates

When more than one reaction occurs in a reactor we need a way to add up the cumulative effects of all the reactions on the generation or consumption of a species. For example, suppose the following reactions occur in a reactor:



Let's also suppose that in a period of 1 s you're somehow able to observe that the first reaction occurs three times, the second reaction occurs six times, and the third reaction occurs just once. To find the net change in A molecules during that 1 s period we can simply add up all the changes we observed to get

$$\begin{pmatrix} \text{net change} \\ \text{of A} \\ \text{particles in} \\ 1 \text{ s} \end{pmatrix} = \begin{pmatrix} 3 \text{ molecules} \\ A \\ \text{consumed} \\ \text{in R1} \end{pmatrix} + \begin{pmatrix} 12 \\ \text{molecules} \\ A \\ \text{consumed} \\ \text{in R2} \end{pmatrix} + \begin{pmatrix} 1 \text{ molecule} \\ A \text{ generated} \\ \text{in R3} \end{pmatrix}$$

Our use of the words “consumed” and “generated” imply negative signs in front of the 3 and 12 with a positive sign in front of the 1, so we have

$$\begin{pmatrix} \text{net change} \\ \text{of A} \\ \text{particles in} \\ 1 \text{ s} \end{pmatrix} = -3 - 12 + 1 = -14,$$

so we see that there's a net consumption of A equal to 14 A molecules in 1 s.

The way we went about calculating each term was to look at how many times each reaction occurred, then multiply this number by the quantity of A consumed or generated each time the reaction occurs. The latter number is the stoichiometric coefficient of A in each reaction, and if we let any reaction be represented by the indexing variable  $j$  then we can rewrite our

word equation as

$$\left( \begin{array}{c} \text{net change} \\ \text{of A} \\ \text{particles in} \\ 1 \text{ s} \end{array} \right) = \sum_{j=1}^q \left[ \left( \begin{array}{c} \text{number of} \\ \text{times reaction } j \\ \text{occurs in 1 s} \end{array} \right) \times \left( \begin{array}{c} \text{stoichiometric} \\ \text{coefficient of A} \\ \text{in reaction } j \end{array} \right) \right]$$

where  $q$  is the number of reactions we're working with, in this case 3. Finally, let's replace the phrase "net change of A in 1 s" with "net rate of A" because there really was no reason to pick a time window of just 1 s as opposed to any other number, and similarly we'll replace "number of times reaction  $j$  occurs in 1 s" with "rate of reaction  $j$ ." These changes lead us to

$$\left( \begin{array}{c} \text{net rate} \\ \text{of A} \end{array} \right) = \sum_{j=1}^q \left[ \left( \begin{array}{c} \text{rate of} \\ \text{reaction} \\ j \end{array} \right) \times \left( \begin{array}{c} \text{stoichiometric} \\ \text{coefficient of A} \\ \text{in reaction } j \end{array} \right) \right]$$

We now make our last generalization: there was no particular reason to choose species A instead of any other species, so we can replace A with any arbitrary species as represented by the letter  $i$  to get

$$\left( \begin{array}{c} \text{net rate} \\ \text{of } i \end{array} \right) = \sum_{j=1}^q \left[ \left( \begin{array}{c} \text{rate of} \\ \text{reaction} \\ j \end{array} \right) \times \left( \begin{array}{c} \text{stoichiometric} \\ \text{coefficient of } i \\ \text{in reaction } j \end{array} \right) \right] \quad (3.11)$$

which is a general expression for the net rate of  $i$  which participates in multiple reactions. We can rewrite Equation (3.11) with the usual symbols to yield a more useful equation for the net rate of any species as

$$\boxed{r_i = \sum_{j=1}^q \nu_{ij} r_j} \quad (3.12)$$

where  $r_i$  is the net rate of species  $i$ ,  $r_j$  is the rate of reaction  $j$ , and  $\nu_{ij}$  is the stoichiometric coefficient of species  $i$  in reaction  $j$ . Pay particular attention to the difference between  $r_i$  and  $r_j$  because it's easy to confuse them: the net rate of species  $i$  can be written as  $r_i$  for any of the  $Z$  species in the reactor, whereas the rate of reaction for reaction  $j$  can be written as  $r_j$  for any of the  $q$  reactions occurring in the reactor.

The reason this equation is boxed is because we will always come back to this equation whenever we're dealing with more than one reaction, regardless of what reactor we're studying. In fact, there's no reason we can't *always* start with Equation (3.12) when analyzing a

reactor because it will always be correct, even if there's only one reaction of interest. In the case of just one reaction then Equation (3.12) reduces to a rearranged form of Equation (3.6).

### Example 3.4: Writing net rates

Assume each of the following elementary reactions take place in a reactor:



Write the net rates of A, B, and C assuming the rates are written as molar concentrations.

### Solution

We're of course going to start by writing Equation (3.12) in its general form,

$$r_i = \sum_{j=1}^q \nu_{ij} r_j,$$

and to get an idea of what size system we're working with, we'll note that there are  $N = 3$  species (A, B, and C) and  $q = 3$  reactions (R1, R2, and R3). We'll refer to the reactions as 1, 2, and 3 just so we don't have to carry around the extra "R." We can also write each of the three elementary rate laws as

$$r_{j=1} = r_1 = k_1 C_A$$

$$r_{j=2} = r_2 = k_2 C_A^2$$

$$r_{j=3} = r_3 = k_3 C_B C_C$$

Let's start writing net rates with species A since that's first in the alphabet. The net rate of A is

$$\begin{aligned} r_{i=A} = r_A &= \sum_{j=1}^3 \nu_{Aj} r_j \\ &= \nu_{A1} r_1 + \nu_{A2} r_2 + \nu_{A3} r_3 \end{aligned}$$

We now replace each of the  $\nu_{A_j}$  with the stoichiometric coefficients for A in each of the reactions, which for reaction R1, R2, and R3 are  $-1$ ,  $-2$ , and  $+1$ :

$$\begin{aligned} r_A &= \nu_{A1} r_1 + \nu_{A2} r_2 + \nu_{A3} r_3 \\ &= -r_1 - 2r_2 + r_3 \end{aligned}$$

Finally, we substitute the rates of all three reactions as we wrote them earlier to yield

$$r_A = -k_1 C_A - 2k_2 C_A^2 + k_3 C_B C_C,$$

and we refer to this equation as the net rate of A. We can perform similar processes for species B to get

$$\begin{aligned} r_{i=B} = r_B &= \sum_{j=1}^3 \nu_{Bj} r_j \\ &= \nu_{B1} r_1 + \nu_{B2} r_2 + \nu_{B3} r_3 \\ &= r_1 + 0 - r_3 \\ &= k_1 C_A - k_3 C_B C_C \end{aligned}$$

and for species C to get

$$\begin{aligned} r_{i=C} = r_C &= \sum_{j=1}^3 \nu_{Cj} r_j \\ &= \nu_{C1} r_1 + \nu_{C2} r_2 + \nu_{C3} r_3 \\ &= 0 + r_2 - r_3 \\ &= k_2 C_A^2 - k_3 C_B C_C \end{aligned}$$

### 3.7 Rates of reversible reactions can be related to $K_a$

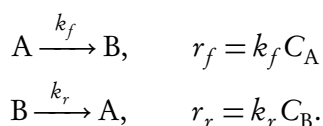
We explored reversible reactions all throughout Chapter 2 and they're going to continue to be a topic of discussion because they're so prevalent. Recall that when we defined the reaction  $A \rightleftharpoons B$  to be at equilibrium we were careful to state that the forward and reverse reactions

never stopped, but rather that they became equal such that the net rate of consumption of A (or generation of B) was zero.<sup>9</sup>

To write the rate of a reversible reaction we can split it into two parts, the forward reaction and the reverse reaction. One way to denote these two reactions and their corresponding rate constants is by using the letters  $f$  and  $r$ , but you may also come across texts that use nomenclature such as  $+1$  and  $-1$  or  $f$  and  $b$  (for backwards); they're all equivalent. Thus we can rewrite the reversible reaction



as the set of (elementary) reactions



Assuming equilibrium has been achieved we know  $r_f = r_r$ , or

$$k_f C_A = k_r C_B,$$

which can be rearranged to yield

$$\frac{k_f}{k_r} = \frac{C_B}{C_A}$$

We use the special symbol  $K_C$  for the ratio  $k_f/k_r$  when concentrations are written as molar concentrations, and we call it an equilibrium constant:

$$K_C \equiv \frac{k_f}{k_r}. \quad (3.14)$$

We could have simply defined Equation (3.14) right from the outset of this section, but we chose to show this derivation so that you can see where the idea came from. Ultimately, however, this is simply a definition and it's not predicated on the idea that the system must be at equilibrium, which means we can use it whenever we want. We make this definition because it's a useful way to replace the reversible rate constant  $k_r$ —which is unknown—to the equilibrium constant  $K_a$ —which is generally assumed to be known. To show this relationship, first we'll relate the rate  $r$  for Equation (3.13) to  $K_C$ , and then we'll relate  $K_C$  to our other equilibrium constant  $K_a$ .

<sup>9</sup>Refer back to Figure 2.1 if this idea isn't clear.

Let's tackle the easy one first, the relationship of  $K_C$  to the rate  $r$  for Equation (3.13). The net rate of Equation (3.13) is simply the difference of the forwards and reverse reactions, or

$$r = r_f - r_r \quad (3.15)$$

Remember that we're no longer restricting ourselves to equilibrium and that's why we don't set this equation equal to zero. Next, we substitute elementary rate laws to get

$$r = k_f C_A - k_r C_B. \quad (3.16)$$

Now we rearrange Equation (3.14) to get  $k_r = k_f / K_C$ , which we substitute into Equation (3.16) to yield

$$r = k_f \left( C_A - \frac{C_B}{K_C} \right). \quad (3.17)$$

The reason Equation (3.17) is preferable over Equation (3.16) is because it has only one unknown constant ( $k_f$ ) instead of two, which means when it comes time to experimentally investigate this reaction we have one less variable to figure out. If for some reason you happen to know the forward and reverse rate constants you can still use Equation (3.16) as the rate, but most of the time we'll be in a position where we only know the forward rate constant and therefore we'll make use of Equation (3.17).

For the more generalized reversible reaction



we can extend this same analysis to write the rate as

$$r = k \left( C_A^a C_B^b - \frac{C_C^c C_D^d}{K_C} \right), \quad (3.19)$$

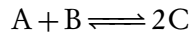
where

$$K_C = \frac{C_C^c C_D^d}{C_A^a C_B^b} = \prod_{i=A}^Z C_i^{\nu_i}. \quad (3.20)$$

Notice that we've dropped the subscript  $f$  on  $k_f$  in Equation (3.19) because it's usually implied that the only known rate constant is the forward rate constant. You can use the general form of Equation (3.19) to write the rate of most any reversible reaction.

**Example 3.5: Writing a reversible rate**

For the reaction



write  $r$ ,  $K_C$ ,  $r_A$ ,  $r_B$ , and  $r_C$  in terms of at most the variables  $k$  and any  $C_i$ .

**Solution**

Starting with Equation (3.19) we have

$$r = k \left( C_A C_B - \frac{C_C^2}{K_C} \right)$$

where

$$K_C = \frac{C_C^2}{C_A C_B}.$$

To get the rates of A and B we go back to Equation (3.6) to see that we need simply multiply  $r$  by any stoichiometric coefficient to get the corresponding species rate  $r_i$ , so we have

$$\begin{aligned} r_A &= \nu_A r = -k \left( C_A C_B - \frac{C_C^2}{K_C} \right), \\ r_B &= \nu_B r = -k \left( C_A C_B - \frac{C_C^2}{K_C} \right), \\ r_C &= \nu_C r = 2k \left( C_A C_B - \frac{C_C^2}{K_C} \right). \end{aligned}$$

Now let's do the last part, relating  $K_C$  to  $K_a$ . Just by inspecting the two general expressions for both variables,

$$K_a = \prod_{i=A}^Z a_i^{\nu_i}, \quad (2.32)$$

and

$$K_C = \prod_{i=A}^Z C_i^{\nu_i}, \quad (3.20)$$



we see that the only difference between the two is that we've replaced all the  $a_i$  in Equation (2.32) with  $C_i$  in Equation (3.20). The next step is therefore to link  $C_i$  to  $a_i$ , and we'll divide our approach based on whether we're dealing with liquids or gases.

### 3.7.1 For pure liquids $K_C = K_a C_T^\delta$

For liquids the molar concentration  $C_i$  is related to the total molar concentration  $C_T$  by the liquid mole fraction  $x_i$  as

$$C_i = x_i C_T, \quad (3.21)$$

where  $C_T = \sum C_i$ . Substituting Equation (3.21) into Equation (3.20) yields

$$\begin{aligned} K_C &= \prod_{i=A}^Z C_i^{\nu_i} \\ K_C &= \prod_{i=A}^Z (x_i C_T)^{\nu_i} \\ K_C &= \prod_{i=A}^Z x_i^{\nu_i} \prod_{i=A}^Z C_T^{\nu_i} \\ K_C &= K_a C_T^\delta, \end{aligned} \quad (3.22)$$

where in the last line we identified  $x_i = a_i$  using Table 2.3 and

$$\prod_{i=A}^Z C_T^{\nu_i} = C_T^c C_T^d C_T^{-a} C_T^{-b} = C_T^{c+d-a-b} = C_T^\delta$$

which used the definition of  $\delta$  from Equation (2.13) in the last equality.

Importantly, **watch your units!**  $K_a$  is a dimensionless quantity but  $K_C$  has different units based on the value of  $\delta$ .

### 3.7.2 For ideal gases $K_C = K_a (P^\circ/RT)^\delta$

For ideal gases the molar concentration  $C_i$  is related to the species partial pressure  $P_i$  as

$$P_i = C_i RT, \quad (3.23)$$

which we substitute into Equation (3.20) to yield

$$K_C = \prod_{i=A}^Z C_i^{\nu_i} = \prod_{i=A}^Z \left( \frac{P_i}{RT} \right)^{\nu_i}. \quad (3.24)$$

We now take the somewhat unusual step of multiplying the term in parentheses by  $P^\circ/P^\circ = 1$  to get

$$\begin{aligned} K_C &= \prod_{i=A}^Z \left( \frac{P_i}{RT} \frac{P^\circ}{P^\circ} \right)^{\nu_i} \\ K_C &= \prod_{i=A}^Z \left( \frac{P_i}{P^\circ} \right)^{\nu_i} \prod_{i=A}^Z \left( \frac{P^\circ}{RT} \right)^{\nu_i} \\ K_C &= K_a \left( \frac{P^\circ}{RT} \right)^\delta, \end{aligned} \quad (3.25)$$

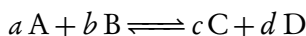
where in the last line we identified  $P_i/P^\circ = a_i$  using Table 2.3 and we used the definition of  $\delta$  as we did with liquids.

Importantly, **watch your units!**  $K_a$  is a dimensionless quantity but as with liquids  $K_C$  has different units based on the value of  $\delta$ . Worse for gases, the units of  $P^\circ$  are often different than the units specified in a problem so you'll have to check carefully your units in Equation (3.25).

### 3.8 Reversible rates can be written in terms of partial pressures

As mentioned in Section 3.2 it's somewhat commonplace for gas-phase reaction rates to be written in terms of partial pressures, and that means we have to make a few minor adjustments to the expressions from Section 3.7. Fortunately all of the general steps are the same so we'll just skip to the end results.

For the generic, reversible, gas-phase reaction



the rate in terms of partial pressures is

$$r = k \left( P_A^a P_B^b - \frac{P_C^c P_D^d}{K_p} \right),$$

where

$$K_p = \frac{P_C^c P_D^d}{P_A^a P_B^b} = \prod_{i=A}^Z P_i^{\nu_i}. \quad (3.26)$$

The variable  $K_p$  is yet another equilibrium constant and can be related to  $K_a$  by following the

same approach as in Section 3.7.2 to yield

$$K_P = K_a (P^\circ)^\delta.$$

Once again, **watch your units!**  $K_a$  is a dimensionless quantity but the units of  $K_P$  depend on the value of  $\delta$ .

We've done a fair amount of work on reversible reactions, their rates, and how they relate to the equilibrium constant  $K_a$  and it would certainly be understandable if it was all a bit fuzzy at first! The relevant relationships have been summarized in Table 3.1 to help you quickly identify the information you'll need when presented with a reversible reaction.

Table 3.1: Summary table for commonly used reaction rates for the generic reversible reaction  $aA + bB \rightleftharpoons cC + dD$ .  $C_T = \sum C_i$ ,  $\delta = \sum \nu_i$ .

Conc. variable	Rate expression	Phase	Equilibrium constant
$C_i$	$r = k \left( C_A^a C_B^b - \frac{C_C^c C_D^d}{K_C} \right)$	liquid	$K_C = \prod C_i^{\nu_i} = K_a C_T^\delta$
		gas	$K_C = \prod C_i^{\nu_i} = K_a (P^\circ / RT)^\delta$
$P_i$	$r = k \left( P_A^a P_B^b - \frac{P_C^c P_D^d}{K_P} \right)$	gas	$K_P = \prod P_i^{\nu_i} = K_a (P^\circ)^\delta$



## Chapter 4

# The Continuously Stirred Tank Reactor, or CSTR

“Stirred, not shaken.”

Dikko Henderson, offering James Bond the wrong kind of martini. *You Only Live Twice* (1967)

We finally get to start analyzing a real reactor! Let’s start with the good stuff right away: the general form of a **material balance** for the continuously stirred tank reactor (CSTR) is

$$n_i = n_{i0} + r_i V \quad (4.1)$$

and the general form of the **energy balance** for the CSTR is

$$0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q - W_s \quad (4.2)$$

The reason we start off with these two equations is because you should know that you can *always* come back to these two equations to solve *any* CSTR problem in this book. We’re certainly going to derive these expressions so that you can understand their limitations and assumptions, but try to keep in mind that every problem that follows, including every special CSTR case we’ll see in Chapter 5, are all applications, rearrangements, or simplifications of Equation (4.1) and Equation (4.2).

The first thing we’ll do is derive the material and energy balances for the CSTR, which we refer to as the *design equations* for the reactor, followed by a brief discussion of the  $Q$  and  $W_s$ ,

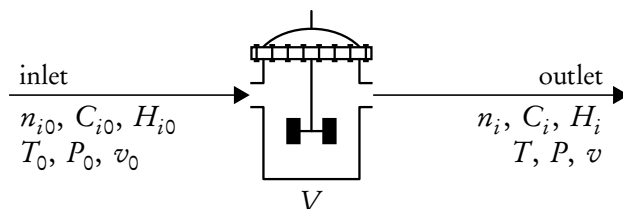


Figure 4.1: Basic diagram and variables associated with the CSTR where  $n_i$  is molar flow rate of component  $i$ ,  $C_i$  is molar concentration (mole per volume),  $H_i$  is molar enthalpy of component  $i$ ,  $T$  is temperature,  $P$  is pressure, and  $v$  is volumetric flow rate. The subscript 0 indicates an inlet term; no such subscript indicates an outlet term except for the volume of the reactor,  $V$ , which has no subscript but refers to the reactor itself.

terms, and then we'll use the remainder of the chapter to explore example problems. We'll look at problems that can be solved with pencil and paper, followed by problems that require MATLAB. In the next chapter we'll look at some special cases of Equations (4.1) and (4.2) that pop up regularly, but just try to keep in mind that although those special cases will make it seem as though we've got several different ways of analyzing CSTRs, there's really just these two equations that we're using over and over again.

## 4.1 The CSTR material balances are algebraic equations

A CSTR is most commonly used for liquid-phase processes and at its simplest is something like a stirred bucket with inlet and outlet streams as sketched in Figure 4.1. We'll denote all inlet quantities with a subscript 0 and all outlet quantities without such a subscript, thereby establishing symbols like  $n_{i0}$  for the inlet molar flow rate of species  $i$  and  $n_i$  for the outlet molar flow rate of species  $i$ . The molar concentration  $C_i$  (mole per volume) is simply the molar flow rate of  $i$  divided by the total volumetric flow rate  $v$ , or

$$C_i = \frac{n_i}{v}, \quad (4.3)$$

with a similar expression for the inlet. For the moment we'll only deal with liquids in CSTRs, and if we assume that these liquids are both incompressible and have negligible change in density upon reacting then the inlet and outlet volumetric flow rates are the same,

$$v_0 = v \quad (\text{liquids only}) \quad (4.4)$$

such that the  $v$  in the denominator of Equation (4.3) can be written as either  $v$  or  $v_0$ . Also, in all cases that we'll encounter the volume  $V$  of the reactor will be constant, which for liquids

is easily achieved with either some pumps and a level controller, or one pump on the inlet and the outlet pipe placed some height above the bottom of the reactor to give the desired volume. The volume  $V$  and volumetric flow rate  $v$  are related through the residence time  $\tau$  by

$$\tau = \frac{V}{v_0}, \quad (4.5)$$

where  $\tau$  represents the average time a molecule spends in the reactor. Generally larger residence times allow the reaction to proceed to a higher degree of completion, and can be achieved by using either large reactors or small volumetric flow rates (or both).

The stirring mechanism is typically some kind of motor-driven paddle wheel or propeller, although anything that provides agitation can count as a mixer (e.g., a magnetic stir bar in a lab beaker is ok too). We'll encounter more variations on CSTR construction and their impacts on the design equation as we go, but a simple "stirred bucket" is a good picture to keep in your head for now. The key feature of the stirring mechanism is that it keeps the contents of the reactor **well-mixed**, and the practical implication of this feature is that **outlet stream has the same temperature and composition as the reactor itself**.

The material balance for any species around any unit always starts life as

$$\text{Acc} = \text{In} - \text{Out} + \text{Gen} - \text{Cons}$$

which we'll rewrite as

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}$$

where "Change" has replaced "Generation" and "Consumption" but is closely related: positive change is generation and negative change is consumption.

At steady state the accumulation term is zero and we often rearrange the resulting material balance to the general form

$$\text{Out} = \text{In} + \text{Change},$$

and if you think back to your material and energy balances course you'll recall that we express "change" by using the extent of reaction ( $\xi$ ) to get the familiar equation

$$n_i = n_{i0} + \sum_{j=1}^q \nu_{ij} \xi_j. \quad (4.6)$$

where  $\nu_{ij}$  is the stoichiometric coefficient of species  $i$  in reaction  $j$ ,  $\xi_j$  is the extent of reaction for reaction  $j$ , and the sum is over all  $Q$  reactions. This form is still valid for a CSTR and you can always come back to it if you think it'll help you work through a problem. However,

going forward it's going to be quite rare to see  $\xi$  used explicitly in any equation because we're going to replace it with the more useful net reaction rate,  $r_i$ .

We can develop a new form of the CSTR material balance comparing the units on the  $v_i \xi$  terms in the old material balance,

$$v_i \xi \doteq \frac{\text{mol } i}{\text{mol reacted}} \times \frac{\text{mol reacted}}{\text{time}} \doteq \frac{\text{mol } i}{\text{time}}$$

to the units on the rate term  $r_i$ ,

$$r_i \doteq \frac{\text{mol } i}{\text{volume} \times \text{time}}.$$

Evidently  $r_i$  has the same units as  $v_i \xi$  except for the volume term in the denominator, and since the CSTR is well-mixed we can simply multiply  $r_i$  by the reactor volume  $V$  to get a quantity  $r_i V$  with the same units as  $v_i \xi$ .

Ever so conveniently we've already defined  $r_i$  as the *net* rate of reaction for species  $i$ , which means there's no further summation work we have to do to account for multiple reactions because the summation is performed each time we write an expression for  $r_i$  (see Equation (3.12) in Section 3.6 for a review). Therefore we can replace the  $\sum v_{ij} \xi_j$  term in Equation (4.6) with  $r_i V$  to get the updated CSTR material balance in terms of reaction rates as

$$\boxed{n_i = n_{i0} + r_i V} \quad (4.1)$$

as we saw at the beginning of the chapter. Remember that we can usually write Equation (4.1) **once for each species** involved in a process.

Equation (4.1) is quite similar to expressions for  $n_i$  that were introduced in Chapter 2, first as a function of  $\xi$  as

$$n_i = n_{i0} + v_i \xi, \quad (2.10)$$

and later as a function of conversion  $X_A$  as

$$n_i = n_{A0} \left( \Theta_i - \frac{v_i}{v_A} X_A \right). \quad (2.35)$$

The critical difference between Equations (2.10) and (2.35) and Equation (4.1) is that the former are expressions of stoichiometry which are valid for *any* reactor, whereas the latter is a material balance valid only for CSTRs. Equation (4.1) can be considered an “improvement” over Equations (2.10) and (2.35) because it provides a more direct connection to the real world: it's based on a material balance of a reactor which can actually be built, and it relies on more



readily available experimental data through  $r_i$ .

Before we move on to the energy balance, we can rearrange Equation (4.1) to root-finding form to get

$$0 = n_i - n_{i0} - r_i V$$

which is a form that will be useful when we want to solve the material balance using a function like `fsolve`.

## 4.2 The CSTR energy balance is an algebraic equation

The steady-state, open system energy balance that you'd have learned from your material and energy balances course (as well as thermodynamics) was likely something along the lines of

$$\Delta H = Q + W_S, \quad (4.7)$$

where  $Q$  is energy added to the system by a thermal gradient (heat) and  $W_S$  is shaft work added to the system. In this text we take heat added to the system and shaft work done *on* the system to be positive quantities. The enthalpy difference term  $\Delta H$  in Equation (4.7) is the difference between the outlet and inlet enthalpies, or

$$\Delta H = \sum_{i=A}^Z n_i H_i - \sum_{i=A}^Z n_{i0} H_{i0}, \quad (4.8)$$

where  $H_i$  is the molar enthalpy (e.g.,  $\text{J} \cdot \text{mol}^{-1}$ ) of component  $i$  at the reactor outlet and  $H_{i0}$  is the molar enthalpy of component  $i$  at the reactor inlet.

We'll look at  $Q$  more in the next section, and in nearly all cases the shaft work added to a CSTR by the mixer is negligible compared to other energy terms,<sup>1</sup> but we're going to need to play around with the  $\Delta H$  term a little bit before we get the energy balance into a usable form. To keep things a little more simple we'll focus on a CSTR in which only a single reaction occurs, but we'll see that it doesn't get terribly more complicated even for multiple reactions.

To get things started let's rearrange the general material balance in Equation (4.1) to an expression for  $n_i$  as

$$n_i = n_{i0} + r_i V, \quad (4.9)$$

<sup>1</sup>Exceptions include intense mixing as in a blender, or mixing of highly viscous fluids like polymers. Neither of these will be addressed in this text.

which we can now use to eliminate  $n_i$  from Equation (4.8) to yield

$$\Delta H = \sum_{i=A}^Z (n_{i0} + r_i V) H_i - \sum_{i=A}^Z n_{i0} H_{i0}. \quad (4.10)$$

We can expand the sums and rearrange a little bit to group similar terms as

$$\begin{aligned} \Delta H &= \sum_{i=A}^Z n_{i0} H_i + \sum_{i=A}^Z r_i V H_i - \sum_{i=A}^Z n_{i0} H_{i0} \\ &= \sum_{i=A}^Z n_{i0} (H_i - H_{i0}) + r V \sum_{i=A}^Z v_i H_i \end{aligned} \quad (4.11)$$

where we also used  $r_i = v_i r$  in the last step as defined in Equation (3.6), and brought the quantity  $r V$  outside the sum since it's the same  $r V$  for every term in the sum. We can also use our definition of  $\Delta M$  from Equation (2.22) to replace the second sum with the **enthalpy of reaction**  $\Delta_r H$ , or

$$\Delta H = \sum_{i=A}^Z n_{i0} (H_i - H_{i0}) + r V \Delta_r H, \quad (4.12)$$

where

$$\Delta_r H = \sum_{i=A}^Z v_i H_i \quad (4.13)$$

is the **enthalpy of reaction at the reactor (outlet) temperature**. We use the subscript  $r$  on  $\Delta_r$  to distinguish it from the  $\Delta H$  that's in the energy balance, but soon we'll drop the subscript because we won't need to make the distinction anymore.

We've now got to think back to the general concept of dealing with enthalpy calculations: we can only calculate *differences* in enthalpy between two states, never an absolute enthalpy. That means we need to pick a reference state and calculate all the  $H_i$  terms in Equation (4.11) relative to that reference state. For reasons we'll see soon, the most convenient reference state is the elements at 25 °C and 1 bar in whatever state of aggregation they occur naturally under those conditions (e.g.,  $\text{O}_2(\text{g})$  instead of  $\text{O}$ ,  $\text{N}_2(\text{g})$  instead of  $\text{N}$ , and so on).

Conceptually, the enthalpy calculation we're going to do for every  $H_i$  in Equation (4.11) will proceed in two steps: assemble the molecule of interest from some combination of the appropriate elements at 25 °C and 101.325 kPa, then adjust the temperature, pressure, and phase of the molecule to the appropriate process condition. Another way you may have seen such a

process written is

$$\left[ \begin{array}{c} \text{elements at} \\ \text{at } 25^\circ\text{C,} \\ 101.325 \text{ kPa} \end{array} \right] \xrightarrow[\text{molecule}]{\text{assemble}} \left[ \begin{array}{c} \text{molecule } i \\ \text{at } 25^\circ\text{C,} \\ 101.325 \text{ kPa} \end{array} \right] \xrightarrow[\text{heat or cool}]{\text{change phase,}} \left[ \begin{array}{c} \text{molecule } i \\ \text{at } T, P \end{array} \right] \quad (4.14)$$

(reference state)  (process state)

The enthalpy change of the “assembly” step is just the standard enthalpy of formation for that species,  $\Delta H_{f,i}^\circ$ , which you can find in Appendix D for many different molecules. Remember that  $\Delta H_{f,i}^\circ$  depends on the molecule’s state of aggregation (its phase, like liquid or gas), and to avoid needing to change phases in the next step let’s assume that we’ll always be able to assemble the molecules into the same phase as needed for the process condition. The practical consequence of this assumption is that we’ll always need to have an enthalpy of formation for the correct state of aggregation in order to use any of the equations derived hereafter, but that’s usually not too big of a problem.

The other assumptions we’ll make apply to the second step, changing phase and heating or cooling the molecule to the process state. To greatly simplify this calculation we’ll make three assumptions:

1. The phase of a molecule does not change from inlet to outlet,
2. If the molecule is in the gas phase then it behaves ideally,
3. If the molecule is in the liquid phase then the process operates isobarically.

The practical consequence of these three assumptions is that the enthalpy change for the second step—from the molecule at  $25^\circ\text{C}$  and  $101.325 \text{ kPa}$  to the molecule at the process state—can always be calculated as a sensible heat, which means we’ll need only evaluate the integral of a constant-pressure heat capacity  $C_p$ , or

$$\left[ \begin{array}{c} \Delta H \text{ to adjust molecule} \\ \text{from standard state to} \\ \text{process state} \end{array} \right] = \int_{25^\circ\text{C}}^T C_p dT.$$

If we further assume that **heat capacity is always constant**<sup>2</sup> then this integral simply becomes a heat capacity multiplied by temperature difference, so our original calculation procedure in

---

<sup>2</sup>This is assumed *everywhere* in this text!

Equation (4.14) simplifies to

$$\left[ \begin{array}{c} \text{elements at} \\ \text{at } 25^\circ\text{C,} \\ 101.325 \text{ kPa} \end{array} \right] \xrightarrow{\Delta H_{f,i}^\circ} \left[ \begin{array}{c} \text{molecule } i \\ \text{at } 25^\circ\text{C,} \\ 101.325 \text{ kPa} \end{array} \right] \xrightarrow{C_{P,i}(T-25^\circ\text{C})} \left[ \begin{array}{c} \text{molecule } i \\ \text{at } T, P \\ \text{(process state)} \end{array} \right] \quad (4.15)$$

which means that every single  $H_i$  term in Equation (4.11) can be calculated using an expression like

$$H_i = \Delta H_{f,i}^\circ + C_P(T - 25^\circ\text{C}) \quad (4.16)$$

For example, each term in the first sum on the right-hand side of Equation (4.11) becomes

$$\begin{aligned} n_{i0}(H_i - H_{i0}) &= n_{i0}[\Delta H_{f,i}^\circ + C_P(T - 25^\circ\text{C}) - \Delta H_{f,i}^\circ - C_P(T_0 - 25^\circ\text{C})] \\ &= n_{i0}C_{P,i}(T - T_0) \end{aligned} \quad (4.17)$$

Substituting into our  $\Delta H$  expression from Equation (4.12) yields

$$\Delta H = \sum_{i=A}^Z n_{i0}C_{P,i}(T - T_0) + rV\Delta_r H. \quad (4.18)$$

If we happened to know the enthalpy of reaction  $\Delta_r H$  at the reactor (outlet) temperature then we could stop with Equation (4.18), but usually we don't have this information so we need to apply Equation (4.16) to  $\Delta_r H$  in order to get it into a more usable form. Substituting the form of Equation (4.16) into Equation (4.13) yields

$$\begin{aligned} \Delta_r H &= \sum_{i=A}^Z \nu_i H_i \\ &= \sum_{i=A}^Z \nu_i [\Delta H_{f,i}^\circ + C_{P,i}(T - 25^\circ\text{C})] \\ &= \sum_{i=A}^Z \nu_i \Delta H_{f,i}^\circ + \sum_{i=A}^Z \nu_i C_{P,i}(T - 25^\circ\text{C}) \\ &= \Delta_r H^\circ + \Delta C_P(T - 25^\circ\text{C}) \end{aligned} \quad (4.19)$$

where

$$\Delta_r H^\circ = \sum_{i=A}^Z \nu_i \Delta H_{f,i}^\circ \quad (4.20)$$

is the **standard state enthalpy of reaction** at 25 °C and 101.325 kPa, and

$$\Delta C_P = \sum_{i=A}^Z \nu_i C_{P,i}. \quad (4.21)$$

Equation (4.19) also implies a general way to adjust enthalpies of reaction from any reference temperature  $T_R$  to any other temperature  $T$  as

$$\Delta_r H = \Delta_r H^\circ + \Delta C_P (T - T_R), \quad (4.22)$$

where  $T_R = 25^\circ\text{C}$  if  $\Delta_r H^\circ$  was calculated using tabulated values of  $\Delta H_{f,i}^\circ$  as in Equation (4.20). Even more generally you can use Equation (4.22) if you're given  $\Delta_r H$  at some temperature  $T_1$  and need to adjust it to another temperature  $T_2$  as

$$\Delta_r H(T_2) = \Delta_r H(T_1) + \Delta C_P (T_2 - T_1). \quad (4.23)$$

It took us a while but we're finally in a place to do some damage to the energy balance in Equation (4.7). Substituting Equation (4.17) for  $\Delta H$  and rearranging to root-finding form<sup>3</sup> yields

$$0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + r V \Delta_r H - Q - W_S, \quad (4.24)$$

which can be extended easily to cases of multiple reactions simply by summing up the  $r \Delta_r H$  term over all  $Q$  reactions as

$$r \Delta_r H \xrightarrow[\text{mult. reactions}]{\text{extend to}} \sum_{j=1}^q r_j \Delta_r H_j, \quad (4.25)$$

so that the CSTR energy balance for one reaction in Equation (4.24) becomes the **general energy balance for a CSTR** that was introduced at the beginning of the chapter,

$$\boxed{0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q - W_S} \quad (4.2)$$

where we've dropped the subscript  $r$  on  $\Delta_r H$  since at no other point in our work with CSTRs will we ever again need to work with anything *other* than the enthalpy of reaction. The equations that most commonly show up when using Equation (4.2) are the ones that correspond

<sup>3</sup>We do this in anticipation of using `fsolve` to solve CSTR problems.

to Equations (4.20) to (4.22), reproduced here for convenience and without the subscript  $r$ :

$$\Delta H^\circ = \sum_{i=A}^Z \nu_i \Delta H_{f,i}^\circ \quad (4.20)$$

$$\Delta C_P = \sum_{i=A}^Z \nu_i C_{P,i} \quad (4.21)$$

$$\Delta H = \Delta H^\circ + \Delta C_P (T - T_R) \quad (4.22)$$

There are several temperature and sum terms that show up in the energy balance and its “support” equations, and it’s important to keep them arranged correctly in your mind:

**Temperature terms:**  $T$  is the reactor and outlet temperature,  $T_0$  is the inlet temperature, and  $T_R$  is the temperature at which  $\Delta H^\circ$  is evaluated (usually 25 °C).

**Summed terms:** The first sum in Equation (4.2) is over all *species* because the index variable is  $i$  while the second sum is over all *reactions* because the index variable is  $j$ .

The last thing to do before we can actually use the energy balance in Equation (4.2) is to figure out how to deal with the heat and shaft work terms,  $Q$  and  $W_S$ .

### 4.3 Shaft work is negligible and heat is transferred by a heat exchanger

As mentioned previously, we’re not going to look at problems or scenarios in this text wherein the shaft work term  $W_S$  is comparable to either the enthalpy or heat terms, so our energy balance in Equation (4.2) will always be used with  $W_S = 0$ . There are exceptions, such as with exceptionally intense mixing or mixing of highly viscous materials like polymers, but generally it’s a safe initial assumption to neglect shaft work in CSTRs.

Heat, on the other hand, can almost never be ignored because it’s one of the primary ways we regulate a reaction, or one of the primary ways we find out that we have *failed* to regulate a reaction. When we say “heat” we’re referring to the transfer of energy due to a temperature difference between the reactor and its surroundings, which in practice lumps reactors into one of two broad categories:

**Adiabatic reactors**, which are reactors for which there is no heat transfer to or from the surroundings. This is most easily achieved by insulating the reactor.

**Non-adiabatic reactors**, which are reactors equipped with a heat exchanger to transfer energy between the reactor contents and a utility fluid such as steam (for heating) or chilled water (for cooling).

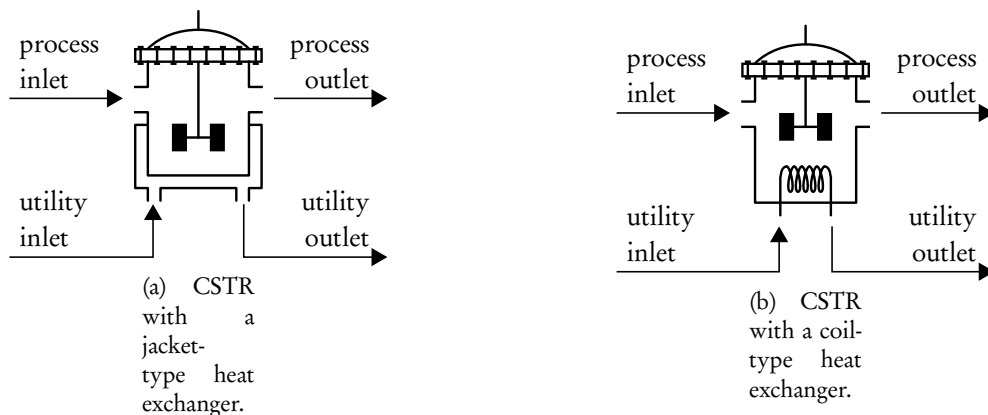


Figure 4.2: Heat exchangers attached to CSTRs are usually either jackets or coils. Process inlets and outlets are unchanged. Utility streams, often steam or cooling water, are provided to the heat exchanger and absorb energy from or provide energy to the reactor contents. No reactions occur in the heat exchangers and there is no mixing of process and utility fluids.

Adiabatic reactors are relatively simple to visualize since they're essentially a reactor wrapped with insulation, and they're relatively straightforward to analyze in terms of the energy balance because we simply set  $Q = 0$  in Equation (4.2). In fact we'll see later see that adiabatic conditions lead to particularly simple and equivalent expressions of the energy balance for many different reactors.

Non-adiabatic reactors typically have a heat exchanger integrated into the reactor design as shown in Figure 4.2, and the most common heat exchanger designs are jacket-type or coil-type. A jacket-type heat exchanger (Figure 4.2a) wraps around the bulk of the reactor exterior whereas a coil-type heat exchanger (Figure 4.2b) has a tube of conductive material—usually copper or stainless steel, or more exotic materials depending on chemical compatibility—inserted into and coiled within the reactor.

The energy  $Q$  transferred from the process fluid (*i.e.*, the fluid inside the reactor) to the utility fluid (*i.e.*, the fluid flowing through the heat exchanger) is proportional to  $A$ , the area available for heat transfer, and  $T_u - T$ , the temperature difference between the process and utility fluids. If we multiply the product of these two terms by  $U$ , an experimentally-determined “fudge factor,” we get

$$Q = UA(T_u - T), \quad (4.26)$$

where  $Q$  and  $T$  are the same as in Equation (4.2),  $T_u$  is the temperature of the utility and the fudge factor  $U$  is called the **overall heat transfer coefficient**. You'll learn (or have learned) *much* more about both Equation (4.26) and  $U$  in your heat transfer course but suffice it to

Table 4.1: Order-of-magnitude estimates for the individual heat transfer coefficient  $h$  for calculating the overall heat transfer coefficient  $U$  with Equation (4.27). A “phase change” refers to condensation or vaporization of saturated water.

Fluid	$h$ ( $\text{W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$ )
Gases	60
Organic liquids	1000
Liquid water	1500
Phase change	2500

say that it’s a proportionality factor that takes into account the thermal characteristics of the process fluid, the utility fluid, and the material and geometry of the heat exchanger. The fluids are usually the dominant terms in such an accounting and so we can approximate  $U$  as

$$\frac{1}{U} \approx \frac{1}{h} + \frac{1}{h_u}, \tag{4.27}$$

where  $h$  and  $h_u$  are the individual heat transfer coefficients of the process and utility streams. It’s quite a challenge all by itself to evaluate  $h$  but you can get an idea of the general order-of-magnitude of  $h$  for various fluids from Table 4.1.

**Example 4.1: Estimating the overall heat transfer coefficient**

A reaction involving the isomerization of liquid heptane is to be temperature-controlled by a heat exchanger using chilled water as a coolant. Estimate  $U$  for this system.

**Solution**

The process fluid is an organic liquid for which  $h = 1000 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$  according to Table 4.1. The utility is liquid water for which  $h_u = 1500 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$  by the same table. From Equation (4.27) we have

$$\frac{1}{U} \approx \frac{1}{1000} + \frac{1}{1500} \Rightarrow U \approx 600 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}.$$

If  $T_u$  is constant then addition of a heat exchanger doesn’t propose a significant complication of the CSTR energy balance: simply substitute Equation (4.26) into Equation (4.2) and solve as needed for the problem of interest. If  $T_u$  is not constant or if more details about the utility stream are needed then you’ll have to couple the CSTR energy balance with an energy



balance on the utility, which is

$$-Q = -UA(T_u - T) = m_u \Delta \hat{H}_u = \begin{cases} m_u C_{P,u} (T_u - T_{u0}) & \text{(no phase change)} \\ m_u \lambda_u & \text{(phase change)} \end{cases} \quad (4.28)$$

where  $m_u$  is the mass flow rate of the utility,<sup>4</sup>  $C_{P,u}$  is the (constant) heat capacity of the utility,  $T_{u0}$  and  $T_u$  are the inlet and outlet utility temperatures, and  $\lambda_u$  is the enthalpy of vaporization of the utility.

For utilities without phase change Equation (4.28) reveals how good or bad it is to assume an isothermal utility. Rearranging Equation (4.28) for the utility temperature difference gives

$$T_u - T_{u0} = \frac{-Q}{m_u C_{P,u}},$$

so we see that if we want the inlet and outlet utility temperatures to be the same (an isothermal utility) then we can either make the flow rate ( $m_u$ ) huge or choose a coolant liquid with a very high heat capacity ( $C_{P,u}$ ), or both. Water is the most common coolant meaning that  $C_{P,u}$  is usually fixed around  $4.2 \text{ kJ} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$ , which leaves the most common method of assuring isothermal utility operation to be increasing  $m_u$ .

## 4.4 CSTR problems can be solved by hand calculations

All the problems in this section are ones that are typical of CSTR problems that can be solved by hand. The solution procedure for each problem follows the same four step process:

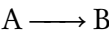
1. Start with the material and energy balances, Equations (4.1) and (4.2).
2. Identify the rate law using the tools from Chapter 3.
3. Gather any additional information, like enthalpies of formation, heat capacities, etc.
4. Substitute 2 and 3 into the material and energy balances, rearrange for the desired variable, and solve.

### Example 4.2: Isothermal CSTR sizing

A feed of pure A at  $3 \text{ L} \cdot \text{s}^{-1}$  is provided to an isothermal CSTR in which the elemen-

<sup>4</sup>A molar flow rate is also ok, as long as the units work out to be consistent with the units on  $Q$ .

tary, liquid-phase reaction



takes place with  $k = 0.003 \text{ s}^{-1}$  (constant). Determine the size of the CSTR needed to convert 75% of the feed, and compare this volume to an everyday object.

**Solution**

The system is isothermal so we'll ignore the energy balance and focus on only the material balance, Equation (4.1). We can set up a stoich table with two components as follows:

Sp.	$\nu_i$	$r_i = \sum \nu_{ij} r_j$	$n_{i0}$	$n_i = n_{i0} + r_i V$
A	-1	$-r$	$n_{A0}$	$n_A = n_{A0} - r V$
B	1	$r$	0	$n_B = r V$

Notice that each entry for  $n_i$  is an exercise in writing Equation (4.1), but it can be convenient to leave it in a fairly general form when it's in the stoich table.

We've been tasked with finding a volume  $V$  and the stoich table suggests that one approach is to use the material balance for either A or B. Since we're also provided a conversion of A as a process specification we'll focus our attention on A first. We're also given a volumetric flow rate  $v$  which suggests that we'll want to convert from molar flow rates to concentrations as

$$n_{A0} = v_0 C_{A0} \quad \text{and} \quad n_A = v_0 C_A,$$

where we assumed  $v = v_0$  in the second expression since the reaction is liquid-phase. The reaction is elementary and irreversible so the reaction rate  $r$  is

$$r = k C_A$$

and substituting these into the material balance for A (which is the  $n_i$  column for species A in the stoich table) yields

$$v_0 C_A = v_0 C_{A0} - k C_A V,$$

from which  $V$  can be isolated as

$$V = \frac{v_0(C_{A0} - C_A)}{kC_A}.$$

We have yet to make use of the additional process specification regarding conversion of A, and we do so now. To convert 75% of the feed means

$$0.75 = 1 - \frac{n_A}{n_{A0}} = 1 - \frac{v_0 C_A}{v_0 C_{A0}} = 1 - \frac{C_A}{C_{A0}}$$

or

$$C_A = 0.25C_{A0},$$

which can be substituted into the material balance to get

$$V = \frac{v_0(1 - 0.25)}{0.25k} \doteq \frac{\text{L s}}{\text{s}} \doteq \text{L}.$$

Substituting  $v_0 = 3 \text{ L} \cdot \text{s}^{-1}$  and  $k = 0.003 \text{ s}^{-1}$  yields

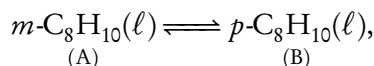
$$V = \frac{3 \times 0.75}{0.25 \times 0.003} = 3000 \text{ L}.$$

3000 L is not terribly large: if you hold your arms out and imagine a cylinder of your height with a diameter from fingertip to fingertip, that's about 3000 L.

Two elements of Example 4.2 will come up regularly: Equations (4.3) and (4.4) are a good way to convert between the molar flow rates that appear in the material and energy balances and the molar concentrations that appear in rate laws, and conversion specifications are a means of relating inlet and outlet molar flow rates or molar concentrations.

### Example 4.3: Isomerization in a CSTR

Liquid-phase isomerization of *m*-xylene to *p*-xylene is to take place at 120 °C in an isothermal, jacketed CSTR according to the elementary, reversible reaction



for which  $A = 0.021 \text{ s}^{-1}$  and  $E_a = 27 \text{ kJ} \cdot \text{mol}^{-1}$ . If  $2 \text{ kg} \cdot \text{s}^{-1}$  of pure *m*-C<sub>8</sub>H<sub>10</sub> is to be

processed in a CSTR with 8% conversion, determine

- (a) the reactor volume required,
- (b) the energy that must be added to or withdrawn from the reactor to maintain isothermal reactor operation,
- (c) the length of 1" OD copper coil needed for a heat exchanger if saturated steam is to be supplied as a utility at 200 °C, and
- (d) the mass flow rate of the saturated steam utility.

The density of both xylenes is about  $870 \text{ kg} \cdot \text{m}^{-3}$ .

### Solution

A quick note before we start: for any CSTR the temperature of the outlet stream is always the same as the temperature of the reactor itself. This problem specifies an *isothermal* CSTR, which means that the temperature of the inlet stream is the same as the temperature of the outlet stream. These ideas are subtly different—the first is a consequence of the well-mixed approximation, the second is an additional assumption about operating conditions—and should be kept separate in your mind.

(a) Once again we start with the material and energy balances for a CSTR, Equations (4.1) and (4.2):

$$\begin{aligned}
 n_A &= n_{A0} + r_A V \\
 n_B &= n_{B0} + r_B V \\
 0 &= \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q - W_S.
 \end{aligned}$$

Let's gather up some data that we'll need going forward and organize it in a stoich table. In anticipation of needing them for equilibrium and energy calculations we'll also gather up some property data from Table D.3:

Sp.	$\nu_i$	$r_i$	$n_{i0}$	$n_i$	$C_p$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )	$\Delta G_f^\circ$ (kJ · mol <sup>-1</sup> )
A	-1	-r	$n_{A0}$	$n_{A0} - rV$	185	-25.4	107.6
B	1	r	0	$rV$	183	-24.4	110.6
$\delta = 0$					$\Delta M = -2$	1	3

Notice that the bottom row of the stoich table is a convenient place to calculate various sums and the  $\Delta M$  quantities for later use. We could have also included the sums for  $n_{T0}$  and  $n_T$  but they happen to not be necessary just now. Moving on to thermodynamics, by Section 3.7.1 with  $\delta = \sum \nu_i = 0$  the equilibrium constant in the rate law is

$$K_C = K_a C_T^\delta = K_a \approx K_0 K_1 = 0.33.$$

The inlet molar flow rate A is

$$n_{A0} = \frac{m_{A0}}{MM_A} = \frac{2 \text{ kg} \cdot \text{s}^{-1}}{106 \text{ kg} \cdot \text{kmol}^{-1}} = 18.90 \text{ mol} \cdot \text{s}^{-1}$$

and the total inlet volumetric flow rate is

$$v_0 = \frac{m_A}{\rho_A} = \frac{2 \text{ kg} \cdot \text{s}^{-1}}{870 \text{ kg} \cdot \text{m}^{-3}} \times \frac{1000 \text{ L}}{\text{kg}} = 2.30 \text{ L} \cdot \text{s}^{-1}.$$

Since this is a liquid stream we assume that the inlet and outlet volumetric flow rates are the same, or  $v = v_0$ . Based on the 8% conversion requirement we can calculate the outlet molar flow rate of A as

$$n_A = (1 - 0.08) n_{A0} = 17.40 \text{ mol} \cdot \text{s}^{-1}$$

and from stoichiometry the outlet molar flow rate of B as

$$n_B = 0.08 n_{A0} = 1.50 \text{ mol} \cdot \text{s}^{-1}.$$

The reactor (and outlet) molar concentrations of A and B are

$$C_A = \frac{n_A}{v} = \frac{n_A}{v_0} = \frac{17.40 \text{ mol} \cdot \text{s}^{-1}}{2.30 \text{ L} \cdot \text{s}^{-1}} = 7.57 \text{ mol} \cdot \text{L}^{-1}$$

$$C_B = \frac{n_B}{v} = \frac{n_B}{v_0} = \frac{1.50 \text{ mol} \cdot \text{s}^{-1}}{2.30 \text{ L} \cdot \text{s}^{-1}} = 0.65 \text{ mol} \cdot \text{L}^{-1}$$

The rate law for the reversible reaction is

$$r = k \left( C_A - \frac{C_B}{K_C} \right),$$

where

$$k = A \exp \left[ \frac{-E_a}{RT} \right] = 0.0210 \exp \left[ \frac{-2.7 \times 10^4}{(8.314)(393.15)} \right] = 5.43 \times 10^{-6} \text{ s}^{-1}$$

Substituting into  $C_A$ ,  $C_B$ ,  $k$ , and  $K_C$  into  $r$  gives the reaction rate as

$$r = -r_A = r_B = (5.43 \times 10^{-6}) \left( 7.57 - \frac{0.65}{0.33} \right) = 3.03 \times 10^{-5} \text{ mol} \cdot \text{L}^{-1} \cdot \text{s}^{-1},$$

and from either material balance the volume  $V$  is

$$V = \frac{n_{A0} - n_A}{-r_A} = \frac{18.90 - 17.40}{3.03 \times 10^{-5}} = \boxed{4.95 \times 10^4 \text{ L}},$$

which is pretty big! An average passenger vehicle has exterior dimensions of about  $1.5 \text{ m} \times 1.5 \text{ m} \times 4.5 \text{ m} \approx 12000 \text{ L}$ , so this reactor is about as big as four average passenger vehicles.

**(b)** To determine  $Q$  we first expand each of the sums in the energy balance to get

$$0 = rV\Delta H - Q,$$

where we've customarily dropped  $W_s$  and eliminated all terms containing  $(T - T_0)$  because the system is isothermal (remember that  $T$  is the outlet temperature,  $T_0$  is the inlet temperature, and in an isothermal system  $T = T_0$ ). Rearranging for  $Q$  gives

$$Q = rV\Delta H.$$

Remember that  $\Delta H$  needs to be evaluated **at the reactor temperature**, which means we'll need to use Equation (4.23) to adjust  $\Delta H$  from 25 °C to 120 °C as

$$\begin{aligned}\Delta H(T_2) &= \Delta H(T_1) + \Delta C_p(T_2 - T_1) \\ &= 1 \text{ kJ} \cdot \text{mol}^{-1} + (-2 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1})(120^\circ\text{C} - 25^\circ\text{C})/1000 \\ &= 0.81 \text{ kJ} \cdot \text{mol}^{-1}.\end{aligned}$$

Also keep in mind that this adjustment is consistent with an isothermal reactor because this adjustment is from a *reference* state to a *process* state. Substituting for  $Q$  gives

$$\begin{aligned}Q &= r V \Delta H = (3.03 \times 10^{-5})(4.95 \times 10^4)(0.81) \\ &= \boxed{1.22 \text{ kJ} \cdot \text{s}^{-1}}.\end{aligned}$$

Notice that this is a positive number meaning that energy must be supplied to the reactor to maintain isothermal operation. This is consistent with this isomerization reaction being endothermic.

(c) Since we're given the outer diameter (OD) and asked to find a length, the approach will be to find the area  $A$  for heat exchange by Equation (4.28) with  $T_u = 200^\circ\text{C}$  (constant because the steam is saturated and is therefore condensing to provide energy to the process stream), then using simple geometry to find the length  $L$  from

$$A = \pi D L,$$

where  $D$  is the outer diameter of the pipe. With condensing water as the utility and an organic liquid on the process side the overall heat transfer coefficient  $U$  can be estimated from Equation (4.27) as

$$\frac{1}{U} \approx \frac{1}{1000} + \frac{1}{2500} \Rightarrow U \approx 714.29 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}.$$

Rearranging Equation (4.28) gives the area  $A$  as

$$A = \frac{-Q}{-U(T_u - T)} = \frac{-1.22 \times 1000}{-714.29(200 - 120)} = 0.021 \text{ m}^2,$$

which gives a coil length of 1" OD tubing as

$$L = \frac{A}{\pi D} = \frac{0.021}{3.14 \times (1 \times 2.54/100)} = \boxed{0.267 \text{ m.}}$$

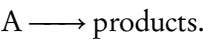
(d) The mass flow rate of the utility can be determined from Equation (4.28) if we know the enthalpy of vaporization of condensing saturated steam at 200 °C, which from a (saturated) steam table is about  $\lambda_u = -1940 \text{ kJ} \cdot \text{kg}^{-1}$ . Rearranging Equation (4.28) for  $m_u$  yields

$$m_u = \frac{-Q}{\lambda_u} = \frac{-1.22 \text{ kJ} \cdot \text{s}^{-1}}{-1940 \text{ kJ} \cdot \text{kg}^{-1}} = \boxed{0.63 \text{ g} \cdot \text{s}^{-1}}.$$

That’s pretty tiny! We could probably get away with assuming isothermal behavior for such a low conversion and small enthalpy of reaction and not introduce too much error in doing so.

**Example 4.4: Measuring activation energy in a CSTR**

Species A undergoes thermal degradation in an isothermal CSTR to form products according to the elementary reaction



When the inlet temperature is 45 °C the conversion of A is 6%, but when the inlet temperature is raised to 80 °C the conversion of A is measured to be 19%. What is the activation energy of the reaction?

**Solution**

The stoichiometric table for this problem will only contain one component because “products” is used as a placeholder to describe all sorts of degradation products.

Sp.	$\nu_i$	$r_i = \sum \nu_{ij} r_j$	$n_{i0}$	$n_i$
A	-1	-r	$n_{A0}$	$n_A = n_{A0} + r_A V$

Our analysis starts at the general material balance for a CSTR, Equation (4.1), as

$$n_A n_{A0} + r_A V,$$



but we ignore the energy balance since the system is operating isothermally and there appears no need to calculate  $Q$  for this problem. Let's make some substitutions of  $n_A = v_0 C_A$ ,  $n_{A0} = v_0 C_{A0}$ ,  $C_A = C_{A0}(1 - X_A)$ , and for our elementary reaction  $r = -r_A = k C_A$ , so that our material balance becomes (after a little rearranging)

$$k = \frac{v_0}{V} \frac{X_A}{1 - X_A}.$$

We can write this expression twice, once for  $T_1 = 45^\circ\text{C}$  and again for  $T_2 = 80^\circ\text{C}$ , then take the ratio of those two expressions to get

$$\frac{k(T_2)}{k(T_1)} = \frac{k_2}{k_1} = \frac{X_{A2}}{1 - X_{A2}} \frac{1 - X_{A1}}{X_{A1}}.$$

Substituting  $X_{A1} = 0.06$  and  $X_{A2} = 0.19$  yields  $k_2/k_1 = 3.67$ . Recall that Equation (3.10) gave exactly this ratio as

$$3.67 = \frac{k(T_2)}{k(T_1)} = \exp \left[ \frac{E_a}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right],$$

where  $R = 8.314 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ ,  $T_1 = 318 \text{ K}$ , and  $T_2 = 353 \text{ K}$ . From this equation we can find  $E_a = 34.7 \text{ kJ} \cdot \text{mol}^{-1}$ .

Notice that surprisingly little information was needed to solve this problem even though in the first few steps of the solution we made lots of substitutions that involved variables that we didn't know (e.g.,  $v_0$ ,  $V$ , etc.). That's a classic feature of many problems that require hand solutions.

## 4.5 CSTR problems can be solved in MATLAB

The class of problems that can be solved by hand is fairly small, often including isothermal systems with just one—usually irreversible—reaction. If we want to expand our analytical skills to more complicated systems then we need a more robust (and reliable) method of solving systems of nonlinear equations, and for that we turn to MATLAB, and more specifically, to the `fsolve` function.

Recall from Sections 1.5 and 1.8 that the general structure of a script to use `fsolve` for root finding is as follows:

```
1 % yourScriptName.m
```

```

2 % Script portion
3 x0 = 0;
4 x_sol = fsolve(@fun, x0);
5
6 % Local function
7 function out = fun(x)
8 out = x - cos(x); % root-finding form
9 end

```

Recall also a few of the syntax requirements of this approach: the actual call to `fsolve` comes *before* the definition of the local function, the local function comes at the very bottom of the script, and the local function contains the function in root-finding form. The way to interpret this structure is that `fsolve` will guess a value of  $x$  (the input to the local function), evaluate the value of `out` (the function in root-finding form), and if `out` is close to zero then `fsolve` return the guessed value of  $x$ . If `out` is not close to zero then `fsolve` will continue guessing  $x$  values until it either solves the problem or fails and gives up.

#### Example 4.5: Isothermal CSTR sizing with MATLAB

Repeat Example 4.2 but use MATLAB to solve for  $V$ .

#### Solution

After working on the material balance a little bit we found

$$V = \frac{v_0 X_A}{k(1 - X_A)}$$

which can be rearranged to root-finding form as

$$0 = \frac{v_0 X_A}{k(1 - X_A)} - V$$

and solved with `fsolve` as follows:

```

1 cstrIsothermalSizeMatlab.m
2 V = fsolve(@fun, 1000); % guess V=1000 L initially
3 fprintf('V = %.1f L\n', V);
4
5 function out = fun(V)
6 % Constants
7 v0 = 3; % L/s
8 k = 0.003; % 1/s

```

```

9| Xa = 0.75;
10
11| % Equations
12| out = v0*Xa/(k*(1-Xa)) - V;
13| end

```

Notice that within the local function fun it's not necessary to define the variable  $V$  because it's the input to fun and therefore exists within the function's workspace. Running this script produces the following output:

```

>> cstrIsothermalSizeMatlab

V = 3000.0 L

```

Unsurprisingly this is the same value we found in Example 4.2; the only difference is that we asked fsolve to do the math.

Example 4.5 is unusual in the sense that if we made it as far as rearranging the material balance to

$$V = \frac{v_0 X_A}{k(1 - X_A)}$$

then we probably didn't need MATLAB to do the math of substituting  $v_0$ ,  $X_A$ , and  $k$ , and even if we thought MATLAB was necessary then we probably didn't need to go as far as using fsolve because we could simply ask MATLAB to calculate  $V$  directly using the equation above.

More commonly we use MATLAB when it's either difficult or impossible to algebraically rearrange the material and energy balances to solve for the variables of interest. Fortunately the material and energy balance equations for a CSTR—Equations (4.1) and (4.2)—are already given in root-finding form, and that wasn't an accident: in the vast majority of CSTR problems we'll be in a position to use fsolve to solve these equations for one or more unknowns. A fairly general modification of the code above to handle CSTR problems is as follows:

```

1| % yourScriptName.m
2| % Script portion
3| Xa = fsolve(@fun, Xa_initial);
4|
5| % Local function
6| function out = fun(Xa)
7| % calculate other stuff here
8| out = [nA - nA0 - rA*V; % material balance on A
9|       nB - nB0 - rB*V; % material balance on B
10|       sum(ni0.*Cpi.*(T-T0)) + V*sum(rj.*dHj) - Q - Ws]; % eb
11| end

```

Notice that the variable out is a *column* vector, which it must always be when multiple equations are being solved. For this script fsolve will guess values of  $X_a$  (starting at  $X_{a\_initial}$ ) until two material balances—one for A, one for B—and the energy balance are satisfied. There's a fair amount of missing material in the `%calculate other stuff here` line but it's missing material that will be specific to the problem you're attempting to solve.

#### Example 4.6: Isomerization in a CSTR using MATLAB

Repeat Example 4.3, Part (a) but use MATLAB to find  $V$ .

#### Solution

We need solve only the material balance on A to find  $V$ , so the general structure of our solution script will be as follows:

```
1 % cstrIsomerizationMatlab
2 V = fsolve(@fun, 1000); % guess V=1000 L initially
3 fprintf('V = %.1f L\n', V);
4
5 function out = fun(V)
6 % calculate other stuff here
7 out = nA - nA0 - rA*V; % MB on A
8 end
```

The majority of work comes when we have to fill in the missing material on Line 6: we have to go from knowing nothing except  $V$  (because  $V$  is an input to fun) to calculating out on Line 7. A convenient way to approach such work is to go backwards: define each variable within the material balance as either a constant or an equation, then check to see if each of the variables so written can actually be calculated. If not, keep defining new variables until all variables are either defined as constants or as equations which use those constants.

For example, looking at Line 8 of the script above it appears that we'll need to define the variables  $n_A$ ,  $n_{A0}$ , and  $r_A$ , so the first few lines added to the local function could be

```
5 function out = fun(V)
6 nA0 = mA0/MM; % mol/s
7 nA = nA0*(1-Xa); % mol/s
8 rA = k*(Ca-Cb/Kc); % mol/L-s
9 out = nA - nA0 - rA*V; % MB on A
10 end
```

Notice that we could have also written  $n_{A0} = 18.90$ ; if we had done math in our head,

but you're strongly advised to break yourself of that habit because it's *far* easier to troubleshoot code when each step is written out clearly and with as few "mental substitutions" as possible. We can finish our script by continuing to define variables until we end up with something like the following script:

```

1 % cstrIsomerizationMatlab
2 V = fsolve(@fun, 1000); % guess V=1000 L initially
3 fprintf('V = %.1f L\n', V);
4
5 function out = fun(V)
6 % Constants
7 A = 0.021; % 1/s
8 Ea = 27e3; % J/mol
9 mA0 = 2; % kg/s
10 rhoA = 870; % kg/m3
11 MM = 106; % g/mol or kg/kmol
12 Xa = 0.08;
13 T = 120+273; % K
14 R = 8.314; % J/mol-K
15 dH = -24.4 + 25.4; % kJ/mol
16 dG = 110.6 - 107.6; % kJ/mol
17
18 % Algebraic equations
19 v0 = mA0/rhoA * 1000; % L/s
20 v = v0; % for liquids
21 nA0 = mA0/MM*1000; % mol/s
22 nA = nA0*(1-Xa); % mol/s
23 nB = nA0*Xa; % mol/s
24 Ca = nA/v; % mol/L
25 Cb = nB/v; % mol/L
26
27 % Rate equations
28 K0 = exp(-dG*1e3/(R*298.15));
29 K1 = exp(dH*1e3/R*(1/298.15-1/T));
30 Kc = K0*K1; % mol/L
31 k = A*exp(-Ea/(R*T)); % 1/s
32 rA = -k*(Ca-Cb/Kc); % mol/L-s
33
34 % Material balance
35 out = nA - nA0 - rA*V; % MB on A
36 end

```

Running this script produces the following output:

```

>> cstrIsomerizationMatlab

V = 50217.1 L

```

That’s a little more than 1% higher than the value we found previously, and the discrepancy can be attributed to round-off errors in Example 4.3.

Also notice that this appears to be quite a bit of code and that there are ways to simplify it, but as stated previously you’re *strongly encouraged* to get in the habit of writing code this way because it’s easy to find problems. For example, the equilibrium constant  $K_C$  could have been defined as a constant with a line like  $K_C = 0.33$  instead of going all the way back to  $\Delta G_{f,i}^\circ$ , but then when it comes time to debug such code the calculations to get to  $K_C$  have to be checked *by hand*, which is tedious!

**Example 4.7: Optimizing selectivity in a CSTR**

The following elementary, liquid-phase reactions take place in an isothermal, 15 L CSTR:

$A \longrightarrow B$	$k_{298\text{K}} = 0.2\text{ s}^{-1}$	$E_a = 35\text{ kJ} \cdot \text{mol}^{-1}$
$B \longrightarrow 2\text{ D}$	$k_{298\text{K}} = 0.1\text{ s}^{-1}$	$E_a = 25\text{ kJ} \cdot \text{mol}^{-1}$
$A + \text{D} \longrightarrow \text{U}$	$k_{298\text{K}} = 0.04\text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1}$	$E_a = 40\text{ kJ} \cdot \text{mol}^{-1}$

where product D is desired and product U is undesired. An aqueous solution of A is fed at  $10\text{ L} \cdot \text{min}^{-1}$  and contains 0.5 mol A per liter of solution. Determine the reactor temperature which optimizes the production of the desired component D, assuming the solution remains in the liquid phase from 250 K to 450 K. Provide at least one disadvantage of maximizing the production of D in this way.

**Solution**

This is a great example to make use of a **for** loop because we’re in a situation where, if we could just solve the system *once* for any individual  $T$ , then we can envision repeating this solution over and over for different  $T$  across the range of 250 K to 450 K and identifying the maximum production of D. We’ll take the approach of first solving the problem *once*, then showing how to modify the solution script to solve the problem lots of times.

Our approach starts with another stoichiometric table, this time with a few extra columns because we’ve got multiple reactions. We’ll refer to the reactions in the order given above as reactions  $j = 1$ ,  $j = 2$ , and  $j = 3$ .

Sp	$\nu_1$	$\nu_2$	$\nu_3$	$n_{i0}$	$r_i = \sum \nu_{ij} r_j$	$n_i = n_{i0} + r_i V$
A	-1	0	-1	$v_0 C_{A0}$	$r_A = -r_1 - r_3$	$n_A = 5 + (-r_1 - r_3)V$
B	1	-1	0	0	$r_B = r_1 - r_2$	$n_B = (r_1 - r_2)V$
D	0	2	-1	0	$r_D = 2r_2 - r_3$	$n_D = (2r_2 - r_3)V$
U	0	0	1	0	$r_U = r_3$	$n_U = r_3 V$

Notice that the stoich table makes quick work of identifying net rates and applying Equation (3.12). This is one of the table's key advantages when working with multireaction problems.

Since there are multiple reactions we don't have the option of solving for  $X_A$  so we'll use `fsolve` to solve for the molar flow rates directly, which means we'll have to construct a vector of all four flow rates and a vector of all four material balances (see Section 1.8 to review this syntax). The solution script to solve for a *single* temperature is as follows:

```

1 % cstrOptimize.m
2 Y0 = [0 0 0 0]; % [nA, nB, nC, nD] guesses, mol/s
3 Y = fsolve(@fun, Y0);
4 fprintf('nD = %.3f mol/s\n', Y(4));
5
6 function out = fun(Y)
7 % Constants
8 T = 350; % K
9 nA = Y(1); % mol/s
10 nB = Y(2); % mol/s
11 nD = Y(3); % mol/s
12 nU = Y(4); % mol/s
13 Ca0 = 0.5; % mol/L
14 v0 = 10/60; % L/s
15 V = 15; % L
16 nB0 = 0; % mol/s
17 nD0 = 0; % mol/s
18 nU0 = 0; % mol/s
19 kR = [0.2 0.1 0.04]; % s, L, mol
20 Ea = [35 25 40]*1e3; % J/mol
21 R = 8.314; % J/mol-K
22 nA0 = Ca0*v0; % mol/s
23 v = v0; % liquid phase
24
25 % Net rates
26 Ca = nA/v; % mol/L
27 Cb = nB/v; % mol/L

```

```

28 Cd = nD/v; % mol/L
29 k = kR.*exp(Ea/R*(1/298 - 1./T)); % s, L, mol
30 r1 = k(1)*Ca; % mol/L-s
31 r2 = k(2)*Cb; % mol/L-s
32 r3 = k(3)*Ca*Cd; % mol/s
33 rA = -r1 - r3; % molA/s
34 rB = r1 - r2; % molB/s
35 rD = 2*r2 - r3; % molD/s
36 rU = r3; % molU/s
37
38 % Balance equations
39 mbA = nA - nA0 - rA*V; % mol/s
40 mbB = nB - nB0 - rB*V; % mol/s
41 mbD = nD - nD0 - rD*V; % mol/s
42 mbU = nU - nU0 - rU*V; % mol/s
43 out = [mbA; mbB; mbD; mbU];
44 end

```

Running this script from the command line produces the following result:

```

>> cstrOptimize

nD = 0.014 mol/s

```

One approach at this point is to simply change  $T$  on Line 8 to lots of different values, writing down the resulting  $n_D$  each time, and then plotting the result. That's exceedingly tedious, however, so instead we'll use the `for` loop as mentioned earlier. The general idea of this approach was introduced in Section 1.4, and we'll also need the ability to pass extra input arguments to `fsolve` as described in Section 1.7.

First let's modify the local function `fun` to accept multiple inputs. We change the function definition to

```

6 function out = fun(Y, T)

```

and we delete Line 8 so that temperature is now an input to the function. We also change the script portion at the top of the file to be

```

1 % cstrOptimize.m
2 Y0 = [0 0 0 0]; % [nA, nB, nC, nD] guesses, mol/s
3 T = linspace(250, 450); % K
4 Y = zeros(4, length(T));
5 for i = 1:length(T)
6     Y(:, i) = fsolve(@(Y)fun(Y,T(i)), Y0);
7 end

```



Here, we've created a vector of evenly spaced values of  $T$  from 350 K to 450 K (Line 3), preallocated a storage matrix  $Y$  to hold one row for each molar flow rate (Line 4), and used a `for` loop to solve the system of equations at each value of  $T$  (Lines 5-7).

The resulting molar flow rates are shown in Figure 4.3 and the maximum flow rate of  $n_D$  is  $0.122 \text{ mol} \cdot \text{s}^{-1}$  at 334.8 K. One disadvantage of operating at this temperature is that production of the undesired compound U is also large, which could make downstream separation equipment more expensive. Separation considerations might favor a lower temperature where less D and less U are produced, but the selectivity of D to U is higher (more D is produced per unit U).

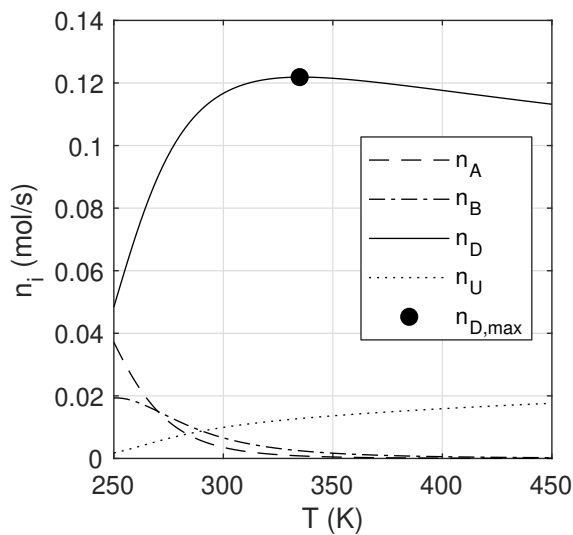


Figure 4.3: Outlet molar flow rates as a function of temperature for Example 4.7. The maximum flow rate for  $n_D$  is  $0.122 \text{ mol} \cdot \text{s}^{-1}$  at 334.8 K and was found using the `max` function.

#### Example 4.8: Liquid-phase isomerization of xylenes

Within a jacketed CSTR having a residence time of 17 min, the following network of elementary, reversible reactions takes place between the three isomers of xylene:

Reaction	$A \text{ (s}^{-1}\text{)}$	$E_a \text{ (kJ} \cdot \text{mol}^{-1}\text{)}$
$p\text{-C}_8\text{H}_{10} \rightleftharpoons o\text{-C}_8\text{H}_{10}$	0.019	26.8
$m\text{-C}_8\text{H}_{10} \rightleftharpoons o\text{-C}_8\text{H}_{10}$	0.013	27.4
$m\text{-C}_8\text{H}_{10} \rightleftharpoons p\text{-C}_8\text{H}_{10}$	0.021	27.1

An equimolar mixture of ortho-xylene, meta-xylene, and para-xylene at  $24 \text{ mol} \cdot \text{s}^{-1}$ , a total volumetric flow rate of  $3 \text{ L} \cdot \text{s}^{-1}$ , and  $80^\circ\text{C}$  is provided to the reactor. The reactor is heated by hot water ( $95^\circ\text{C}$ , 10 gpm) flowing though a half-pipe jacket around the outside of the reactor ( $A = 2.7 \text{ m}^2$ ).

Determine the outlet molar flow rate of each species and the temperatures of the reactor and utility stream.

Solution

Before beginning the analysis a basic stoich table and relevant physical properties, the latter from Table D.3, can be assembled as follows:

Sp	$\nu_1$	$\nu_2$	$\nu_3$	$n_{i0}$	$r_i$	$C_p$ ( $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ )	$\Delta H_f^\circ$ ( $\text{kJ} \cdot \text{mol}^{-1}$ )	$\Delta G_f^\circ$ ( $\text{kJ} \cdot \text{mol}^{-1}$ )
<i>m</i> -C <sub>8</sub> H <sub>10</sub>	0	-1	-1	24	$-r_2 - r_3$	185	-25.4	107.6
<i>p</i> -C <sub>8</sub> H <sub>10</sub>	-1	0	1	24	$-r_1 + r_3$	183	-24.4	110.6
<i>o</i> -C <sub>8</sub> H <sub>10</sub>	1	1	0	24	$r_1 + r_2$	187	-24.4	110.0
$\Delta M_1 =$						4	0.0	-0.6
$\Delta M_2 =$						2	1.0	2.4
$\Delta M_3 =$						-2	1.0	3.0

Notice we’ve omitted water from the stoichiometric table because it’s only present as a utility and utility streams don’t directly contact process streams. Eventually we’ll still need to know the heat capacity of the water so that we can solve the energy balance on the utility stream. We have to use lots of equations in the solution process so let’s briefly summarize where they’re coming from:

Variable	Defined in	Description
$v$	Equation (4.4)	Liquid volumetric flow rate
$C_i$	Equation (4.3)	Molar concentration for each species
$V$	Equation (4.5)	Residence time
$r$	Equation (3.19)	Rate law for reversible reaction
$k$	Equation (3.9)	Forward rate constant for each reaction
$K_a$	Equation (2.29)	Shortcut van't Hoff equation
$K_C$	Equation (3.20)	Equilibrium constant for reversible reaction
$r_i$	Equation (3.12)	Net reaction rate for each species
$h$	Table 4.1	Individual heat transfer coefficients
$U$	Equation (4.27)	Overall heat transfer coefficient
$Q$	Equation (4.26)	Heat transfer to utility
$\Delta H$	Equation (4.22)	$\Delta H$ adjustment for EB
mb	Equation (4.1)	Material balance for each species
eb	Equation (4.2)	Energy balance for reactor
eb_u	Equation (4.28)	Energy balance on utility stream

In the script below we've made extensive use of MATLAB's vector and matrix operations to illustrate how they simplify the code; see Appendix B to review these concepts if the extension of the general equations to matrix formulations is not clear. The only significant unit conversion is to get the mass flow rate of the water (utility) stream from the specification of 10 gallons per minute (gpm) as

$$\frac{10 \text{ gal}}{\text{min}} \frac{1 \text{ min}}{60 \text{ s}} \frac{1 \text{ m}^3}{264.17 \text{ gal}} \frac{1000 \text{ kg}}{1 \text{ m}^3} \frac{1 \text{ kmol}}{18 \text{ kg}} \frac{1000 \text{ mol}}{1 \text{ kmol}} = 35.05 \text{ mol} \cdot \text{s}^{-1}$$

It's also notable that  $\Delta H$  is implied to be constant through our use of Equation (2.29) to calculate  $K_{s_c}$  (and from this,  $K_C$ ), yet the  $\Delta H$  we use in the energy balance will be adjusted for temperature dependence using Equation (4.22). To assume constant  $\Delta H$  in one place and temperature-dependent  $\Delta H$  in another is something of a contradiction but one that we'll make as-needed. The correct way is to go all the way back to the polynomial forms for heat capacities and re-derive all relevant equations, a process that's tedious and doesn't provide a large correction in most cases.

Onwards to the solution script itself:

```
1 % cstrXylenes.m
2 Y0 = [8; 8; 8; 80+273; 95+273]; % [M P O T T_u]
```

```

3 Y = fsolve(@fun, Y0);
4 fprintf('[M P O] = [%3f %3f %3f] mol/s\n', Y(1:3));
5 fprintf('[T Tu] = [%1f %1f] degC\n', Y(4:5)-273);
6
7 function out = fun(Y)
8 n = Y(1:3); % mol/s
9 T = Y(4); % K
10 T_u = Y(5); % K
11
12 % Constants
13 tau = 17*60; % s
14 v0 = 3; % L/s
15 T0 = 80+273; % K, process T0
16 T_u0 = 95+273; % K, utility T0
17 area = 2.5; % m2, heat exchanger area
18 A = [0.019; 0.013; 0.021]; % 1/s
19 Ea = [26.8; 27.4; 27.1]*1e3; % J/mol
20 dHf = [-25.4; -24.4; -24.4]*1e3; % J/mol, [M; P; O]
21 dGf = [107.6; 110.6; 110.0]*1e3; % J/mol, [M; P; O]
22 Cp = [185; 183; 187]; % J/mol-K, [M; P; O]
23 nu = [0 -1 -1; % M in [r1 r2 r3]
24       -1 0 1; % P in [r1 r2 r3]
25       1 1 0]; % O in [r1 r2 r3]
26 n0 = [8; 8; 8]; % [M; P; O], mol/s
27 h = [1000; 1500]; % [process; utility], W/m2-K
28 m_u = 10/60/264.17*1000/18*1000; % mol/s, utility
29 Cp_u = 75; % J/mol-K
30 R = 8.314; % J/mol-K
31
32 % Thermodynamics
33 dH = nu' * dHf; % [r1; r2; r3], J/mol
34 dG = nu' * dGf; % [r1; r2; r3], J/mol
35 dCp = nu' * Cp; % [r1; r2; r3] J/mol-K
36 K0 = exp(-dG/(R*298)); % [r1; r2; r3]
37 K1 = exp(dH/R*(1/298-1/T)); % [r1; r2; r3]
38 Kc = K0.*K1; % [r1; r2; r3]
39 dH = dH + dCp*(T-298); % [r1; r2; r3]
40 U = 1/(1/h(1) + 1/h(2)); % W/m2-K
41 Q = U*area*(T_u-T); % J/s
42
43 % Rate expressions
44 V = tau*v0; % L
45 C = n/v0; % mol/L, [M; P; O]
46 k = A.*exp(-Ea/(R*T)); % [r1; r2; r3], 1/s
47 r(1,1) = k(1)*(C(2)-C(3)/Kc(1)); % mol/L-s, P <=> O
48 r(2,1) = k(2)*(C(1)-C(3)/Kc(2)); % mol/L-s, M <=> O
49 r(3,1) = k(3)*(C(1)-C(2)/Kc(3)); % mol/L-s, M <=> P

```

```

50
51 % Material and energy balances
52 mb = n0 - n + nu*r*V; % mol/s, [M; P; O]
53 eb = sum(n0.*Cp*(T-T0)) + V*sum(r.*dH) - Q; % J/s
54 eb_u = m_u*Cp_u*(T_u-T_u0) + Q; % J/s
55 out = [mb; eb; eb_u];
56 end

```

Running this script produces the following output:

```

>> cstrXylenes

[M P O] = [8.053 7.958 7.989] mol/s
[T Tu] = [82.7 90.5] degC

```

The conversion is quite small and is headed in the direction of meta-xylene. For exceptionally long residence times the outlet composition approaches the equilibrium composition, and to decide how long is “long enough” try playing around with tau on Line 13.

## 4.6 Solutions from fsolve should be checked for stability

One of the trickier parts of solving large systems with fsolve is to figure out how to make an initial guess for each variable. Here are a few general guidelines that can help you in your initial guess procedure:

**Assume the value is positive.** Most of the quantities we solve for most often—molar or mass flow rates, temperatures, conversions—are positive quantities.

**Start with low conversion and no temperature change.** If absolutely no other information is available, start with a small conversion (on the order of 10% or less) and assume the reactor temperature is the same as the inlet stream temperature.

**Solve a simpler problem first.** Suppose  $X$  and  $T$  are unknown in a non-isothermal system. First solve the *isothermal* system to get a guess for  $X$ , then assume this value of  $X$  is constant and solve the *non-isothermal* with this value of  $X$  to get a guess for  $T$ . Use these two values of  $X$  and  $T$  as initial guesses for the original non-isothermal system.

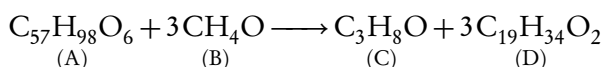
Whatever your guessing procedure, the solution provided by fsolve should be *stable*. A stable solution is one that doesn’t depend too much on the initial guess,<sup>5</sup> and the easiest way to check

<sup>5</sup>Multiple solutions are possible but aren’t an indicator of an unstable system.

for stability is to try a few different guesses to see if the solution changes. If the solution changes significantly for small variations in the initial guess then the solution is not stable. Let's set up an example and then take a look at its stability afterwards.

### Example 4.9: Biodiesel from soybean oil

Soybean oil is a triglyceride ( $C_{57}H_{98}O_6$ ), which is a molecule that looks like three, long-chain hydrocarbons attached to a glycerol-like backbone. When mixed with methanol ( $CH_4O$ ) and a catalyst, the long-chain hydrocarbons can be broken off to form biodiesel ( $C_{19}H_{34}O_2$ ) and glycerol ( $C_3H_8O$ ) according to the reaction



Suppose you're interested in producing small batches of biodiesel according to this reaction and you've set up a 5 gallon bucket as your CSTR. The feed to the reactor is  $1.5 \text{ kg} \cdot \text{min}^{-1}$  soybean oil and—because nobody warned you about how dangerous it is—an equimolar flow of pure methanol. The total volumetric flow rate of the feed is 0.5 gpm at  $40^\circ\text{C}$ . The reaction is endothermic so you've rigged up a makeshift heater which you guess has  $UA = 24 \text{ kJ} \cdot \text{min}^{-1} \cdot \text{K}^{-1}$  and  $T_u = 50^\circ\text{C}$  (constant).

Determine the soybean oil conversion and temperature of this reactor. Additional information about this reaction<sup>a</sup> is provided below. Comment on the reasonableness of your result.

$$\begin{aligned} -r_A &= kC_A^2 \\ k &= 0.14 \text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1} \text{ at } 50^\circ\text{C} \\ E_a &= 49.8 \text{ kJ} \cdot \text{mol}^{-1} \end{aligned}$$

<sup>a</sup>Adapted from Nouredini, H.; Zhu, D. Kinetics of transesterification of soybean oil. JAOCS, 74 (1997) 1457-1463.

### Solution

As always, we start with a stoichiometric table to organize information given in the problem statement:

Sp.	$\nu_i$	$n_{i0}$	$r_i$	$n_i$	$C_p$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )
A	-1	$n_{A0}$	$-r$	$n_{A0} - rV$	1782	-2431
B	-3	$3n_{A0}$	$-3r$	$3n_{A0} - 3rV$	81	-238.7
C	1	0	$r$	$rV$	222	-670.0
D	3	0	$3r$	$3rV$	580	-605.0

Next up are applications of the general material and energy balances for a CSTR, Equations (4.1) and (4.2). Since we need to solve for  $X_A$  and  $T$  simultaneously we're going to need fsolve, and since there are two variables to solve for we're going to need the procedures developed in Section 1.8. Our general solution script, with just the bare bones of solving—but not analyzing—the problem will look like this:

```

1 % cstrBiodiesel.m
2 Y0 = [0.5 30+273]; % [Xguess, Tguess]
3 Y = fsolve(@fun, Y0); % [Xsol, Tsol]
4
5 function out = fun(Y)
6 X = Y(1);
7 T = Y(2);
8 % more code here
9 out(1,1) = nA - nA0 - rA*V;
10 out(2,1) = (nA0*CpA+nB0*CpB)*(T-T0) + r*V*dH - Q;
11 end

```

where out(1,1) is the material balance for A in root-finding form, and out(2,1) is the energy balance in root-finding form. A few things to keep in mind before writing the completed code:

- The initial guess for  $X_A$  is a shot in the dark. The initial guess for  $T$  assumes the reactor is isothermal, which is consistent with a small conversion.
- To ensure dimensional consistency we must pick one set of units and stick with it. The set of units used in this solution are minutes, liters, joules, kilograms, moles, and kelvin. It's good to get into the habit of adding comments to indicate units.
- This example makes use of vectorized code in some places to make the code more compact, such as through the use of the sum function and elementwise operator .\* for multiplication.

Onwards to the coded solution:

```

1 % cstrBiodiesel.m
2 Y0 = [0.1 40+273]; % [Xguess, Tguess]
3 Y = fsolve(@fun, Y0); % [Xsol, Tsol]
4 fprintf('X = %.3f, T = %.1f K\n', Y(1), Y(2));
5
6 function out = fun(Y)
7 X = Y(1);
8 T = Y(2); % K
9
10 % Constants
11 V = 18.93; % L (5 gal)
12 v0 = 1.893; % L/min (0.5 gpm)
13 mA0 = 1.5; % kg/min
14 kR = 0.014*60; % L/mol-min
15 TR = 50+273; % K
16 T0 = 40+273; % K
17 Ea = 49.8e3; % J/mol
18 nu = [-1 -3 1 3]; % [a b c d]
19 Cp = [1782 81 222 580]; % J/mol-K
20 dHf = [-2431 -238.7 -670.0 -605.0]*1e3; % J/mol
21 R = 8.314; % J/mol-K
22
23 % Rate equations
24 nA0 = mA0/879*1000; % mol/min
25 nB0 = 3*nA0; % mol/min
26 nA = nA0*(1-X); % mol/min
27 dH = sum(nu.*dHf) + sum(nu.*Cp)*(T-298); % J/mol
28 Ca = nA/v0; % mol/L
29 k = kR*exp(Ea/R*(1/TR-1/T)); % L/mol-min
30 rA = -k*Ca.^2; % mol/L-min
31
32 % Heat exchanger equations
33 UA = 24e3; % J/min-K
34 Tu = 50+273; % K, constant
35 Q = UA*(Tu-T); % J/min
36
37 % Balance equations
38 out(1,1) = nA - nA0 - rA*V; % MB on A, mol/min
39 out(2,1) = (nA0*Cp(1)+nB0*Cp(2))*(T-T0) - rA*V*dH - Q; % J/min
40 end

```

Running this script at the command line produces the following output:

```

1 >> cstrBiodiesel.m
2
3 Equation solved, fsolve stalled.

```



```

4
5 X = 0.499, T = 301.2 K

```

Right away we should be suspicious of this result because `fsolve` told us that it “stalled”, which means that `fsolve` is not as confident of its result as it could be. If we vary our initial guess of  $X$  from  $X_{\text{guess}} = 0$  to  $0.9$  in increments of  $0.05$ , and vary our initial guess of  $T$  from  $T_{\text{guess}} = 250$  K to  $350$  K in increments of  $5$  K, the solution values of  $X$  and  $T$  vary as shown in Figure 4.4.

Optimally the value of  $X_A$  and  $T$  will not vary at all with the initial guess of either  $X_A$  or  $T$ , but from Figure 4.4 there is significant variation, such as values of  $X_A$  spanning from less than  $-2$  all the way up to  $1$ . Similarly the solution values of  $T$  range from about  $250$  K all the way up to  $350$  K, so we should have no confidence in any of these results.

The reason for the variation in this particular problem is that the material and energy balances are *poorly scaled*. If we place a breakpoint on Line 40, run the script, and check the value of `out` from the (debugger enabled) command prompt we’ll see

```

K>> format long; disp(out); format
      1e+05 *

      0.000000966385826
      1.140606196852031

```

What this output is telling us is that the numerical value of the material balance in root-finding form (Line 38) is around  $0.1$ , but the numerical value of the energy balance in root-finding form

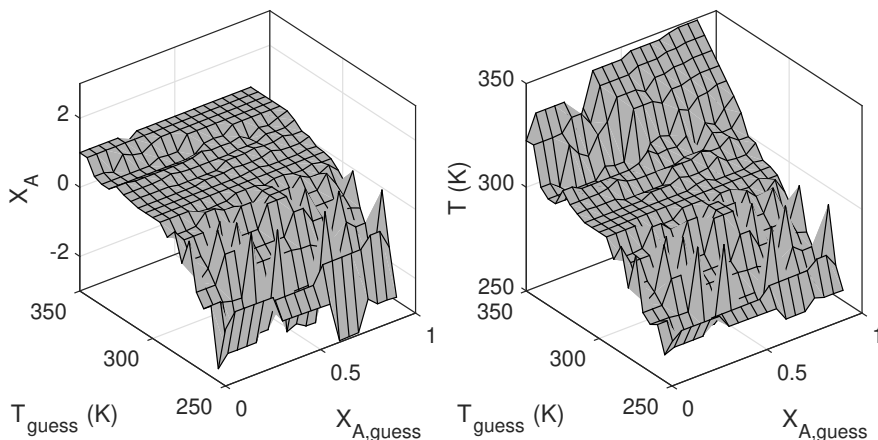


Figure 4.4: Solution values of  $X_A$  and  $T$  in Example 4.9 with variations in the initial guess of  $X_A$  (as  $X_{A,\text{guess}}$ ) and  $T$  (as  $T_{\text{guess}}$ ).

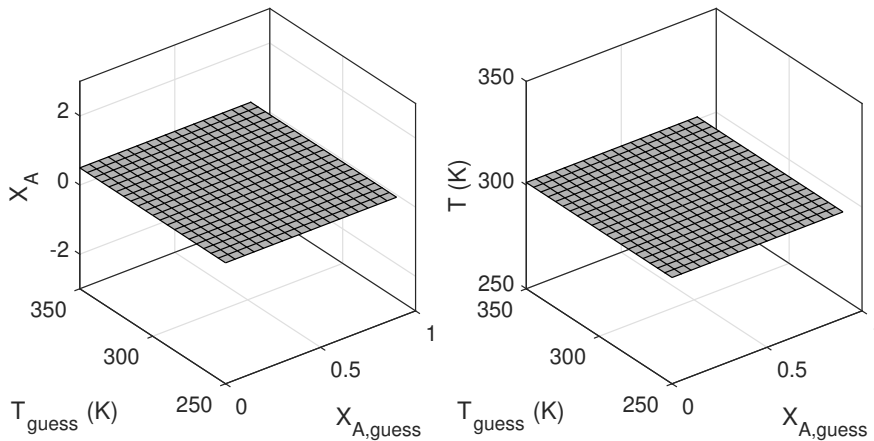


Figure 4.5: Stability of Example 4.9 after scaling the material and energy balances by  $10$  and  $10^{-5}$ . Regardless of initial guess, `fsolve` finds a solution of  $X_A = 0.4985$  and  $T = 301.2\text{ K}$ , which indicates that the solution is now stable.

(Line 39) is about  $10^5$ . This is problematic because the quality of the guess (as evaluated by `fsolve`'s algorithms) is dominated by the energy balance: any guess always produces a small value of the material balance, so any guess is viewed as “good.”

There's a simple way to correct poor scaling once you identify it: multiply each of the equations by a *scaling factor* to make the two values' scales close to order 1. For example, we could multiply the material and energy balances of Example 4.9 by scaling factors of  $10$  and  $10^{-5}$  by modifying the last few lines of our script to be

```
38 out(1,1) = nA - nA0 - rA*V; % MB on A, mol/min
39 out(2,1) = (nA0*Cp(1)+nB0*Cp(2))*(T-T0) - rA*V*dH - Q; % J/min
40 out = out .* [10; 1e-5]; % scaling factors for [mb; eb]
41 end
```

If we now repeat the initial guess variation process that we used to produce Figure 4.4, we'll produce plots as shown in Figure 4.5 which reveal almost no variation: any initial guess combination of  $X_A$  and  $T$  gives the same solution of  $X_A = 0.4985$  and  $T = 301.2\text{ K}$ , which means that this solution is stable.

As you may have noticed in some of the problems of this chapter, the vector (and matrix) capabilities of MATLAB can really shine in these types of problems because they essentially put a cap on how difficult the code can be: when written in vector form, the balance equations automatically scale to the size of the problem. More details about how to set up problems in this way are provided in Appendix B but before you review that material you should try this example in whatever way you can to reproduce the results (*e.g.*, write out all of the relevant equations).

# Chapter 5

## Special Cases of the CSTR

“A very simple case, not that I’m particularly bright sir.”

Lieutenant Columbo, *Columbo*, “How to Dial a Murder” (S07E04, 1978)

### 5.1 A fluidized bed reactor can be modeled as a gas-phase CSTR

A great number of industrially-relevant reactions take place in the gas phase and often require the use of a solid-phase catalyst to achieve useful rates. Catalysts are typically micro- or nano-sized metals—Pt, Rh, Ni, etc.—attached to an inert support such as a polymer or ceramic. Somewhat confusingly, both the catalyst and support are often referred to as “catalyst” or “catalyst particle,” but that’s how it goes sometimes. When the reacting species are in a phase different from the catalyst, the process is called *heterogeneous catalysis* and is an area of ongoing research.

The classic CSTR that we learned about in Chapter 4 is best suited for liquid phase materials but the equations of a CSTR can be applied anywhere the two primary assumptions of the CSTR hold—specifically, anywhere we’ve got a well-mixed, constant volume reactor. One such reactor is the *fluidized bed reactor*, wherein the reacting species enters through the bottom of the reactor and flows through a constant-height particle bed as in Figure 5.1. Under appropriate flow conditions the drag force experienced by the particles is enough to match or slightly exceed gravitational forces and so the particles lift and become “fluidized.” The entering stream can be liquid or gas phase and similar flow patterns will develop, but for this section we’ll assume the entering stream is gas phase only.

The random motion of the particles couples with the ever-changing flow patterns of the gas to provide mixing throughout the volume of particles, and if we assume that the reaction

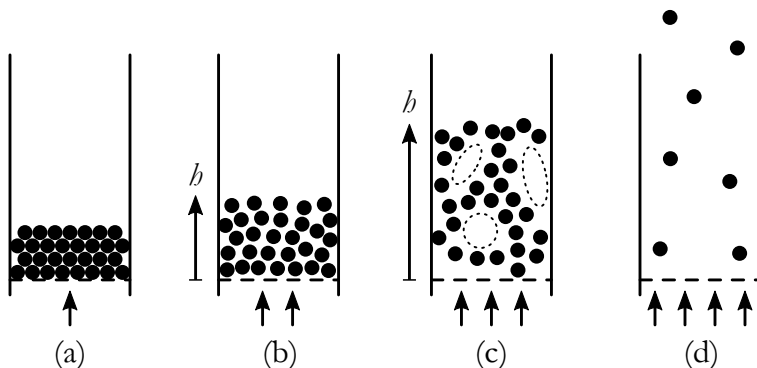


Figure 5.1: Different flow patterns within fluidized beds, where gravity is acting down and gas enters from the bottom. (a) *Fixed bed*. At low gas velocities the particles are fixed. (b) *Fluidized bed*. At moderate gas velocities the catalyst particles are lifted and circulate up to a height  $h$ . (c) *Bubbling bed*. At still higher gas velocities the bed height  $h$  increases and voids form within the bed. (d) *Transport*. At extreme gas, catalyst velocities particles are conveyed out of the reactor.

proceeds uniformly throughout this volume then we can write the material and energy balances for a fluidized bed just as we did for the general CSTR in Chapter 4,

$$n_i = n_{i0} + r_i V \quad (4.1)$$

and

$$0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q - W_s. \quad (4.2)$$

In both of these equations  $V = hA_c$  where  $h$  is the height of the fluidized bed and  $A_c$  is the cross-sectional area of the reactor. The actual structure of the reactor is nearly always larger than  $V$  because empty space is provided above the bed to allow for fluctuations in bed height,  $h$ . In this regime the rate of reaction is nearly zero, however, because the catalyst is absent.

When dealing with gases instead of liquids there are no differences in either interpretation or calculation of any terms in Equations (4.1) and (4.2), except for the following three tricky requirements:

**Tricky Requirement for Gases 1:** When looking up  $\Delta H_{f,i}^\circ$  to calculate  $\Delta H$ , you must be sure to get the values for the gas phase instead of the liquid phase. Table D.2 provides such values for many common chemicals.

**Tricky Requirement for Gases 2:** When calculating the volumetric flow rates  $v$  or  $v_0$  you'll likely need the ideal gas law (see below). The expression for  $v$  in particular can be lengthy.

**Tricky Requirement for Gases 3:** Due to Requirement 2, expressions for  $C_i$  can become complicated because they're calculated from  $v$  as  $C_i = n_i/v$ .

There's not much detail to add to Requirement 1 because it's just changing which resource you use for one step of a calculation. For Requirements 2 and 3 we'll use our old friend the ideal gas law as

$$P_0 v_0 = n_{T0} R T_0 \quad \text{or} \quad P v = n_T R T, \quad (5.1)$$

where  $v$  is the volumetric flow rate of the stream,  $n_T$  is the total molar flow rate of the stream, the variables subscripted with 0 refer to properties of the inlet stream, and the variables not subscripted with 0 refer to properties of the outlet stream.<sup>1</sup>

As a small aside, Equation (5.1) is the same ideal gas law you've used in previous courses and you can use all the properties of ideal gases as you've used them before as well. For example, the average molar mass  $\overline{MM}$  of an ideal gas mixture is

$$\overline{MM} = \sum_{i=A}^Z y_i MM_i, \quad (5.2)$$

where  $y_i = n_i/n_T$  is the mole fraction of species  $i$  in the stream. The partial pressure of species  $i$  in the stream is  $P_i = y_i P$ , and Equation (5.1) can be rewritten in terms of partial pressures as

$$P_{i0} v_0 = n_{i0} R T_0 \quad \text{or} \quad P_i v = n_i R T. \quad (5.3)$$

Getting back to Requirement 2, we can relate the outlet volumetric flow rate  $v$  to the inlet volumetric flow rate  $v_0$  by taking the ratio of Equation (5.1) for the outlet to the inlet as

$$\frac{P v}{P_0 v_0} = \frac{n_T R T}{n_{T0} R T_0} \quad (5.4)$$

and rearranging for  $v$  to yield

$$v = v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0} \quad (5.5)$$

where we've eliminated  $R$  because it's the same  $R$  for both streams. Equation (5.5) is an important equation for gas-phase systems and will pop up again throughout this text. Should a time ever come where you need more precision than the ideal gas law offers through Equation (5.5) you'll have to repeat the process of relating  $v$  to  $v_0$  with the equation of state of your choice.<sup>2</sup>

<sup>1</sup>Except for the gas constant  $R$  which is always the same  $R$  everywhere.

<sup>2</sup>For example, you could write  $P v = z n R T$  and  $P_0 v_0 = z_0 n_{T0} R T_0$  for the outlet and inlet streams and use your favorite equation of state to find  $z$  and  $z_0$ .

Finally, as noted in Requirement 3 we've got to revisit our use of the molar concentration  $C_i$  because it was defined as

$$C_i = \frac{n_i}{v}$$

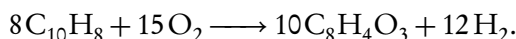
which obviously depends on the volumetric flow rate  $v$  that we just discussed. The approximation  $v = v_0$  that we used for liquids is a terrible approximation for gases, so instead we'll substitute Equation (5.5) into our definition of  $C_i$  to get

$$C_i = \frac{n_i}{v} = \frac{n_i}{v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0}}, \quad (5.6)$$

which indicates that we'll obviously have a little more writing to do whenever we're working with gases compared to liquids since  $C_i$  can appear in many ways within the rate laws. Avoiding the lengthiness of Equation (5.6) is one of the many reasons why gas-phase rate laws and concentrations are often expressed in terms of partial pressures instead of molar concentrations.<sup>3</sup>

### Example 5.1: Catalytic cracking of naphthalene in a fluidized bed

DeMaria and coworkers<sup>a</sup> found that one of the ways naphthalene ( $C_{10}H_8$ ) and oxygen react over a vanadium oxide catalyst yields phthalic anhydride ( $C_8H_4O_3$ ) according to the simplified reaction



Within an isothermal, isobaric fluidized bed reactor (2 in diameter, 15 ft bed height) the reaction rate was approximated as  $r = kC_N$ , where  $C_N$  is the molar concentration of naphthalene and  $k$  was measured to be  $4.0 \text{ s}^{-1}$  at  $370^\circ\text{C}$ .

If a feed gas contains 1 mol% naphthalene and the balance air is provided to the reactor at 1 atm, what is the outlet mole fraction of phthalic anhydride for a residence time of 20 s at  $370^\circ\text{C}$ ?

<sup>a</sup>DeMaria, F.; Longfield, J.E.; Butler, G. Catalytic Reactor Design. Ind. Eng. Chem. 1961, 53, 4, 259-266.

<sup>3</sup>One of the other reasons is because "it's always been done that way," which is both a terrible justification and a very good justification since a significant amount of literature exists using partial pressures.

**Solution**

Let's get started with a stoichiometric table to organize some information:

Sp.	$\nu_i$	$n_{i0}$	$r_i$	$n_i$
$\text{C}_{10}\text{H}_8$	-8	$0.01n_{T0}$	$-8r$	$0.01n_{T0} - 8rV$
$\text{O}_2$	-15	$0.99(0.21)n_{T0}$	$-15r$	$0.99(0.21)n_{T0} - 15rV$
$\text{C}_8\text{H}_4\text{O}_3$	10	0	$10r$	$10rV$
$\text{H}_2$	12	0	$12r$	$12rV$
$\text{N}_2$	0	$0.99(0.79)n_{T0}$	0	$0.99(0.79)n_{T0}$

For the inlet molar flow rates of oxygen and nitrogen we've assumed "air" is 21 mol%  $\text{O}_2$  and 79 mol%  $\text{N}_2$ . Naturally we'll return to the general material balance, Equation (4.1), to solve this isothermal problem but we've got some work to do in terms of getting values to start with. The volume of the reactor is

$$V = \frac{1}{4}\pi D^2 h = \frac{1}{4}\pi (2 \text{ in})^2 (15 \text{ ft}) = 9.267 \text{ L},$$

the inlet volumetric flow rate is

$$v_0 = \frac{V}{\tau} = \frac{9.27 \text{ L}}{20 \text{ s}} = 0.463 \text{ L} \cdot \text{s}^{-1},$$

and the total inlet molar flow rate is

$$n_{T0} = \frac{P_0 v_0}{RT_0} = \frac{(1 \text{ atm})(0.463 \text{ L} \cdot \text{s}^{-1})}{R(643 \text{ K})} = 8.788 \text{ mmol} \cdot \text{s}^{-1}$$

The volumetric flow rate of the outlet stream is given by Equation (5.5), which simplifies to

$$v = v_0 \frac{n_T}{n_{T0}}$$

for this isobaric, isothermal reactor. We shouldn't expect  $n_T = n_{T0}$  because  $\delta = \sum \nu_i = 10 + 12 - 8 - 15 = -1$ , which means that we lose 1 mol each time the reaction proceeds as written above. With these ideas in mind we can now approach the problem using fsolve just as we have previously, with the only variation being a new calculation for  $v$  to allow us to calculate  $C_i$  according to Equation (5.6). Note that we're going to

solve all five material balances because the problem asked us to calculate an outlet mole fraction, so it'll be convenient if we have all five outlet molar flow rates.

```

1 % fluidizedBed.m
2 Y0 = [1 1 0 0 10]; % mmol/s, [N O2 PA H2 N2]
3 Y = fsolve(@fun, Y0);
4 fprintf('x_PA = %.3f\n', Y(3)/sum(Y));
5
6 function out = fun(Y)
7 n = Y; % mmol/s, [N O2 PA H2 N2]
8
9 % constants
10 D = 2*2.54/100; % m
11 h = 15*12*2.54/100; % m
12 tau = 20; % s
13 P0 = 1; % atm
14 T0 = 370+273; % K
15 R = 0.082; % L-atm/mol-K
16 y0 = [0.01, 0.2079, 0, 0, 0.7821]; % [N O2 PA H2 N2]
17 nu = [-8 -15 10 12 0]; % [N O2 PA H2 N2]
18 k = 4; % 1/s at 370 C
19
20 % algebraic equations
21 V = 0.25*pi*D^2*h*1000; % L
22 v0 = V/tau; % L/s
23 nT0 = P0*v0/(R*T0)*1000; % mmol/s
24 n0 = y0*nT0; % mmol/s, [N O2 PA H2 N2]
25
26 % rate equations
27 v = v0*sum(n)/nT0; % L/s
28 C = n/v; % mmol/L, [N O2 PA H2 N2]
29 r = nu*k*C(1); % mmol/L-s, [N O2 PA H2 N2]
30
31 % balance equations
32 out(1,1) = n(1) - n0(1) - r(1)*V; % mmol/s, N
33 out(2,1) = n(2) - n0(2) - r(2)*V; % mmol/s, O2
34 out(3,1) = n(3) - n0(3) - r(3)*V; % mmol/s, PA
35 out(4,1) = n(4) - n0(4) - r(4)*V; % mmol/s, H2
36 out(5,1) = n(5) - n0(5) - r(5)*V; % mmol/s, N2
37 end

```

Running this script from the command line produces the following output:

```

>> fluidizedBed.m
x_PA = 0.012

```

which means the outlet mole fraction of phthalic anhydride is about 1.2 mol%. That's not particularly high but then again the conversion of naphthalene was more than 99%,



which is fantastic.

Also, this was a good example of how we can make use of MATLAB's vector operations to simplify the code. We could go a little further by replacing Lines 32 through 36 with a single line as

```
32 out = n - n0 - r*V; % mmol/s, [N O2 PA H2 N2]
```

which produces the same vector of material balances that we had before. You can see more tricks to simplify your code in Appendix B.

## 5.2 Rewriting equations in terms of conversion can be useful

Sometimes when we're limited to solving problems by hand we can make a few simplifications which enable us to rewrite the material and energy balances in terms of the conversion of a single component,  $X_A$ . As we'll see in Chapter 11 such expressions also enable quick estimates of multi-unit systems as well. Always keep in mind, however, that the seemingly "new" ideas of this chapter are simply special cases of the general material and energy balances, and that you can always go back to those equations to correctly analyze a system.

We start by rearranging the general CSTR material balance, Equation (4.1), for species A to isolate  $V$  on one side and everything else on the other side,

$$V = \frac{n_A - n_{A0}}{r_A}. \quad (5.7)$$

The reason for isolating  $V$  is because we're often interested in calculating the cost of a reactor, and cost is closely related to size. Recalling the definition of conversion from Equation (2.33) we can rewrite this material balance as

$$V = \frac{n_{A0}X_A}{-r_A}, \quad (5.8)$$

which is a particularly convenient form for solving CSTR problems by hand (hence the box around the equation) as well as for identifying trends that may not be readily apparent from Equation (4.1). For example, we can see from Equation (5.8) that the volume of a reactor gets bigger if either conversion goes up or reaction rate goes down, a conclusion that should be intuitively sensible.

Similarly, the general CSTR energy balance can be rewritten in terms of conversion by

substituting Equation (5.8) into Equation (4.2) and rearranging for  $X_A$  to get

$$X_A = \frac{\sum_{i=A}^Z \Theta_i C_{P,i} (T - T_0) - \frac{Q}{n_{A0}}}{\frac{\Delta H}{v_A}}. \quad (5.9)$$

It's important to keep in mind that Equations (5.8) and (5.9) are just special cases of Equations (4.1) and (4.2); there's no new information here. Except for problems written explicitly to be solved by Equation (5.8) you can always go back to Equation (4.1) if it's more convenient, and you shouldn't try to use both Equation (4.1) and Equation (5.8) at the same time because they're different forms of the same material balance. No equivalent form of Equations (5.8) and (5.9) exists for species other than A because if we're using either of these equations then the implication is that we're *primarily* interested in species A.

The general workflow when attempting to use Equations (5.8) and (5.9) will be to express  $r_A$  as a function of  $X_A$ , substitute the resulting expression into Equation (5.8), and then solve Equation (5.8) for one unknown, usually  $V$  or  $X_A$ ; if necessary then we also use Equation (5.9) to aid in this solution process. Since most rate equations are functions of molar concentration we'll also need to express these various  $C_i$  terms as functions of conversion as well. If we substitute Equation (2.35) into the general definition of  $C_i$  we get

$$C_i = \frac{n_i}{v} = \frac{n_{A0} \left( \Theta_i - \frac{v_i}{v_A} X_A \right)}{v}, \quad (5.10)$$

and now we have to decide what to do with the  $v$  term depending on whether we're dealing with liquids or gases.

### 5.2.1 $C_i(X_A)$ for liquids

For liquids we assume  $v = v_0$  so that Equation (5.10) becomes

$$C_i = C_{A0} \left( \Theta_i - \frac{v_i}{v_A} X_A \right) \quad (5.11)$$

For species A Equation (5.11) simplifies to

$$C_A = C_{A0} (1 - X_A) \quad (5.12)$$

but there are no similar simplifications for other species because  $\Theta_i$  and  $v_i/v_A$  vary from case to case. Should we need the total outlet molar flow rate we can still use Equation (2.37) as

$$n_T = n_{T0} - n_{A0} \frac{\delta}{v_A} X_A, \quad (2.37)$$

and should we need the total outlet molar concentration we can combine Equation (2.37) with the definition of molar concentration to get

$$C_T = \frac{n_T}{v} = \frac{n_{T0} - n_{A0} \frac{\delta}{v_A} X_A}{v_0} = C_{T0} - C_{A0} \frac{\delta}{v_A} X_A. \quad (5.13)$$

### 5.2.2 $C_i(X_A)$ for gases

For gases we've already seen that the ideal gas law provides a general expression for  $v$  as

$$v = v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0} \quad (5.5)$$

We need to work on the  $n_T/n_{T0}$  term because  $n_T$  will generally be a function of conversion. Substituting Equation (2.37) into  $n_T/n_{T0}$  yields

$$\frac{n_T}{n_{T0}} = \frac{n_{T0} - n_{A0} \frac{\delta}{v_A} X_A}{n_{T0}} = 1 - y_{A0} \frac{\delta}{v_A} X_A,$$

where  $y_{A0}$  is the inlet mole fraction of A. Substituting this expression for  $n_T/n_{T0}$  into Equation (5.5) and the result into Equation (5.10) yields

$$C_i = C_{A0} \frac{\Theta_i - \frac{v_i}{v_A} X_A}{\left(1 - y_{A0} \frac{\delta}{v_A} X_A\right) \frac{P_0}{P} \frac{T}{T_0}}. \quad (5.14)$$

Unlike with liquids there's not much simplification that can be performed for Equation (5.14), even for species A. Indeed Equation (5.14) highlights why most gas-phase problems solved using Equation (5.8) tend to be simple: the approach in general only works for single-reaction systems because we're expressing all other molar flow rates as functions of  $X_A$ , and Equation (5.14) can get pretty nasty unless the system is also isobaric and isothermal with  $\delta = 0$ . While it's true that not many systems obey all of these conditions, it's also true that a fair amount can be learned about general relationships between rates, volumes, and conversions by *assuming* the system is simple enough to be modeled under these conditions.

**Example 5.2: Isomerization in a CSTR, revisited**

Repeat Example 4.3, Part (a) but calculate the conversion possible with a 5000 L reactor.

**Solution**

This problem is a good one for Equation (5.8) because it's isothermal, isobaric, liquid-phase, and there's only one reaction, so we start with the material balance as

$$V = \frac{n_{A0}X_A}{-r_A},$$

and from Example 4.3 we also know

$$n_{A0} = 18.90 \text{ mol} \cdot \text{s}^{-1}$$

$$v_0 = 2.30 \text{ L} \cdot \text{s}^{-1}$$

$$K_C = 0.33$$

$$k = 5.43 \times 10^{-6} \text{ s}^{-1}$$

$$r_A = -k \left( C_A - \frac{C_B}{K_C} \right).$$

Substituting the rate into the material balance yields

$$V = n_{A0}X_A \left[ k \left( C_A - \frac{C_B}{K_C} \right) \right]^{-1},$$

and it looks like we know almost everything in here except for  $C_A$  and  $C_B$ . For these two quantities we go to Equation (5.11) to get

$$C_A = C_{A0}(1 - X_A)$$

$$C_B = C_{A0} \left( \Theta_B - \frac{(1)}{(-1)} X_A \right) = C_{A0} X_A,$$

where we made use of the “pure A” feed specification to identify that  $\Theta_B = n_{B0}/n_{A0} = 0$ . As an aside, now is a good time to point out that much of this information can be neatly summarized with additional or modified columns in the stoichiometric table as follows:

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$C_i$
A	-1	$n_{A0}$	1	$-r$	$C_{A0}(1-X_A)$
B	1	0	0	$r$	$C_{A0}X_A$

Substituting into the material balance yields

$$V = n_{A0}X_A \left[ k \left( C_{A0}(1-X_A) - \frac{C_{A0}X_A}{K_C} \right) \right]^{-1},$$

which after a little rearranging and simplification can get down to

$$V = \frac{v_0}{k} \frac{X_A}{1 - X_A - \frac{X_A}{K_C}}$$

or

$$V = \frac{v_0}{k} \frac{X_A}{1 - 4.03X_A},$$

where we used  $K_C = 0.33$ . We can rearrange one more time to isolate  $X_A$  as

$$X_A = \frac{\tau k}{1 + 4.03\tau k},$$

where  $\tau = V/v_0 = 5000 \text{ L} / 2.30 \text{ L} \cdot \text{s}^{-1} = 2174 \text{ s}$ . Before we substitute numbers any further let's check the units and try to estimate what  $X_A$  should be. The units of  $\tau$  are time (in this case, seconds) and the units of  $k$  are inverse time (in this case,  $\text{s}^{-1}$ ), so their product is dimensionless and therefore  $X_A$  is dimensionless too (as it should be). From Example 4.3 we know that we get about 8% conversion in a 50000 L reactor and since the reactor for this problem is quite a bit smaller at just 5000 L we therefore expect a smaller conversion, probably on the order of 1% or less. Substituting numbers yields

$$X_A = \frac{(2174 \text{ s})(5.43 \times 10^{-6} \text{ s}^{-1})}{1 + 4.03(2174 \text{ s})(5.43 \times 10^{-6} \text{ s}^{-1})} = 0.0113,$$

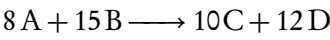
which is about 1.1%.

### Example 5.3: Naphthalene cracking, revisited

Repeat Example 5.1 using hand calculations instead of MATLAB.

Solution

As a brief review the reaction is



with  $k = 4.0\text{ s}^{-1}$  and  $\tau = 20\text{ s}$ . This is another good one for Equation (5.8) because the reactor is isothermal, isobaric, and there's only one reaction. One potential problem is that it's a gas-phase reaction so the expressions for  $v$  and  $C_i$  as functions of  $X_A$  might be tricky. We also have to specify what we mean by species A, and the obvious choice here is naphthalene because it's most likely the more expensive reagent (oxygen, after all, appears to be coming from the air for this system).

From our previous work we can fill in some preliminary numbers in our stoichiometric table so that we don't have to repeat quite so many calculations:

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$n_i$
A	-8	0.0879	1	$-8r$	$n_{A0}(1 - X_A)$
B	-15	1.8269	20.79	$-15r$	$n_{A0}(20.79 - (15/8)X_A)$
C	10	0	0	$10r$	$n_{A0}(10/8)X_A$
D	12	0	0	$12r$	$n_{A0}(12/8)X_A$
inert	0	6.8728	78.21	0	6.8731
$\delta = -1 \quad n_{T0} = 8.7880$					$n_T = n_{T0} - n_{A0}(1/8)X_A$

Substituting the rate law  $r_A = -8r = -8kC_A$  into the material balance of Equation (5.8) yields

$$V = \frac{n_{A0}X_A}{-r_A} = \frac{n_{A0}X_A}{8kC_A},$$

into which we must now substitute the aforementioned mess of Equation (5.14), which for  $C_A$  in this problem yields

$$C_A = C_{A0} \frac{1 - X_A}{1 - 0.01 \frac{(-1)}{(-8)} X_A} = \frac{n_{A0}}{v_0} \frac{1 - X_A}{1 - 0.0013 X_A}.$$

Substituting into our material balance and rearranging a little bit yields

$$8\tau k = 640 = \frac{X_A(1 - 0.0013 X_A)}{1 - X_A},$$

which is one equation for one unknown. Take a moment to notice how little information we needed from the problem statement to get to this point: a residence time, a reaction rate, a little stoichiometry, and knowing the initial mole fraction of A. Everything else canceled out!

We can go about solving this equation algebraically or perhaps more conveniently on a graphing calculator, but whatever your choice you should end up with  $X_A = 0.9984$ —identical to the 0.9984 we calculated in Example 5.1—from which

$$y_C = \frac{n_C}{n_T} = \frac{n_{A0}(10/8)X_A}{n_{T0} - n_{A0}(1/8)X_A} = \frac{0.0879(10/8)0.9984}{8.7880 - 0.0879(1/8)0.9984} = 0.0125,$$

which, as expected, is very close to the same 1.2 mol% phthalic anhydride we found previously.

Comparing Example 5.1 to Example 5.3, we see that there are advantages and disadvantages to both the MATLAB approach and the hand calculation approach but that we end up with the same answer either way. When using MATLAB we had to calculate a few intermediate quantities like  $V$  that weren't necessary in the hand calculations, but with the hand calculations we had to be more careful with our algebra, especially when calculating  $C_A$  and  $y_C$ .

### 5.3 Multiple steady states can exist in CSTRs

Let's recall the general CSTR energy balance from Equation (4.2),

$$0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q \quad (4.2)$$

where we've dropped the shaft work term  $W_s$  because it's usually small compared to the other terms. Only one of the remaining terms,  $V \sum r_j \Delta H_j$ , has anything to do with an energy change due to chemical reaction, so let's isolate that one on the left side and everything else on the right side to get

$$-V \sum_{j=1}^q r_j \Delta H_j = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) - Q. \quad (5.15)$$

The reason for the negative sign on the left-hand side instead of any other way is that most reactions are exothermic, so the product  $-V r \Delta H$  will be generally be positive. For the same reason we also associate the product  $-V r \Delta H$  with energy *generated* by chemical reaction and

we give it the symbol  $G(T)$  to remind us that it's both “energy generated” and a function of temperature.

When energy is released in an exothermic reaction it's got to go somewhere and Equation (5.15) tells us that there are only two options: it can be removed from the reactor by the outward flow of process fluid, represented by the  $\sum n_{i0} C_{p,i} (T - T_0)$  term, or it can be removed by the utility fluid represented by the  $Q = UA(T_u - T)$  term.<sup>4</sup> We give these terms the symbol  $R(T)$  to remind us that it's both “energy removed” and a function of temperature.

Given these definitions of  $G(T)$  and  $R(T)$ , Equation (5.15) becomes

$$\underbrace{-V \sum_{j=1}^q r_j \Delta H_j}_{G(T)} = \underbrace{\sum_{i=A}^Z n_{i0} C_{p,i} (T - T_0) - Q}_{R(T)}, \quad (5.16)$$

or

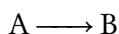
$$G(T) = R(T). \quad (5.17)$$

Equation (5.17) is an equivalent way of stating the CSTR energy balance in the absence of shaft work. To solve an energy balance problem for a CSTR we could either plot the right-hand side of Equation (4.2) and look for places where the curve crosses zero, or we could plot  $G(T)$  and  $R(T)$  on the same axis and look for places where the curves intersect. Since Equation (5.17) is simply a rearrangement of Equation (4.2), the temperatures which satisfy Equation (4.2) will be the same as the temperatures which satisfy Equation (5.17).

Let's look at an example of how to use Equation (5.17) to solve a problem.

#### Example 5.4: Multiple steady states in a jacketed CSTR

The aqueous, elementary reaction



takes place in a jacketed, 18 L CSTR. The reactor feed is  $5 \text{ mol}_A \cdot \text{s}^{-1}$  and  $5 \text{ mol} \cdot \text{s}^{-1}$  water at a total volumetric flow rate of  $0.8 \text{ L} \cdot \text{s}^{-1}$  and  $30^\circ\text{C}$ . Determine the following:

- (a) An expression for conversion as a function of temperature,  $X_A(T)$ .
- (b) Expressions for  $G(T)$  and  $R(T)$ .

<sup>4</sup>By “remove” we mean the energy that used to be stored in chemical bonds is transferred to the various vibrational, translational, and rotational energies of the process or utility fluids which are then transported out of the reactor.



(c) The reactor temperature and conversion of A.

Additional information has been provided below.

$$k = 0.04 \text{ s}^{-1} \text{ at } 50^\circ\text{C}$$

$$E_a = 80 \text{ kJ} \cdot \text{mol}^{-1}$$

$$\Delta H = -30 \text{ kJ} \cdot \text{mol}^{-1} \text{ (constant)}$$

$$C_{P,A} = C_{P,B} = C_{P,H_2O} = 75 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$$

$$UA = 1500 \text{ W} \cdot \text{K}^{-1}$$

$$T_u = 10^\circ\text{C} \text{ (constant)}$$

### Solution

Let's write quick stoichiometric table to organize information before we start on the solutions themselves. The system is aqueous so we can use the liquid-phase expression for  $C_i$  given in Equation (5.11). To aid our calculations we'll also include a column for each of the  $\Theta_i$  terms in our table:

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$C_i$
A	-1	5	1	$-r$	$C_{A0}(1 - X_A)$
B	1	0	0	$r$	$C_{A0}X_A$
H <sub>2</sub> O	0	5	1	0	$C_{A0}$

(a) The reaction is liquid-phase and there's only one reaction so this is another good opportunity to use the material balance expressed in terms of conversion, all the more so since the problem explicitly asked for  $X_A(T)$ . Starting from Equation (5.8) and substituting the rate law as  $r = -r_A = kC_A$  yields

$$V = \frac{n_{A0}X_A}{-r_A} = \frac{n_{A0}X_A}{k(T)C_{A0}(1 - X_A)},$$

where we used Equation (5.11) for A in the last step, and we explicitly indicated the temperature dependence of  $k$  as  $k(T)$  to remind ourselves that this is where temperature

comes into the material balance. Rearranging for  $X_A$  yields

$$X_A(T) = \frac{\tau k(T)}{1 + \tau k(T)},$$

where

$$k(T) = 0.04 \exp \left[ \frac{E_a}{R} \left( \frac{1}{323} - \frac{1}{T} \right) \right].$$

(b) For  $G(T)$  and  $R(T)$  we turn to Equation (5.16), which gives

$$G(T) = -Vr\Delta H = -Vk(T)C_{A0}(1 - X_A(T))\Delta H$$

and

$$\begin{aligned} R(T) &= \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) - Q \\ &= n_{T0} C_P (T - T_0) - UA(T_u - T) \end{aligned}$$

where  $X_A(T)$  and  $k(T)$  were defined in the previous part of this problem, the species subscript on  $C_P$  was dropped because all species have the same heat capacity in this problem (which also allows us to use  $n_{T0}$ ), and Equation (4.26) was used for  $Q$ .

Notice that  $G(T)$  is a fairly complicated function of  $T$  because of the exponential terms in  $k(T)$  and  $X_A(T)$ , but while  $R(T)$  is indeed a function of  $T$  it can be rearranged to the somewhat simpler form

$$R(T) = (n_{T0} C_P + UA)T - (n_{T0} C_P T_0 + UA T_u),$$

which shows that  $R(T)$  is a linear function of  $T$  with slope  $n_{T0} C_P + UA$  and intercept  $-(n_{T0} C_P T_0 + UA T_u)$ . We'll come back to this observation later.

(c) To solve for reactor temperature we use the condition  $G(T) = R(T)$  according to Equation (5.17), which suggest that we plot  $G(T)$  and  $R(T)$  and look for intersections as the temperatures which satisfy the energy balance. Since the reactor has lots of water in it we'll only look in the range of about 0 °C to 100 °C or 273 K to 373 K. Plotting over this range produces the figure shown below.

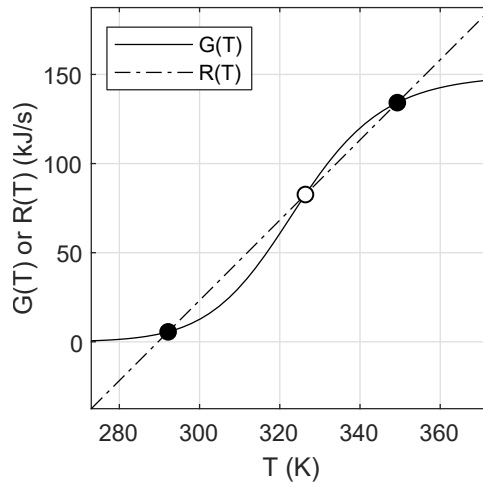


Figure 5.2: Intersections between  $G(T)$  and  $R(T)$  for Example 5.4 reveal solutions to the energy balance at about  $T = 349$  K, 326 K, and 292 K.

At the temperatures of the intersection points we can evaluate  $X_A$  as shown in the table below.

$T(K)$	$X_A(T)$ (%)
349	89.5
326	55.1
292	3.7

The reason for different marker styles for each of these points will be discussed in the next section.

What we see in Example 5.4 is one of the neatest things about CSTRs: it's possible to have more than one steady state! These are absolutely real states; they're not some abstract mathematical concept that doesn't actually correlate to something in the real world. If you had this reactor in front of you under these conditions it could be operating happily at *any* of these three steady states, but which one it's at depends critically on *where it's been before*. To learn more about this we need to know more about how the reactor behaves when it's operating at each of these steady states.

## 5.4 Steady states can be stable, unstable, or metastable

Let's imagine we're operating a CSTR at one of the steady states shown in Figure 5.2 at a temperature we'll denote as  $T_{SS}$ , the steady-state temperature. As you might expect—and as you'll

learn more about when you get to your process controls course—it's well-nigh impossible to operate with all parameters at a *precisely* constant value. Instead, values and the measurements we make of those values tend to fluctuate randomly around their steady state values (*c.f.* the equilibrium composition from Figure 2.1).

To look more closely at the different kinds of steady states that are possible we'll denote fluctuations in reaction temperature as  $+\delta T$  and  $-\delta T$ , where  $+\delta T$  represents a brief increase in reactor temperature and  $-\delta T$  represents brief decrease in reactor temperature. We don't care exactly how these fluctuations arise—a control valve could open or close a little bit too much, someone could drop an ice cube in the reactor, whatever—but what we do care about is how the energy balance says the reactor will respond to these fluctuations. We've already taken the time to define  $G(T)$  and  $R(T)$  as the rate of energy generated and removed from the reactor, and it's the ratio of these two quantities that tell us about how the reactor will respond to a temperature fluctuation.

As mentioned above, let's begin our analyses of reactor responses assuming we're at  $T_{SS}$  where  $G(T) = R(T)$ , and recognize that a fluctuation in  $T$ —as either  $+\delta T$  or  $-\delta T$ —can only result in one of two outcomes:

**The reactor temperature increases** because  $G(T) > R(T)$ , which states that the rate of energy generation exceeds the rate of energy removal, or

**The reactor temperature decreases** because  $R(T) > G(T)$ , which states that the rate of energy removal exceeds the rate of energy generation.

Before going on you should take a moment to try to draw a picture in your mind of what's happening in the physical system that supports these two outcomes because they'll be important going forward.

The response of the reactor to a fluctuation depends on both the direction of the fluctuation—as either  $+\delta T$  or  $-\delta T$ —and the relative magnitude of  $G(T)$  and  $R(T)$  as shown in Figure 5.3. In this figure, arrows indicate how the reactor temperature  $T$  will respond to a fluctuation: the arrows can either point *towards* the original steady state temperature  $T_{SS}$ , which we consider to be a stable response, or *away* from  $T_{SS}$ , which we refer to as an unstable response. In Figure 5.3(a), for example, a negative fluctuation of  $-\delta T$  results in a condition where  $G(T) > R(T)$ . In this state energy generation exceeds energy removal and the reactor will tend to heat up, thereby returning to the original  $T_{SS}$ . Similarly, a positive fluctuation of  $+\delta T$  results in a condition where  $R(T) > G(T)$  and in this state energy removal exceeds energy generation, meaning that the reactor will tend to cool off and again return to the original  $T_{SS}$ . Under these conditions we see that any fluctuation is always accompanied by a response

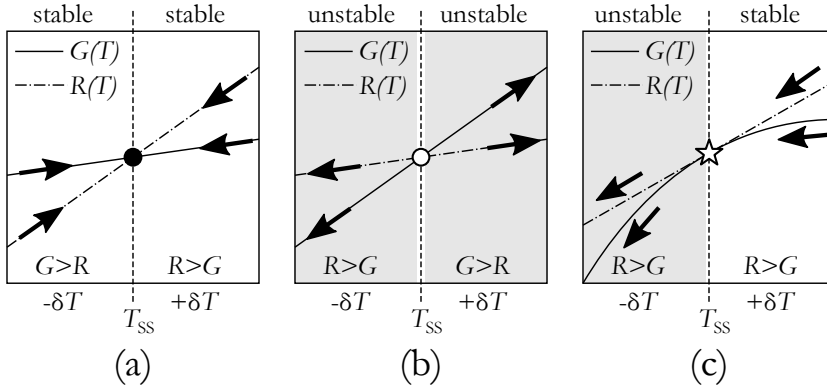


Figure 5.3: Behavior of reactor temperature  $T$  in response to fluctuations  $+\delta T$  and  $-\delta T$ . Arrows indicate how reactor temperature  $T$  responds to fluctuations: stable points (●) are defined as those for which *both* a positive fluctuation results in cooling because  $R(T) > G(T)$  and a negative fluctuation results in heating because  $G(T) > R(T)$ . Unstable (○) and metastable (☆) points have none or one of these properties, respectively.

which returns the reactor to its original  $T_{SS}$ , so we call this kind of steady state *stable* and indicate its stability with a filled circle marker (●).

Figure 5.3(b) indicates precisely the opposite case: a negative fluctuation of  $-\delta T$  produces a condition where  $R(T) > G(T)$ , causing the reactor to cool yet more; a positive fluctuation of  $+\delta T$  produces a condition where  $G(T) > R(T)$ , causing the reactor to heat yet more. The combination of these two unstable responses indicates that even the smallest fluctuation will result in the reactor temperature moving away from  $T_{SS}$ , so we call this kind of steady state *unstable* and mark it with an open circle marker (○). Trying to operate a reactor at an unstable steady state is like trying to balance a sharpened pencil on its tip: it's possible, but even the smallest perturbation—a gnat lands on one side, somebody breathes too hard, the table shakes ever so slightly—will cause it to fall over.

The third and final type of steady state is shown in Figure 5.3(c). In response to a negative fluctuation the reactor continues to cool because  $R(T) > G(T)$ , but in response to a positive fluctuation the reactor also cools, again because  $R(T) > G(T)$ . At this steady state the reactor is stable in response to positive fluctuations but unstable in response to negative fluctuations, and we call this “somewhat stable” state *metastable*<sup>5</sup> and mark it with an open star (☆). Trying to operate a reactor at a metastable point is like trying to balance a sharpened pencil on its tip next to a vertical wall: you can push the pencil as hard as you want *towards* the wall and the pencil will remain upright, but even the smallest perturbation *away* from the wall and it will fall over.

<sup>5</sup>Sometimes this is also called the *bistable* steady state.

Applying this logic to Figure 5.2 from Example 5.4 we see that the steady states at 349 K and 292 K are stable steady states (●), whereas the steady state at 326 K is an unstable steady state (○). The desired operating point is most likely the steady state at 349 K because the conversion is highest at this point, but how does one go about getting the reactor into this state? The answer requires us to assume which of the operational parameters we can control when the reactor is running, such as flow rates and temperatures. Most importantly, we must be *exceedingly* careful about how we manipulate these variables because they can cause unsafe conditions to develop with little to no warning!

## 5.5 Multiple steady states lead to ignition and extinction behavior

On most reactors the things we can directly manipulate tend to be flow rates and temperatures because both can be controlled reasonably well by a combination of a sensor and valve: the sensor senses either flow rate or temperature and in response a controller opens or closes a valve—perhaps one to directly control the flow itself, or one to control the flow of a utility stream in a heat exchanger. Other variables such as heat exchanger area or reactor volume  $V$  may fluctuate during operation—such as if the reactor is a fluidized bed reactor—but generally are not controlled directly.

Suppose the inlet temperature  $T_0$  is one of the variables we can manipulate in Example 5.4. Within the solution to Example 5.4 it was noted that  $R(T)$  is a linear function of  $T$  with slope  $n_{T_0}C_p + UA$  and intercept  $-(n_{T_0}C_p T_0 + UAT_u)$ , and if we look closely at the expression for  $G(T)$  we see that  $T_0$  doesn't appear. Therefore variations in  $T_0$  affect only  $R(T)$  and only by changing its intercept, which corresponds to shifting  $R(T)$  left and right (without changing its slope) on Figure 5.2. The appearance and classification of steady states at different values of  $T_0$  are shown in Figure 5.4, and evidently there are quite a few to investigate!

Let's begin our analysis of the CSTR in Example 5.4 in the top row of Figure 5.4 where  $T_0 = 350$  K and just one stable steady state exists around 370.4 K. Small fluctuations in reaction temperature will not shift the reactor off this stable steady state. However, if we choose a slightly cooler inlet temperature  $T_0$  then the straight line of  $R(T)$  shifts a little bit to the left, and if we choose  $T_0$  to be about 323 K then we'll find a second steady state as shown in the second row of Figure 5.4. At this point  $R(T)$  is tangent to  $G(T)$  and the result is a metastable state at about 308.4 K but even if we were able to somehow operate the reactor at 308.4 K the slightest perturbation to this metastable state would result in a temperature increase until the reactor temperature reaches the upper steady state at about 359.7 K.

We can continue choosing colder and colder  $T_0$  values to produce the remaining rows of Figure 5.4: with  $T_0 = 300$  K we have three steady states—two stable and one unstable; with

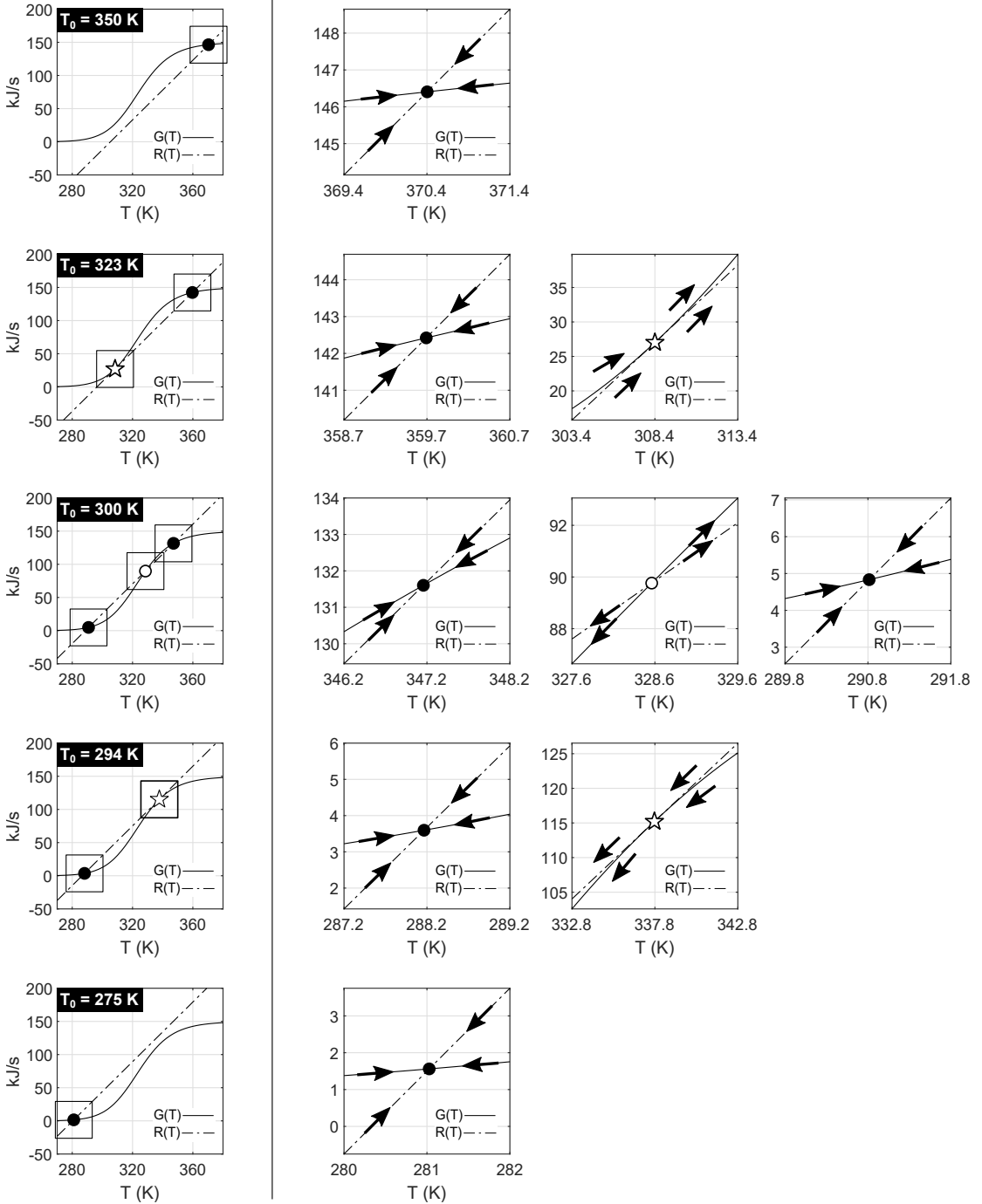


Figure 5.4: Appearance and classification of steady states as inlet temperature  $T_0$  is varied in Example 5.4. The plots in the *left* column show  $G(T)$  and  $R(T)$  over a large temperature range; the plots in the *right* column show details of the various intersection points. Steady states are categorized as either stable (●), unstable (○), or metastable (☆) as described in Figure 5.3.

$T_0 = 294\text{ K}$  we have two steady states—one stable and one metastable; and with  $T_0 = 275\text{ K}$  we have just one steady state.

The language of the preceding paragraphs was chosen carefully because we were simply calculating steady states for various  $T_0$ ; we were *not* implying a particular manner of operating the reactor. Let's now think about what would happen if we attempted to operate the reactor in the same pattern that we just described in our calculations: a most remarkable behavior will result! The reactor will *not* smoothly vary from upper steady state temperatures, through metastable and unstable temperatures, and finally settle at a lower steady state temperature. Instead, the reactor will gradually cool along the upper steady state temperatures until a metastable point is reached, at which time the reactor temperature will suddenly and abruptly drop to the lower steady state.

The reason for the abrupt decrease in temperature is due to the stability of the steady states and is shown in the left figure of Figure 5.5. When the inlet temperature is initially  $T_0 = 350\text{ K}$  then the reactor is operating at a steady state temperature of about  $370.4\text{ K}$ . As  $T_0$  is cooled we continue to operate at steady states only slightly cooler than  $370.4\text{ K}$ . Eventually a metastable state at  $308.4\text{ K}$  is possible when  $T_0 = 323\text{ K}$  but this state is inaccessible to the reactor because it would be operating a stable steady state of  $359.7\text{ K}$ , and any perturbations will not result in

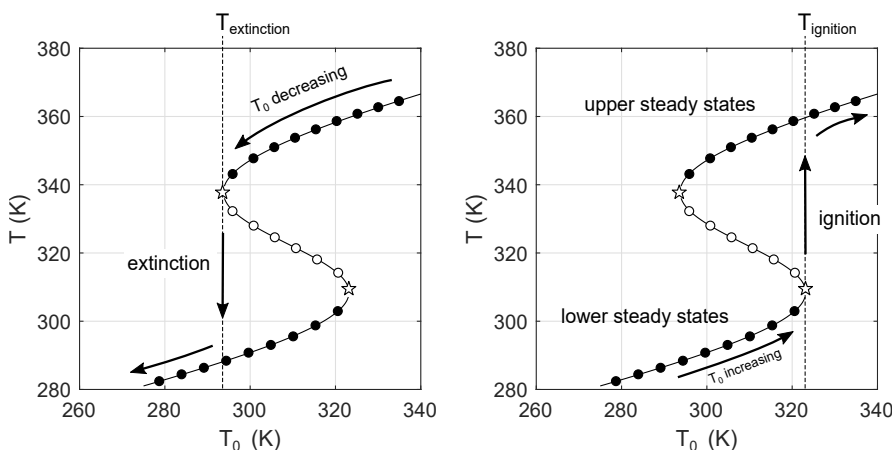


Figure 5.5: Ignition and extinction behavior in a CSTR. (*left*) As  $T_0$  increases from an initially low temperature so too does  $T$  increase along the lower steady state line until it reaches the metastable point ( $\star$ ) at about  $T_0 = 323\text{ K}$ , whereupon  $T$  abruptly increases from about  $T = 310\text{ K}$  to  $T = 360\text{ K}$  as it shifts to the upper steady state line. This abrupt increase is called *ignition* and the temperature  $T_0$  at which it occurs is called the *ignition temperature*. (*right*) A similar process occurs when  $T_0$  begins from an initially high temperature but here the abrupt decrease from about  $T = 340\text{ K}$  to  $T = 290\text{ K}$  when  $T_0 = 294\text{ K}$  is called *extinction*, and the temperature at which it occurs is called the *extinction temperature*.



a shift away from this stable state.

As  $T_0$  continues to fall additional unstable and stable states appear but these states are inaccessible to the reactor for the same reason that the aforementioned metastable state was unavailable: the reactor will always be operating at a stable state and any perturbations will not result in a shift away from these stable states.

The sudden shift in reactor temperature  $T$  occurs when  $T_0$  is eventually lowered to 294 K because at this temperature the reactor would be operating in a metastable state: even the slightest fluctuation will result in the reactor temperature suddenly and abruptly falling to the lower steady state temperature at about 288.2 K. This abrupt decrease is called *extinction* and the temperature  $T_0$  at which it occurs is called the *extinction temperature*. Since the rate constant  $k$  varies exponentially with temperature this sudden drop from 340 K to 290 K results in a dramatic decrease in reactor conversion: the reaction is “extinguished.”

A similar behavior is shown in the right panel of Figure 5.5 for a case in which the reactor is initially operated at low  $T_0$ . As the inlet temperature is slowly raised the reactor operates at the corresponding lower steady state until the metastable state at  $T_0 = 323$  K is encountered, at which point even the smallest fluctuation pushes the reactor suddenly and abruptly from  $T = 310$  K to  $T = 360$  K. This dramatic increase is called *ignition* and the temperature at which it occurs is called the *ignition temperature*. Ignition events should not be underestimated: the dramatic increase in reactor temperature can be exceptionally dangerous!

## 5.6 Multiple steady states occur in the laboratory and in nature

When multiple steady states, ignition, and extinction are first encountered there can be a tendency to think that they’re the result of a carefully chosen set of parameters that may or may not be physically reasonable. That’s a reasonable caution to have because it’s always a good idea to keep in mind how physically reasonable a calculation or analysis truly is, but in the case of multiple steady states they’re quite real and they can be studied in a number of different systems. A few examples have been collected here to give you an idea of multiple steady states appear in the laboratory and in nature.

Some of the earliest work to confirm the existence of multiple steady states in CSTRs was done in the lab by Vejtasa and Schmitz<sup>6</sup> working with a reaction between sodium thiosulfate and hydrogen peroxide in water. Kalthoff and Vortmeyer<sup>7</sup> demonstrated similar phenomena in a long, thin tubular reactor in which the catalytic oxidation of ethane was performed. Marb

<sup>6</sup>S.A. Vejtasa; R.A. Schmitz. An experimental study of steady state multiplicity and stability in an adiabatic stirred reactor. AIChE J. 1970, 16, 3, 410-419.

<sup>7</sup>O. Kalthoff; D. Vortmeyer. Ignition/extinction phenomena in a wall cooled fixed bed reactor. Chem. Eng. Sci. 1980, 35, 1637-1643

and Vortmeyer<sup>8</sup> again demonstrated multiple steady states 1987 with the same reaction but with a reactor closer to the fluidized bed reactor described in Section 5.1.

More recently, the existence of multiple steady states in biological systems has been investigated, and within this field the phenomena are referred to as “bistability” and “switch-like behavior.” Here, the multiple steady states arise from the interactions between multiple chemical reactions. Craciun, Tang, and Feinberg<sup>9</sup> investigated bistability in enzyme-catalyzed reaction networks, and in 2015 Ashkenasy and coworkers<sup>10</sup> demonstrated “switch” behavior in a synthetic, non-enzymatic chemical reaction network intended to mimic the behavior observed in cells.

## 5.7 Cooling ability can be dangerously overestimated

The ignition/extinction behavior which can occur in systems with multiple steady states benefits from an appropriate pairing of eyebrow-raising terminology (ignition) with potential for havoc (a rapid increase in temperature). A more insidious pairing is also identifiable in the forms of  $G(T)$  and  $R(T)$ , neglect of which can lead to devastating and deadly runaway reactions. Let’s take a moment to review two chemical accidents and then appreciate their link to our current discussion. These two accidents were analyzed and reconstructed by the US Chemical Safety Board (CSB), an independent federal agency charged with investigating industrial chemical accidents.<sup>11</sup>

### MFG incident: 154 injured, hundreds evacuated

MFG Chemical, Inc. (MFG) in Dalton, Georgia was contracted to produce triallyl cyanurate (TAC) by reacting allyl alcohol and cyanuric chloride in the presence of a catalyst. A by-product of the reaction was hydrogen chloride which could be neutralized by addition of caustic soda. Process engineers knew the neutralization produced significant heat but had not considered the potential for heat release from thermal decomposition reactions which began to occur at about 100 °F (38 °C)

To develop their procedures MFG made several test batches in a 30 gallon (110 L) reactor but only one of these batches added the entire quantity of reactants all at once, as production-

<sup>8</sup>C.M. Marb; D. Vortmeyer. Multiple steady states of a crossflow moving bed reactor: theory and experiment. Chem. Eng. Sci. 1988, 43, 4, 811-819.

<sup>9</sup>G. Craciun; Y. Tang; M. Feinberg. Understanding bistability in complex enzyme-driven reaction networks. PNAS. 2006, 103, 23, 8697-8702.

<sup>10</sup>R. Mukherjee; R. Cohen-Luria; N. Wagner; G. Ashkenasy. A bistable switch in dynamic thiodepsipeptide folding and template-directed ligation. Ang. Chem. 2015, 54, 12452-12456.

<sup>11</sup>United States Chemical Safety Board, csb.gov. Information provided here was collected from Reports 2004-09-I-GA (MFG incident) and 2006-04-I-NC (Synthron).

scale manufacturing would do. With such a small reactor and adequate cooling the reactor temperature never exceeded 50 °F (10 °C). This test batch also used impure (recycled) allyl chloride instead of the pure allyl chloride used at production scale.

Around 7:30 pm on April 12, workers loaded a production-scale quantity of allyl chloride into a 4000 gallon (15 000 L) reactor equipped with jacketed heat exchanger connected to a temporary chiller to provide heat removal. The temperature gradually rose from 32 °F (0 °C) to 72 °F (22 °C) as the allyl chloride was added. Ten minutes later the temperature hit 103 °F (40 °C); the last observed reactor temperature was 124 °F (51 °C).

Dense, white vapor blew out a gasket around an access port and began spraying the surrounding area. About 30 s later the overpressure rupture disc blew open, venting additional white vapor. The toxic vapors would continue to escape for nearly 24 hours until the reactor was finally resealed the next day around 1:00 pm. Hundreds of families from the surrounding area were evacuated and more than 150 individuals received decontamination and treatment for chemical exposure.

### **Synthron incident: 1 dead, 14 injured**

In late January, 2006, Synthron, LLC in Morganton, North Carolina began production for an order of a little over 6,000 lb (2700 kg) of an acrylic polymer, about 12% more than would normally be made in one of their single, 1500 gallon (5700 L) reactors. As part of normal production, heat from the polymerization reaction boiled the reactor mixture. The resulting hot gas was sent to a heat exchanger where it was condensed to a liquid and returned to the reactor, a process which also provided cooling to the reactor. The reactor was also equipped with a jacketed heat exchanger for heating and emergency cooling.

The relatively modest increase in total material as well as other changes in the ratio of solvents and monomers had an outsized effect on heat production during the reaction, more than doubling the maximum heat output. As the solvent boiling rate increased the fixed-sized heat exchanger was overwhelmed, unable to completely condense the resulting vapors or return a continuous stream of subcooled liquid back into the reactor for temperature control.

After addition of the polymer initiator to the reaction vessel the pressure in the reactor ruptured an access port, accompanied by a loud hissing sound. After most workers within the building had evacuated a senior operator donned a respirator, entered the building, and started an emergency cooling water flow to the reactor jacket. A short time after the operator exited, the building exploded, injuring over a dozen employees gathered around the building. A nearby maintenance supervisor received burns to his entire body and was evacuated by helicopter to a burn center. He succumbed to his injuries five days later.

### Appreciate the connection

CSB noted several root-causes contributing to both disasters including a lack of hazard recognition, poorly documented process safety information, inadequate corporate oversight of process safety, and failure to implement risk management programs. The connection to our present material is made by the following observation: in both incidents inadequate cooling led to a loss of control. Restated more succinctly: in both incidents  $G(T) > R(T)$ , although the forms for  $G(T)$  and  $R(T)$  would differ slightly for the batch reactors involved in these incidents compared to the CSTRs we've been studying here.

Minor differences aside, let's consider again the energy balance we used to define  $G(T)$  and  $R(T)$ ,

$$\underbrace{-V \sum_{j=1}^q r_j \Delta H_j}_{G(T)} = \underbrace{\sum_{i=A}^Z n_{i0} C_{p,i} (T - T_0) - UA(T_u - T)}_{R(T)}. \quad (5.16)$$

Notice that  $G(T)$  depends on the *volume* of the reactor—or more specifically, the volume of material within the reactor,  $V$ . The cooling and regulatory capacity of  $R(T)$ , however, depends on *surface area* as  $A$ . When MFG scaled up from 30 gallon test batches to a 4000 gallon production batch the ratio of area available for heat transfer,  $A$ , to reactor volume,  $V$ , decreased drastically and resulted in a loss of control. When Synthron increased the amount of material within their reactor the surface area-to-volume ratio again decreased, and again resulted in a loss of control. This simple geometric relationship is so crucial but so easily overlooked that it's worth taking a moment to state it plainly and simply:

Heat produced by a reaction depends on **volume**. Cooling capacity depends on **area**. The ratio of surface area to volume decreases when size is increased and this can lead to a loss of control.

The dependence of  $G(T)$  on  $V$  and  $R(T)$  on  $A$  wasn't a central feature of our previous ignition/extinction discussion but now we see its importance. Indeed the same scaling features are at work in many heating and cooling systems. We have a responsibility as professional engineers to ourselves, our coworkers, and our community to understand and appreciate these factors so that these accidents, or ones like it, never occur again.

## Chapter 6

# MATLAB II: Ordinary Differential Equations

“...it is an impressive illustration of the power of numerical analysis that such an easy calculation should give such precise results.”

Richard Feynman, *The Feynman Lectures*, Vol. 1, Ch. 9 (1963)

In previous chapters we saw how analytical (“closed form”) solutions to the material and energy balances of a CSTR were difficult if not impossible to find but that we could instead approximate a solution numerically without too much trouble using the `fsolve` function. In the next few chapters we’ll find ourselves in a similar situation with ordinary differential equations (ODEs) instead of algebraic equations, and as with the algebraic equations we’ll be able to solve a few of them by hand but we’ll have to solve the tougher ones using a numerical approximation scheme implemented by MATLAB’s `ode45` function.

This chapter is dedicated to introducing the `ode45` function so if you’re already familiar with how to use this function then you can safely skip this chapter.

### 6.1 A difference equation can approximate a differential equation

Think way back to your first calculus class and you’ll remember that the formal definition of the derivative of a function  $n(V)$  at  $V$  is

$$n'(V) = \left. \frac{dn}{dV} \right|_V = \lim_{\Delta V \rightarrow 0} \frac{n(V + \Delta V) - n(V)}{\Delta V},$$

where  $n$  and  $V$  were chosen instead of the usual  $f$  and  $x$  because they’re the variables that show up more often in subsequent chapters. If instead of allowing  $V$  to approach 0 we instead let  $V$

be small but finite then we can *approximate* the exact derivative with the difference equation

$$n'(V) = \left. \frac{dn}{dV} \right|_V \approx \frac{n(V + \Delta V) - n(V)}{\Delta V},$$

which can be rearranged to

$$n(V + \Delta V) = n(V) + \Delta V \left. \frac{dn}{dV} \right|_V \quad (6.1)$$

which is called a *difference* equation, or a *finite difference* equation because  $\Delta V$  is finite. In anticipation of using Equation (6.1) to calculate something in MATLAB, an equivalent way of writing it is

$$n_{i+1} = n_i + \Delta V \left. \frac{dn}{dV} \right|_i, \quad (6.2)$$

where we've defined  $n(V)$  to be the  $i$ -th value of  $n$  as  $n_i$ , and  $n(V + \Delta V)$  to be the  $i$ -th+1 value of  $n$  as  $n_{i+1}$ . Assuming that we have an expression for  $dn/dV$  and knowledge of  $n$  when  $V = 0$ , we can use Equation (6.2) to approximate  $n(V)$  by the following method:

1. Start with the initial condition  $n_{i=0}$  at  $V = 0$ .
2. Use Equation (6.2) with a small  $\Delta V$  to calculate a new value of  $n$  at  $n_{i+1} = n_1$ .
3. Increment  $V$  by  $\Delta V$ .
4. Repeat Steps 2-3 to the desired final value of  $V$ .

If we follow these four steps then we've approximated the solution to a differential equation by using a finite difference approximation.

#### Example 6.1: The finite difference approximation

Use Equation (6.2) with  $\Delta V = 0.1$  to plot the solution of

$$\frac{dn}{dV} = -kn, \quad n(0) = 1$$

from  $V = 0$  to  $V = 2$  with  $k = 2$ . Compare the approximate solution to the exact solution.

#### Solution

Let's do a few of the calculations by hand before putting them in MATLAB to really go at it. Initially we've got

Step	$V$	$n$
0	0	1

which is simply the initial condition (aptly named, as you can see). Now we use Equation (6.2) to start calculating new values of  $n$ . The next value of  $n$  will be evaluated at  $V + \Delta V = 0 + 0.1 = 0.1$  to get

$$n(0.1) = n(0) + 0.1(-k \times 1) = 0.8$$

so that our table of values is now

Step	$V$	$n$
0	0.0	1.000
1	0.1	0.800

The next value of  $n$  will be evaluated at  $V + \Delta V = 0.1 + 0.1 = 0.2$  to get

$$n(0.2) = n(0.1) + 0.1(-k \times 0.8) = 0.640$$

so that our table of values is now

Step	$V$	$n$
0	0.0	1.000
1	0.1	0.800
2	0.2	0.640

Repeating this pattern a few more times gives

Step	$V$	$n$
0	0.0	1.000
1	0.1	0.800
2	0.2	0.640
3	0.3	0.512
4	0.4	0.410
$\vdots$		
18	1.8	0.018
19	2.0	0.012

To build the MATLAB script to repeat this calculation, notice that we have to evaluate the expression for  $dn/dV$  lots and lots of times at different values of  $n$ . In anticipation of  $dn/dV$  being much more complicated than what we've got here, let's use a local function like

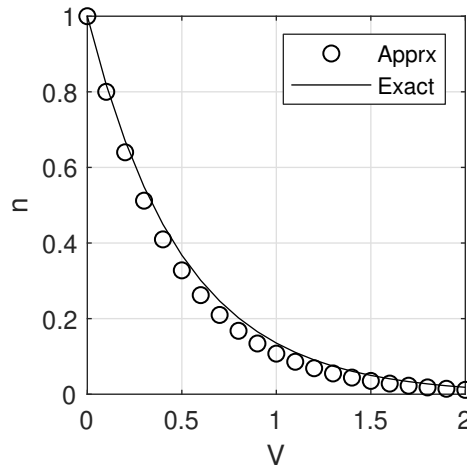
```
15 function dndV = fun(n)
16 k = 2;
17 dndV = -k*n;
18 end
```

even though it's not strictly necessary for this problem. We can then use a `for` loop to call this local function lots and lots of times in order to produce the table given above:

```
1 % finiteDiff.m
2 Vmax = 2;
3 dV = 0.1;
4 V = 0:dV:Vmax;
5 n = zeros(size(V));
6 n(1) = 1; % initial condition
7 for i = 1:(length(V)-1)
8   n(i+1) = n(i) + dV*dndV(n(i));
9 end
10 plot(V, n)
11 hold on
12 plot(V, exp(-2*V))
13 hold off
14
15 function dndV = fun(n)
16 k = 2;
17 dndV = -k*n;
18 end
```



On Line 12 we calculate the exact solution as  $n = \exp(-kV)$  to produce the plot shown below, but in most cases that won't be an option because we won't be able to produce an analytic (exact) solution (if we could then we wouldn't need the finite difference approximation in the first place).



The results in Example 6.1 are actually pretty good considering how simple an approximation Equation (6.2) is and how easy it is to implement! The important idea here is that no matter how complicated the differential, the difference equation of Equation (6.2) allows us to calculate  $n(V)$  fairly painlessly.

## 6.2 MATLAB's ode45 function can solve differential equations

We can improve our finite difference approximation in two central ways:

1. Use a better approximation than Equation (6.2).
2. Use a better step size.

The word “better” was chosen instead of “smaller” because to improve on the finite difference approximation will require extra computational steps so it's advantageous to choose a small step size where the derivative is large but a large step size where the derivative is small. An approach with variable step sizes is said to have *adaptive step size*.

Fortunately for us MATLAB has bundled all of these improvements into one easy-to-use function called `ode45`.<sup>1</sup> There are alternative solvers like `ode15s` or `ode23s` to handle special

<sup>1</sup>The “45” in `ode45` should be interpreted as “four-five” instead of “forty five” because the algorithm uses methods for which the total accumulated error is on the order of  $(\Delta V)^4$  and  $(\Delta V)^5$ . You can learn more about the algorithms by checking the MATLAB documentation or searching the web for “runge kutta 4 5.”

types of problems but `ode45` is a good, general-purpose solver that will be able to handle almost everything in this book. You can learn more about choosing appropriate ODE solvers by typing `doc choose_ode_solver` at the command prompt.

The most common syntax for `ode45` that we'll use is

```
[Vsol, nsol] = ode45(@fun, Vspan, n0)
```

where `@fun` is a handle to the local function `fun` which calculates  $dn/dV$  at any value of  $n$  and  $V$ , `Vspan` is a two element vector specifying the minimum and maximum of the solution range, `n0` is the initial condition of  $n(V = 0)$ , and `Vsol` and `nsol` are the solution vectors for  $V$  and  $n$ . The local function `fun` could have equally been called `dndV` but we'll see that it's more readable if we use `dndV` as the output of `fun`, so we establish this convention here and stick with it for the rest of the text.

### Example 6.2: Basic usage of `ode45`

Repeat Example 6.1 using `ode45`.

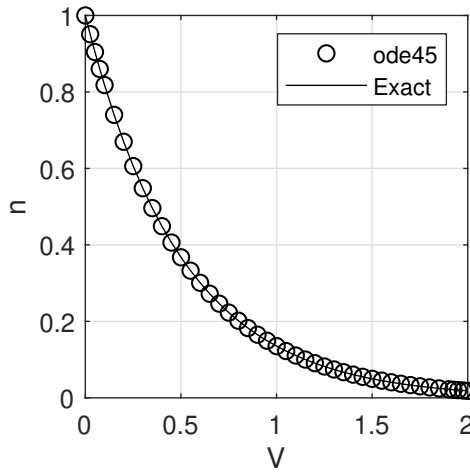
### Solution

The usage of `ode45` closely follows the way Example 6.1 approached the problem: we create a script with initial conditions at the top, we solve the problem with `ode45` instead of the `for` loop in Example 6.1, we do any plotting or analysis using the solution vectors `Vsol` and `nsol`, and at the bottom of the script is the local function `fun`. The local function `dndV` has to take two inputs,  $V$  and  $n$ , even if they're not used because this is part of the syntax that `ode45` expects.

```
1 % ode45basic.m
2 Vmax = 2;
3 n0 = 1;
4 Vspan = [0 Vmax];
5 [Vsol, nsol] = ode45(@fun, Vspan, n0);
6 plot(Vsol, nsol, 'ko')
7 hold on
8 plot(linspace(0,Vmax), exp(-2*linspace(0,Vmax)), 'k')
9 hold off
10
11 function dndV = fun(V, n)
12 k = 2;
13 dndV = -k*n;
14 end
```

Running this script produces the plot shown below. For all practical purposes

the numerical approximation of ode45 perfectly matches the exact (analytic) solution, which is common for the problems we'll encounter.



After you run the `ode45basic.m` script in Example 6.2 you might notice that the solution vectors `Vsol` and `nsol` are both 45-by-1, yet we never specified how many points to return in either of these vectors; MATLAB simply chose points within the solution that produced what it thought was a nice-looking plot. The points at which MATLAB actually solved the ODE are called *solution points* and they vary from problem to problem because, as mentioned previously, the `ode45` function uses *adaptive step sizes*. This means it picks points at a high density where it needs more precision and a low density where it does not.

### 6.3 Output points from ode45 can be made different from solution points

Suppose we want to know the value of  $n$  at  $V = 1$  in Example 6.2: by inspecting `Vsol` we can find `Vsol(22)` and `Vsol(23)` to be 0.9505 and 1.0005, but there's no element of `Vsol` that's *exactly* 1.0000. Or, as another example, suppose we wish to know `nsol` at 100 equally spaced points between 0 and `Vmax` because we want to use the solution in some kind of analysis that requires equal spacing between solution points.

In either case we're in a position where `Vsol` and `nsol` aren't the way we want them because the default syntax of `ode45` doesn't specify where `ode45` provides its output points. However, we can modify the inputs and outputs to `ode45` to give us the *appearance* of control, at least over what's provided as output data from `ode45` (always remember that we cannot change the underlying solution points). Here's how it works: we ask `ode45` to solve our ODE, it does

so by choosing whatever solution points it thinks is best, and—if we’ve asked it to do so—it will *interpolate* between the solution points to high accuracy to provide data at the desired output points. We can accomplish this feat in one of two equivalent ways.

### 6.3.1 Spans containing more than two values provide outputs at those values

The span vector `Vspan` can include more than just the minimum and maximum values of  $V$ , and if it does then `ode45` will do the following:

1. It will solve the problem as though the span consisted of only two elements, the minimum and maximum of the provided vector.
2. The solution points will be *interpolated* to provide outputs which correspond to the points in the original span vector.

For example, were we to change the span vector in Example 6.2 to

```
4 Vspan = 0:0.5:Vmax;
```

With this input `ode45` will provide `Vsol` and `nsol` as 5-by-1 vectors where `Vsol` is the same as `Vspan`. The plot now becomes the one shown in Figure 6.1, where the returned values are evaluated only  $n(0)$ ,  $n(0.5)$ ,  $n(1.0)$ ,  $n(1.5)$ , and  $n(2.0)$ . Despite appearing to consist of just five values, these five values are no less accurate because the underlying solution points are the same regardless of how many elements are provided in `Vspan`.

### 6.3.2 A solution object can be returned instead of vectors

Let’s return to the original code of Example 6.2 with the standard syntax for `Vspan` as `Vspan = [0 Vmax]`. If we ask for only a single variable as the output from `ode45` as

```
5 sobj = ode45(@fun, Vspan, n0);
```

then `sobj` will be a *solution object* which contains information about the solution to the ODE.<sup>2</sup> We can find the solution to our ODE at any value of  $V$  by providing `sobj` and the value to the `deval` function, such as

```
>> deval(sobj, 0.5)

ans =

    0.3679
```

<sup>2</sup>The information includes `sobj.x` and `sobj.y`, which are the mysterious solution points that have been previously hidden.

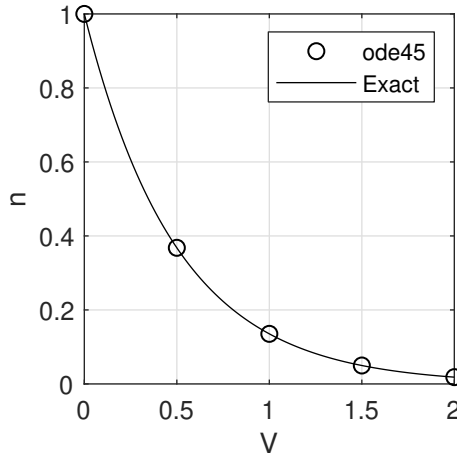


Figure 6.1: Plot from `ode45basic.m` in Example 6.2 with `Vspan = 0:0.5:Vmax`. The returned values are evaluated only at the points in `Vspan` but are no less accurate compared to the exact solution because the underlying solution points are the same regardless of `Vspan`.

which provides the value  $n(0.5) = 0.3679$ . We can also go the other way around and find the value of  $V$  which corresponds to some value of  $n$  by using the `fsolve` function. For example, suppose we wish to find the value of  $V$  at which  $n(V) = 0.6$ : we can use `fsolve` with `deval` as

```
>> fsolve(@(V)deval(sObj,V)-0.6, 0.1)

ans =
    0.2554
```

to find that  $n(0.2554) = 0.6$ .

We can also pass vectors to `deval` to evaluate `sObj` at several points, such as

```
>> nPoints = deval(sObj, linspace(0, Vmax));
>> plot(linspace(0, Vmax), nPoints)
```

which produces `nPoints` as a 1-by-100 vector and plots it.

## 6.4 The ode45 function can accept additional input parameters

As with `fsolve`, the `ode45` function can accept additional input arguments which can be helpful if you'd like to find the solution to an ODE at several values of a parameter. The general syntax is

```
[Vsol, nsol] = ode45(@(V,n)fun(V,n,param), Vspan, n0);
```

where `param` is any parameter that you'd like to pass to the `dndV` function. The local function also has to accept this parameter, so the function definition must be

```
function dndV = fun(V,n,param)
```

This approach is particularly useful when coupled with a `for` loop to quickly solve the ODE at several values of `param`.

### Example 6.3: Passing an extra input parameter to `ode45`

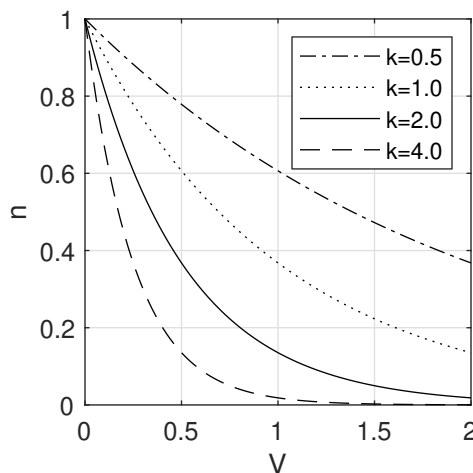
Solve the ODE from Example 6.2 for  $k = 0.5$ ,  $k = 1.0$ ,  $k = 2.0$ , and  $k = 4.0$ . Compare the solutions on a single plot.

#### Solution

The script below creates a vector  $k$  to store the desired values of  $k$ , then uses a `for` loop to iterate through each value of  $k$ . Within each iteration the current value of  $k$  is passed to `dndV` as a parameter, `ode45` provides a solution as a solution object, and `deval` is used to evaluate the solution at 100 equally spaced points between 0 and  $V_{\max}$ . The resulting vector is stored as one of the columns in a storage matrix `nsol`.

```
1 % ode45_for.m
2 k = [0.5 1.0 2.0 4.0];
3 Vmax = 2;
4 n0 = 1;
5 Vspan = linspace(0, Vmax)';
6 nsol = zeros(length(Vspan), length(k)); % 100-by-4
7 for i = 1:length(k)
8     [~, nsol(:,i)] = ode45(@(V,n)fun(V,n,k(i)), Vspan, n0);
9 end
10 plot(Vspan, nsol)
11 legend('k=0.5', 'k=1.0', 'k=2.0', 'k=4.0')
12
13 function dndV = fun(V, n, k)
14 dndV = -k*n;
15 end
```

Running this script produces the plot shown below. Note that  $k$  is no longer defined within the local function `dndV` as it was in Example 6.2 because it's now provided as an input to the function.



The other tricky part of this problem is that within the `for` loop we have to come up with a way to store the solution vector. The way presented here ensures that all solution vectors have exactly 100 points, each point corresponding to one of the points in the `Vspan` vector. Can you come up with a way to mimic this behavior using solution objects and the `deval` function?

## 6.5 The local ODE function can return multiple outputs

The usual syntax for the local ODE function file is

```
1 function dxdy = fun(x, y)
```

which designates only one output as `dxdy`. So long as this derivative term is *always* returned as the first output, you can also ask the local ODE function to return multiple outputs by using a syntax like

```
1 function [dxdy, myExtraOutput] = fun(x,y)
```

where `myExtraOutput` is whatever additional output you'd like to return. This strategy can be useful to save some typing if there's a parameter calculated as part of the solution process that you're interested in plotting after `ode45` has completed its analysis.

For example, consider the following script:

```
1 [V, nA] = ode45(@fun, [0 5], 2);
2
3 function dnAdV = fun(V, nA)
4 k = 0.9; % 1/ min
```

```

5 Kc = 1.14;
6 v0 = 10; % m3/ min
7 nA0 = 2; % mol/ min
8
9 v = v0; % m3/ min
10 nB = nA0 - nA; % mol/ min
11 Ca = nA/v; % mol/m3
12 Cb = nB/v; % mol/m3
13 rA = -k*(Ca -Cb/Kc); % mol/m3 - min
14 dnAdV = rA; % mol/m3 - min
15 end

```

Here, ode45 is solving the differential equation

$$\frac{dC_A}{dV} = r_A$$

and along the way it needs to calculate  $C_B$  as Cb on Line 12. If we wanted to plot  $C_B(V)$  then we've got two options: repeat some of the code we've got inside fun but do it after we call ode45, or ask fun to also return Cb as an extra output and then evaluate fun after calling ode45. You can compare these two approaches in Figure 6.2. For this particular example the code savings is only a few lines but for more complex examples the code savings can easily be 10-20 lines, which means 10-20 few lines where coding mistakes can be made!

## 6.6 The ode45 function can solve systems of ODEs

Sometimes the solution of one ODE depends on the solution of another ODE, and vice versa. Such ODEs are called *coupled* and since there's more than one ODE this is called a *system* of ODEs. For example, the differential equations<sup>3</sup>

$$\frac{dx}{dt} = z + 4 \quad (6.3)$$

$$\frac{dz}{dt} = -3x \quad (6.4)$$

constitute a system of ODEs because there are two ODE functions, and that system is *coupled* because the  $dx/dt$  is a function of  $z$  and  $dz/dt$  is a function of  $x$ . To solve this system using

<sup>3</sup>The author feels no shame in using the same sample system that you can find on the MATLAB Academy because the author helped write the content in that module.



```

1 [V,nA]=ode45(@fun,[0,5],2);
2 v0 = 10;
3 nA0 = 2;
4 v = v0;
5 nB = nA0 - nA;
6 Cb = nB/v;
7 plot(V, Cb)
8
9 function dnAdV=fun(V,nA)
10 k = 0.9;
11 Kc = 1.14;
12 v0 = 10;
13 nA0 = 200;
14 v = v0;
15 nB = nA0 - nA;
16 Ca = nA/v;
17 Cb = nB/v;
18 rA = -k*(Ca -Cb/Kc);
19 dnAdV = rA;
20 end

```

```

1 [V,nA]=ode45(@fun,[0,5],2);
2 [~, Cb] = fun(V,nA);
3
4
5
6
7 plot(V, Cb)
8
9 function [dnAdV,Cb]=fun(V,nA)
10 k = 0.9;
11 Kc = 1.14;
12 v0 = 10;
13 nA0 = 2;
14 v = v0;
15 nB = nA0 - nA;
16 Ca = nA/v;
17 Cb = nB/v;
18 rA = -k*(Ca -Cb/Kc);
19 dnAdV = rA;
20 end

```

Figure 6.2: The approach on the **left** duplicates some the code within fun in order to calculate Cb after the ODE is solved on Line 1. The approach on the **right** assigns Cb as an additional output to the local function on Line 9, then calls the local function on Line 2 after the ODE is solved.

ode45 we imagine the variables  $x$  and  $z$  to be “bundled” into a vector  $Y$  as

$$Y = \begin{bmatrix} x \\ z \end{bmatrix}, \quad (6.5)$$

the initial conditions  $x_0$  and  $z_0$  to be “bundled” into a vector  $Y_0$  as

$$Y_0 = \begin{bmatrix} x_0 \\ z_0 \end{bmatrix},$$

and the differentials  $dx/dt$  and  $dz/dt$  to be “bundled” into a vector  $dY/dt$  as

$$\frac{dY}{dt} = \begin{bmatrix} dx/dt \\ dz/dt \end{bmatrix} = \begin{bmatrix} z + 4 \\ -3x \end{bmatrix}. \quad (6.6)$$

The inputs to ode45 are mostly unchanged; the only noticeable difference is that you’ll need to provide  $Y_0$  as a column vector of initial conditions rather than as a single (scalar) initial condition. The outputs from ode45, assuming that you use the two-output syntax, are slightly changed: the vector of solution points for the independent variable  $t$  is the same as for a single ODE, but the vector of solution points for the dependent variable  $Y$  is now a *matrix* where each *column* of the matrix corresponds to one of the dependent variables.

There are two important changes to the local function file to keep in mind. First, the function definition will be something like

```
function dYdt = fun(t,Y)
```

and since  $Y$  is a column vector consisting of  $x$  and  $z$  as defined in Equation (6.5) we can access the variables  $x$  and  $z$  by indexing into  $Y$  as

```
function dYdt = fun(t,Y)
x=Y(1);
z=Y(2);
```

Second, for simple reasons of syntax the output from the local function must *always* be a column vector; a row vector will produce an error. For example, we could define the necessary local function for the ODE system above as

```
1 function dYdt = fun(t,Y)
2 x = Y(1);
3 z = Y(2);
4 dxdt = z+4;
5 dzdt = -3*x;
6 dYdt = [dxdt; dzdt]; % <- column vector!
```

```
7 | end
```

Let's see how to put all this together to actually solve the system.

#### Example 6.4: Solving a system of coupled ODEs with ode45

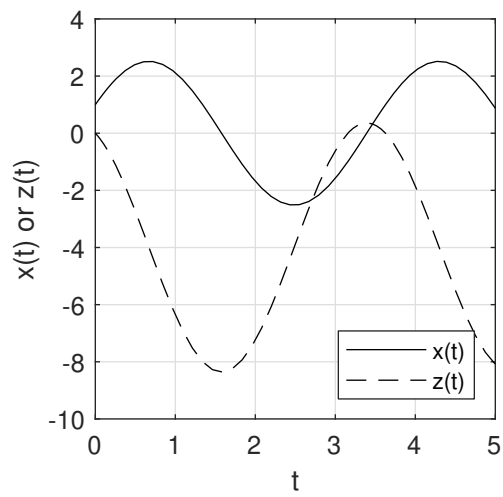
Solve the system of ODEs given in Equations (6.3) and (6.4) from  $t = 0$  to  $t = 5$ , subject to the boundary conditions  $x_0 = 1$  and  $z_0 = 0$ . Plot  $x(t)$  and  $z(t)$ .

#### Solution

The following script solves the system and plots the solution. Take special note of the column vectors on Lines 4 and 14 because this is the required syntax for systems of ODEs.

```
1 % ode45sys.m
2 x0 = 1;
3 z0 = 0;
4 Y0 = [x0; z0]; % <- column vector!
5 tspan = [0 5];
6 [tSol, YSol] = ode45(@fun, tspan, Y0);
7 hold on
8 plot(tSol, YSol(:, 1));
9 plot(tSol, YSol(:, 2));
10 hold off
11 legend('x(t)', 'z(t)')
12
13 function dYdt = fun(t,Y)
14 x = Y(1);
15 z = Y(2);
16 dxdt = z+4;
17 dzdt = -3*x;
18 dYdt = [dxdt; dzdt]; % <- column vector!
19 end
```

Running this script produces the plot shown below. Additional parameters can be passed to the fun function in exactly the same way as shown in Section 6.4.



## Chapter 7

### The Plug Flow Reactor, or PFR

“It’s a series of tubes!”

United States Senator Ted Stevens (R-Alaska, 2006)

We now introduce the third of four primary reactors, the *plug flow reactor* or PFR. As we did with the CSTR, let’s start with the good stuff right away: the general form of a material balance for the plug flow reactor is

$$\boxed{\frac{dn_i}{dV} = r_i} \quad n_i(V=0) = n_{i0} \quad (7.1)$$

and the general form of the energy balance for the PFR is

$$\boxed{\frac{dT}{dV} = \frac{\delta Q - \sum_{j=1}^q r_j \Delta H_j}{\sum_{i=A}^Z n_i C_{P,i}}} \quad T(V=0) = T_0 \quad (7.2)$$

As with the CSTR, the reason we start off with these two equations is because you should know that you can *always* come back to these two equations to solve *any* PFR problem in this book. We’re certainly going to derive these expressions so that you can understand their limitations and assumptions but try to keep in mind that every problem that follows, including every special PFR case we’ll see in Chapter 8, are all applications, rearrangements, or simplifications of Equations (7.1) and (7.2).

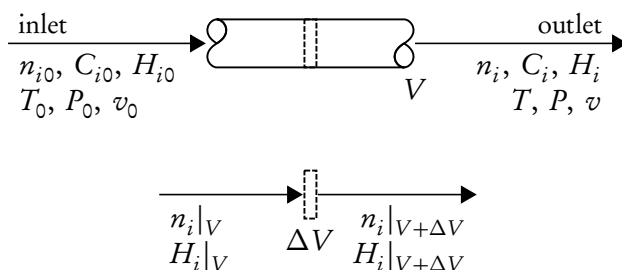


Figure 7.1: Basic diagram and variables associated with the PFR; heat  $Q$  and shaft work  $W_s$  are omitted for clarity. (top) Macroscale of the full reactor of volume  $V$ . (bottom) Detail of infinitesimal slice of volume  $\Delta V$  beginning after the fluid has been exposed to any arbitrary volume  $V$ .

## 7.1 The PFR material balances are ordinary differential equations

A basic illustration of the plug flow reactor is shown in Figure 7.1 and it's shouldn't be over-analyzed: PFRs, at their most basic, are just tubes through which flows a fluid undergoing chemical reaction. Not shown in this figure are heat exchangers and mixers, which are addressed in the next few sections. PFRs are equally suited to liquid and gas-phase reactions, and catalyst can be deposited on the walls of the PFR.<sup>1</sup>

The key assumption of a plug flow reactor is in the first two words, *plug flow*: it's assumed that there is no radial or axial variation in the fluid's velocity profile. Under such an assumption each fluid element travels at exactly the same rate and doesn't mix with any of its neighboring fluid elements.

To develop the material balance for the PFR let's start with the infinitesimal slice of reactor volume shown in the bottom part of Figure 7.1. The location of the slice along the volume of the reactor is unimportant, but at whatever point we choose to make the slice the fluid has already experienced a volume  $V$ . The slice has a small volume  $\Delta V$  such that the exiting fluid has now experienced a volume  $V + \Delta V$ , just a little bit more volume than the entering fluid. Since the slice is tiny we assume its contents are uniform so that we can write the general material balance for any species,

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}$$

as

$$0 = n_i|_V - n_i|_{V+\Delta V} + r_i \Delta V, \quad (7.3)$$

where the vertical bars  $|_V$  and  $|_{V+\Delta V}$  indicate quantities evaluated at the entrance and exit of

<sup>1</sup>Catalyst can also be piled up within the volume of the reactor, which is the topic of Chapter 9.

our infinitesimal slice. Rearranging and dividing by  $\Delta V$  yields

$$\frac{n_i|_{V+\Delta V} - n_i|_V}{\Delta V} = r_i \quad (7.4)$$

In the limit of  $\Delta V \rightarrow 0$  the left-hand side becomes a derivative,

$$\lim_{\Delta V \rightarrow 0} \frac{n_i|_{V+\Delta V} - n_i|_V}{\Delta V} = \frac{dn_i}{dV} \quad (7.5)$$

so that the general material balance for any species  $i$  within a plug flow reactor becomes

$$\frac{dn_i}{dV} = r_i \quad (7.6)$$

Equation (7.6) is a first order, ordinary differential equation and therefore requires a boundary or initial condition. Most commonly we have information about the inlet flow of each species so we use an initial condition of  $n_i$  at  $V = 0$ , or

$$\boxed{\frac{dn_i}{dV} = r_i} \quad n_i(V = 0) = n_{i0} \quad (7.1)$$

as we saw at the beginning of the chapter. Remember that we can write Equation (7.1) once for each species involved in a process. Also keep in mind that for each material balance we write on species  $i$ , that material balance needs its own initial condition.

## 7.2 The PFR energy balance is an ordinary differential equation

The PFR energy balance is developed in a manner similar to the material balance, which is to start with the infinitesimal slice of reactor volume shown in the bottom part of Figure 7.1. For this volume at steady state, the general balance expression

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}$$

becomes

$$0 = \sum_{i=A}^Z [n_i H_i]_V - \sum_{i=A}^Z [n_i H_i]_{V+\Delta V} + \delta W_S \Delta V + \delta Q \Delta V, \quad (7.7)$$

where  $\delta W_S \Delta V$  is the shaft work done *on* the volume element and  $\delta Q \Delta V$  is the heat entering the volume element. The terms  $\delta W_S$  and  $\delta Q$  are the shaft work and heat *per unit volume*, which is why they're denoted as  $\delta Q$  instead of just  $Q$  and  $\delta W_S$  instead of just  $W_S$ . The terms

$\delta Q$  and  $\delta W_s$  should be interpreted as one term each, not the products of  $\delta$  and  $Q$  or  $\delta$  and  $W_s$ . For reasons that will be discussed in the next section, the shaft work term  $\delta W_s \Delta V$  is nearly always zero—not just small compared to the other terms, but actually zero—so we'll omit it going forward.

Rearranging Equation (7.7), dividing by  $\Delta V$ , and taking the limit as  $\Delta V \rightarrow 0$  produces

$$\frac{d}{dV} \left( \sum_{i=A}^Z n_i H_i \right) = \delta Q \quad (7.8)$$

Applying the chain rule,

$$\sum_{i=A}^Z \left( n_i \frac{dH_i}{dV} \right) + \sum_{i=A}^Z \left( H_i \frac{dn_i}{dV} \right) = \delta Q \quad (7.9)$$

and at this point we need to bring in some additional relationships in order to make more progress. Recall from your thermodynamics course that we can approximate the change in enthalpy as

$$dH_i = C_{p,i} dT, \quad (7.10)$$

which is valid when there are no phase changes, is exact for ideal gases, and is a fair approximation for liquids at low pressures. The PFR material balance also gives

$$\frac{dn_i}{dV} = r_i = \nu_i r, \quad (7.1)$$

and substituting Equations (7.1) and (7.10) into Equation (7.9) yields

$$\sum_{i=A}^Z \left( n_i C_{p,i} \frac{dT}{dV} \right) + \sum_{i=A}^Z (r \nu_i H_i) = \delta Q. \quad (7.11)$$

If the sum terms are written out for all  $N$  species we'll see that the terms  $dT/dV$  and  $r$  are the same in every one of the sum terms, which means we can factor them out as

$$\frac{dT}{dV} \sum_{i=A}^Z (n_i C_{p,i}) + r \sum_{i=A}^Z (\nu_i H_i) = \delta Q, \quad (7.12)$$

and within this equation we can use Equation (4.13) to identify  $\sum \nu_i H_i = \Delta H$ . Substituting



this relationship into Equation (7.12) and rearranging to isolate  $dT/dV$  yields

$$\frac{dT}{dV} = \frac{\delta Q - r\Delta H}{\sum_{i=A}^Z n_i C_{P,i}}, \quad (7.13)$$

which is the energy balance for PFRs in which only a single reaction occurs. To extend Equation (7.13) to account for multiple reactions requires only modification of  $r\Delta H$  in the same way we did in Chapter 4,

$$r\Delta H \xrightarrow[\text{mult. reactions}]{\text{extend to}} \sum_{j=1}^q r_j \Delta H_j, \quad (4.25)$$

where the index  $j$  runs from 1 to the number of reactions,  $q$  (not to be confused with heat  $Q$  or heat per unit volume  $\delta Q$ ). Substituting this modification for multiple reactions makes Equation (7.13) become the general energy balance for a PFR that was introduced at the beginning of the chapter,

$$\boxed{\frac{dT}{dV} = \frac{\delta Q - \sum_{j=1}^q r_j \Delta H_j}{\sum_{i=A}^Z n_i C_{P,i}}} \quad T(V=0) = T_0 \quad (7.2)$$

which can be written somewhat more compactly as

$$\frac{dT}{dV} = \frac{\delta Q - \sum r_j \Delta H_j}{\sum n_i C_{P,i}}, \quad T(V=0) = T_0 \quad (7.14)$$

where the sum indices were omitted for clarity. Naturally if you want to use Equation (7.14) instead of Equation (7.2) you'll have to remember that the sum in the numerator is over all  $j$  reactions and the sum in the denominator is over all  $i$  components.

### 7.3 Shaft work is negligible and heat is transferred by a heat exchanger

In our drawings of CSTRs—see Figures 4.1 and 4.2—we often showed a stirring paddle to represent some kind of a mixer or agitator, which is a central feature of the continuous *stirred* tank reactor. As tubes, however, PFRs are not geometrically well-suited to having stirring paddles inserted through the tube wall, and given the length of most PFRs it's also not practical to insert a stirring paddle axially near the inlet or outlet. For these reasons the shaft work term  $\delta W_s$  is zero in nearly all PFR designs because no mixer is present. If mixing is required then

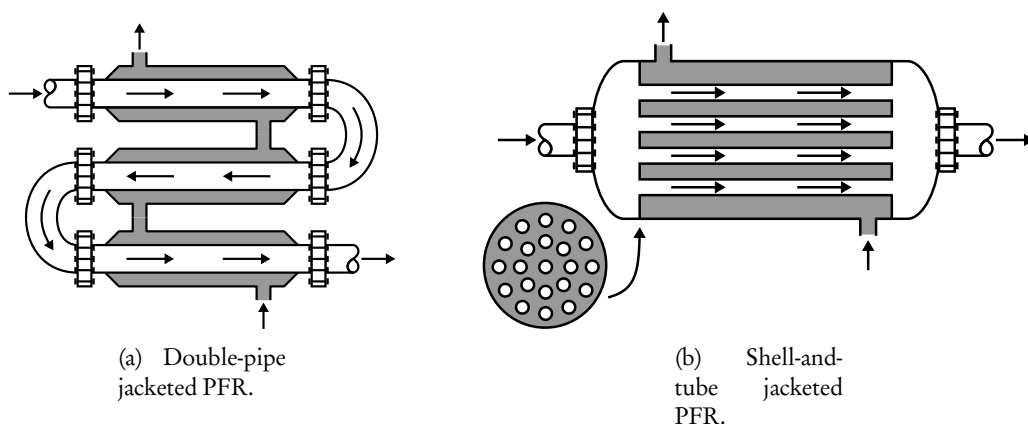


Figure 7.2: Heat exchange in PFRs is usually accomplished using either a double-pipe arrangement or a shell-and-tube arrangement. No reactions occur in the heat exchangers and there is no mixing of process (white) and utility (gray) fluids. The flow direction of the utility stream is important if the utility stream temperature is not constant.

structures called *static mixers*—things like baffles, twists, and turns—can be installed throughout the volume of the reactor.

Heat, as with the CSTR, can almost never be ignored because it's one of the primary ways we regulate a reaction. PFRs can be operated adiabatically simply by wrapping them in an appropriate insulation, and if a heat exchanger is required it's nearly always a jacketed heat exchanger as opposed to a coiled heat exchanger. Figure 7.2 shows two common geometries for jacketed PFRs, the double-pipe heat exchanger and the shell-and-tube heat exchanger.

In the double-pipe design shown in Figure 7.2a, an inner tube is surrounded by an outer tube, the process stream (white) flowing through the inner tube and the utility stream (gray) flowing through the annulus between the two tubes. If particularly large volumes are required, multiple double-pipe designs can be connected in series.

In the shell-and-tube design shown in Figure 7.2b, a main supply line of process fluid (white) enters a series of parallel tubes called a *tube bank*. Each tube in the bank is assumed to receive an equal flow from the main supply and to operate under identical conditions. The tube bank is contained within a larger tube called a shell; the utility fluid (gray) flows in the space outside the tubes and within the shell. The circular inset figure in Figure 7.2b shows the inlet plate which directs the entering fluid into the tube bank; a similar plate is installed at the outlet of the tube bank.

To develop an expression for  $\delta Q$  we have to first figure out how much energy enters (or leaves) the tiny volume element shown in Figure 7.1. Denoting this amount of energy as  $Q_{\Delta V}$  and assuming it *enters* the small slice  $\Delta V$ , we can use Equation (4.26) to write an analogous

expression for  $Q_{\Delta V}$  as

$$Q_{\Delta V} = UA_{\Delta V}(T_u - T), \quad (7.15)$$

where  $A_{\Delta V}$  is the area available for heat transfer to the tiny volume element  $\Delta V$ , and the other variables are the same as defined previously (c.f. Section 4.3). The variable  $A_{\Delta V}$  can be generally defined as

$$A_{\Delta V} = \frac{dA}{dV} \Delta V = a \Delta V, \quad (7.16)$$

where  $a$  is the area available for heat exchange per unit volume of the reactor. If you have a strange reactor design for which the heat transfer area is not a fixed proportion of the reactor volume (e.g., the pipe diameter changes along the length of the reactor) then you'll have to come back to Equation (7.16) to derive a general expression for  $a$ . However, by far the most common geometry for a PFR is a cylinder of constant diameter  $d$  and length  $L$ , for which the volume is

$$V = \frac{1}{4} \pi d^2 L = \frac{d}{4} (\pi d L) = \frac{d}{4} A,$$

and from which we calculate  $a$  as

$$a = \frac{dA}{dV} = \frac{4}{d}. \quad (\text{cylindrical PFR}) \quad (7.17)$$

Substituting this expression for  $a$  into Equation (7.15) yields

$$Q_{\Delta V} = Ua(T_u - T)\Delta V = \delta Q \Delta V, \quad (7.18)$$

where

$$\boxed{\delta Q = Ua(T_u - T)} \quad (7.19)$$

with  $a = 4/d$  for cylindrical PFRs. Notice the unit of  $a$  is  $\text{length}^{-1}$ , which is somewhat unusual, but the units of  $\delta Q$  are  $\text{energy} \cdot \text{volume}^{-1} \cdot \text{time}^{-1}$ , which exactly matches the units of  $r\Delta H$  in the numerator of Equation (7.2). The overall heat transfer coefficient  $U$  can be estimated from the individual heat transfer coefficients  $h$  using Equation (4.27) and Table 4.1 in the same manner as we did for CSTRs.<sup>2</sup>

The problem of how to deal with non-constant utility temperatures is one that requires enough effort to resolve that we're going to put it off for another chapter. For this chapter we'll always assume that the utility temperature  $T_u$  is constant, which as you'll recall from

<sup>2</sup>To use Equation (4.27) with PFRs requires us to assume the tubes have negligible thermal resistance, either due to high thermal conductivity, thin walls, or both. You'll learn (or have learned) how to correct  $U$  for tubes and pipes of non-negligible thermal resistance in your heat transfer course.

Table 7.1: Stoichiometric tables for PFRs cannot be written with all the same information as similar tables for CSTRs.

(a) For **CSTRs**, stoichiometric tables for **may include** outlet stream properties as functions of the CSTR material balance, Equation (4.1), as  $n_i = n_{i0} + r_i V$ .

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$n_i = n_{A0} \left( \Theta_i + \frac{\nu_i}{\nu_A} X_A \right)$	$n_i = n_{i0} + r_i V$
A	$\nu_A$	$n_{A0}$	1	$\nu_A r$	$n_A = n_{A0}(1 - X_A)$	$n_A = n_{A0} + \nu_A r V$
B	$\nu_B$	$n_{B0}$	$\Theta_B$	$\nu_B r$	$n_B = n_{A0}(\Theta_B - (\nu_B/\nu_A)X_A)$	$n_B = n_{B0} + \nu_B r V$

(b) For **PFRs**, stoichiometric tables **should not include** properties calculated by the PFR material balance Equation (7.1). Properties calculated by stoichiometry relationships—namely those involving  $X_A$ —may still be included.

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$n_i = n_{A0} \left( \Theta_i + \frac{\nu_i}{\nu_A} X_A \right)$
A	$\nu_A$	$n_{A0}$	1	$\nu_A r$	$n_A = n_{A0}(1 - X_A)$
B	$\nu_B$	$n_{B0}$	$\Theta_B$	$\nu_B r$	$n_B = n_{A0}(\Theta_B - (\nu_B/\nu_A)X_A)$

Section 4.3 can be achieved with either large utility flow rates, high heat capacity utility fluids, or utility fluids which undergo phase change.

### 7.4 PFR problems can be solved by hand calculations

The category of PFR problems that can be solved by hand calculations using Equations (7.1) and (7.2) is fairly small. Nearly all such problems are isothermal because otherwise we’re left with a system of coupled ODEs (*c.f.* Section 6.6) which generally cannot be solved by hand calculations. In addition to being isothermal the problems are often constrained to have constant volumetric flow rate,  $v = v_0$ , by one of two restrictions: the process fluid is liquid-phase, or the process is gas-phase but isothermal and isobaric with  $\delta = 0$  (*c.f.* Equation (5.5)).

If they can be worked by hand the integrals associated with the solution of the PFR material balances given in Equation (7.1) are usually one of the indefinite integral forms given in Appendix A. Later we’ll see that the material balance can be easier to solve by hand if it’s written in terms of conversion  $X_A$  rather than molar flow rate  $n_i$  because the lower bound of the integral is nearly always 0, and in this case one of the definite integrals provided in Appendix A can be useful. However as we saw in Chapter 5 the use of  $X_A$  instead of  $n_i$  in material balances is a special case of the general material balance so we defer that topic to the next chapter as well.

As with CSTRs a stoichiometric table is again a great place to start every problem. Im-

portant to note, however, is that any columns in the table related to outlet stream properties can *only* be expressed as functions of stoichiometry but *never* as functions of rates or material balances. The reason for this is that Equations (7.1) and (7.2) are differential equations, not algebraic equations. Table 7.1 compares the two forms of the stoichiometric table. Since we won't write material balances in terms of conversion again until Chapter 8 the stoichiometric tables we write in this chapter typically won't include more than stoichiometric coefficients, inlet flows, and net rates of reaction.

### Example 7.1: Isothermal PFR sizing

A feed of pure A at  $3 \text{ L} \cdot \text{s}^{-1}$  is provided to an isothermal PFR in which the elementary, liquid-phase reaction



takes place with  $k = 0.003 \text{ s}^{-1}$  (constant). Determine the size of the PFR needed to convert 75% of the feed, and compare this volume to the volume calculated in Example 4.2 as well as an everyday object.

### Solution

Let's write a short stoichiometric table to summarize some of the problem statement:

Sp.	$\nu_i$	$n_{i0}$	$r_i$	$n_i$
A	-1	$n_{A0}$	$-r$	$n_{A0}(1 - X_A)$
B	1	$n_{B0}$	$r$	$n_{A0}X_A$

The system is isothermal so we'll ignore the energy balance and focus on only the material balance, Equation (7.1) for species A,

$$\frac{dn_A}{dV} = r_A,$$

where

$$r_A = -r = -kC_A = -\frac{k}{v}n_A = -\frac{k}{v_0}n_A.$$

Substituting the rate law into the material balance yields

$$\frac{dn_A}{dV} = -\frac{k}{v_0} n_A,$$

subject to the initial condition  $n_A(V=0) = n_{A0}$ . This is a separable differential equation which can be solved by moving all  $n_A$  terms to the  $dn_A$  side and everything else to the  $dV$  side, or

$$\frac{dn_A}{n_A} = -\frac{k}{v_0} dV.$$

Notice that  $-k/v$  is constant and could have been placed on either side of this equation, but conventionally we put it with the volume term. Integrating both sides we have

$$\int_{n_{A0}}^{n_A} \frac{dn_A}{n_A} = -\int_0^V \frac{k}{v_0} dV,$$

where we're being a little sloppy with our notation because the variable of integration should not also appear in the limits of the integral,<sup>a</sup> but we write it this way to save time and avoid introducing yet more notation. Evaluating both integrals and applying the boundary conditions yields

$$\ln \frac{n_A}{n_{A0}} = -\frac{k}{v_0} V,$$

We know  $k$  and  $v$  because they were given in the problem statement but we don't know  $n_A$  or  $n_{A0}$ . However, we know that 75% of the pure A fed is converted to B, or

$$n_A = (1 - 0.75) n_{A0} = 0.25 n_{A0},$$

which can be rearranged to

$$\frac{n_A}{n_{A0}} = 0.25$$

and substituted into the material balance to yield

$$\ln 0.25 = -\frac{k}{v_0} V.$$

Solving for  $V$  yields

$$V = -\frac{v_0}{k} \ln 0.25 = -\frac{3 \text{ L} \cdot \text{s}^{-1}}{0.003 \text{ s}^{-1}} \ln 0.25 = 1386 \text{ L}.$$

That's less than *half* the volume of the CSTR we found previously, and all we did was change from a tank to a tube! If the tube or pipe has a diameter roughly the same as your head (about 1 ft) then it needs to be about 60 ft long to achieve this volume, which is about the length of a medium-size classroom.

---

<sup>a</sup>For example, the variable of integration is sometimes denoted with a prime as  $dV'$  instead of  $dV$ .

It's possible to design a single, long tube as suggested at the end of Example 7.1 as a series of bends as in Figure 7.2a, but a more compact design is to use a series of parallel, straight tubes in the shell-and-tube design shown in Figure 7.2b.

### Example 7.2: Isothermal shell-and-tube PFR

Assume the reactor from Example 7.1 was to be constructed not from a single large tube but rather in a shell-and-tube design as shown in Figure 7.2b. If each tube is 2 in-sch 40 pipe (pronounced “two inch schedule forty”) with a length of 20 ft,

- (a) determine the number of tubes ( $N_{\text{tube}}$ ) needed, and
- (b) show that this parallel tube arrangement achieves the same conversion as a single, large tube of the same total volume.

### Solution

Whenever you see the word “schedule” or the abbreviation “sch” in reference to a pipe, you're dealing with a nominal (*i.e.*, arbitrary) pipe size specification which means the “2 in” doesn't really mean “two inches”. Checking any number of online sources indicates that 2 in-sch 40 pipe has an outer diameter of 2.375 in and a wall thickness of 0.154 in, which means its inner diameter is  $2.375 - 2 \times 0.154 = 2.067$  in, or 52.5 mm. The inner diameter is the one through which process fluid flows, so this is the dimension of interest.

(a) A single 2 in-sch 40 pipe with a length of 20 ft (6096.0 mm) has a volume of

$$V_{\text{one tube}} = \frac{1}{4}\pi d^2 L = \frac{1}{4}\pi (52.5 \text{ mm})^2 (6096.0 \text{ mm}) = 13.197 \text{ L},$$

and to obtain a total volume of at least 1386 L we therefore need at least

$$N_{\text{tube}} = \frac{V}{V_{\text{one tube}}} = \frac{1386 \text{ L}}{13.197 \text{ L}} = 105.02 \approx 105 \text{ tubes},$$

where the number of tubes was rounded to the nearest integer because partial tubes can't be used in the shell-and-tube geometry (typically we round up, but here the numbers came out close to enough that rounding down should be ok).

(b) As mentioned in Section 7.3 one of the assumptions of the shell-and-tube design is that each tube in the bank receives an equal flow from the main supply. Therefore the "main supply" of  $3 \text{ L} \cdot \text{s}^{-1}$  is divided equally among all 105 tubes so that the flow rate within each tube is

$$\frac{3 \text{ L} \cdot \text{s}^{-1}}{105 \text{ tubes}} = 0.029 \text{ L} \cdot \text{s}^{-1} \text{ per tube}.$$

For a single tube with  $V = 1386 \text{ L}$  we saw that  $n_A/n_{A0} = 0.25$ , so our goal is to find the ratio  $n_A/n_{A0}$  for a single, small tube with  $V = 13.197 \text{ L}$  and show that it's the same number. Substituting the volumetric flow rate of a small tube into the integrated material balance from Example 7.1 yields

$$\ln \frac{n_A}{n_{A0}} = -\frac{0.003}{0.029} 13.197 \Rightarrow \frac{n_A}{n_{A0}} = 0.255,$$

which is close to 0.25 as expected, the difference being rounding of numbers throughout the calculation. As an exercise, try to show that  $kV/v_0$  for a single, large tube is algebraically equivalent to  $kV/v_0$  for one pipe within a shell-and-tube reactor, and in so doing you'll also show that exactly the same conversion is achieved in either design.

It might seem like 105 tubes is quite a few and that the corresponding shell might be exceptionally (even unreasonably) large but if the tubes are staggered on a triangular pattern as shown below then a shell of only about 30 in is needed to contain the tubes—a diameter roughly equal to the length of your outstretched arm (see below).



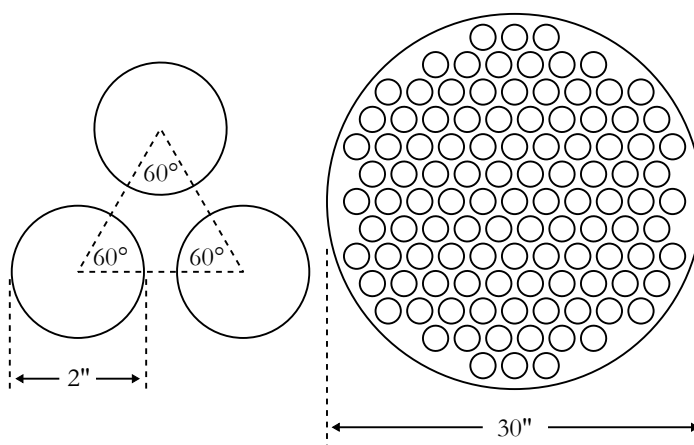
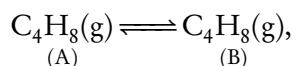


Figure 7.3: Sample tube sheet layout for Example 7.2. (left) Enlarged view of a triangular tube pattern with space between tubes for utility fluid to flow. (right) Tube sheet hole pattern for 109 tubes—slightly more than the 105 needed—within a 30 in (2.5 ft) diameter shell.

As noted previously, the integrals associated with Equation (7.1) can rapidly grow beyond hand solutions. The next example is close to the limit of what can be reasonably expected of a hand calculation when it comes to PFRs.

### Example 7.3: Isomerization in a PFR

The uncatalyzed, gas-phase isomerization of  $10 \text{ m}^3 \cdot \text{min}^{-1}$  pure cis-2-butene ( $\text{C}_4\text{H}_8$ , A) to trans-2-butene ( $\text{C}_4\text{H}_8$ , B) takes place according to the reversible, elementary reaction



in an isothermal, isobaric PFR at  $350^\circ\text{C}$  and  $101.325 \text{ kPa}$ . If the forward rate constant is  $0.023 \text{ min}^{-1}$ ,

- calculate the reactor volume  $V$  needed to achieve 40% conversion and compare this volume to an everyday object, and
- create a side-by-side figure with the molar flow rate profiles of both species on the left plot, and the equilibrium and reactor conversion profiles on the right plot.<sup>a</sup>

<sup>a</sup>A “profile” is a plot of that quantity as a function of reactor volume.

**Solution**

Before we get to the solution let's make a stoichiometric table to organize the available information:

Sp.	$\nu_i$	$n_{i0}$	$r_i$	$n_i$	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )	$\Delta G_f^\circ$ (kJ · mol <sup>-1</sup> )
A	-1	$n_{A0}$	$-r$	$n_{A0}(1 - X_A)$	-7.4	65.4
B	1	0	$r$	$n_{A0}X_A$	-11	63.2
$\delta = 0$					$\Delta M = -3.6$	-2.2

(a) The material balance for A is given in Equation (7.1) as

$$\frac{dn_A}{dV} = \nu_A r = r_A, \quad n_A(V=0) = n_{A0}$$

where

$$r = k \left( C_A - \frac{C_B}{K_C} \right) = \frac{k}{v} \left( n_A - \frac{n_B}{K_C} \right).$$

The general approach will be to solve the material balance to get an expression for  $V$  as a function of  $n_A$ , and then to apply the known conversion of 40% to find  $n_A$  followed by  $V$ . One thing to keep in mind is that we must never calculate  $n_A$  until *after* we integrate the material balance because  $n_A$  is itself the variable of integration.

From Equation (3.25) the equilibrium constant  $K_C$  is

$$K_C = K_a (P^\circ / RT)^\delta = K_a \approx K_{sc} = K_0 K_1$$

since  $\delta = 1 - 1 = 0$ . From Table D.2 at  $T_R = 298.15$  K at 101.325 kPa we have

$$\Delta H_R = (-1)(-7.4) + (1)(-11) = -3.6 \text{ kJ} \cdot \text{mol}^{-1}$$

$$\Delta G_R = (-1)(65.4) + (1)(63.2) = -2.2 \text{ kJ} \cdot \text{mol}^{-1}$$

from which

$$K_C = K_0 K_1 = \exp \left[ \frac{-\Delta G_R}{RT_R} \right] \exp \left[ \frac{\Delta H_R}{R} \left( \frac{1}{T_R} - \frac{1}{623 \text{ K}} \right) \right] \approx 1.14$$

which is constant because the reactor temperature is constant. From Equation (5.5) we

find that

$$v = v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0} = v_0$$

because the system is isothermal and isobaric, and the total number of moles is constant ( $n_T = n_{T0}$ ) since  $\delta = 0$ . Finally, from Equation (2.35) we note that  $n_B$  is directly proportional to  $n_A$  as

$$n_B = n_{A0} X_A = n_{A0} \frac{n_{A0} - n_A}{n_{A0}} = n_{A0} - n_A.$$

With these terms the material balance for A becomes

$$\frac{dn_A}{dV} = -\frac{k}{v_0} \left( n_A - \frac{n_{A0} - n_A}{1.14} \right),$$

which after a little bit of rearranging yields a separable differential equation

$$\int_{n_{A0}}^{n_A} \frac{dn_A}{c_1 n_A + c_2} = - \int_0^V \frac{k}{v_0} dV$$

where

$$c_1 = 1 + \frac{1}{K_C} \approx 1.88$$

and, using the ideal gas law as  $n_{A0} = P_0 v_0 / RT_0 = 195.58 \text{ mol} \cdot \text{min}^{-1}$ ,

$$c_2 = -\frac{n_{A0}}{K_C} \approx -171.56 \text{ mol} \cdot \text{min}^{-1}.$$

These integrals are available in Appendix A as Equation (A.5) for the left side and Equation (A.2) for the right side. Integrating and applying the boundary conditions yields

$$-\frac{v_0}{c_1 k} \ln \frac{c_1 n_A + c_2}{c_1 n_{A0} + c_2} = V. \quad (7.20)$$

All the quantities to calculate  $V$  are known except  $n_A$ , which can now be found using the conversion of 40% to be  $n_A = n_{A0}(1 - X_A) = 117.35 \text{ mol} \cdot \text{min}^{-1}$  (remember, we can't use the 40% conversion specification until *after* we integrate the material balance).

The volume of the reactor is therefore

$$V = \frac{-10}{(1.88)(0.023)} \ln \frac{(1.88)(117.35) - 171.56}{(1.88)(195.58) - 171.56} = \boxed{320.5 \text{ m}^3}$$

An average sewer pipe that you might find next to a street has a diameter of about 1 m, and to achieve this volume would need to be more than 400 m long, about the length of four football fields! Somebody should look into using a catalyst to speed up the rate of reaction!

(b) To create profile plots we need to know how each variable— $n_A$ ,  $n_B$ ,  $X_A$ , and  $X_A^{\text{eq}}$ —depends on  $V$ . In the previous part of this problem we found  $n_A(V)$  to be Equation (7.20) and as part of the solution to that part we also noted that  $n_B = n_{A0} - n_A$  by stoichiometry. Once we've calculated  $n_A(V)$  we can also calculate the conversion  $X_A(V)$  as  $1 - n_A(V)/n_{A0}$ . The equilibrium conversion  $X_A^{\text{eq}}$  can be calculated using Equation (2.32) as

$$K_{sc} \approx \prod_{i=A}^Z a_i^{v_i} = \frac{y_B}{y_A} = \frac{n_B}{n_A} = \frac{X_A^{\text{eq}}}{1 - X_A^{\text{eq}}},$$

which can be rearranged to yield

$$X_A^{\text{eq}} = \frac{K_{sc}}{1 + K_{sc}} = 0.533 \quad (\text{constant}),$$

which is constant because the reactor is isothermal. The script below will generate the appropriate plots, which are shown beneath the script.

```

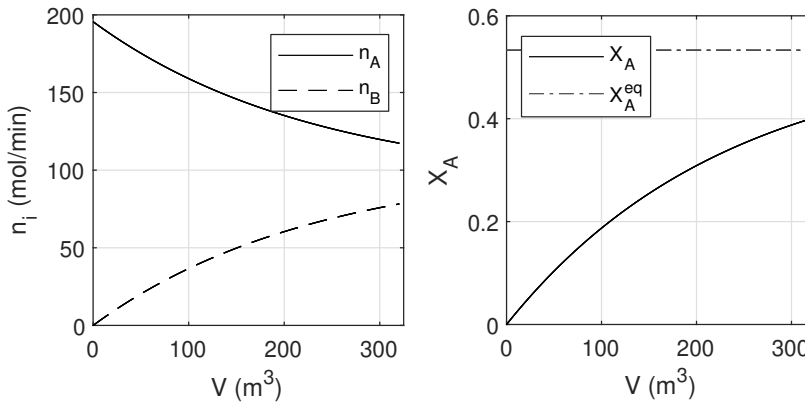
1 % pfrIsomerization.m
2 c1 = 1.88;
3 c2 = -171.56; % mol/min
4 v0 = 10; % m3/min
5 k = 0.023; % 1/min
6 nA0 = 195.58; % mol/min
7 V = linspace(0, 320.5); % m3
8 nA = ((c1*nA0+c2)*exp(-k*c1*V/v0)-c2)/c1; % mol/min
9 nB = nA0-nA; % mol/min
10 Xa = 1-nA/nA0;
11 Xa_eq = 0.533;
12
13 subplot 121; hold on
14 plot(V, nA, 'k')
15 plot(V, nB, 'k--')
16 hold off
17 subplot 122; hold on

```

```

18 plot(V, Xa, 'k')
19 yline(Xa_eq, 'k-.');
20 hold off

```



From this plot we can see that we're indeed approaching equilibrium because all curves are beginning to flatten off, but even with a volume of more than 300 m³ the reactor has only achieved about 75% (0.4/0.533) of the equilibrium conversion.

From these two examples we see that a large number of relationships—volumetric flow rates, rate laws, equilibrium constants, stoichiometry—are unchanged compared to the CSTR analyses from the previous two chapters. We'll see a few more examples of how to solve PFRs by hand calculation when we look at the special case of Equations (7.1) and (7.2) written in terms of conversion  $X_A$ , but now we look at how to solve these material and energy balances more generally using MATLAB.

## 7.5 PFR problems can be solved in MATLAB

When a differential equation or system of differential equations becomes too complicated to solve by hand calculation we can approximate the differential equation by a difference equation as described in Section 6.1. Such an approximation has been performed and stored in MATLAB's `ode45` function which we can use to solve individual (Section 6.2) or coupled (Section 6.6) differential equations.

As a brief review of Chapter 6, suppose we wish to solve the generic differential equation

$$\frac{dx}{dt} = f(x, t), \quad x(t=0) = x_0$$

So that we can write code that actually runs, let's assume  $f(x, t) = x + t$ ,  $x_0 = 0$ , and that we wish to solve this differential from  $t = 0$  to  $t = 5$ . The way to set up a script using `ode45` to solve this differential equation is as follows:

```

1 % yourScriptName.m
2 % Script portion
3 x0 = 0;
4 tspan = [0 5];
5 [xSol, tSol] = ode45(@fun, tspan, x0);
6
7 % Local (ODE) function
8 function dxdt = fun(t, x)
9     f = x + t;
10    dxdt = f;
11 end

```

This structure should look quite similar to the structure we used to find roots with `fsolve` way back in Section 1.5, and that's not an accident: the organization was deliberately chosen to be similar so that patterns knowledge from how to approach CSTR problems can be transferred to PFR problems.

There are a few things to note about this generic script:

- Lines 3 and 4 create the initial condition and span over which the ODE will be solved.
- Line 5 is the line that actually solves the ODE (see Sections 6.3 to 6.5 for additional modifications to outputs and inputs for `ode45`).
- Lines 8 through 11 define a local function which represents the differential equation. The output variable out is typically the right-hand side of the ODE. The variables  $x$  and  $t$  are available for calculation purposes because they're inputs to the local function (and they must *always* be provided as inputs to the local function).

Let's see how to use this approach to solve a PFR problem, and we'll start with one for which we know the exact (analytical) solution so that we can compare the results.

#### Example 7.4: Isomerization in a PFR with MATLAB

Use MATLAB to solve Example 7.3 and compare the volumes needed to achieve 75% conversion. Also create a plot of  $X_A(V)$  for both the analytic and numerical (`ode45`) solutions.

## Solution

The differential equation we wish to solve is the material balance for A,

$$\frac{dn_A}{dV} = r_A, \quad n_{A0} = 195.58 \text{ mol} \cdot \text{min}^{-1}$$

and we wish to solve this ODE until 75% conversion is obtained. We don't initially know how much volume is required so we'll simply pick a large placeholder value, say  $500 \text{ m}^3$ , which we can use to get started with our solution.

The beginnings of a script to solve this ODE are as follows:

```
1 % pfrIsomMatlab.m
2 nA0 = 195.58; % mol/min
3 Vspan = [0 500]; % m3
4 [V, nA] = ode45(@fun, Vspan, nA0);
5
6 function dnAdV = fun(V, nA)
7 % TODO
8 dnAdV = rA;
9 end
```

It's good to get into the habit of setting up a script this way because it establishes the solution itself (Lines 2-4), and the local function (Lines 6-9) has been set up in a way that we can interpret as follows: given any value of  $V$  and  $n_A$ , the local function `fun` needs to calculate the corresponding value of  $dn_A/dV$ , which we know from the general material balance to be equal to  $r_A$ . If *any* information other than  $V$  and  $n_A$  is need to calculate  $dn_A/dV$  then it needs to be defined within the local function.

Another good habit to get into when developing the local function for tubular reactors is to *work backwards*: we need to end up with an expression for  $r_A$  by the time the code reaches Line 8, so let's add a line between 7 and 8 that calculates  $r_A$  in a general way, like this:

```
6 function dnAdV = fun(V, nA)
7 % TODO
8 rA = -k*(Ca-Cb/Kc); % mol/m3-min
9 dnAdV = rA; % mol/m3-min
10 end
```

Now we ask ourselves if all the new variables we introduced on Line 8— $k$ ,  $C_a$ ,  $C_b$ , and  $K_c$ —have all been defined: if so then we can try to run the script, and if not then we have to keep adding equations and constants until they're all defined. We know  $k$  and  $K_c$  are

constant for this problem, so let's add those as constants as well as expressions for  $C_a$  and  $C_b$ :

```

6 function dnAdV = fun(V, nA)
7 % TODO
8 k = 0.023; % 1/min
9 Kc = 1.14;
10
11 Ca = nA/v; % mol/m3
12 Cb = nB/v; % mol/m3
13 rA = -k*(Ca-Cb/Kc); % mol/m3-min
14 dnAdV = rA; % mol/m3-min
15 end

```

Notice also that units have been added as comments to each line, which is yet another good habit to practice because you can catch *lots* of simple mistakes just by checking units.

Looking at the local function above, we're still missing expressions for  $n_A$ ,  $n_B$ , and  $v$ . A key point to keep in mind is that  $n_A$  is an input to the local function and therefore we do not need to define or calculate it; it'll already exist as a variable when the local function is called by ode45. That still leaves  $n_B$  and  $v$ , which will complete the local function when added:

```

6 function dnAdV = fun(V, nA)
7 k = 0.023; % 1/min
8 Kc = 1.14;
9 v0 = 10; % m3/min
10 nA0 = 195.58; % mol/min
11
12 v = v0; % m3/min
13 nB = nA0 - nA; % mol/min
14 Ca = nA/v; % mol/m3
15 Cb = nB/v; % mol/m3
16 rA = -k*(Ca-Cb/Kc); % mol/m3-min
17 dnAdV = rA; % mol/m3-min
18 end

```

We also had to add  $n_{A0}$  so that we can calculate  $n_B$  because the local function doesn't know what its value is ( $n_{A0}$  is an input to ode45 on Line 4, but it's not available to the local function `fun`). With these latest additions our local function is now complete so the `%TODO` reminder, which was just a note to ourselves that we're not done with the function, can be removed.

Our completed script, including a plot of conversion, is as follows:

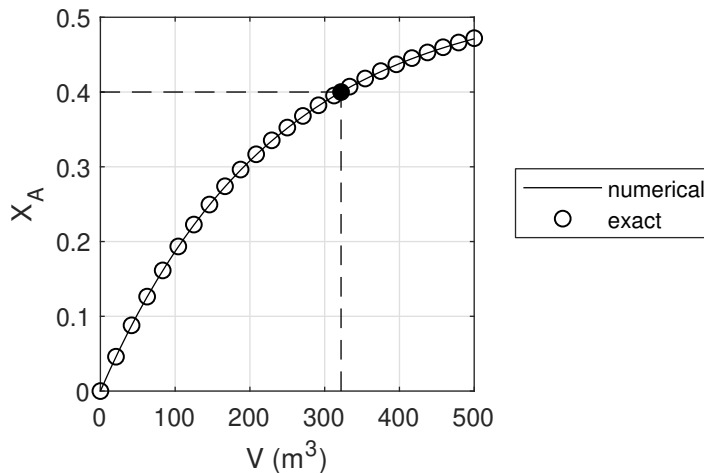


```

1 % pfrIsomMatlab.m
2 nA0 = 195.58; % mol/min
3 Vspan = [0 500]; % m3
4 [V, nA] = ode45(@fun, Vspan, nA0);
5 plot(V, 1-nA/nA0)
6
7 function dnAdV = fun(V, nA)
8 k = 0.023; % 1/min
9 Kc = 1.14;
10 v0 = 10; % m3/min
11 nA0 = 195.58; % mol/min
12
13 v = v0; % m3/min
14 nB = nA0 - nA; % mol/min
15 Ca = nA/v; % mol/m3
16 Cb = nB/v; % mol/m3
17 rA = -k*(Ca-Cb/Kc); % mol/m3-min
18 dnAdV = rA; % mol/m3-min
19 end

```

To find the volume needed to achieve 40% conversion we can inspect the resulting plot<sup>a</sup> to find where  $X_A = 40\%$ , which as shown below occurs at about  $321.9 \text{ m}^3$ , in good agreement with the analytic value of  $320.5 \text{ m}^3$ . The profile of  $X_A$  also shows good agreement between the numerical (ode45) and exact solutions.



<sup>a</sup>You could simply zoom in on the intersection point, or if you're feeling clever you could use a combination of an ODE solution object, the `deval` function, and `fsolve` to get a more accurate estimate.

In subsequent examples the scripts won't be developed in quite as much step-by-step detail as in Example 7.4 but you should return to this example any time you're stuck and not quite

sure where to begin on a PFR problem, or even on a CSTR problem because the same general approach works well there too.

Example 7.4 also illustrates another common experience when using ode45: sometimes the quantity of interest isn't exactly what's provided as an output from ode45 and some additional analysis is required. In this particular case, we were interested in the volume at which 40% conversion was achieved, yet ode45 provide us only with information about how  $n_A$  varied with  $V$ . We had to do additional analysis in the form of calculating and plotting  $X_A$ , followed by an inspection of the results.

As described in Section 6.6 the ode45 function can also be extended to solve multiple, coupled ODEs (also called a *system* of ODEs), and these occur commonly in multireaction systems.

### Example 7.5: Enzyme kinematics in a PFR

An *enzyme* is a large biomolecule which catalyzes a reaction, usually with exceptionally high selectivity. A common mechanism for enzyme-based catalysis consists of two elementary reactions,



where S is a reactant called the *substrate*, E is the unbound (free) enzyme, ES is a structure called the *enzyme-substrate complex*, and P is the product.

Suppose you're working in a lab and you have 6 ft of some 1/4" (ID) tygon tubing that you plan to use as an isothermal PFR for this enzyme reaction. Plot the molarity (M, mol · L<sup>-1</sup>) of each species as a function of reactor length if the feed to the reactor is 2 gpm of an aqueous solution containing 10<sup>-7</sup> M enzyme and 10<sup>-3</sup> M substrate.

The rate constants are  $k_f = 10^4$  L · mol<sup>-1</sup> · s<sup>-1</sup>,  $k_r = 1$  s<sup>-1</sup>, and  $k_p = 15$  s<sup>-1</sup>.

### Solution

A stoichiometric table will be nice here because this is a multi-reaction problem and stoich tables are particularly helpful in the calculation of net rates.

Sp.	$\nu_{i,1}$	$\nu_{i,2}$	$n_{i0}$	$r_i$
E	-1	1	$v_0 C_{E0}$	$-r_1 + r_2$
S	-1	0	$v_0 C_{S0}$	$-r_1$
ES	1	-1	0	$r_1 - r_2$
P	0	1	0	$r_2$

The rates of each individual reaction in the table are

$$r_1 = r_f - r_r = k_f C_E C_S - k_r C_{ES}$$

$$r_2 = k_p C_{ES}.$$

Since the PFR is isothermal we again begin with the general material balance, Equation (7.1), as

$$\frac{dn_i}{dV} = r_i, \quad n_i(V=0) = n_{i0},$$

where we've used the subscript  $i$  because we've got one material balance per species. Obviously we've got species E, S, ES, and P, but water (W) is also present because the feed is an aqueous solution, although in this particular problem it's not necessary to include the water because it's inert.

The material balance is written in terms of moles but the feed conditions and requested analysis are in terms of concentration (molarity), so the general approach of the problem will be as follows:

1. Do some unit conversions to go from molarity to moles.
2. Use the moles and ode45 to solve the problem.
3. Do some more unit conversions to go back to molarity from moles.

The basic structure of the solution script begins by following the syntax from Section 6.6, like so:

```

1 % pfrEnzyme.m
2 % TODO: unit conversions for M->mol
3 Y0 = [E0; S0; ES0; P0]; % mol/s
4 Vspan = [0 Vmax]; % L
5 [V,Y] = ode45(@fun, Vspan, Y0);
6 % TODO: unit conversions for mol->M
7 % TODO: plot molarity using semilogy
8
9 function dYdV = fun(V, Y)
```

```

10 nE = Y(1); % mol/s
11 nS = Y(2); % mol/s
12 nES = Y(3); % mol/s
13 nP = Y(4); % mol/s
14
15 % TODO: get r1, r2
16
17 dnEdV = -r1 + r2; % mol/s
18 dnSdV = -r1; % mol/s
19 dnESdV = r1 - r2; % mol/s
20 dnPdV = r2; % mol/s
21 dYdV = [dnEdV; dnSdV; dnESdV; dnPdV];
22 end

```

Notice that there are still a number of %TODO statements but the general structure of the code is not wildly different from what we saw in Example 7.4: the initial conditions and volume span are defined on Lines 3 and 4, the system of ODEs is solved with ode45 on Line 5, and the local function begins on Line 9.

One assumption that we'll make going forward: the feed is quite dilute so its properties will be well-approximated as those of pure water. We also assume those properties to be constant since the system is isothermal and concentrations are small. Additional unit conversions and calculations have been performed in the completed script below.

```

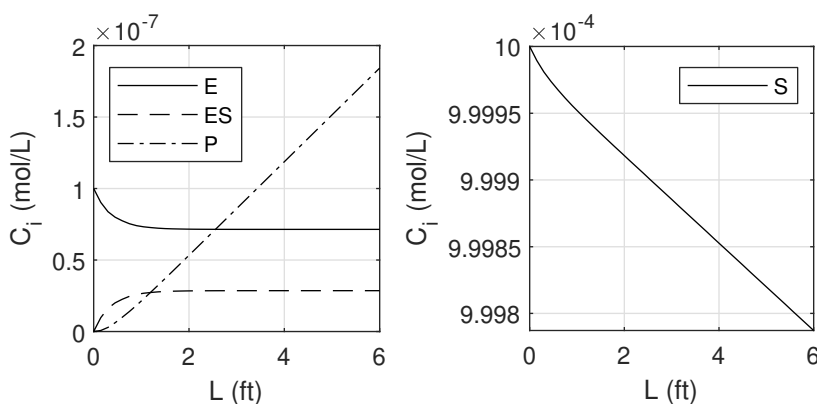
1 % pfrEnzyme.m
2 v0 = 2/60*3.79; % 2 gal/min -> L/s
3 E0 = 1e-7*v0; % mol/s, enzyme
4 S0 = 1e-3*v0; % mol/s, substrate
5 d = 1/4*2.54; % 1/4" -> cm
6 L = 6*12*2.54; % 6 ft -> cm
7
8 Y0 = [E0; S0; 0; 0]; % mol/s, [E S ES P]
9 Vspan = [0, 1/4*pi*d^2*L/1000]; % L
10 [V,Y] = ode45(@fun, Vspan, Y0);
11
12 L = V*1000/(1/4*pi*d^2)/(2.54*12); % L -> ft
13 C = Y/v0; % mol/s -> mol/L
14 subplot(1,2,1); plot(L, [C(:,1), C(:, 3:4)])
15 subplot(1,2,2); plot(L, C(:, 2))
16
17 function dYdV = fun(V, Y)
18 nE = Y(1); % mol/s
19 nS = Y(2); % mol/s
20 nES = Y(3); % mol/s
21
22 v0 = 2/60*3.79; % 2 gal/min -> L/s

```

```

23 v = v0; % L/s
24 Ce = nE/v; % mol/L
25 Cs = nS/v; % mol/L
26 Ces = nES/v;
27 kf = 1e4; % L/mol-s
28 kr = 1; % 1/s
29 kp = 15; % 1/s
30 r1 = kf*Ce*Cs - kr*Ces; % mol/L-s
31 r2 = kp*Ces; % mol/L-s
32
33 dnEdV = -r1 + r2; % mol/s
34 dnSdV = -r1; % mol/s
35 dnESdV = r1 - r2; % mol/s
36 dnPdV = r2; % mol/s
37 dYdV = [dnEdV; dnSdV; dnESdV; dnPdV];
38 end

```



Two plots were used because the molarities of the enzyme, enzyme-substrate complex, and product are four orders of magnitude smaller than that of the substrate,  $10^{-7}$  compared to  $10^{-3}$ . We also see from the left part that the concentrations of the enzyme and enzyme-substrate complex stabilize within about the first foot of the reactor.

The concentration of product is quite low, but that's not unusual for biochemical production. The concentration of substrate is nearly unchanged and could even be approximated as constant, which indeed represents a common approach for the experimental determination of rate laws and rate constants: make one reactant far in excess of the other so that its concentration is essentially constant, greatly simplifying the rate expressions.

One additional point about the previous example is worth noting: one rate constant was

several orders of magnitude higher than the other two, as was one concentration compared to the other three. When order-of-magnitude differences in such terms are present the ode45 solver can sometimes fail to provide a solution, instead returning noisy or incorrect results. These kinds of differential equations are called *stiff*, a concept similar to those of stability and scaling introduced in Section 4.6.

Unlike algebraic expressions, differential equations *cannot* be scaled by arbitrary factors to improve performance. Should you encounter a stiff ODE, try the following:

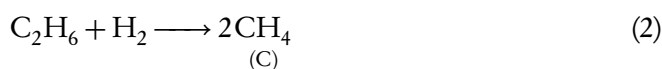
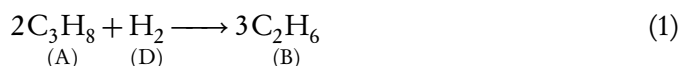
1. Check your unit conversions! A factor of  $10^3$  one way or the other from a missed unit conversion can wreak havoc on your solution.
2. Try one of the stiff ODE solvers like ode15s or ode45s.

If neither of these are successful then you're in a bit of a pickle! You may need to consult more advanced texts on numerical solution of differential equations to get yourself out of it.

Let's look at one more example of how to use MATLAB to solve coupled ODEs with ode45, this time with a non-isothermal system.

### Example 7.6: Thermal cracking in a PFR

Thermal cracking uses high temperatures and pressures to break long-chain hydrocarbons into short-chain hydrocarbons, such as propane into ethane and methane according to the gas-phase reactions



The rate laws for both reactions are first-order in both the hydrocarbon and hydrogen concentrations with Arrhenius parameters given below. The feed stream contains  $\text{C}_3\text{H}_8$  with 150% excess  $\text{H}_2$  according to reaction 1, with a total molar flow rate of  $0.1 \text{ kmol} \cdot \text{s}^{-1}$  at  $550^\circ\text{C}$  and 2.5 MPa. The feed enters a shell-and-tube style PFR with 137 tubes, each tube with a diameter of 2 cm and length of 14 m.

Compare reactor profiles for a reactor operated adiabatically versus jacketed with saturated, liquid water at  $350^\circ\text{C}$  (constant), and comment on the safety of operating the reactor in these two ways.

Reaction	$A \text{ (m}^3 \cdot \text{mol}^{-1} \cdot \text{s}^{-1}\text{)}$	$E_a \text{ (kJ} \cdot \text{mol}^{-1}\text{)}$
(1)	0.05	19.1
(2)	0.10	25.6

### Solution

This problem represents the peak complexity of PFR problems: non-isothermal with multiple, gas-phase reactions. To write code as efficiently as possible we'll use the vector and matrix capabilities introduced in Appendix B, which also tends to improve the readability of the code. Given such expected complexity we'll naturally find a stoichiometric table to be a great help in organizing the information provided in the problem statement:

Sp.	Formula	$\nu_{i,1}$	$\nu_{i,2}$	$n_{i0}$	$r_i$	$C_p$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )
A	C <sub>3</sub> H <sub>8</sub>	-2	0	$n_{A0}$	$-2r_1$	165	-104.7
B	C <sub>2</sub> H <sub>6</sub>	3	-1	0	$3r_1 - r_2$	116	-83.8
C	CH <sub>4</sub>	0	2	0	$2r_2$	68	-74.5
D	H <sub>2</sub>	-1	-1	$n_{D0}$	$-r_1 - r_2$	30	0
		$\delta_1 = 0$	$\delta_2 = 0$			$\Delta M_1 = -12$	-42.0
						$\Delta M_2 = -10$	-65.2

We'll also use a little bit of trickery from Section 6.4 to pass an extra input argument to the local function to control whether the reactor is adiabatic or jacketed, and within the local function we'll use a `switch` statement to control how the  $\delta Q$  term is evaluated. In this way we can get away with writing just one local function but calling it twice with two different inputs.

The adiabatic case is obviously  $\delta Q = 0$ , and for the jacketed case we need to calculate  $\delta Q$  according to Equation (7.19). The overall heat transfer coefficient  $U$  for gas on the tube side and phase change (boiling water) on the shell side is  $58.6 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}$  by Equation (4.27), but the area per volume available for heat transfer  $a$  is a little trickier.

The total volume of the reactor is

$$V_{\text{tot}} = N_t \left( \frac{1}{4} \pi d^2 L \right) = \frac{d}{4} (N_t \pi d L),$$

where  $N_t$  is the number of tubes,  $L$  is the length of a single tube, and  $d$  is the diameter of a single tube. The total surface area available for heat transfer is

$$A_{\text{tot}} = N_t (\pi d L) = \frac{4}{d} V_{\text{tot}},$$

where we've substituted the expression for  $V_{\text{tot}}$  after the second equality. We now calculate  $a$  according to Equation (7.16) as

$$a = \frac{dA_{\text{tot}}}{dV_{\text{tot}}} = \frac{4}{d},$$

where  $d$  is the diameter of a *single* tube. That's unusual given that we're dealing with a bank of tubes, and it's exactly what we found for a single tube in Equation (7.17)! Tricky!

All physical properties are taken from Table D.2, we use the heat capacity at 900 K since that's close to the feed, and we use the ideal gas law wherever molar or volumetric flow rates are necessary. As a refresher from your material and energy balance courses, the relevant percent excess calculation is

$$\begin{aligned} n_{T0} &= n_{A0} + n_{D0} = n_{A0} + (1 + \%_{xs}) n_{D0}^{st} \\ &= n_{A0} + (1 + \%_{xs}) \left( \frac{1}{2} n_{A0} \right) \\ &= \left( 1 + \frac{1}{2} (1 + \%_{xs}) \right) n_{A0} \end{aligned}$$

where  $n_{D0}^{st}$  is the stoichiometric amount of  $H_2$  needed per mole  $C_3H_8$  according to reaction (1), and  $\%_{xs}$  is the percent excess of D (here, 1.5 for 150%).

The script below produces plots of the three hydrocarbons and reactor temperature. Keep in mind that Appendix B was used to reduce the length of the code but all the same steps we've seen before—adjusting the volumetric flow rate, calculating enthalpies of reaction, finding net rates, and so forth—are all present. Study this example well because it represents a general, flexible approach that can be modified to match the needs of nearly any PFR problem.



```

1 % pfrNonIso.m
2 T0 = 550+273.15; % K
3 nT0 = 100; % mol/s
4 xsD = 1.5;
5 nA0 = nT0/(1+0.5*(1+xsD)); % mol/s
6 nD0 = nT0-nA0; % mol/s
7 Y0 = [nA0; 0; 0; nD0; T0]; % [C3H8 C2H6 CH4 H2 T]
8
9 Ntube = 137;
10 d = 0.02; % m
11 L = 14; % m
12 Vspan = linspace(0, Ntube*0.25*pi*d^2*L); % m3
13
14 [~,Yad]=ode45(@(V,Y)fun(V,Y,'adiabatic'),Vspan,Y0);
15 [~,Yjk]=ode45(@(V,Y)fun(V,Y,'jacketed'),Vspan,Y0);
16 subplot(2,2,1); plot(Vspan,[Yad(:,1),Yjk(:,1)]);
17 subplot(2,2,2); plot(Vspan,[Yad(:,2),Yjk(:,2)]);
18 subplot(2,2,3); plot(Vspan,[Yad(:,3),Yjk(:,3)]);
19 subplot(2,2,4); plot(Vspan,[Yad(:,5),Yjk(:,5)]);
20
21 function dYdV = fun(V,Y,type)
22 R = 8.314; % m3-Pa/mol-K = J/mol-K
23 T0 = 550+273.15; % K
24 P0 = 2.5e6; % Pa
25 nT0 = 100; % mol/s
26 n = Y(1:4); % mol/s
27 T = Y(5); % K
28 dHf = [-104.7; -83.8; -74.5; 0]*1e3; % J/mol
29 Cp = [165; 116; 68; 30]; % J/mol-K
30 nu = [-2 0; 3 -1; 0 2; -1 -1];
31 A = [0.05; 0.1]; % 1/s
32 Ea = [19.1; 25.6]*1e3; % J/mol
33
34 v0 = nT0*R*T0/P0; % m3/s
35 v = v0*(sum(n)/nT0)*(T/T0); % m3/s
36 C = n/v; % mol/m3
37 k = A.*exp(-Ea/(R*T)); % 1/s
38 dHref = nu' * dHf; % kJ/mol
39 dCp = nu' * Cp; % J/mol-K
40 dH = dHref + dCp*(T-298.15); % kJ/mol
41 r = [k(1)*C(1)*C(4); k(2)*C(2)*C(4)]; % mol/m3-s
42
43 switch type
44     case 'adiabatic'
45         dQ = 0;
46     case 'jacketed'
47         d = 0.02; % m

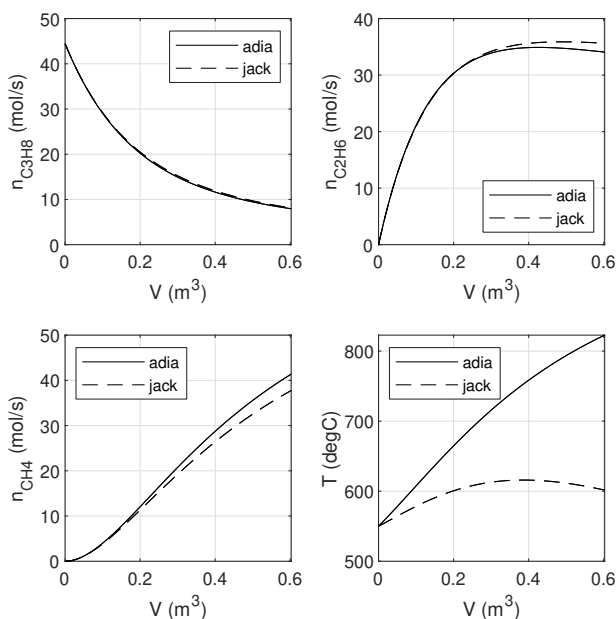
```

```

48     dQ = 58.6*(4/d)*(350+273.15-T); % J/m3-s
49 end
50
51 dndV = nu * r; % mol/s-m3
52 dTdV = (dQ - sum(r.*dH))/sum(n.*Cp); % K/m3
53 dYdV = [dndV; dTdV]; % [mol/s-m3; K/m3]
54 end

```

As shown in the figure below there's evidently no appreciable difference in the outlet flow rates of each species except that the adiabatic reactor produces slightly more short-chain hydrocarbons. However, there's a considerable difference between the outlet temperatures: the jacketed reactor temperature is more than 200 °C cooler than the adiabatic reactor. This will likely be a less expensive reactor to build because more common materials will be capable of operating at this temperature, and will be safer to operate at lower temperature and with the additional control afforded by the cooling water.



The code used in Example 7.6 is good to study because it uses several elements that were introduced in Chapter 6 and appendix B. Try to answer the following questions for yourself:

1. On Lines 14 and 15, why was the first output argument suppressed with the ~ symbol?
2. On Lines 14 and 15, why was @(V,Y) used instead of @(V,Y,type)?
3. On Lines 16 through 19, why were the variables Yad and Yjk able to be concatenated?  
Put another way, why did they have the same number of rows?

4. Write out the matrix algebra performed on Lines 38 and 51.
5. On Line 53, why is there only a single `dndV` instead of `dnAdV`, `dnBdV`, and so forth?

After you answer these questions you'll have a strong grasp of the advanced coding principles used here, but remember that you can always calculate values one-at-a-time and still come up with the right answer. It's not in any way *required* that you use the matrix algebra and other tricks that were used here, but they certainly make many operations more *convenient*.



# Chapter 8

## Special Cases of the PFR

“Yet I have now spent the better part of two years on the trail of the Internet’s physical infrastructure, following that wire from the backyard. I have confirmed with my own eyes that the Internet is many things, in many places. But one thing it most certainly is, nearly everywhere, is, in fact, a series of tubes.”

Andrew Blum, *Tubes: A Journey to the Center of the Internet* (2012)

### 8.1 Rewriting equations in terms of conversion can be useful

As we saw with CSTRs, it can sometimes be useful to express the material and energy balances for PFRs in terms of the conversion of a single species, A. This tends to simplify the integrals that need to be solved and can allow us to make comparisons of some quantities more easily. The outlet molar flow rate  $n_A$  is related to  $X_A$  through Equation (2.33) as

$$n_A = n_{A0}(1 - X_A), \quad (2.33)$$

from which we identify the derivative  $dn_A/dV$  as

$$\frac{dn_A}{dV} = -n_{A0} \frac{dX_A}{dV}. \quad (8.1)$$

Substituting Equation (8.1) into Equation (7.1) yields

$$\boxed{\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}}} \quad X_A(V=0) = 0 \quad (8.2)$$

Some of the great utility of Equation (8.2) comes from its initial condition because having one of the boundaries in an integral be zero usually simplifies the result of that integral considerably.

In order to use Equation (8.2) we also have to express  $r_A$  as a function of  $X_A$  which usually requires use of either

$$C_i = C_{A0} \left( \Theta_i - \frac{\nu_i}{\nu_A} X_A \right) \quad (\text{liquids}) \quad (5.11)$$

or

$$C_i = C_{A0} \frac{\Theta_i - \frac{\nu_i}{\nu_A} X_A}{\left( 1 - y_{A0} \frac{\delta}{\nu_A} X_A \right)^{\frac{P_0}{P} \frac{T}{T_0}}} \quad (\text{gases}) \quad (5.14)$$

within the rate expression. Given the complexity of Equations (5.11) and (5.14), hand calculations for PFRs involving  $X_A$  are limited to fairly simple, but nevertheless instructive, systems.

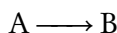
The PFR energy balance for a single reaction, Equation (7.13), can also be written in terms of  $X_A$  by using Equation (2.35) to write all  $n_i$  in terms of  $X_A$  to get

$$\frac{dT}{dV} = \frac{\delta Q - r \Delta H}{n_{A0} \left[ \sum_{i=A}^Z \Theta_i C_{P,i} - \frac{X_A}{\nu_A} \Delta C_P \right]} \quad T(V=0) = T_0 \quad (8.3)$$

where  $\Delta C_P \equiv \sum \nu_i C_{P,i}$  according to Equation (2.22). Although Equation (8.3) can be simplified quite a bit, the coupled solution of Equations (8.2) and (8.3) almost invariably requires solution by numerical approximation (e.g., by use of ode45).

### Example 8.1: Isothermal PFR sizing, revisited

In Example 7.1 a feed of pure A at  $3 \text{ L} \cdot \text{s}^{-1}$  was provided to an isothermal PFR in which the elementary, liquid-phase reaction



took place with  $k = 0.003 \text{ s}^{-1}$  (constant). Determine again the size of the PFR needed to convert 75% of the feed but this time use Equation (8.2) as the starting point for the material balance instead of Equation (7.1).

### Solution

Let's start again with the stoichiometric table:

Sp.	$\nu_i$	$n_{i0}$	$r_i$	$n_i$
A	-1	$n_{A0}$	$-r$	$n_{A0}(1-X_A)$
B	1	$n_{B0}$	$r$	$n_{A0}X_A$

The system is isothermal so we'll ignore the energy balance and focus only on the material balance for species A,

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}}, \quad X_{A0} = 0.$$

The rate expression is

$$r_A = -r = -kC_A = -kC_{A0}(1-X_A),$$

where Equation (5.11) was used in the last equality. Substituting  $r_A$  into the material balance yields

$$\frac{dX_A}{dV} = \frac{kC_{A0}(1-X_A)}{n_{A0}}$$

or, after noting  $n_{A0} = v_0C_{A0}$ ,

$$\frac{dX_A}{dV} = \frac{k}{v_0}(1-X_A).$$

This is a separable differential equation, so we can separate terms and integrate to get

$$\int_0^{X_A} \frac{dX_A}{(1-X_A)} = \int_0^V \frac{k}{v_0} dV,$$

which evaluates to

$$-\ln(1-X_A) = \frac{k}{v_0} V$$

or

$$V = -\frac{v_0}{k} \ln(1-X_A).$$

Given  $k = 0.003 \text{ s}^{-1}$ ,  $v_0 = 3 \text{ L} \cdot \text{s}^{-1}$ , and  $X_A = 0.75$ , we calculate the necessary volume

as

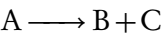
$$V = -\frac{3 \text{ L} \cdot \text{s}^{-1}}{0.003 \text{ s}^{-1}} (1 - 0.75) = 1386 \text{ L},$$

which matches exactly the volume found in Example 7.1.

Let’s also take a look at how non-isothermal PFRs can be set up using Equation (8.3), which although somewhat more straightforward in some senses than Equation (7.2) nonetheless requires simultaneous solution of coupled ODEs, and therefore also requires MATLAB.

**Example 8.2: Adiabatic decomposition in a PFR**

A feed of pure, gas-phase A at  $3 \text{ L} \cdot \text{s}^{-1}$ ,  $180^\circ\text{C}$ , and  $200 \text{ kPa}$  is fed to a  $100 \text{ L}$ , adiabatic, isobaric PFR. Within the reactor the elementary decomposition reaction



takes place. Calculate the conversion of A and outlet temperature of the reactor. Additional information is provided below.

$$k = 0.02 \text{ s}^{-1} \text{ at } 25^\circ\text{C}$$

$$E_a = 9.7 \text{ kJ} \cdot \text{mol}^{-1}$$

$$\Delta H = -13.7 \text{ kJ} \cdot \text{mol}^{-1} \text{ (const.)}$$

$$C_{p,\text{A}} = C_{p,\text{B}} = 80 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$$

$$C_{p,\text{C}} = 29 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$$

**Solution**

A stoichiometric table will help us organize the information given in the problem:

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$n_i$	$C_p$ ( $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ )
A	-1	$n_{\text{A}0}$	1	$-r$	$n_{\text{A}0}(1 - X_{\text{A}})$	80
B	1	0	0	$r$	$n_{\text{A}0}X_{\text{A}}$	80
C	1	0	0	$r$	$n_{\text{A}0}X_{\text{A}}$	29
$\delta = 1$						$\Delta C_p = 29$



The material balance on A is

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}}, \quad X_{A0} = 0$$

where the rate expression  $r_A$  is

$$r_A = \nu_A r = -k C_A,$$

the rate constant  $k$  is

$$k = 0.02 \exp \left[ \frac{E_a}{R} \left( \frac{1}{T_R} - \frac{1}{T} \right) \right] \quad \text{where } T_R = 25^\circ\text{C} = 298\text{ K},$$

the concentration of A according to Equation (5.14) is

$$C_A = C_{A0} \frac{1 - X_A}{(1 + X_A)} \frac{T_0}{T},$$

and the inlet molar flow rate of A is

$$n_{A0} = \frac{P_0 v_0}{RT_0}.$$

The system is adiabatic so  $\delta Q = 0$  in the energy balance, and  $\Delta H$  was noted as constant. Additional terms in the energy balance are

$$\sum_{i=A}^Z \Theta_i C_{P,i} = \Theta_A C_{P,A} = C_{P,A}$$

and

$$\Delta C_P = C_{P,C} + C_{P,B} - C_{P,A}$$

so that Equation (8.3) becomes

$$\frac{dT}{dV} = \frac{-r \Delta H}{n_{A0} [C_{P,A} - X_A \Delta C_P]}, \quad T_0 = 180^\circ\text{C}.$$

The material and energy balances are coupled ODEs which are readily solved using ode45 as shown in the script below.

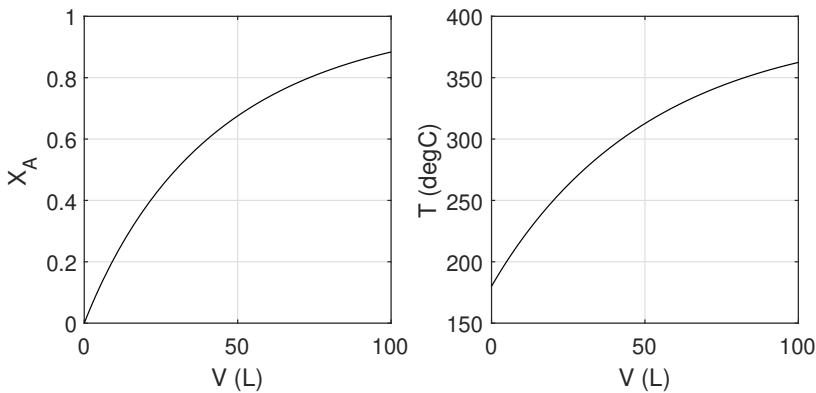
```
1 % pfrAdiaDecomp.m
```

```

2 Y0 = [0; 180+273]; % [Xa, T]
3 Vspan = [0 100]; % L
4 [V,Y] = ode45(@fun, Vspan, Y0);
5 subplot 121; plot(V, Y(:, 1))
6 subplot 122; plot(V, Y(:, 2)-273)
7
8 function dYdV = fun(V, Y)
9 Xa = Y(1);
10 T = Y(2); % K
11
12 v0 = 3; % L/s
13 T0 = 180+273; % K
14 P0 = 200; % kPa
15 R = 8.314; % kPa-L/mol-K, J/mol-K
16 Ea = 9.7e3; % J/mol
17 dH = -13.7e3; % J/mol
18 CpA = 80; % J/mol-K
19 dCp = 29+80-80; % J/mol-K
20
21 nA0 = P0*v0/(R*T0); % mol/s
22 Ca0 = nA0/v0; % mol/L
23 Ca = Ca0*(1-Xa)/(1+Xa)*T0/T; % mol/L
24 k = 0.02*exp(Ea/R*(1/298- 1/T)); % 1/s
25 r = k*Ca; % mol/L-s
26 rA = -r; % mol/L-s
27
28 dXadV = -rA/nA0; % 1/L
29 dTdV = -r*dH/(nA0*(CpA - Xa*dCp)); % K/L
30 dYdV = [dXadV; dTdV];
31 end

```

Running this script produces the plots shown below, from which we can identify the conversion of A as about 88.4% and the outlet reactor temperature as about 362 °C. As a comparison, try this problem using the general material and energy balances, Equations (7.1) and (7.2), to see which one you think is easier to set up and solve.



Also notice that it can be a little tricky to keep track of all the negative signs floating around, especially in the concentration terms and the energy balance. You'll have to develop your own habits for bookkeeping such terms, but you can use a little intuition in some places, like with the energy balance: the reaction is exothermic and takes place within an adiabatic (well-insulated) reactor, so we know the temperature has to go up.

## 8.2 Pressure drop occurs in PFRs

Pressure is the usual driving force that we look to whenever we want to move a fluid from one location to another, but whenever a fluid with non-zero viscosity is flowing some of its energy is lost due to friction. This loss manifests itself as a rise in temperature and a fall in pressure, and we call the fall in pressure the *pressure drop*. Pressure drop is important because it's directly related to how big a pump needs to be in order to provide a desired flow rate, and pumps are expensive to purchase, operate, and maintain.

A simple force balance<sup>1</sup> coupled with an adjustable coefficient will give us a useful expression for estimating pressure drops in PFRs, and that force balance begins with Figure 8.1. There are three forces acting on a cylindrical fluid element: the upstream force  $F_u$  pushes on the element in the direction of flow, and the downstream force  $F_d$  and drag force  $F_D$  push on the element in the direction opposite of flow. At steady state the force balance on the cylindrical element is

$$0 = F_u - F_d - F_D, \quad (8.4)$$

and if the pressure is uniform across the tube then both the upstream and downstream forces

<sup>1</sup>See Chapter 6 in Bird, R.B.; Stewart, W.E.; Lightfoot, E.N. Transport Phenomena, 2nd ed. John Wiley and Sons, New York (2002).

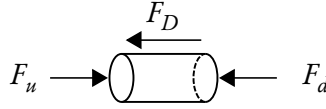


Figure 8.1: Force balance on a cylindrical fluid element.  $F_d$  and  $F_u$  are the upstream and downstream forces due to the surrounding fluid, and  $F_D$  is the drag force due to the wall.

can be represented as pressure times area, or

$$F_D = (P_u - P_d)A_c, \quad (8.5)$$

where  $A_c = \frac{1}{4}\pi d^2$  is the cross-sectional area of the cylinder (pipe),  $d$  is the diameter of the pipe,  $P_u$  and  $P_d$  are the upstream and downstream pressures, and  $F_D$  was moved to the left-hand side.

In order to make further progress we now have to think about what sorts of thing contribute to drag forces, and there are two: area (more area should mean more drag) and kinetic energy (higher velocities should mean more drag too). In true engineering fashion, we'll now assume that the drag force is equal to the product of area and kinetic energy, adjusted by an experimentally determined coefficient as

$$F_D = \left( \begin{array}{c} \text{area of} \\ \text{liquid- solid} \\ \text{contact} \end{array} \right) \times \left( \begin{array}{c} \text{characteristic} \\ \text{kinetic} \\ \text{energy} \end{array} \right) \times \left( \begin{array}{c} \text{fudge} \\ \text{factor} \end{array} \right) = A_w K f, \quad (8.6)$$

where  $A_w = \pi d L$  is called the *wetted area* and is the contact area between the moving fluid and stationary pipe,  $K = \frac{1}{2}\rho u^2$  is a characteristic kinetic energy<sup>2</sup> of a stream with density  $\rho$  flowing at average velocity  $u$ , and  $f$  is a non-dimensional, experimentally-determined coefficient called the *Fanning friction factor*.

Substituting Equation (8.6) into Equation (8.5) and rearranging for  $P_d$  yields

$$P_d = P_u - f K \frac{A_w}{A_c}, \quad (8.7)$$

and substituting for  $K$ ,  $A_c$ , and  $A_w$  as noted in the text above yields

$$P_d = P_u - 4f \left( \frac{1}{2}\rho u^2 \right) \frac{\pi d L}{\pi d^2}. \quad (8.8)$$

<sup>2</sup>You'll learn more about characteristic quantities in your fluid mechanics class. Briefly, it's a value whose magnitude establishes a reasonable scale for the quantity in that problem. In the case of fluid mechanics the quantity  $1/2\rho u^2$  occurs regularly.

We have to make a few substitutions in order to get Equation (8.8) into a form that plays well with the variables introduced in Chapter 7 for the plug flow reactor, namely that the length  $L$  down the reactor is proportional to the reactor volume  $V$  as

$$L = \frac{V}{A_c} = \frac{4V}{\pi d^2},$$

the average fluid velocity  $u$  is proportional to the volumetric flow rate  $v$  as

$$u = \frac{v}{A_c} = \frac{4v}{\pi d^2},$$

and the bulk density  $\rho$  is proportional to the total mass flow rate  $m_T$  and volumetric flow rate  $v$  as

$$\rho = \frac{m_T}{v}.$$

It's important to note that while  $v$  can vary according to Equation (5.5) in the case of gases, the total mass flow rate  $m_T$  is always *constant* regardless of fluid phase and the presence or absence of chemical reaction. Substituting these three relationships into Equation (8.8) and doing a bit of algebra yields

$$P_d = P_u - \left( v f \frac{128 m_T}{\pi^3 d^7} \right) V, \quad (8.9)$$

from which the derivative  $dP_d/dV$  can be identified as

$$\frac{dP_d}{dV} = -v f \frac{128 m_T}{\pi^3 d^7}. \quad (8.10)$$

Lastly, we note that our selection of the location at which we measure the downstream pressure  $P_d$  was arbitrary, so we replace  $P_d$  with the more familiar  $P$  to get

$$\boxed{\frac{dP}{dV} = -v f \frac{128 m_T}{\pi^3 d^7}}, \quad P(V=0) = P_0 \quad (8.11)$$

which is the equation we'll use to examine pressure drop effects in plug flow reactors. It's important to keep in mind that Equation (8.11) isn't a fundamental equation of fluid mechanics; it's an *empirical* expression which represents how and upon which variables we expect  $dP/dV$  to depend, multiplied by the experimentally-determined Fanning friction factor  $f$ .

The friction factor  $f$  usually ranges from about 0.001 to 0.1 and can be calculated from several different approximations you'll encounter in your fluid mechanics course, one of which<sup>3</sup>

<sup>3</sup>See Equation (6.2-12) in Bird, Stewart, and Lightfoot (*op. cit.*). Be careful not to confuse the Fanning friction

is

$$f = \frac{0.0791}{\text{Re}^{1/4}}, \quad (8.12)$$

where the variable  $\text{Re}$  is the *Reynolds number*,

$$\text{Re} = \frac{\rho u d}{\mu} = \frac{4m_T}{\pi d \mu}, \quad (8.13)$$

where  $\mu$  is the viscosity of the fluid, usually on the order of about  $10^{-3} \text{ Pa} \cdot \text{s}$  for liquids and  $10^{-5} \text{ Pa} \cdot \text{s}$  for gases. The Reynolds number is *always* dimensionless, which makes keeping track of units essential. We often assume the viscosity  $\mu$  in Equation (8.13) to be constant and therefore Equation (8.12) typically yields a constant friction factor. If the temperature change is more than a few hundred degrees celsius then this approximation should be revisited by evaluating the temperature dependence of  $\mu$ .

You should always keep in mind that Equation (8.12) is an *experimentally-observed* approximation: it's not derived from anything except experience.<sup>4</sup> In this case, Equation (8.12) is applicable for flows in “hydraulically smooth” pipes, where the size of surface imperfections is much less than the diameter of the pipe, and is valid up to about  $\text{Re} \sim 10^5$ .

To calculate pressure drop in PFRs we include Equation (8.11) for  $dP/dV$  in our system of coupled differential equations, using Equations (8.12) and (8.13) for  $f$  and  $\text{Re}$  if necessary, and then solve the system using `ode45`. Pressure drop exists for liquids and can be estimated using Equation (8.11) but such estimation is largely irrelevant for reacting systems because nearly all relevant parameters in liquids are independent of pressure—density, concentrations, rate constants, etc. Therefore Equation (8.11) will find application mainly in gas-phase systems.

### Example 8.3: Pressure drop in a PFR

Include the effects of pressure drop for the system described in Example 7.4 assuming a pipe diameter of 0.5 m and a reactor volume fixed at  $321.9 \text{ m}^3$ . Also assume the gas viscosity is approximately constant and equal to  $10^{-5} \text{ Pa} \cdot \text{s}$ . Compare the conversion and pressure profiles when the pressure drop is and is not included.

### Solution

factor with the Moody (or Darcy) friction factor. The Moody friction factor is, by convention, four times larger than the Fanning friction factor and commonly appears in emotional charts called “Moody diagrams.”

<sup>4</sup>And there's nothing wrong with that!

Let's focus first on the major changes we'll need to make to the solution script in Example 7.4:

1. Pressure is no longer constant, which means that  $v$  will vary along the length of the reactor.
2. The change in pressure along the length of the reactor is given by Equation (8.11), an ODE which will have to be solved simultaneously with the material balances.

Just as we did in Example 7.4, we begin by working backwards, first including the differential for pressure drop according to Equation (8.11) and then adding lines until each term in Equation (8.11) is defined. We're also working with a system of ODEs now, so we need the tools developed in Chapter 6 to handle such systems. With line numbers referring to the finished script in Example 7.4, the local function will change from

```
7 function dnAdV = fun(V, nA)
18 dnAdv = rA; % mol/m3-min
19 end
```

to a system of differential equations as

```
7 function dYdV = fun(V, Y)
8 nA = Y(1); % mol/min
9 P = Y(2); % kPa
20 dnAdv = rA; % mol/m3-min
21 dPdV = -v*f*128*mT/(pi^3 * d^7); % kPa/m3
22 dYdV = [dnAdv; dPdV];
23 end
```

Clearly we'll need additional lines to define the new variables  $f$ ,  $mT$ , and  $d$ , being careful about units<sup>a</sup> as we go about adding these variables to ensure that when we're all done the quantity  $dP/dV$  will have units of  $\text{kPa} \cdot \text{m}^{-3}$ . We'll also have to modify  $v$  from

```
13 v = v0; % m3/min
```

to include pressure effects according to Equation (5.5) as

```
13 v = v0*P0/P; % m3/min
```

Notice that we didn't include the  $T/T_0$  term because the system is isothermal, nor the  $n_T/n_{T0}$  term because  $\delta = 0$ , but inclusion of these two terms would have been a straightforward modification to Line 13 in a similar manner. The final change is to modify the call to `ode45` from

```

2 nA0 = 195.58; % mol/min
3 Vspan = [0 500]; % m3
4 [V, nA] = ode45(@fun, Vspan, nA0);

```

to a syntax for solving a system of ODEs as

```

2 Y0 = [195.58; 101.325]; % [nA0; P0], [mol/min; kPa]
3 Vspan = [0 321.9]; % m3
4 [V, Y] = ode45(@fun, Vspan, Y0);

```

Naturally there's still some work to do in terms of ensuring dimensional consistency everywhere, and we have to add the appropriate plotting commands. If we wanted to get fancy we could also include a `switch` statement as we did in Example 7.6 to control whether or not the pressure drop is calculated, but we could also simply run the script twice—once with  $dP/dV = 0$  and again with  $dP/dV$  defined by Equation (8.11)—and create the needed plots as we go. Only the script that calculates pressure drop is shown below.

```

1 % pfrPdrop.m
2 Y0 = [195.58; 101.325]; % [mol/min; kPa]
3 Vspan = [0 321.9]; % m3
4 [V, Y] = ode45(@fun, Vspan, Y0);
5 subplot 121; plot(V, 1-Y(:,1)/195.58);
6 subplot 122; plot(V, Y(:,2));
7
8 function dYdV = fun(V, Y)
9 nA = Y(1); % mol/min
10 P = Y(2); % kPa
11 k = 0.023; % 1/min
12 Kc = 1.14;
13 v0 = 10; % m3/min
14 nA0 = 195.58; % mol/min
15 P0 = 101.325; % kPa
16
17 % Pressure drop terms
18 d = 0.5; % m
19 mu = 1e-5; % Pa-s
20 mT = nA0*56e-3; % kg/min, const
21 Re = 4*mT/(pi*d*mu)/60; % non-dim
22 f = 0.0791/(Re^(1/4)); % non-dim
23
24 v = v0*P0/P; % m3/min
25 nB = nA0 - nA; % mol/min
26 Ca = nA/v; % mol/m3
27 Cb = nB/v; % mol/m3
28 rA = -k*(Ca-Cb/Kc); % mol/m3-min

```

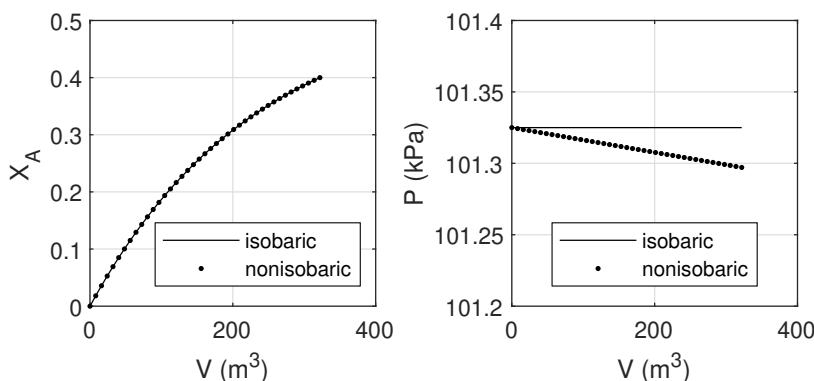


```

29 dnAdV = rA; % mol/m3-min
30 dPdV = -v*f*128*mT/(pi^3 * d^7); % kg/min2-m4
31 dPdV = dPdV/(60^2)/1000; % kPa/m3
32 dYdV = [dnAdV; dPdV];
33 end

```

Notice the unit conversions on Lines 20, 21, and 31: always be careful to check for dimensional consistency because the  $dPdV$  term can easily be off by many orders of magnitude. The completed plot is shown below and demonstrates why we often ignore pressure drop in PFRs: it's practically zero and has had no noticeable impact on conversion. Pressure drop can occasionally be important for shell-and-tube designs, but any tube larger than about the size of your hand will likely have a small pressure drop if the reactor isn't exceedingly long.



"You can choose any consistent set of units; it doesn't always have to be the set used here.

### 8.3 PFR jackets can be co-flow or counter-flow

When PFR heat exchangers were introduced in Section 7.3 we noted that the case of non-constant utility temperatures was one that requires enough effort to resolve that it would be left to another chapter, and we've now arrived at that chapter. The discussion that follows draws on the same energy balance introduced in Section 7.2 but is simplified considerably because there is no reaction in the utility fluid. For simplicity the discussion is restricted to a single, double-pipe heat exchanger as in Figure 8.2a, but the resulting equations require no modification for use with shell-and-tube heat exchangers as in Figure 7.2b.

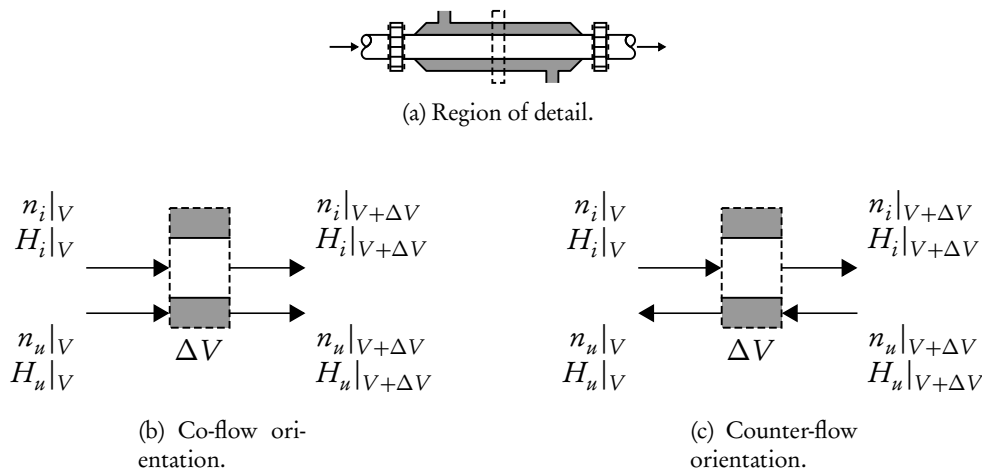


Figure 8.2: Analysis of a single, double-pipe heat exchanger. (a) Region of detail. The process fluid (white) flows from left to right; the utility fluid (gray) can flow in either the same or opposite direction of the utility fluid. (b) When the process and utility fluids flow in the same direction the orientation is called *co-flow*. (c) When the process and utility fluids flow in the opposite direction the orientation is called *counter-flow*.

### 8.3.1 Co-flow heat exchangers can be solved like any other PFR

When the process and utility fluid flow in the same direction, the heat exchanger is said to be operating in *co-flow* orientation. We approach the energy balance on such a system by analyzing a tiny slice across the reactor and heat exchanger together as shown in Figure 8.2b. Remember that the utility fluid is flowing through the annulus created by surrounding the process pipe with another pipe: when viewed in two dimensions—as it is in Figure 8.2b—it appears as though there is utility fluid “above” and “below” the process fluid, but this is simply a limitation of drawing a cylindrical object in two dimensions.

Recall from Section 7.2 that the general energy balance on the *process* fluid started in its general form as

$$0 = \sum_{i=A}^Z [n_i H_i]_V - \sum_{i=A}^Z [n_i H_i]_{V+\Delta V} + \delta Q \Delta V, \quad (7.7)$$

where the work term was omitted as is tradition. Implied in this expression is that energy is transferred by heat *from* the utility fluid *to* the process fluid, which is why the  $\delta Q \Delta V$  term is positive. That’s also consistent with how  $\delta Q$  was later defined as

$$\delta Q = Ua(T_u - T), \quad (7.19)$$

from which we see that  $\delta Q > 0$  when  $T_u > T$  as expected.

The energy balance on the *utility* fluid in co-flow orientation begins from a similar form as Equation (7.7) but the development is greatly simplified because there's only one component and no chemical reaction in the utility stream. The corresponding energy balance for the utility stream is therefore

$$\begin{aligned} 0 &= \text{In} - \text{Out} \\ &= [n_u H_u]_V - [n_u H_u]_{V+\Delta V} - \delta Q \Delta V, \end{aligned} \quad (8.14)$$

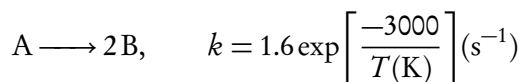
where  $n_u$  is the utility flow rate,  $H_u$  is the utility enthalpy, and  $\delta Q \Delta V$  is the same small amount of energy added by heat transfer that we saw in Equation (7.7) except with a *negative* sign because—simply by the way we've organized the problem—the energy is *leaving* the utility fluid and *entering* the process fluid. Following the same general approach as in Section 7.2, and assuming that  $n_u$  and  $C_{P,u}$  are constant, the energy balance on the utility stream in co-flow orientation becomes

$$\boxed{\frac{dT_u}{dV} = \frac{-\delta Q}{n_u C_{P,u}}} \quad T_u(0) = T_{u0} \quad (\text{co-flow}) \quad (8.15)$$

where  $\delta Q$  is given by Equation (7.19) above. The molar utility flow rate  $n_u$  can be replaced with a mass flow rate  $m_u$  as long as the units of  $C_{P,u}$  are modified appropriately. Equation (8.15) is an additional ODE that must be solved as part of the system of ODEs for the process fluid.

#### Example 8.4: Co-flow heat exchange in a PFR

The irreversible, elementary, liquid-phase decomposition of a hydrocarbon



takes place in a shell-and-tube PFR constructed from a series of 5 cm tubes with a total volume of  $0.8 \text{ m}^3$ . Pure A is provided to the reactor at  $0.11 \text{ kmol} \cdot \text{s}^{-1}$  and  $125^\circ\text{C}$ ; the residence time is 1 min. Steam at  $200^\circ\text{C}$  and  $0.25 \text{ kg} \cdot \text{s}^{-1}$  provides heat for the endothermic decomposition ( $\Delta H = 22.9 \text{ kJ} \cdot \text{mol}^{-1}$ , constant). The heat capacities of A and B are  $180$  and  $90 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ , and the heat capacity of steam is  $2 \text{ kJ} \cdot \text{kg}^{-1} \cdot ^\circ\text{C}^{-1}$ .

Plot the conversion, reactor temperature, and jacket temperature profiles for this reactor.

Solution

There's only one reaction and the problem asked us to plot conversion, so this is a good problem to practice using the material and energy balances written for conversion, Equations (8.2) and (8.3). We'll start a stoichiometric table, then a material balance for species A, followed by the energy balance for the process stream, and finish up with the energy balance for the utility fluid.

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$C_i$	$C_p$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )
A	-1	0.11	1	-r	$C_{A0}(1 - X_A)$	180
B	1	0	0	r	$2C_{A0}X_A$	90

$\Delta C_p = 0$

Material balance on A

The material balance is

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}} = \frac{r}{n_{A0}},$$

where  $n_{A0} = 0.11 \text{ kmol} \cdot \text{s}^{-1}$ , the rate expression is

$$r = kC_A = 1.6C_{A0}(1 - X_A)\exp\left[\frac{-3000}{T}\right],$$

and the inlet molar concentration of A is

$$C_{A0} = \frac{n_{A0}}{v_0} = \frac{n_{A0}\tau}{V},$$

where  $V = 0.8 \text{ m}^3$ .

Energy balance on process stream

The energy balance, after appropriate simplifications, is

$$\frac{dT}{dV} = \frac{\delta Q - r\Delta H}{n_{A0}C_{p,A}},$$

where  $\Delta H = 22.9 \text{ kJ} \cdot \text{mol}^{-1}$ ,  $C_{p,A} = 180 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ , and the change in heat capacity is

$$\Delta C_p = 2C_{p,B} - C_{p,A} = 2(90) - 180 = 0,$$

which is why the denominator in the energy balance is considerably simpler than usual. The heat transfer term is

$$\delta Q = Ua(T_u - T).$$

and Equation (4.27) provides an estimate of the overall heat transfer coefficient for a steam-hydrocarbon pairing as

$$U = \left[ \frac{1}{1000} + \frac{1}{60} \right]^{-1} = 56.6 \text{ W} \cdot \text{m}^{-2} \cdot \text{K}^{-1}.$$

Finally, Equation (7.17) provides  $a$  for a cylindrical PFR as

$$a = \frac{4}{d} = \frac{4}{0.05 \text{ m}} = 80 \text{ m}^{-1},$$

where we've used the diameter  $d$  of a single tube (see Example 7.6). Notice that this is one of the advantages of the shell-and-tube design: the heat transfer is increased because  $d$  is generally smaller than for a double-pipe arrangement.

### Energy balance on utility stream

The energy balance for the utility stream is given by Equation (8.15),

$$\frac{dT_u}{dV} = \frac{-\delta Q}{m_u C_{P,u}},$$

subject to the initial condition  $T_u(0) = T_{u0} = 200^\circ\text{C}$ . The flow term  $n_u$  in the denominator was replaced with  $m_u$  since the utility flow was given on a mass basis, as was its heat capacity. The heat term  $\delta Q$  is exactly the same as defined above for the process stream.

### Solution and analysis

```

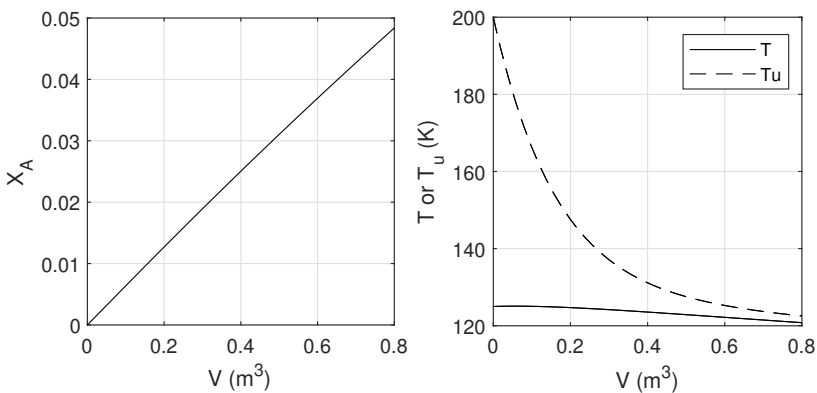
1 % coflow.m
2 Y0 = [0; 125+273; 200+273]; % [Xa0; T0; Tu0]
3 Vrange = [0, 0.8]; % m3
4 [V, Y] = ode45(@fun, Vrange, Y0);
5 subplot 121; plot(V, Y(:,1))
6 subplot 122; plot(V, Y(:,2:3)-273)
7
8 function dYdV = fun(V, Y)
9 Xa = Y(1);
10 T = Y(2); % K
11 Tu = Y(3); % K
12 nA0 = 0.11e3; % mol/s

```

```

13 tau = 60; % s
14 V = 0.8; % m3
15 dH = 22.9e3; % J/mol
16 CpA = 180; % J/mol-K
17 mu = 0.25; % kg/s
18 Cpu = 2e3; % J/kg-C
19
20 Ca0 = nA0*tau/V; % mol/m3
21 Ca = Ca0*(1-Xa); % mol/m3
22 k = 1.6*exp(-3000/T); % 1/s
23 r = k*Ca; % mol/m3-s
24
25 U = 1/(1/100+1/60); % W/m2-K
26 a = 4/0.05; % 1/m
27 dQ = U*a*(Tu-T); % J/s-m3
28
29 dXadV = r/nA0; % 1/m3
30 dTdV = (dQ-r*dH)/(nA0*CpA); % K/m3
31 dTudV = -dQ/(mu*Cpu); % K/m3
32 dYdV = [dXadV; dTdV; dTudV];
33 end

```



There are several interesting regions in the reactor: initially the steam temperature is high, leading to a large  $\delta Q$  and an increase in process stream temperature. The steam temperature rapidly falls, however, until the cooling effect of the endothermic reaction exceeds the heat transferred to the process stream and the reactor temperature falls. Near the end of the reactor the process and utility streams approach thermal equilibrium at about 120 °C.

There are quite a few interesting design questions that can be answered with small changes to the script above. See if you can figure out how to answer the following questions by chang-

ing no more than *one* line of the script in Example 8.4 (*answers are provided in the footnotes*):

1. What happens if the system is operated adiabatically? <sup>5</sup>
2. Would constant utility temperature be beneficial? <sup>6</sup>
3. What utility flow rate is needed for approximately constant utility temperature? <sup>7</sup>
4. What happens if the system is operated as a heat exchanger (no reaction) instead of a reactor? <sup>8</sup>

Another option may have occurred to you: what happens if the direction of the utility stream is reversed, from co-flow to counter-flow? That's an excellent suggestion but requires a bit more effort to evaluate!

### 8.3.2 Counter-flow heat exchangers require an iterative approach

To answer the question posed above regarding utility flow in the opposite direction of process flow we have to go back to the energy balance, which starts with Figure 8.2c. The a counter-flow orientation is nearly identical to what we saw for the co-flow orientation, with two significant changes: in the derivation the *inlet* terms for the utility are now those evaluated at  $V + \Delta V$  and the *outlet* terms are now those evaluated at  $V$ , and we now know  $T_u$  at the end of the reactor (at  $V$ ) instead of at the entrance to the reactor (at  $V = 0$ ). The corresponding energy balance for the utility stream is therefore

$$0 = \text{In} - \text{Out} \\ = [n_u H_u]_{V+\Delta V} - [n_u H_u]_V - \delta Q \Delta V, \quad (8.16)$$

which is similar to Equation (8.14) except the signs on the utility enthalpy terms are now reversed. Notice that the sign on the  $\delta Q$  terms is unchanged because we still approach the problem as though heat is transferred *from* the utility *to* the process fluid.

The resulting differential equation for the energy balance on the utility stream in counter-flow orientation becomes

$$\boxed{\frac{dT_u}{dV} = \frac{\delta Q}{n_u C_{P,u}}} \quad T_u(V) = T_{u0} \quad (\text{counter-flow}) \quad (8.17)$$

<sup>5</sup>Set  $dQ=0$  on Line 27.

<sup>6</sup>Set  $dT_{ud}V=0$  on Line 31.

<sup>7</sup>Set  $\mu_u$  on Line 17 high enough to see almost no change in  $T_u$ .

<sup>8</sup>Set  $r=0$  on Line 23.

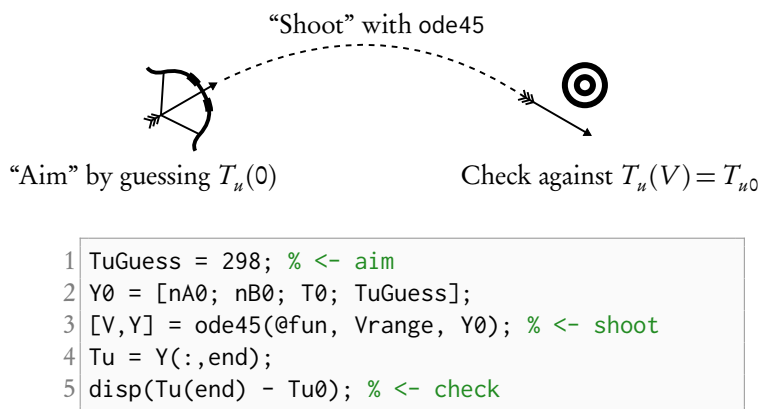


Figure 8.3: Basic steps of the “shooting” method for counter-flow PFR heat exchangers: aim, shoot, check. The image on top is the conceptual origin of the phrase “shooting method” and the pseudo-code underneath provides a basic structure for implementing the shooting method in a MATLAB script.

Importantly, the “initial” condition has also changed: we now know  $T_u$  at the *end* of the reactor (at  $V$ ) to be  $T_{u0}$  because the utility fluid enters from the opposite reactor side compared to where the process fluid enters. We’re now in a sticky situation because `ode45` requires knowledge of all dependent variables at the *entrance* to the reactor (at  $V = 0$ ), yet we don’t know  $T_u(V = 0)$ .

The solution is an iterative approach called the *shooting method*, so named because of the steps involved.<sup>9</sup> There are three steps to the method that are summarized in Figure 8.3:

**Aim:** Guess a value of  $T_u$  at  $V = 0$  and use this in the vector of initial conditions,  $Y0$ .

**Shoot:** Use `ode45` to solve the system as usual.

**Check:** Compare the calculated value of  $T_u$  at the end of the reactor to the known inlet utility temperature,  $T_{u0}$ . If they’re close then the method stops; if not then the method repeats with a new guess.

The agreement between the calculated value at  $Tu(end)$  and the known value  $Tu0$  can be considered acceptable by whatever criteria seems reasonable for the problem, but in most cases agreement of better than  $1^\circ\text{C}$  is acceptable for manual iteration (*i.e.*, for cases where you’re manually entering guesses, running the script, and making new guesses). If high accuracy is required then the process can be modified to work with `fsolve` (see Section 1.5), which automates the guessing procedure.

<sup>9</sup>It would be entirely appropriate to call the shooting method a *guess-and-check* method, but evidently whoever came up with “shooting method” thought it sounded more impressive.



**Example 8.5: Counter-flow heat exchange in a PFR**

Repeat Example 8.4 but with a counter-flow orientation for the heat exchanger. Show a few iterations of the shooting method and continue until convergence to less than 1 °C. Compare the conversion and temperature profiles of each reactor.

**Solution**

There are only a few changes which need to be made to the solution script from Example 8.4:

- The first few lines of the script need to be changed to the shooting method as outlined in Figure 8.3.
- On Line 31, the sign needs to be switched.

The script below demonstrates these changes, with large parts of unchanged code omitted for clarity.

```

1 % counterflow.m
2 TuGuess = 150; % degC
3 Y0 = [0; 100+273; TuGuess+273]; % [Xa0; T0; Tu0]
4 Vrange = [0, 0.8]; % m3
5 [V, Y] = ode45(@fun, Vrange, Y0);
6 fprintf('Tu(V) = %.1f C\n', Y(end,end)-273);
7
8 function dYdV = fun(V, Y)
31 dTudV = dQ/(mu*Cpu); % K/m3
33 end

```

This script allows us to execute the shooting method by guessing values for  $T_u(V=0)$  on Line 2, running the script, and inspecting the value of  $T_u(V)$  (at the end of the reactor). Ideally the displayed value will be close to 200 °C because that's the temperature at which the steam enters the jacket. Running the script above with a guessed value of 150 °C for  $T_u(V=0)$  produces the following output:

```

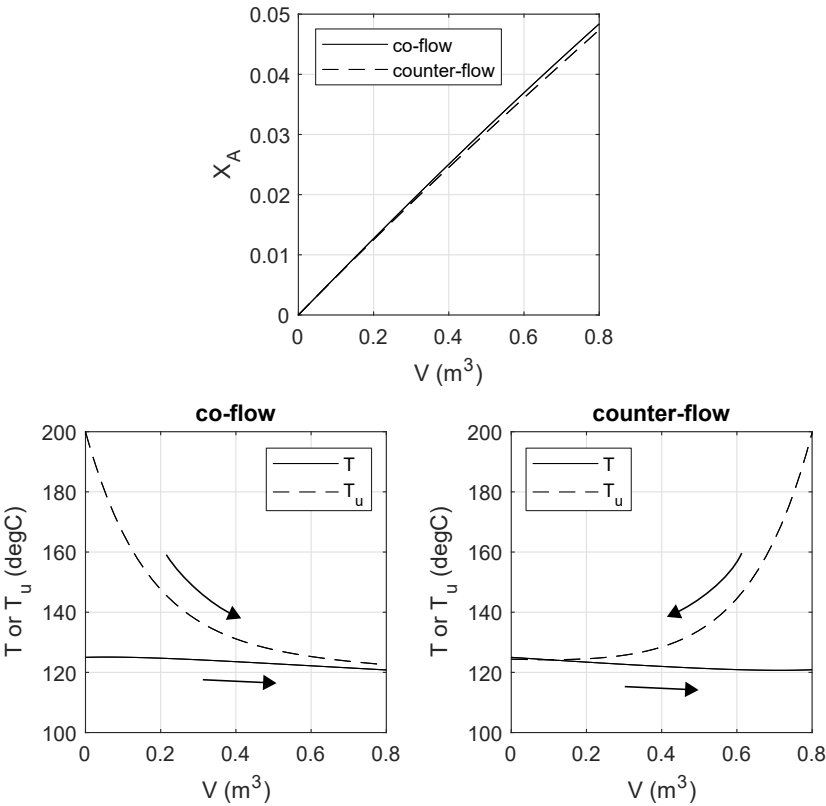
1 >> counterflow
2 Tu(V) = 3031.2 C

```

Evidently 3031.2 °C is quite a bit off from 200 °C, so we need to guess again. The table below shows a series of guesses (iterations) which show fairly rapid convergence to the solution value near 125 °C:

Iteration	TuGuess	Tu(V)
1	150	3031.2
2	100	-2498.5
3	125	266.0
$\vdots$	$\vdots$	$\vdots$
9	124.4	199.7

With just nine guesses the agreement improved to less than 1 °C, which is close enough for most applications and therefore we stop guessing. It’s important to keep in mind what we just calculated: the steam *enters* the reactor at  $V$  with a temperature of 200 °C, flows in the opposite direction of the process stream, and *exits* the reactor at  $V = 0$  with a temperature of about 124.4 °C. The requested profiles are shown in the figure below.



Evidently the counter-flow orientation provides lower conversion than the co-flow, but for this particular application the differences are small. The arrows on the temperature profile plots indicate the direction of flow: the co-flow plot is identical to the one shown

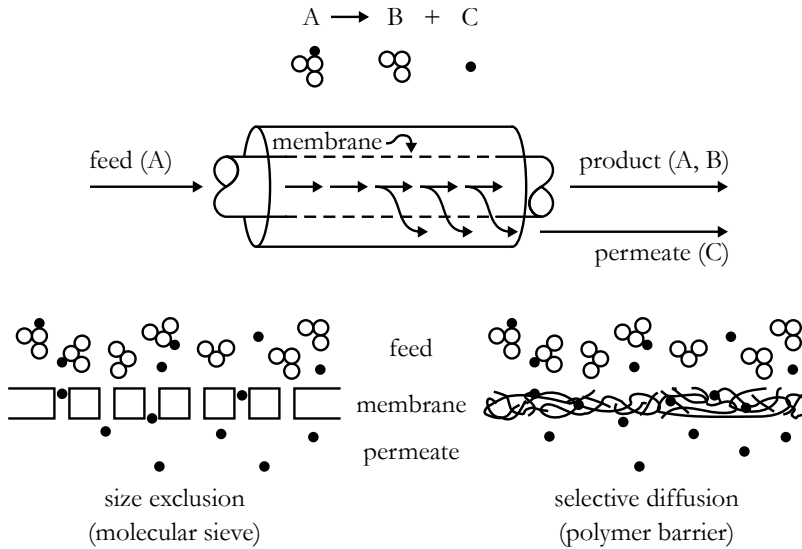


Figure 8.4: Schematic of a PFR membrane reactor. (*top*) Double-pipe arrangement with a permeable membrane separating the feed stream from the permeate stream. (*bottom*) Membranes often function by size exclusion—also called a molecular sieve—or by selective diffusion—typically through a polymer slab.

in Example 8.4 but the counter-flow plot has opposite flow orientation. As expected, the known steam temperature of 200 °C occurs at 0.8 m<sup>3</sup> for the counter-flow orientation, as opposed to occurring at 0 m<sup>3</sup> for the co-flow orientation.

Interestingly, the two temperature profiles for the counter-flow orientation cross near the entrance to the reactor. Can you explain how this is possible? Is it possible for the temperature profiles to cross in co-flow orientation?

## 8.4 Membrane reactors are PFRs with permeable walls

The last special case of the PFR to be analyzed here is the *membrane reactor* sketched in Figure 8.4. The structure is similar to the double-pipe heat exchanger from Figure 7.2a: A inner, feed stream pipe is surrounded by an outer pipe, and one species can selectively cross the barrier which separates the inner pipe from the outer pipe. This barrier is called a *membrane* and depending on the desired operation the membrane can allow species in the process stream to exit to the permeate stream, or it can allow fresh reactant to enter the process stream along the length of the reactor. Two commons methods of constructing a membrane are by *size exclusion*—pores in solid structures are shaped such that only certain molecules can pass, also called a molecular sieve—and *selective diffusion*—one molecule selectively adsorbs to and diffuses

through a solid slab, often a polymer or metallic barrier.

A useful model for describing the rate  $R_i$  at which a species  $i$  crosses from the process to the permeate stream defines a driving force as the concentration difference between the process and permeate streams, or

$$R_i = k_c (C_i - C_i^{\text{perm}}), \quad (8.18)$$

where  $C_i^{\text{perm}}$  is the concentration of  $i$  in the permeate stream, and  $k_c$  is an experimentally-determined constant called the *diffusion coefficient*. Each species will have its own  $k_c$  but for simplicity only one species is assumed to cross the membrane in any significant amount, so no subscript is placed on  $k_c$ . The units of  $R_i$  are the same as  $r_i$ , amount of  $i$  per unit time per unit volume.

The material balance for species  $i$  must include the effects of  $R_i$ , and as defined in Equation (8.18) or Equation (8.20) the term  $R_i$  is taken as positive when the species is exiting the process stream, so the material balance on the exiting species becomes

$$\boxed{\frac{dn_i}{dV} = r_i - R_i} \quad (8.19)$$

The material balance on all other species are unchanged from Equation (7.1), and no boundary conditions are changed. Often the concentration of  $i$  in the permeate stream is kept small enough to safely assume  $C_i \gg C_i^{\text{perm}}$  so that the driving force is entirely governed by the concentration in the process stream, or

$$R_i \approx k_c C_i \quad (i \text{ removed}) \quad (8.20)$$

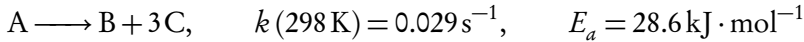
If species  $i$  is instead *fed* rather than *removed* by the membrane then the concentration in the permeate stream is kept large enough to safely assume  $C_i \ll C_i^{\text{perm}}$  so that the driving force is entirely governed by the permeate concentration, or

$$R_i \approx -k_c C_i^{\text{perm}} \quad (i \text{ fed}) \quad (8.21)$$

An energy balance on the membrane reactor is certainly possible but presents additional challenges because mass transfer across the membrane also leads to energy transfer. This situation is complicated enough that it's omitted from this discussion; within this text any membrane reactor can be assumed to operate isothermally and isobarically.

**Example 8.6: Dehydrogenation in a membrane reactor**

Hydrogen is one of the easier molecules to separate with a membrane because it's considerably smaller than most other molecules. Suppose the dehydrogenation of cyclohexane ( $\text{C}_6\text{H}_{12}$ , A) to benzene ( $\text{C}_6\text{H}_6$ , B) and hydrogen ( $\text{H}_2$ , C) proceeds quickly to equilibrium according to the elementary reaction



within an isothermal, isobaric, 40 L membrane reactor at 575 K and 300 kPa. Hydrogen is removed by a nanoporous membrane according to Equation (8.20) with  $k_c = 0.3\text{ s}^{-1}$ .

Compare the molar flow rate profiles of the membrane reactor to an unmodified PFR on the basis of pure A fed to the reactor at  $1\text{ mol} \cdot \text{s}^{-1}$ .

**Solution**

Naturally we organize the information with a stoichiometric table to get started:

Sp.	Formula	$\nu_i$	$n_{i0}$	$r_i$	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )	$\Delta G_f^\circ$ (kJ · mol <sup>-1</sup> )
A	$\text{C}_6\text{H}_{12}$	-1	1	$-r$	-123.3	31.9
B	$\text{C}_6\text{H}_6$	1	0	$r$	82.9	129.6
C	$\text{H}_2$	3	0	$3r$	0	0
		$\delta = 3$		$\Delta M = 206.2$		97.7

Now let's establish the material balances and work our way from there. Cyclohexane (A) and benzene (B) both follow the usual PFR material balance of Equation (7.1) because they're not involved with the membrane, but hydrogen (C) will follow Equation (8.20) because it's exiting the process stream through the reactor. Therefore we have to solve the system of ODEs defined as

$$\begin{aligned} \frac{dn_A}{dV} &= r_A = \nu_A r & n_A(0) &= 1\text{ mol} \cdot \text{s}^{-1} \\ \frac{dn_B}{dV} &= r_B = \nu_B r & n_B(0) &= 0 \\ \frac{dn_C}{dV} &= r_C - R_C = \nu_C r - R_C & n_C(0) &= 0 \end{aligned}$$

The elementary, reversible rate law is

$$r = k \left( C_A - \frac{C_B C_C^3}{K_C} \right),$$

where

$$k = 0.029 \exp \left[ \frac{28.6 \text{ kJ} \cdot \text{mol}^{-1}}{R} \left( \frac{1}{298 \text{ K}} - \frac{1}{575 \text{ K}} \right) \right] = 7.542 \text{ s}^{-1}.$$

The reactor is isothermal and isobaric so there's no difference between  $T$  and  $T_0$  or  $P$  and  $P_0$ , but we shouldn't expect  $n_T = n_{T0}$  because (a)  $\delta \neq 0$  and (b) there's a membrane selectively removing one of the species. The concentration of each species can therefore be calculated as  $C_i = n_i/v$  where

$$v = v_0 \frac{n_T}{n_{T0}} = \frac{n_{T0} R T_0}{P_0} \frac{n_T}{n_{T0}} = 15.94 n_T \quad (\text{L} \cdot \text{s}^{-1}).$$

Since  $\delta \neq 0$  the equilibrium constant  $K_C$  is a little trickier: the equilibrium constant  $K_a$  according to Equation (2.29) is

$$K_a \approx K_{sc} = K_0 K_1 = 1.892,$$

and the equilibrium constant  $K_C$  according to Equation (3.25) is

$$K_C = K_a \left( \frac{P^\circ}{RT} \right)^\delta = 1.892 \left( \frac{101.325 \text{ kPa}}{R \times 575 \text{ K}} \right)^3 = 1.8 \times 10^{-5} \text{ mol}^3 \cdot \text{L}^{-3}.$$

As was pointed out in Section 3.7.2,  $K_a$  is always dimensionless but if  $\delta \neq 0$  then  $K_C$  is not dimensionless.

All other parameters are known so we can construct our solution script as follows:

```

1 % membrane.m
2 Y0 = [1 0 0]; % [A B C], mol/s
3 Vspan = [0 40]; % L
4 [V,Y] = ode45(@fun, Vspan, Y0);
5 plot(V,Y)
6
7 function dYdV = fun(V,Y)
8 n = Y; % [A B C], mol/s
9 k = 7.542; % 1/s
10 Kc = 1.8e-5; % (mol/L)^3
11 kc = 0.3; % 1/s

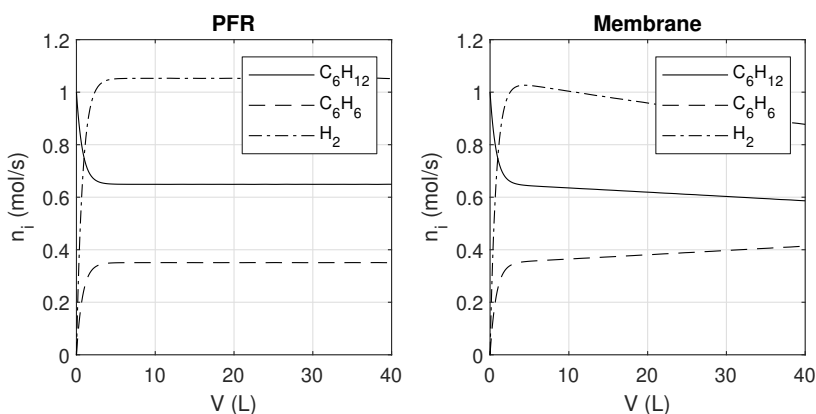
```

```

12
13 v = 15.94*sum(n); % L/s
14 C = n/v; % mol/L
15 r = k*(C(1) - C(2)*C(3)^3/Kc); % mol/L-s
16 Rc = kc*C(3); % mol/L-s
17
18 dnAdV = -r;
19 dnBdV = r;
20 dnCdV = 3*r - Rc;
21 dYdV = [dnAdV; dnBdV; dnCdV];
22 end

```

To create a plot for the standard PFR we can simply change Line 16 to  $R_c = 0$  to “stop” transport across the membrane, which is how the plots below were generated.



From the standard PFR profiles we see that the reaction rapidly approaches equilibrium: within about 5 L the net forward reaction essentially halts. From the membrane reactor profiles we see the effects and timescales associated with transport of hydrogen across the membrane: as hydrogen is removed the equilibrium constant shifts towards the products, so more cyclohexane is consumed and more benzene is produced.

The hydrogen removal process is slow, however: the remaining 90% of the reactor (about 35 L) increases the benzene output from about  $0.35 \text{ mol} \cdot \text{s}^{-1}$  to  $0.41 \text{ mol} \cdot \text{s}^{-1}$ , an increase of about 20%. On the other hand, an increase of 20% is fantastic!





## Chapter 9

### The Packed Bed Reactor, or PBR

“Whoa, déjà vu.”

Neo, *The Matrix* (1999)

At long last we come to the last of our four primary reactor designs, the *packed bed reactor* or PBR. As is tradition we start with the general form of the material balance for a PBR, which is

$$\boxed{\frac{dn_i}{dW} = r'_i} \quad n_i(W=0) = n_{i0} \quad (9.1)$$

and the general form of the energy balance for the PBR is

$$\boxed{\frac{dT}{dW} = \frac{\delta Q' - \sum_{j=1}^q r'_j \Delta H_j}{\sum_{i=A}^Z n_i C_{P,i}}} \quad T(W=0) = T_0 \quad (9.2)$$

The quote above was not chosen as a joke:<sup>1</sup> the material and energy balances for the PBR should look remarkably similar to the material and energy balances for the PFR, Equations (7.1) and (7.2). Indeed the only apparent difference is to replace  $V$  with  $W$  and put prime marks on  $r_i$  and  $\delta Q$ , and such a minor modification to the PFR equations should tell you two things:

1. It's easy to remember the PBR design equations because they're almost identical to the PFR design equations, and

---

<sup>1</sup>Well, not *entirely* as a joke.

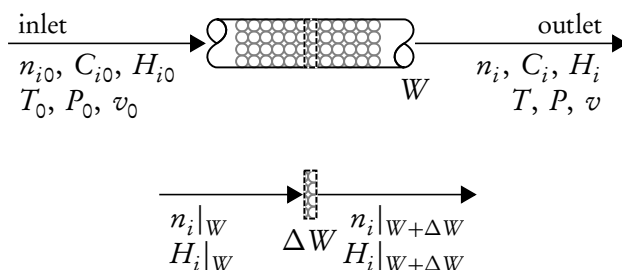


Figure 9.1: Basic diagram and variables associated with the PBR; heat  $Q$  and shaft work  $W_s$  are omitted for clarity. *(top)* Macroscale of the full reactor containing catalyst weight  $W$ . The small circles indicate catalyst particles. *(bottom)* Detail of infinitesimal slice of catalyst weight  $\Delta W$  beginning after the fluid has been exposed to any arbitrary weight  $W$ .

2. The problems and solutions for PBRs are going to look quite similar to what we saw with PFRs.

Especially because of the second point, you'll notice in this chapter and the next that the steps of derivations in this chapter follow closely the steps we took in Chapters 7 and 8, so if you're reading this book from cover to cover then you might be able to skip the derivations that seem familiar. All the steps are repeated in case you're *not* reading the book from cover to cover<sup>2</sup> and are just interested in the packed bed reactor. You'll also notice that the structures of the two chapters are nearly identical: in many section headers, for example, the acronym "PFR" is simply replaced with "PBR" because the major design elements are identical.

## 9.1 The PBR material balances are ordinary differential equations

A basic illustration of the packed bed reactor is shown in Figure 9.1 and it should look quite similar to the basic illustration of the plug flow reactor in Figure 7.1. The reason they're similar is no secret: if you fill up a plug flow reactor with catalyst particles then you've got yourself a packed bed reactor! Sometimes to serve as a contrast to the fluidized bed reactor from Section 5.1, a packed bed reactor is also called a *fixed bed reactor*.

PBRs, like PFRs, are equally suited to liquid- and gas-phase reactions. The difference is of course the catalyst: PBRs are used *only* when a solid catalyst particle is required. In most cases when we say *catalyst particle* what we really mean is that micro- or nano-sized precious metal catalysts have been deposited on millimeter- to centimeter-sized structures called *catalyst support*, as shown in Figure 9.2. The support material is often a porous, ceramic material with

<sup>2</sup>Gasp!

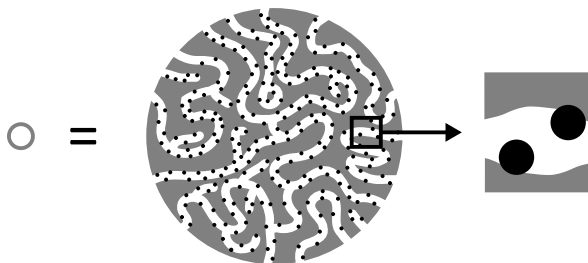


Figure 9.2: Detail of a catalyst particle. The small gray circle (*left*) represents, in most cases, a highly porous structure (*center*). Within the porous structure micro- or nano-scale catalyst particles are attached to the walls (*right*). Within this text the word “catalyst” will be used to refer to the combined structure of catalyst and support.

a large surface area and good thermal and physical properties to withstand the rigors of high temperature, high pressure reactions. For the purposes of our discussion the difference is immaterial and we’ll simply refer to the combined catalyst and support object as “catalyst.”

To develop the material balance for the PBR let’s again start with the infinitesimal slice of reactor volume in the bottom part of Figure 9.1. When we draw this imaginary control volume to have a volume  $\Delta V$ , we also enclose a corresponding amount of catalyst  $\Delta W$  and the two quantities are related as

$$\Delta W = \rho_{\text{bed}} \Delta V, \quad (9.3)$$

where  $\rho_{\text{bed}}$  is called the *bed density* and has units of

$$\rho_{\text{bed}} \doteq \frac{\text{catalyst bed mass (kg, lb, etc)}}{\text{reactor volume (m}^3, \text{ft}^3, \text{etc)}} \quad (9.4)$$

The total weight of catalyst is related to the total reactor volume in almost the same way as

$$W = \rho_{\text{bed}} V \quad (9.5)$$

where we use the phrase “*weight* of catalyst” to mean “*mass* of catalyst” because that’s become the unfortunate trend in the field.

Notice that  $\rho_{\text{bed}}$  is *not* the same as the density of the catalyst particle itself, which we denote as  $\rho_{\text{cat}}$ . For example, suppose the catalyst particles are made of a ceramic material with  $\rho_{\text{cat}} = 2000 \text{ kg} \cdot \text{m}^{-3}$ . If spheres of this ceramic material are dumped into a cylindrical reactor there will be gaps between adjacent spheres such that no more than about 75% of the vessel’s volume is taken up by the ceramic spheres, the remainder being empty space. If the catalyst shape is not spherical or if the catalyst isn’t tightly packed within the reactor volume then the

fraction could be less than 50%.

We refer to the fraction of space taken up by the process stream as the *void fraction* and denote it with the symbol  $\phi$  such that

$$\rho_{\text{bed}} = (1 - \phi)\rho_{\text{cat}} \quad (9.6)$$

If 500 kg of the aforementioned catalyst with  $\rho_{\text{cat}} = 2000 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}$  has a void fraction of  $\phi = 0.6$  then the volume of the reactor needed to hold this catalyst should be

$$V = \frac{W}{\rho_{\text{bed}}} = \frac{W}{(1 - \phi)\rho_{\text{cat}}} = \frac{500 \text{ kg}_{\text{cat}}}{(1 - 0.6)2000 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}} = 0.625 \text{ m}^3.$$

To distinguish between the mass (or mass flow rate) of fluid and the weight of catalyst the subscript “cat” is usually added to terms involving catalyst weight.

Returning to our infinitesimal slice, we assume its contents are uniform so that we can write the general material for any species,

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}$$

as

$$0 = n_i|_V - n_i|_{V+\Delta V} + r_i\Delta V, \quad (9.7)$$

where the vertical bars  $|_V$  and  $|_{V+\Delta V}$  indicate quantities evaluated at the entrance and exit of our infinitesimal slice. We convert from volume to weight by using Equation (9.3) to substitute  $\Delta V = \rho_{\text{bed}}^{-1}\Delta W$  such that the material balance becomes

$$0 = n_i|_W - n_i|_{W+\Delta W} + \frac{r_i}{\rho_{\text{bed}}}\Delta W, \quad (9.8)$$

where the vertical bars indicated quantities evaluated at the entrance and exit of our infinitesimal slice. Rearranging and dividing by  $\Delta W$  yields

$$\frac{n_i|_{W+\Delta W} - n_i|_W}{\Delta W} = \frac{r_i}{\rho_{\text{bed}}} \quad (9.9)$$

In the limit of  $\Delta W \rightarrow 0$  the left-hand side becomes a derivative,

$$\lim_{\Delta W \rightarrow 0} \frac{n_i|_{W+\Delta W} - n_i|_W}{\Delta W} = \frac{dn_i}{dW} \quad (9.10)$$

so that the general material balance for any species  $i$  within a packed bed reactor becomes

$$\frac{dn_i}{dW} = \frac{r_i}{\rho_{\text{bed}}} \quad (9.11)$$

As we saw with the plug flow reactor, Equation (9.11) is a first order, ordinary differential equation and therefore requires a boundary or initial condition. Most commonly we have information about the inlet flow of each species so we use an initial condition of  $n_i$  at  $W = 0$ , or

$$\frac{dn_i}{dW} = \frac{r_i}{\rho_{\text{bed}}}, \quad n_i(W = 0) = n_{i0} \quad (9.12)$$

The term on the right-hand side,  $\rho_{\text{bed}}^{-1} r_i$ , shows up so often that we replace it with  $r'_i$  such that

$$r'_i = \frac{r_i}{\rho_{\text{bed}}} \doteq \frac{\text{amount of } i \text{ reacted}}{\text{catalyst weight} \cdot \text{time}}. \quad (9.13)$$

Substituting Equation (9.13) into Equation (9.12) yields

$$\boxed{\frac{dn_i}{dW} = r'_i} \quad n_i(W = 0) = n_{i0} \quad (9.1)$$

as we saw at the beginning of the chapter. Remember that we can write Equation (9.1) once for each species involved in a process, and that each such balance needs its own initial condition.

## 9.2 The PBR energy balance is an ordinary differential equation

The PBR energy balance is developed in a manner similar to the material balance, which is to start with the infinitesimal slice of reactor volume shown in the bottom part of Figure 9.1. For this volume at steady state, the general balance expression

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}$$

becomes

$$0 = \sum_{i=A}^Z [n_i H_i]_V - \sum_{i=A}^Z [n_i H_i]_{V+\Delta V} + \delta W_s \Delta V + \delta Q \Delta V, \quad (9.14)$$

where  $\delta W_s \Delta V$  is the shaft work done *on* the volume element and  $\delta Q \Delta V$  is the heat entering the volume element. As with the PFR, the terms  $\delta W_s$  and  $\delta Q$  are the shaft work and heat *per unit volume*, which is why they're denoted as  $\delta Q$  instead of just  $Q$  and  $\delta W_s$  instead of

just  $W_S$ . For reasons that will be discussed in the next section, the shaft work term  $\delta W_S \Delta V$  is nearly always zero—not just small compared to the other terms, but actually zero—so we'll omit it going forward.

Dropping  $\delta W_S$  and replacing  $\Delta V$  with  $\rho_{\text{bed}}^{-1} \Delta W$  by Equation (9.3) as we did for the material balance, Equation (9.14) becomes

$$0 = \sum_{i=A}^Z [n_i H_i]_W - \sum_{i=A}^Z [n_i H_i]_{W+\Delta W} + \frac{\delta Q}{\rho_{\text{bed}}} \Delta W. \quad (9.15)$$

Rearranging Equation (9.15), dividing by  $\Delta W$ , and taking the limit as  $\Delta W \rightarrow 0$  yields

$$\frac{d}{dW} \left( \sum_{i=A}^Z n_i H_i \right) = \frac{\delta Q}{\rho_{\text{bed}}} \quad (9.16)$$

Applying the chain rule,

$$\sum_{i=A}^Z \left( n_i \frac{dH_i}{dW} \right) + \sum_{i=A}^Z \left( H_i \frac{dn_i}{dW} \right) = \frac{\delta Q}{\rho_{\text{bed}}} \quad (9.17)$$

and at this point we need to bring in some additional relationships in order to make more progress. Recall from your thermodynamics course that we can approximate the change in enthalpy as

$$dH_i = C_{P,i} dT, \quad (9.18)$$

which is valid when there are no phase changes, is exact for ideal gases, and is a fair approximation for liquids at low pressures. The PBR material balance also gives

$$\frac{dn_i}{dW} = r'_i = \nu_i r', \quad (9.1)$$

and substituting Equations (9.1) and (9.18) into Equation (9.17) yields

$$\sum_{i=A}^Z \left( n_i C_{P,i} \frac{dT}{dW} \right) + \sum_{i=A}^Z (r' \nu_i H_i) = \frac{\delta Q}{\rho_{\text{bed}}}. \quad (9.19)$$

For conciseness we now make the substitution

$$\delta Q' = \frac{\delta Q}{\rho_{\text{bed}}} \quad (9.20)$$

which is analogous to the way we related  $r'_i$  to  $r_i$  in Equation (9.13). Thus Equation (9.19)

becomes

$$\sum_{i=A}^Z \left( n_i C_{P,i} \frac{dT}{dW} \right) + \sum_{i=A}^Z (r' \nu_i H_i) = \delta Q'. \quad (9.21)$$

If the sum terms are written out for all  $N$  species we'll see that the terms  $dT/dW$  and  $r'$  are the same in every one of the sum terms, which means we can factor them out as

$$\frac{dT}{dW} \sum_{i=A}^Z (n_i C_{P,i}) + r' \sum_{i=A}^Z (\nu_i H_i) = \delta Q', \quad (9.22)$$

and within this equation we can use Equation (4.13) to identify  $\sum \nu_i H_i = \Delta H$ . Substituting this relationship into Equation (9.22) and rearranging to isolate  $dT/dW$  yields

$$\frac{dT}{dW} = \frac{\delta Q' - r' \Delta H}{\sum_{i=A}^Z n_i C_{P,i}}, \quad (9.23)$$

which is the energy balance for PBRs in which only a single reaction occurs. To extend Equation (9.23) to account for multiple reactions requires only modification of  $r' \Delta H$  in the same way we did in Chapter 4,

$$r' \Delta H \xrightarrow[\text{mult. reactions}]{\text{extend to}} \sum_{j=1}^q r'_j \Delta H_j, \quad (4.25)$$

where the index  $j$  runs from 1 to the number of reactions  $q$  (not to be confused with heat  $Q$  or heat per unit weight  $\delta Q'$ ). Substituting this modification for multiple reactions makes Equation (9.23) become the general energy balance for a PBR that was introduced at the beginning of the chapter,

$$\boxed{\frac{dT}{dW} = \frac{\delta Q' - \sum_{j=1}^q r'_j \Delta H_j}{\sum_{i=A}^Z n_i C_{P,i}}} \quad T(W=0) = T_0 \quad (9.2)$$

which can be written somewhat more compactly as

$$\frac{dT}{dW} = \frac{\delta Q' - \sum r'_j \Delta H_j}{\sum n_i C_{P,i}}, \quad T(W=0) = T_0 \quad (9.24)$$

where the sum indices were omitted for clarity. Naturally if you want to use Equation (9.24) instead of Equation (9.2) you'll have to remember that the sum in the numerator is over all  $j$

reactions and the sum in the denominator is over all  $i$  components.

### 9.3 Shaft work is negligible and heat is transferred by a heat exchanger

Even more so than with PFRs, shaft work in PBRs is zero because the packed catalyst is fixed in place and spinning a paddle or mixing blade through fixed catalyst particles is a terrible idea. Instead, mixing is usually assumed to be achieved by the complicated twists and turns the reacting fluid must take on its path through the voids between catalyst particles.

Heat exchangers for PBRs are identical to the heat exchangers for PFRs as previously described in Section 7.3 and Figure 7.2: the major styles are either double-pipe or shell-and-tube. An additional concern present in PBRs but absent in PFRs is heat transfer within the catalyst particle itself. As implied by Figure 9.2 the reaction occurs throughout the volume of the porous catalyst particle, but heat transfer within the particle is dominated almost entirely by conduction rather than convection. Consequently, for exothermic reactions the actual temperature of reaction—the temperature within the catalyst particle—tends to be higher than the measured, bulk temperature of the fluid. Simultaneous heat and mass transfer within a catalyst particle are covered in graduate-level reaction engineering courses.

Since the general structure of PBR heat exchangers are identical to PFR heat exchangers we can use all the same expressions for  $\delta Q$ , but as implied by the PBR energy balance we'll find it more convenient to use the “primed” variable  $\delta Q'$  as an energy·catalyst weight<sup>-1</sup>·time<sup>-1</sup> term as opposed to an energy·volume<sup>-1</sup>·time<sup>-1</sup>. Thus

$$\delta Q = Ua(T_u - T) \quad (7.19)$$

becomes

$$\delta Q' = \frac{\delta Q}{\rho_{\text{bed}}} = \frac{Ua}{\rho_{\text{bed}}}(T_u - T), \quad (9.25)$$

where

$$a = \frac{dA}{dV} = \frac{4}{d} \quad (\text{cylindrical PBR}) \quad (7.17)$$

remains our primary expression for  $a$ , and where  $d$  refers to the diameter of the shell in which the catalyst is packed because this shell is the surface area available for heat transfer. Non-constant utility temperatures are handled in a manner analogous to the approach in Section 8.3, which we again delay until the next chapter so that we can focus on the central concepts of packed bed reactor design.



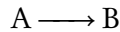
## 9.4 PBR problems can be solved by hand calculations

The category of PBR problems that can be solved by hand calculations using Equations (9.1) and (9.2) is fairly small, restricted mostly to isothermal systems for which only a single ODE—one of the material balances—needs to be solved. We're often constrained to have constant volumetric flow rates as well, either because the fluid is an incompressible liquid or because the fluid is a gas which is isothermal, isobaric, and has  $\delta = 0$  (c.f. Equation (5.5)).

The integrals associated with the solution of PBR material balances given in Equation (9.1) are also generally easier to solve by hand if they're written in terms of conversion  $X_A$  rather than the molar flow rate  $n_i$  because the lower bound of the associated integral is nearly always 0. As we saw in Chapters 4 and 7 the use of  $X_A$  instead of  $n_i$  in material balances is a special case of the general material balances, so we defer that topic to the next chapter.

### Example 9.1: Isothermal PBR sizing

The elementary, gas-phase reaction



takes place in a packed bed reactor according to the rate expression

$$r' = (3.1 \text{ L} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{s}^{-1}) \exp \left[ \frac{-2000}{T(\text{K})} \right] C_A.$$

Determine the catalyst weight necessary for 90% conversion of A if  $1 \text{ mol}_A \cdot \text{s}^{-1}$  and  $3 \text{ mol} \cdot \text{s}^{-1}$  of an inert enter the reactor at 450 K and 1 atm. Estimate the volume of the corresponding catalyst bed if  $\rho_{\text{bed}} = 1400 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}$  and compare this volume to an everyday object.

### Solution

Sp.	$\nu_i$	$n_{i0}$	$r'_i$
A	-1	1	$-r'$
B	1	0	$r'$
inert	0	3	0
$\delta = 0$			

The system is isothermal so we'll ignore the energy balance and focus only on the material balance, Equation (9.1), for species A as

$$\frac{dn_A}{dW} = r'_A = v_A r'$$

subject to the initial condition  $n_{A0} = 1 \text{ mol} \cdot \text{s}^{-1}$ . The rate expression at 450 K is

$$r'(450 \text{ K}) = 3.1 \exp\left[\frac{-2000}{450 \text{ K}}\right] C_A = 0.0364 \text{ L} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{s}^{-1} \times C_A.$$

The concentration of A is

$$C_A = \frac{n_A}{v} = \frac{n_A}{v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0}} = \frac{n_A}{v_0},$$

where Equation (5.5) was used after the second equality. Each of the fractions in the denominator are equal to unity because the system is isothermal ( $T/T_0 = 1$ ), isobaric ( $P_0/P = 1$ ), and  $\delta = 1 - 1 = 0$  ( $n_T/n_{T0} = 1$ ). The inlet volumetric flow rate can be determined from the ideal gas law as

$$v_0 = \frac{n_{T0} R T_0}{P_0} = \frac{(4 \text{ mol} \cdot \text{s}^{-1}) \times R \times (450 \text{ K})}{1 \text{ atm}} = 147.6 \text{ L} \cdot \text{s}^{-1},$$

where we were careful to remember that the *total* inlet molar flow rate,  $n_{T0}$ , is used instead of the *individual* inlet molar flow rate,  $n_{A0}$  or  $n_{I0}$ . Substituting these expressions for  $r'$  and  $v_0$  into the material balance for A yields

$$\frac{dn_A}{dW} = v_A r' = -\frac{0.0364}{147.6} n_A, \quad n_{A0} = 1 \text{ mol} \cdot \text{s}^{-1},$$

which is a separable ODE that can be rearranged and integrated as

$$\int_1^{n_A} \frac{dn_A}{n_A} = - \int_0^W \frac{0.0364}{147.6} dW.$$

Evaluating the both integrals yields

$$\ln \frac{n_A}{1 \text{ mol} \cdot \text{s}^{-1}} = -\frac{0.0364}{147.6} W$$

or

$$W = -\frac{147.6}{0.0364} \ln \frac{n_A}{1 \text{ mol} \cdot \text{s}^{-1}}.$$

With 90% conversion the outlet molar flow rate of A is

$$n_A = n_{A0}(1 - X_A) = 1 \text{ mol} \cdot \text{s}^{-1} (1 - 0.9) = 0.1 \text{ mol} \cdot \text{s}^{-1},$$

and therefore the weight of catalyst is

$$W = -\frac{147.6 \text{ L} \cdot \text{s}^{-1}}{0.0364 \text{ L} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{s}^{-1}} \ln 0.1 = \boxed{9337 \text{ g}_{\text{cat}}}$$

With a bed density  $\rho_{\text{bed}} = 1400 \text{ kg} \cdot \text{m}^{-3}$ , Equation (9.5) provides an estimate of the bed (reactor) volume  $V$  as

$$V = \frac{W}{\rho_{\text{bed}}} = \frac{9.337 \text{ kg}_{\text{cat}}}{1400 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}} = 0.0067 \text{ m}^3 = 6.7 \text{ L},$$

which is about the same size as three or four soda bottles placed end-to-end.

### Example 9.2: Removal of methyl blue from wastewater

Methyl blue, a biological staining agent and generally undesirable component in the watershed, can be broken down by UV light in the presence of titanium dioxide ( $\text{TiO}_2$ ) catalyst particles according to the nonelementary rate expression

$$r = \frac{kC_A}{1 + KC_A},$$

where  $k = 0.05 \text{ min}^{-1}$  and  $K = 9.2 \text{ L} \cdot \text{mmol}^{-1}$ . Determine the weight of  $\text{TiO}_2$  in an isothermal packed bed reactor to remove 99.9% of all methyl blue from a  $170 \text{ L} \cdot \text{h}^{-1}$  wastewater stream containing 40 ppm methyl blue if the bed density is  $280 \text{ g}_{\text{cat}} \cdot \text{L}^{-1}$ . Compare the volume of this reactor to an everyday object.

### Solution

We start again with the general material balance for methyl blue (A) as

$$\frac{dn_A}{dW} = r'_A = v_A r' = -r',$$

but the set of units given in the problem statement need to be worked on a little bit before we can use this material balance. Since the rate expression contains the constant  $K = 9.2 \text{ L} \cdot \text{mmol}^{-1}$  we know the concentration  $C_A$  will need units of  $\text{mmol} \cdot \text{L}^{-1}$  such that the product  $KC_A$  is dimensionless, which is the only way it can be added to the dimensionless number 1 in the denominator of the rate expression. Therefore we have to convert the inlet concentration of 40 ppm methyl blue to  $\text{mmol} \cdot \text{L}^{-1}$ , and since the stream is mostly water we can assume  $40 \text{ ppm} = 40 \text{ mg} \cdot \text{L}^{-1}$  such that

$$C_{A0} = \frac{40 \text{ mg}}{\text{L}} \frac{1 \text{ mmol}}{800 \text{ mg}} = 0.05 \text{ mmol} \cdot \text{L}^{-1},$$

where we used the molar mass of methyl blue as  $800 \text{ mg} \cdot \text{mmol}^{-1}$ . The units of the rate expression will therefore be

$$r \doteq k C_A \doteq k C_{A0} \doteq \text{mmol} \cdot \text{L}^{-1} \cdot \text{min}^{-1},$$

which implies that we need to use the bed density  $\rho_{\text{bed}}$  to get the rate expression in terms of catalyst weight instead of reactor volume. Turning to Equation (9.13) we have

$$r'_A = \frac{r_A}{\rho_{\text{bed}}} = \frac{-r}{\rho_{\text{bed}}} = -\frac{1}{\rho_{\text{bed}}} \frac{k C_A}{1 + K C_A} \doteq \text{mmol} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{min}^{-1},$$

which is consistent with the units of  $dn_A/dW$ . Substituting  $r'_A$  into the material balance yields

$$\frac{dn_A}{dW} = -\frac{1}{\rho_{\text{bed}}} \frac{k C_A}{1 + K C_A}.$$

We could convert each of the  $C_A$  terms on the right to  $n_A$  but we can get away with carrying around a few less terms if we instead relate  $dn_A$  to  $dC_A$  as

$$n_A = v C_A = v_0 C_A \quad \Rightarrow \quad dn_A = v_0 dC_A$$

such that the material balance becomes

$$\frac{dC_A}{dW} = -\frac{v_0}{\rho_{\text{bed}}} \frac{kC_A}{1 + KC_A}.$$

Rearranging and integrating yields

$$\int_{C_{A0}}^{C_A} (C_A^{-1} + K) dC_A = - \int_0^W \frac{k}{v_0 \rho_{\text{bed}}} dW,$$

Evaluating the integrals and rearranging to isolate  $W$  yields

$$W = -\frac{v_0 \rho_{\text{bed}}}{k} \left[ \ln \frac{C_A}{C_{A0}} + K(C_A - C_{A0}) \right].$$

Noting that the volumetric flow rate is  $v_0 = 170 \text{ L} \cdot \text{h}^{-1} = 2.83 \text{ L} \cdot \text{min}^{-1}$ , the required catalyst weight to achieve  $C_A = C_{A0}(1 - 0.999) = 0.00005 \text{ mmol} \cdot \text{L}^{-1}$  is

$$W = -\frac{(2.83)(280)}{0.05} \left[ \ln \frac{0.00005}{0.05} + 9.2(0.00005 - 0.05) \right] \approx 117 \text{ kg}_{\text{cat}},$$

which has a corresponding volume of about

$$V = \frac{W}{\rho_{\text{bed}}} = \frac{117 \times 10^3 \text{ g}_{\text{cat}}}{280 \text{ g}_{\text{cat}} \cdot \text{L}^{-1}} \approx 418 \text{ L}.$$

The average fire hydrant you see on the side of the street is a cylinder about 0.5 m tall with a diameter of about 10 cm, and therefore a volume of about 4 L. To achieve 418 L would require about 105 fire hydrants stacked end-to-end, which would span about 50 m or half the length of a FIFA football field.<sup>a</sup>

<sup>a</sup>A “soccer” field for my American friends.

### Example 9.3: Scale-up of a packed bed reactor

The liquid-phase, elementary reaction  $A \longrightarrow B$  takes place in an isothermal, isobaric packed bed reactor. In a laboratory-scale reactor it was found that 3% conversion was achieved with  $0.5 \text{ kg}_{\text{cat}}$ .

(a) Determine the amount of catalyst needed to achieve 95% conversion in a

production-scale reactor if the inlet volumetric flow rate to the production-scale reactor is twenty times larger than was used in the laboratory-scale reactor.

- (b) If the catalyst has a bed density  $\rho_{\text{bed}} = 950 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}$  and is to be packed into tubes 3 m long with an inner diameter of 20 cm, how many tubes are necessary for the production-scale reactor?

### Solution

A quick stoichiometric table and then we'll get to the solution steps:

Sp.	$\nu_i$	$n_{i0}$	$r'_i$	$n_i$
A	-1	$n_{A0}$	$-r'$	$n_{A0}(1 - X_A)$
B	1	0	$r'$	$n_{A0}X_A$

- (a) The material balance for A is

$$\frac{dn_A}{dW} = r'_A = -kC_A.$$

Making the substitutions

$$n_A = n_{A0}(1 - X_A) \quad \Rightarrow \quad dn_A = -n_{A0}dX_A$$

and

$$C_A = \frac{n_A}{v_0} = \frac{n_{A0}(1 - X_A)}{v_0}$$

simplifies the material balance to

$$\frac{dX_A}{dW} = \frac{k}{v_0}(1 - X_A),$$

which can be rearranged and integrated to yield

$$\int_0^{X_A} \frac{dX_A}{1 - X_A} = - \int_0^W \frac{k}{v_0} dW.$$

Evaluating integrals,

$$-\ln(1 - X_A) = \frac{k W}{v_0}.$$

This expression of the material balance holds in both the laboratory-scale reactor and the production-scale reactor we can identify

$$\frac{k}{v_0} = -\frac{\ln(1 - X_A)}{W} = -\frac{\ln(1 - 0.03)}{0.5 \text{ kg}_{\text{cat}}^{-1}} = 0.061 \text{ kg}_{\text{cat}}^{-1}.$$

The production-scale reactor's inlet volumetric flow rate is twenty times larger than that of the laboratory-scale reactor, or

$$\frac{k}{v_0^{\text{prod}}} = \frac{k}{20 v_0^{\text{lab}}} = \frac{0.061 \text{ kg}_{\text{cat}}^{-1}}{20} = 0.003 \text{ kg}_{\text{cat}}^{-1}.$$

Substituting into the material balance for the production-scale reactor yields

$$W^{\text{prod}} = -\frac{v_0^{\text{prod}}}{k} \ln(1 - X_A^{\text{prod}}) = -\frac{1}{0.003 \text{ kg}_{\text{cat}}^{-1}} \ln(1 - 0.95) = \boxed{998.6 \text{ kg}_{\text{cat}}}$$

(b) The bed volume is

$$V_{\text{bed}} = \frac{998.6 \text{ kg}_{\text{cat}}}{950 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}} = 1.05 \text{ m}^3,$$

which must be packed into individual tubes of volume

$$V_{\text{tube}} = \frac{\pi}{4} d^2 L = 0.094 \text{ m}^3.$$

The number of tubes required for the production-scale reactor is therefore

$$n_{\text{tube}} = \frac{V_{\text{bed}}}{V_{\text{tube}}} = \frac{1.05 \text{ m}^3}{0.094 \text{ m}^3} = 11.2 \approx \boxed{12 \text{ tubes}}$$

where the round-up to 12 was due to the implied requirement of an integer number of tubes.

## 9.5 PBR problems can be solved in MATLAB

As mentioned earlier the class of PBR problems that can be solved by hand is fairly small, limited mostly to isothermal, isobaric systems with just one reaction. While certainly an important class because of its instructional potential, if we want to move beyond that class we'll need to use MATLAB to solve the resulting ODE or set of ODEs. The general layout of the solution script to solve such problems is identical to the one we used for PFRs, repeated here for comprehensiveness.

Suppose you wish to solve the ODE

$$\frac{dn_A}{dW} = f(n_A, W) = 2n_A W, \quad n_{A0} = 1$$

from  $W = 0$  to  $W = 4$ , and then you want to plot  $n_A(W)$ . The following script accomplishes these tasks:

```
1 % yourScriptName.m
2 % Script portion
3 nA0 = 1;
4 Wspan = [0 4];
5 [Wsol, nAsol] = ode45(@fun, Wspan, nA0);
6 plot(Wsol, nAsol)
7
8 % Local (ODE) function
9 function dnAdW = fun(W, nA)
10 dnAdW = 2.*nA.*W;
11 end
```

As with PFRs, we'll start by solving a PBR problem for which we know the analytical solution and then we'll move on to more complicated examples.

### Example 9.4: Removal of methyl blue, revisited

Repeat Example 9.2 but use MATLAB to find the required catalyst weight.

### Solution



First, let's collect the necessary data from the previous problem:

$$\frac{dn_A}{dW} = -r' = \frac{-r}{\rho_{\text{bed}}} = -\frac{1}{\rho_{\text{bed}}} \frac{kC_A}{1 + KC_A}$$

$$C_{A0} = 0.05 \text{ mmol} \cdot \text{L}^{-1}$$

$$\rho_{\text{bed}} = 0.280 \text{ kg}_{\text{cat}} \cdot \text{L}^{-1}$$

$$k = 0.05 \text{ min}^{-1}$$

$$K = 9.2 \text{ L} \cdot \text{mmol}^{-1}$$

$$v_0 = 2.83 \text{ L} \cdot \text{min}^{-1}$$

$$X_A = 0.999$$

With this set of dimensionally-consistent variables and expressions we can move on to the MATLAB script, with one caveat: We don't know exactly how much catalyst weight is required so we simply pick a large value for the upper range of  $W$ . If we achieve 99.9% conversion somewhere in this range then we read off the precise value from the plot (or ask MATLAB to do so), and if we do not achieve the conversion in this range then we increase the upper range of  $W$  until we do. We happen to know from the previous problem that the weight should be around 117 kg, so let's pick 150 kg as our starting point.

```

1 % pbrMethylBlue.m
2 % Numerical solution:
3 v0 = 2.83; % L/min
4 Ca0 = 0.05; % mmol/L
5 nA0 = v0*Ca0; % mmol/min
6 Wspan = [0 150]; % kgcat
7 [Wsol, nAsol] = ode45(@fun, Wspan, nA0);
8 Wval = interp1(nAsol, Wsol, nA0*(1-0.999));
9
10 % Exact solution:
11 Ca = nAsol/v0; % mmol/L
12 rhobed = 0.280; % kgcat/L
13 k = 0.05; % 1/min
14 K = 9.2; % L/mmol
15 Wexact = -v0*rhobed/k*(log(Ca/Ca0)+K*(Ca-Ca0));
16
17 % Plots:
18 hold on
19 plot(Wsol, 1-nAsol/nA0, 'k');
20 plot(Wexact, 1-nAsol/nA0, 'ko');
21 hold off

```

```

22
23 function dnAdW = fun(W, nA)
24 v0 = 2.83; % L/min
25 k = 0.05; % 1/min
26 K = 9.2; % L/mmol
27 rhobed = 0.280; % kgcat/L
28
29 v = v0; % liquid
30 Ca = nA/v; % mmol/L
31 r = k*Ca./(1+K*Ca); % mmol/L-min
32 rprime = r/rhobed; % mmol/kgcat-min
33 rA = -rprime; % mmol/kgcat-min
34
35 dnAdW = rA;
36 end

```

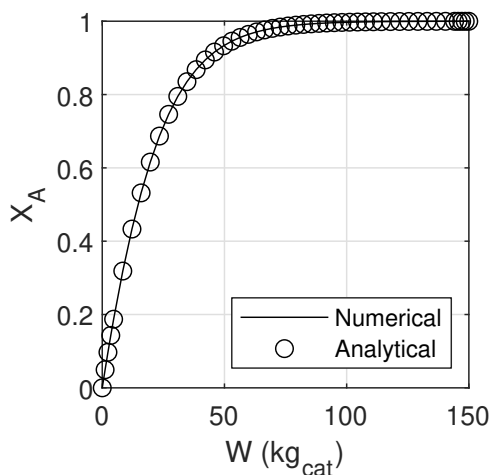
Notice a few things about this script:

**Line 8:** A linear interpolation was performed to find an approximate value of  $W$  for which the conversion of 99.9%. This is one of several different ways to find this value; another way could be to use the ODE solution object (*c.f.* Section 6.3).

**Lines 10-15:** The exact solution from Example 9.2 expressed catalyst weight as a function of concentration,  $W(C_A)$ , so the same form was used here for convenience.  $C_A$  was calculated from the ODE output values on Line 10.

**Lines 19-20:** Conversion  $X_A$  was never explicitly defined but was calculated on an as-needed basis.

Running this script produces the following plots:

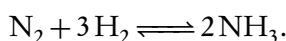


The approximate value of catalyst weight needed to achieve 99.9% conversion of dye is  $116.9 \text{ kg}_{\text{cat}}$ , found on Line 8 as `Wval`. This is in close agreement with the value calculated in Example 9.2, which was reported as about  $117 \text{ kg}_{\text{cat}}$ .

The next example explores a non-isothermal PBR with a nonelementary rate law in terms of partial pressures instead of concentrations. Recall from Section 3.2 that partial pressures are common when investigating gas-phase reactions occurring in the presence of solid-phase catalysts: since many such pairings occur in PBRs, partial pressures are commonly used in the literature.

### Example 9.5: Industrial production of ammonia

Adiabatic, packed bed reactors ( $\rho_{\text{bed}} = 2200 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}$ ) are used to produce ammonia at high temperatures and pressures from hydrogen and nitrogen feedstocks according to the gas-phase reaction



This reaction is non-elementary; its rate law has been estimated by Morud and Skogestad<sup>a</sup> to be

$$r = 4.75 \left( k_f P_{\text{N}_2} \frac{P_{\text{H}_2}^{1.5}}{P_{\text{NH}_3}} - k_r \frac{P_{\text{NH}_3}}{P_{\text{H}_2}^{1.5}} \right)$$

where  $r$  has units of  $\text{kmol} \cdot \text{m}^{-3} \cdot \text{h}^{-1}$ ,  $P_i$  is the partial pressure of species  $i$  in bar, and the forward and reverse rate constants are

$$k_f = 1.79 \times 10^4 \exp \left[ -\frac{87090 \text{ kJ} \cdot \text{kmol}^{-1}}{RT} \right]$$

$$k_r = 2.57 \times 10^{16} \exp \left[ -\frac{198464 \text{ kJ} \cdot \text{kmol}^{-1}}{RT} \right]$$

A feed stream containing 30 mol%  $\text{N}_2$ , 65 mol%  $\text{H}_2$ , and the balance  $\text{NH}_3$  enters such a  $15.2 \text{ m}^3$  ammonia synthesis reactor at  $340^\circ\text{C}$ , 290 bar, and a total molar flow rate of  $30000 \text{ kmol} \cdot \text{h}^{-1}$ . Create side-by-side profiles of all molar flow rates (on the left) and reactor temperature (on the right).

<sup>a</sup>J.C. Morud; S. Skogestad. Analysis of instability in an industrial ammonia reactor. *AIChE J*, 44 (1998) 888-895. Within this reference the original development of the rate expression was attributed to G.F. Froment; K.B. Bischoff, *Chemical Reactor Analysis and Design*, 2nd ed., Wiley, New York (1990).

Solution

First another stoichiometric table for organizational purposes, then onwards to the solution:

Sp.	Formula	$\nu_i$	$n_{i0}$	$r'_i$	$C_p^\circ(700\text{ K})$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )
A	N <sub>2</sub>	-1	0.30 $n_{T0}$	- $r'$	31	0
B	H <sub>2</sub>	-3	0.65 $n_{T0}$	-3 $r'$	29	0
C	NH <sub>3</sub>	2	0.05 $n_{T0}$	2 $r'$	48	-45.9
$\delta = -2$						

With three species-N<sub>2</sub> as A, H<sub>2</sub> as B, and NH<sub>3</sub> as C-and a nonisothermal reactor we'll need to solve a coupled system of four ODEs: three material balances using Equation (9.1) and one energy balance using Equation (9.2). The general equations are as follows:

$$\frac{dn_A}{dW} = r'_A = \nu_A r' = \frac{\nu_A r}{\rho_{bed}} \qquad n_A(W = 0) = n_{A0} = y_{A0} n_{T0}$$
$$\frac{dn_B}{dW} = r'_B = \nu_B r' = \frac{\nu_B r}{\rho_{bed}} \qquad n_B(W = 0) = n_{B0} = y_{B0} n_{T0}$$
$$\frac{dn_C}{dW} = r'_C = \nu_C r' = \frac{\nu_C r}{\rho_{bed}} \qquad n_C(W = 0) = n_{C0} = y_{C0} n_{T0}$$
$$\frac{dT}{dW} = \frac{-r' \Delta H}{\sum n_i C_{p,i}} = \frac{-r \Delta H}{\rho_{bed} \sum n_i C_{p,i}} \qquad T(W = 0) = 340^\circ\text{C} = 613\text{ K}$$

This system of ODEs has catalyst weight  $W$  as its independent variable so we'll also need the total catalyst weight as  $W = \rho_{bed} V = 33\,440\text{ kg}_{cat}$ .

The rate law looks a little intimidating at first, especially since we haven't seen a lot of rate laws up to now that use partial pressure instead of concentration. However, the use of partial pressure actually provides great simplification of our equations because within the ODE function we'll have direct access to each  $n_i$  (because they're dependent variables), and from these values we can quickly calculate the partial pressures of each species as

$$P_i = \frac{n_i}{n_T} P = \frac{n_i}{\sum n_i} P,$$

which isn't so bad after all! The units on  $k_f$  and  $k_r$  can be determined if you're so

inclined but are ultimately unnecessary because the problem statement tells us to use  $P_i$  in bar and that the units of  $r$  are  $\text{kmol} \cdot \text{m}^{-3} \cdot \text{h}^{-1}$ , so the units of  $k_f$  and  $k_r$  are whatever is necessary to make the rate expression dimensionally consistent.

We'll also need to adjust  $\Delta H$  for temperature by using Equation (4.22),

$$\Delta H = \Delta H_R + \Delta C_p(T - T_R),$$

where  $T_R = 298 \text{ K}$ . It's now up to us to put this information into our MATLAB script, and since we've been using MATLAB for a while to solve these problems we're going to use a little more of the vector notation from Appendix B to make our code a little more concise:

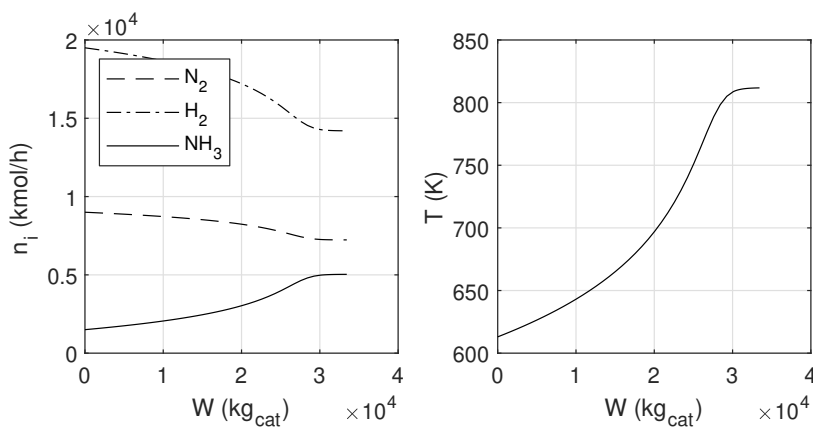
```

1 % ammoniaPBR.m
2 T0 = 340+273; % K
3 nT0 = 30000; % kmol/h
4 y0 = [0.30; 0.65; 0.05]; % [N2 H2 NH3]
5 n0 = nT0*y0;
6 Y0 = [n0; T0]; % [N2 H2 NH3 T]
7 Wspan = [0 33440]; % kgcat
8 [W, Y] = ode45(@fun, Wspan, Y0);
9 subplot 121; plot(W, Y(:, 1:3));
10 subplot 122; plot(W, Y(:, end));
11
12 function dYdW = fun(W, Y)
13 n = Y(1:3); % kmol/h, [N2 H2 NH3]
14 T = Y(4); % K
15 P0 = 290; % bar
16 P = n/sum(n)*P0; % bar, [N2 H2 NH3]
17 rhobed = 2200; % kgcat/m3
18
19 dHf = [0; 0; -45.9]*1e3; % kJ/kmol
20 Cp = [31; 29; 48]; % kJ/kmol-K
21 nu = [-1; -3; 2]; % mol/mol reacted
22 dCp = sum(nu.*Cp); % kJ/kmol-K
23 dH = sum(nu.*dHf) + dCp*(T-298); % kJ/kmol
24
25 f = 4.75;
26 kf = 1.79e4*exp(-87090/(8.314*T));
27 kr = 2.57e16*exp(-198464/(8.314*T));
28 r = f*(kf*P(1)*(P(2)^1.5)/P(3) - kr*P(3)/(P(2)^1.5)); % kmol/kgcat-h
29
30 dndW = nu.*r./rhobed; % kmol_i/kgcat-h
31 dTdW = -r*dH/(rhobed*sum(n.*Cp)); % K/kgcat
32 dYdW = [dndW; dTdW];
33 end

```

Notice that on Lines 19-21 we define constants as vectors in the same pattern that they appear within the vector  $n$ , which itself had its pattern set by the way variables were defined in  $y0$  on Line 4. Line 30 makes use of elementwise multiplication to produce all three material balances in a single expression.

Running the script produces the plot shown below.



Evidently the reacting stream reaches equilibrium after about 30000 kg<sub>cat</sub>, beyond which the net reaction rate approaches zero and the temperature levels off. One way to overcome this limitation could be to pass the outlet stream through a heat exchanger to reduce its temperature, and then send it through another catalyst bed for further conversion. This process of cooling a stream in between reactor beds is called *interstage cooling*, a topic we'll investigate in greater detail in Chapter 11.

Let's look at one final example, this time for methylation of alkenes. In this example we'll again use partial pressures but now with multiple reactions.

### Example 9.6: Methylation of alkenes with methanol

Methylation reactions remove methyl groups from alcohols and place them on a separate, unsaturated hydrocarbon, usually by means of a zeolite catalyst. Svelle and coworkers<sup>a</sup> report the following reaction mechanisms for methylation of ethene ( $C_2H_4$ ), propene ( $C_3H_6$ ), and *n*-butene ( $C_4H_8$ ) with methanol ( $CH_3OH$ ):

$j$	Reaction	$k_j/k_1$ (350 °C)	$E_a$ (kJ · mol <sup>-1</sup> )
1	$\text{C}_2\text{H}_4 + \text{CH}_3\text{OH} \longrightarrow \text{C}_3\text{H}_6 + \text{H}_2\text{O}$	1	109
2	$\text{C}_3\text{H}_6 + \text{CH}_3\text{OH} \longrightarrow \text{C}_4\text{H}_8 + \text{H}_2\text{O}$	17	69
3	$\text{C}_4\text{H}_8 + \text{CH}_3\text{OH} \longrightarrow \text{C}_5\text{H}_{10} + \text{H}_2\text{O}$	50	45

The rate constant  $k_1$  is  $2.6 \times 10^{-4} \text{ mol} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{h}^{-1} \cdot \text{mbar}^{-1}$ , and the rate expression for each reaction is

$$r' = k P_{\text{CH}_3\text{OH}}^0 P_{\text{alkene}}^1,$$

where  $P_{\text{CH}_3\text{OH}}$  is the partial pressure of methanol in mbar, and  $P_{\text{alkene}}$  is the partial pressure of the reactant alkene in each expression ( $\text{C}_2\text{H}_4$ ,  $\text{C}_3\text{H}_6$ , or  $\text{C}_4\text{H}_8$ ) in mbar.

The reactor was a tiny Pyrex tube with an inner diameter of 3 mm and packed with just 3 mg of catalyst. The total feed was  $100 \text{ mL} \cdot \text{min}^{-1}$  at 350 °C and 1 bar; within this stream the partial pressure of methanol, ethene, and propene were 50, 50, and 20 mbar with the balance as inert helium. Since the majority of the feed was inert helium the reactor operates approximately isothermally.

Create profiles of all molar flow rates except helium, as well as rates of reaction for each reaction  $j$ .

---

<sup>a</sup>S. Svelle; P.O. Ronning; S. Kolboe. Kinetic studies of zeolite-catalyzed methylation reactions, Part 1: Coreaction of ethene and methanol. *J. Catalysis* 224 (2004) 115-123. S. Svelle; P.O. Ronning; U. Olsbye; S. Kolboe. Kinetic studies of zeolite-catalyzed methylation reactions, Part 2: Coreaction of propene or n-butene and methanol. *J. Catalysis* 234 (2005) 385-400.

## Solution

The system is isothermal so the only thing we're interested in are the material balances, all of which follow the same general pattern as Equation (9.1),

$$\frac{dn_i}{dW} = r'_i \quad n_i(0) = n_{i0} \quad (9.1)$$

Rather than label species generically as A, B, C, and so forth, we can use somewhat more descriptive labels such as  $\text{C}_2$ ,  $\text{C}_3$ ,  $\text{C}_4$ , and  $\text{C}_5$  for the alkenes,  $m$  for methanol,  $w$  for water, and  $h$  for helium. We'll use the inlet partial pressures  $P_{i0}$  instead of inlet molar flow rates  $n_{i0}$  in our stoichiometric table since those are the quantities provided.

Sp.	Formula	$\nu_{i1}$	$\nu_{i2}$	$\nu_{i3}$	$r'_i$	$P_{i0}$ (mbar)
C <sub>2</sub>	C <sub>2</sub> H <sub>4</sub>	-1	0	0	$-r'_1$	50
C <sub>3</sub>	C <sub>3</sub> H <sub>6</sub>	1	-1	0	$r'_1 - r'_2$	20
C <sub>4</sub>	C <sub>4</sub> H <sub>8</sub>	0	1	-1	$r'_2 - r'_3$	0
C <sub>5</sub>	C <sub>5</sub> H <sub>10</sub>	0	0	1	$r'_3$	0
<i>m</i>	CH <sub>3</sub> OH	-1	-1	-1	$-r'_1 - r'_2 - r'_3$	50
<i>w</i>	H <sub>2</sub> O	1	1	1	$r'_1 + r'_2 + r'_3$	0
<i>h</i>	He	0	0	0	0	880
		$\delta = 0$	0	0		

It gets tedious to write expressions like `rprime` when coding so for brevity we'll drop the prime notation from the code below.

The rate expression and inlet feed compositions were given in terms of partial pressures, so we need to get these quantities in a set of variables consistent with the material balances using one of the following approaches:

- Convert all partial pressure terms in the rate expression and feed conditions to molar flow rates by using the ideal gas law,  $P_i v = n_i RT$  from which  $n_i = P_i v / RT$ .
- Convert the molar flow rate terms in the material balances to partial pressures.

Provided the ideal gas law is a good approximation of this system<sup>a</sup> then either approach should yield exactly the same result. Let's see what it looks like to use the partial pressures in the derivatives (Option 2) since that's not something we've encountered yet and you can always check it against Option 1 at your convenience.

From the ideal gas law we have  $n_i = P_i v / RT$ , from which

$$dn_i = \frac{v}{RT} dP_i = c_0 dP_i$$

where  $v / RT = c_0 = \text{constant}$ , and naturally we'll have to be careful about the units. The material balances can then be rewritten as

$$\frac{dP_i}{dW} = \frac{r'_i}{c_0} \quad P_i(0) = P_{i0}$$

Moving on to the rate expression, notice the oddity of including the partial pressure of



methanol raised to the zero power: anything raised to zero is equal to one, so this term is unnecessary! It's there because the authors wanted to emphasize that they investigated the behavior in response to methanol concentration and didn't simply ignore it. Finally, the rates of each individual reaction  $j$  can be calculated by solving the system of ODEs, then using the substituting the partial pressures back into the rate expressions for each reaction.

Onwards to the code!

```

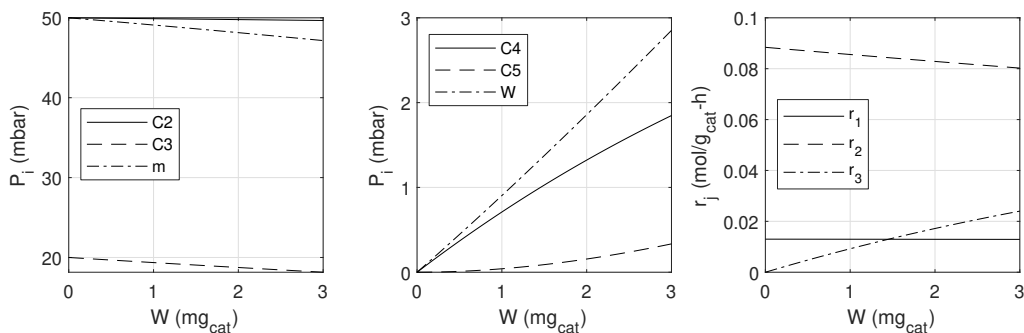
1 % methylation.m
2 % all vectors follow [C2 C3 C4 C5 m w h]
3 P0 = [50; 20; 0; 0; 50; 0; 880]; % mbar
4 Wspan = [0 3e-3]; % gcat
5 [W, P] = ode45(@fun, Wspan, P0);
6
7 W = W*1000; % gcat->mgcat, for plotting
8 subplot 131; plot(W, P(:, [1 2 5]))
9 subplot 132; plot(W, P(:, [3 4 6]));
10
11 k1 = 2.6e-4;
12 r1 = k1*P(:,1); % mol/gcat-h
13 r2 = 17*k1*P(:,2); % mol/gcat-h
14 r3 = 50*k1*P(:,3); % mol/gcat-h
15 r = [r1 r2 r3];
16 subplot 133; plot(W, r);
17
18 function dYdW = fun(W, P)
19 k1 = 2.6e-4; % mol/gcat-h-mbar
20
21 r1 = k1*P(1); % mol/gcat-h
22 r2 = 17*k1*P(2); % mol/gcat-h
23 r3 = 50*k1*P(3); % mol/gcat-h
24
25 T = 350+273; % K
26 v = 100*60; % mL/min->mL/h
27 R = 8.314e-2; % L-bar/mol-K
28 R = R*1000*1000; % mL-mbar/mol-K
29 c0 = v/(R*T); % mol/mbar-h
30
31 dc2 = -r1/c0; % mbar/gcat
32 dc3 = (r1-r2)/c0; % mbar/gcat
33 dc4 = (r2-r3)/c0; % mbar/gcat
34 dc5 = r3/c0; % mbar/gcat
35 dm = -(r1+r2+r3)/c0; % mbar/gcat
36 dw = (r1+r2+r3)/c0; % mbar/gcat
37 dh = 0; % mbar/gcat
38

```

```

39 dYdW = [dc2; dc3; dc4; dc5; dm; dw; dh];
40 end

```



Several features of these plots are noteworthy. First, as implied by the relative scaling of the rate constants, reaction 1 is quite slow and therefore the methylation of ethene is minimal. Second, methanol is consumed primarily through reaction 2 because the rate of reaction 2 is nearly an order-of-magnitude higher than reactions 1 or 3. Third, the composition of methanol is not constant and is not in excess of the other reactants, so its exclusion from the rate expression will eventually be problematic because the rates will continue to be positive despite the absence of methanol. To see this, try running the script with higher catalyst weights: what happens to the partial pressure of  $C_3$  after 10 mg<sub>cat</sub>? After 100 mg<sub>cat</sub>?

---

“What’s the compressibility factor under these conditions?”

# Chapter 10

## Special Cases of the PBR

“Nothing goes over my head. My reflexes are too fast; I would catch it.”

Drax, *Guardians of the Galaxy* (2014)

### 10.1 Rewriting equations in terms of conversion can be useful

This is one of the shortest sections in the book for one simple reason: as we saw in Chapter 9 the material and energy balance equations for the packed bed reactor are nearly identical to those of the plug flow reactor except for a change from volume  $V$  to catalyst mass  $W$  using the bed density  $\rho_{\text{bed}}$  as

$$W = \rho_{\text{bed}} V, \quad (9.5)$$

or

$$dW = \rho_{\text{bed}} dV. \quad (10.1)$$

For example, the material balance for a PFR in terms of conversion is

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}}, \quad (8.2)$$

so if we substitute Equation (10.1) into Equation (8.2) then we get

$$\boxed{\frac{dX_A}{dW} = \frac{-r'_A}{n_{A0}}} \quad X_A(W=0) = 0, \quad (10.2)$$

which is the material balance for species A in a PBR. Similarly, Equation (10.1) can be used with Equation (8.3) to yield

$$\frac{dT}{dW} = \frac{\delta Q' - r' \Delta H}{n_{A0} \left[ \sum_{i=A}^Z \Theta_i C_{P,i} - \frac{X_A}{\nu_A} \Delta C_P \right]} \quad T(W=0) = T_0 \quad (10.3)$$

where  $\delta Q'$  was previously defined as  $\delta Q' = \delta Q / \rho_{\text{bed}}$  in Equation (9.25). Equation (10.3) is the energy balance for a PBR in terms of species A conversion. We can use Equations (10.2) and (10.3) in a manner similar to the way that we used Equations (8.2) and (8.3) back in Chapter 8: if a system can be approximated as having only a single reaction and one of the species can be identified as “important” then Equations (10.2) and (10.3) can potentially be more straightforward to use than the general material and energy balances, Equations (9.1) and (9.2). In particular, we’ll see some clever uses of conversion when we get to multireactor systems in Chapter 11.

As is tradition, let’s revisit a couple PBR examples from Chapter 9 to see how we can solve them using conversion.

### Example 10.1: Isothermal PBR sizing, revisited

Repeat Example 9.1 but find the catalyst weight by using the PBR material balance written in terms of species A.

### Solution

As a reminder, the problem statement asked us to find the catalyst weight necessary to achieve 90% of A for the reaction  $A \longrightarrow B$  if  $1 \text{ mol}_A \cdot \text{s}^{-1}$  and  $3 \text{ mol} \cdot \text{s}^{-1}$  of an inert enter the reactor at 450 K and 1 atm. The rate expression was given as

$$r' = (3.1 \text{ L} \cdot \text{g}_{\text{cat}}^{-1} \cdot \text{s}^{-1}) \exp \left[ \frac{-2000}{T(\text{K})} \right] C_A.$$

Starting with Equation (10.2) we have

$$\frac{dX_A}{dW} = \frac{-r'_A}{n_{A0}} = \frac{-\nu_A r'}{n_{A0}}.$$

Substituting the rate expression into the material balance yields

$$\frac{dX_A}{dW} = \frac{-\nu_A(3.1)\exp[-2000/T]C_A}{n_{A0}},$$

and into this expression we can substitute  $C_A(X_A)$  as

$$C_A = \frac{n_i}{v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0}} = C_{A0}(1 - X_A)$$

to get to

$$\frac{dX_A}{dW} = \frac{-\nu_A(3.1)\exp[-2000/T](1 - X_A)}{v_0}.$$

The total inlet volumetric flow rate is unchanged from Example 9.1 at  $147.6 \text{ L} \cdot \text{s}^{-1}$ , as calculated from the ideal gas law. Substituting this and  $T = 450 \text{ K}$  into the material balance, then rearranging and integrating yields

$$\int_0^{X_A} \frac{dX_A}{1 - X_A} = \int_0^W (2.467 \times 10^{-4} \text{ g}_{\text{cat}}^{-1}) dW.$$

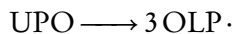
Evaluating these integrals and solving for  $W$  yields

$$W = \frac{-\ln(1 - X_A)}{2.467 \times 10^{-4} \text{ g}_{\text{cat}}^{-1}} = \boxed{9337 \text{ g}_{\text{cat}}}$$

which is identical to what we found in Example 9.1.

### Example 10.2: Catalytic cracking of used palm oil

Used palm oil (UPO) can be converted into a mixture of gasoline, kerosene, and diesel called an “organic liquid product” (OLP) according to the reaction



When this reaction takes place over fresh zeolite catalyst, Ooi *et al*<sup>a</sup> measured the reaction to be first order with  $k = 1.47 \text{ h}^{-1}$  at  $673 \text{ K}$  and  $k = 6.3 \text{ h}^{-1}$  at  $723 \text{ K}$ .

Estimate the mass of catalyst necessary to achieve 10% palm oil conversion if the reaction takes place in a single, 8 cm diameter pipe jacketed with saturated steam at  $400$

°C, and plot the molar flow rates and temperature profiles for the reactor. The fresh feed is pure, used palm oil at  $1 \text{ kg} \cdot \text{s}^{-1}$  and  $150^\circ\text{C}$ . The density of the bed is  $1760 \text{ kg} \cdot \text{m}^{-3}$ , and additional information about palm oil and OLP are provided in the table below.

Property	UPO	OLP
density	$0.9 \text{ kg} \cdot \text{L}^{-1}$	$0.9 \text{ kg} \cdot \text{L}^{-1}$
molar mass	$900 \text{ g} \cdot \text{mol}^{-1}$	$300 \text{ g} \cdot \text{mol}^{-1}$
heat capacity	$1800 \text{ J} \cdot \text{mol}^{-1} \cdot ^\circ\text{C}^{-1}$	$500 \text{ J} \cdot \text{mol}^{-1} \cdot ^\circ\text{C}^{-1}$
enthalpy of formation	$-2400 \text{ kJ} \cdot \text{mol}^{-1}$	$-600 \text{ kJ} \cdot \text{mol}^{-1}$

<sup>a</sup>Y-S. Ooi; R. Zakaria; A.R. Mohamed; S. Bhatia. Catalytic cracking of used palm oil and palm oil fatty acids mixture for the production of liquid fuel: kinetic modeling. Energy and Fuels, 2004 (18) 1555-1561.

Solution

Let’s call UPO species A and OLP species B so that we can simplify our notation a little bit. We can summarize the problem information along with some of the material properties into the more familiar stoichiometric table to help us get started:

Sp.	Abbr.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r'_i$	$C_i$	$C_p$ ( $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ )	$\Delta H_f^\circ$ ( $\text{kJ} \cdot \text{mol}^{-1}$ )
A	UPO	-1	$n_{A0}$	1	$-r'$	$C_{A0}(1-X_A)$	1800	-2400
B	OLP	3	0	0	$r'$	$3C_{A0}X_A$	500	-600

The material and energy balances in terms of palm oil conversion ( $X_A$ ) are

$$\frac{dX_A}{dW} = \frac{-r'_A}{n_{A0}} \tag{10.2}$$

and

$$\frac{dT}{dW} = \frac{\delta Q' - r'\Delta H}{n_{A0} \left[ \sum_{i=A}^Z \Theta_i C_{P,i} - \frac{X_A}{\nu_A} \Delta C_p \right]}. \tag{10.3}$$

The rate per unit catalyst mass is

$$r' = \frac{r}{\rho_{\text{bed}}} = \frac{kC_A}{\rho_{\text{bed}}} = \frac{kC_{A0}(1-X_A)}{\rho_{\text{bed}}} = -r'_A,$$

where the rate constant  $k$  can be determined at any temperature by using Equation (3.10) to first find  $E_a$  from the two given values as

$$\frac{k_2}{k_1} = \exp \left[ \frac{E_a}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right] \quad \Rightarrow \quad E_a = 117.7 \text{ kJ} \cdot \text{mol}^{-1}.$$

Turning now to the energy balance, the heat transfer term  $\delta Q'$  is given by Equation (9.25) as

$$\delta Q' = \frac{Ua}{\rho_{\text{bed}}} (T_u - T),$$

where  $T_u$  is the temperature of the steam,  $a = 4/d = 50 \text{ m}^{-1}$ , and  $U$  is calculated from Equation (4.27) for an organic liquid and phase change pairing as

$$\frac{1}{U} \approx \frac{1}{1000} + \frac{1}{2500} \quad \Rightarrow \quad U = 714 \text{ J} \cdot \text{s}^{-1} \cdot \text{m}^{-2} \cdot \text{K}^{-1}.$$

The enthalpy of reaction  $\Delta H$  and change in heat capacity  $\Delta C_p$  are given by the usual formulae,  $\Delta M = \sum v_i M_i$ . Since pure oil (A) is provided to the reactor  $\Theta_A = 1$  and  $\Theta_B = 0$ .

The material and energy balances constitute a system of couple ODEs so we'll need MATLAB to solve the system for us. Since the amount of catalyst is what we're trying to find, we'll have to guess a value for the upper bound of Wspan, solve the system, then check to see if we've achieved 60% conversion.

```

1 % palmOil.m
2 Y0 = [0 150+273]; % [Xa, T(K)]
3 Wspan = [0 1e3]; % kg
4 [Wsol, Ysol] = ode45(@fun, Wspan, Y0);
5
6 subplot 121; plot(Wsol, Ysol(:,1));
7 subplot 122; plot(Wsol, Ysol(:,2));
8 W = interp1(Ysol(:,1), Wsol, 0.1);
9
10 function dYdW = fun(W, Y)
11 Xa = Y(1);
12 T = Y(2); % K
13
14 % Constants
15 Ea = 117.7e3; % J/mol
16 R = 8.314; % J/mol-K
17 rhobed = 1760; % kg/m3
18 Tu = 400+273; % K

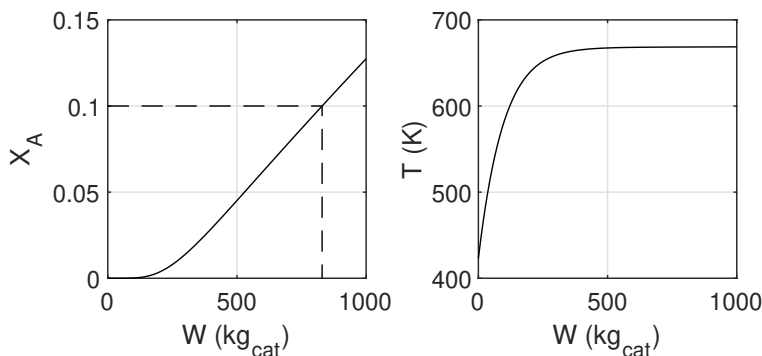
```

```

19 mA0 = 1; % kg/s
20 rhoA = 900; % kg/m3
21 nu = [-1 3];
22 Theta = [1 0];
23 Cp = [1800 500]; % J/mol-C
24 dHf = [-2400 -600]*1e3; % J/mol
25 k1 = 1.47/3600; % 1/s
26 T1 = 673; % K
27
28 % Material balance terms
29 nA0 = mA0*1000/900; % molA/s
30 v0 = mA0/rhoA; % m3/s
31 Ca0 = nA0/v0; % molA/m3
32 k = k1*exp(Ea/R*(1/T1-1./T)); % 1/s
33 rPrime = k*Ca0*(1-Xa)/rhobed; % mol/s-kgcat
34
35 % Energy balance terms
36 dCp = sum(nu.*Cp); % J/mol-C
37 dH = sum(nu.*dHf) + dCp*(T-298); % J/mol
38 a = 4/0.08; % 1/m
39 U = 1/(1/1000+1/2500); % J/s-m2-K
40 dQ = U*a/rhobed*(Tu-T); % J/s-kgcat
41
42 % Differential equations
43 dXadW = rPrime/nA0; % 1/kgcat
44 dTdW = (dQ-rPrime*dH)/...
45 (nA0*(sum(Theta.*Cp)+Xa*dCp)); % K/kgcat
46 dYdW = [dXadW; dTdW];
47 end

```

Notice that on Line 3 an upper bound for Wspan was guessed at 1000 kg, and after inspecting the plot to confirm that at least 10% conversion was achieved then the catalyst required was calculated by linear interpolation on Line 8.



Based on the figure and the linear interpolation approximately 830 kg<sub>cat</sub> is necessary to



achieve 10% conversion of used palm oil according to this mechanism. Since the pipe was 8 cm in diameter we can also estimate the length of pipe as

$$L = \frac{4W}{\pi \rho_{\text{bed}} d^2} = 93.8 \text{ m},$$

which is quite long! As we'll see in the next section the pressure drop across such a length of packed bed would likely be significant so we might try to think of ways to locate a larger diameter pipe to make the reactor length shorter, thereby reducing pressure drop (and therefore reducing operational costs).

## 10.2 Significant pressure drop can occur in PBRs

Pressure drop in PBRs is by far the biggest change from the relationships we've seen for PFRs because we now have to consider the presence of the particles and the resulting increase in surface area. The derivation follows an approach similar to the one we used in Section 8.2, where again we start with a simple force balance on a cylindrical element of fluid,

$$0 = F_u - F_d - F_D, \quad (10.4)$$

where  $F_u$  is the upstream force which pushes on the fluid element in the direction of flow,  $F_d$  is the downstream force which pushes on the fluid element in the direction opposite of flow, and  $F_D$  is the drag force due to all stationary surfaces. The first of many assumptions we'll make in this derivation is that the overwhelming majority of the drag force comes only from the catalyst particles, which means we'll neglect the contribution of the tube walls.

We also assume that the upstream and downstream pressures act only on the fraction of cross-sectional area in which fluid resides, which we estimate as the total cross sectional area scaled by the void fraction,  $A_c \phi$ . The force exerted on either face is therefore no longer  $F = PA_c$  as it was for an empty tube but is now  $F = PA_c \phi$ , where  $\phi$  is the same void fraction defined in Equation (9.6). After a little rearranging Equation (10.4) becomes

$$F_D = (P_u - P_d) A_c \phi. \quad (10.5)$$

As we did for plug flow reactors we assume the drag force within packed bed reactors is proportional to the product of the area of liquid-solid contact with some characteristic kinetic

energy, multiplied by an experimentally determined coefficient as

$$F_D = \left( \begin{array}{c} \text{area of} \\ \text{liquid- solid} \\ \text{contact} \end{array} \right) \times \left( \begin{array}{c} \text{characteristic} \\ \text{kinetic} \\ \text{energy} \end{array} \right) \times \left( \begin{array}{c} \text{fudge} \\ \text{factor} \end{array} \right) = A_w K f, \quad (10.6)$$

but we're going to have to work a little harder on these three terms than we did previously for PFRs. We represent the contact area as the product of the total particle volume,  $A_c L(1 - \phi)$ , and the particle surface area per unit particle volume,  $6/d_p$ , where  $d_p$  is the diameter of the particle. In this way the wetted area  $A_w$  for a packed bed is

$$A_w = \frac{6}{d_p} A_c L(1 - \phi). \quad (10.7)$$

The characteristic kinetic energy  $K$  is similar to the product  $(1/2)\rho u^2$  that we used in Section 8.2 except that the gas velocity  $u$  is scaled by  $\phi$  as  $u \rightarrow u/\phi$  such that

$$K = \frac{1}{2} \rho \left( \frac{u}{\phi} \right)^2. \quad (10.8)$$

Substituting Equations (10.6) to (10.8) into Equation (10.5) and rearranging to isolate  $P_d$  yields

$$P_d = P_u - 3 \frac{1 - \phi}{\phi^3} \frac{L \rho u^2}{d_p} f \quad (10.9)$$

Something unusual happens at this point in the derivation: conventionally, for reasons not entirely clear, the 3 is simply dropped! It goes away! Perhaps it's because the fudge factor  $f$  is going to get complicated enough that it's easier to simply absorb the 3 into  $f$  and worry about it later, but for whatever reason Equation (10.9) is simply rewritten without the 3 as

$$P_d = P_u - \frac{1 - \phi}{\phi^3} \frac{L \rho u^2}{d_p} f \quad (10.10)$$

In order to eventually develop an expression which is comparable to Equation (8.11) we make another arbitrary choice to define a *particle friction factor*  $f_p$  as

$$f_p \equiv \frac{1 - \phi}{\phi^3} f \quad (10.11)$$

such that Equation (10.10) can be rewritten as

$$P_d = P_u - f_p \frac{L \rho u^2}{d_p}. \quad (10.12)$$

As we did with the PFR we now make the substitutions

$$L = \frac{V}{A_c} = \frac{4V}{\pi d^2}, \quad u = \frac{v}{A_c} = \frac{4v}{\pi d^2}, \quad \text{and} \quad \rho = \frac{m_T}{v},$$

where  $V$  is the volume of the bed,  $d$  is the diameter of the bed,  $v$  is the volumetric flow rate of gas, and  $m_T$  is the total mass flow rate (always constant!). After a bit of rearranging Equation (10.12) becomes

$$P_d = P_u - v f_p \frac{64 m_T}{\pi^3 d^6 d_p} V, \quad (10.13)$$

or, switching from bed volume to catalyst weight by using  $W = \rho_{\text{bed}} V$ ,

$$P_d = P_u - v f_p \frac{64 m_T}{\pi^3 d^6 d_p} \frac{W}{\rho_{\text{bed}}}. \quad (10.14)$$

The derivative of  $P_d$  with respect to  $W$ , after dropping the subscript  $d$  on  $P_d$  as we did with PFRs, is therefore

$$\boxed{\rho_{\text{bed}} \frac{dP}{dW} = -v f_p \frac{64 m_T}{\pi^3 d^6 d_p}} \quad P(W=0) = P_0. \quad (10.15)$$

Equation (10.15) is the equation we'll use to calculate pressure drop in packed bed reactors, and in structure its quite similar to Equation (8.11) for PFRs,

$$\frac{dP}{dV} = -v f \frac{128 m_T}{\pi^3 d^7}. \quad (8.11)$$

The presence of  $\rho_{\text{bed}}$  in Equation (10.15) is because we're interested in pressure drop per unit catalyst mass as opposed to per unit bed volume; the numerator contains 64 instead of 128 because of how the wetted area was defined, and because we arbitrarily dropped the 3 from Equation (10.9); and the denominator has  $d^6 d_p$  instead of simply  $d^7$ , again because of the change in the wetted area definition.

The last remaining difference is the particle friction factor  $f_p$ . The friction factor for PFRs

was as a function of the dimensionless Reynolds number,

$$\text{Re} = \frac{\rho u d}{\mu} = \frac{4m_T}{\pi d \mu}, \quad (8.13)$$

and the same convention is followed for the particle friction factor  $f_p$ , albeit with an increased complexity.

The most commonly used form of the particle friction factor is the *Ergun equation*,

$$f_p = \left[ 150 + 1.75 \left( \frac{\text{Re}}{1 - \phi} \right) \right] \frac{(1 - \phi)^2}{\phi^3 \text{Re}}, \quad (10.16)$$

which is obviously somewhat more lengthy than Equation (8.12) for smooth pipes. Equation (10.16) is an *empirical* expression for the friction factor developed by Sabri Ergun in 1952,<sup>1</sup> meaning that it was derived from experimental data. Notice that the  $1.75\text{Re}/(1 - \phi)$  term becomes considerably larger than 150 when Re is larger than about 100, and for this reason the 150 in Equation (10.16) is associated with frictional losses due to viscous forces, whereas the  $1.75\text{Re}/(1 - \phi)$  term is associated with kinetic energy losses.

The Ergun equation is valid in the range of about  $0.1 \leq \text{Re} \leq 200$ , which is fairly small. In 1981 the Nuclear Technology Commission (KTA), part of the German Nuclear Safety Standards Commission (NSSC), published<sup>2</sup> a modified Ergun equation referred to as the “KTA” equation,

$$f_p = \left[ 160 + 3 \left( \frac{\text{Re}}{1 - \phi} \right)^{0.9} \right] \frac{(1 - \phi)^2}{\phi^3 \text{Re}} \quad (10.17)$$

which is not particularly more complicated than Equation (10.16) but is valid over a significantly larger range, about  $1 \leq \text{Re} \leq 100,000$ . Should your needs or curiosity be such that Equations (10.16) and (10.17) are not sufficient, Erdim and coworkers<sup>3</sup> provided an excellent review of more than 30 forms of the particle friction factor. For the purposes of this text, however, we’ll restrict ourselves to only Equation (10.17) for its extended range. We should also keep in mind that Equation (10.17) is recommended only for void fractions between 0.36 and 0.42, and for beds whose length  $L$  is greater than about  $5d_p$ .

Substituting Equation (10.17) into Equation (10.15) produces an absolute mess of an equation; to avoid making algebraic mistakes it’s usually better to leave them separate and simply

<sup>1</sup>S. Ergun. Fluid flow through packed columns. Chem. Eng. Prog. 48 (1952) 89-94.

<sup>2</sup>NSSC KTA, Reactor core design of high-temperature gas-cooled reactors part 3: loss of pressure through friction in pebble bed cores. Germany, 1981.

<sup>3</sup>E. Erdim; O. Akgiray, I. Demir. A revisit of pressure drop-flow rate correlations for packed beds of spheres. Powd. Tech. 283 (2015) 488-504.

Table 10.1: Summary of pressure drop equations for PFRs and PBRs. In both expressions  $v$  is the total volumetric flow rate,  $m_T$  is the total mass flow rate (always constant!),  $d$  is the bed diameter,  $d_p$  is the particle (catalyst) diameter,  $\rho_{\text{bed}}$  is the bed density,  $\phi$  is the void fraction, and  $\text{Re}$  is the Reynolds number given by Equation (8.13) ( $\text{Re} = 4m_T/\pi d\mu$ , where  $\mu$  is the fluid viscosity).

Reactor	Pressure drop	Friction factor
PFR	$\frac{dP}{dV} = -vf \frac{128m_T}{\pi^3 d^7}$	$f = \frac{0.0791}{\text{Re}^{1/4}}$
PBR	$\rho_{\text{bed}} \frac{dP}{dW} = -vf_p \frac{64m_T}{\pi^3 d^6 d_p}$	$f_p = \left[ 160 + 3 \left( \frac{\text{Re}}{1-\phi} \right)^{0.9} \right] \frac{(1-\phi)^2}{\phi^3 \text{Re}}$

calculate Equation (10.17) prior to substitution into Equation (10.15). Assuming the fluid's viscosity  $\mu$  is constant then the Reynolds number is also constant, and therefore  $f_p$  is constant as well.

Simply because there are quite a few expressions floating around, Table 10.1 was constructed as a concise summary of the pressure drop equations we'll use for PFRs and PBRs. Be sure not to switch them: under no flow conditions is the PBR pressure drop expression, Equation (10.15), ever valid for an empty bed (a PFR), and *vice versa*!

### 10.2.1 Absolute pressure is measured relative to vacuum

Since packed beds tend to produce exceptionally large pressure drops we have to think a little bit more closely about the units we're using with pressure. Everywhere in this textbook we use pressure measured on an *absolute* scale, where "absolute" refers to the pressure measured relative to zero. We can achieve zero pressure (hypothetically) by removing all molecules from a container, but similar to how we can define but not actually reach absolute zero, we can defined zero pressure in this way but not actually reach it.

When we want to indicate that a pressure is measured or calculated relative to zero pressure we sometimes include the words "absolute," as in, "*The absolute inlet pressure is 1.2 bar*", or we include the word "absolute" or abbreviation "abs" in the units themselves, as in, "*The inlet pressure is 1.2 bar (absolute)*" or "*The inlet pressure is 1.2 bar<sub>abs</sub>*."

The pressure you're experiencing right now as you read this is referred to as *atmospheric* pressure and is usually reported as 1 atm, 1 bar, 101.325 kPa, or something equivalent. These should all be interpreted as *absolute* pressures because they're measured relative to vacuum.

An alternative scale to atmospheric pressure is *gauge* pressure, which is measured relative to *atmospheric* pressure instead of vacuum. Gauge pressure is important because most of the pressure measurement devices you'll ever encounter report their measurements relative to at-

atmospheric pressure, and thus have units of gauge pressure. Gauge pressure  $P_g$  is related to absolute pressure  $P$  as

$$P = P_g + 1 \text{ atm} \quad (10.18)$$

Since  $P \geq 0$  always, Equation (10.18) implies that  $P_g \geq -1$  always.

Whenever the absolute pressure falls below atmospheric pressure, or equivalently whenever the gauge pressure drops below zero, the system can be said to be “under vacuum” or “under a vacuum.” It’s not uncommon for some intermediate portions of a flow system to be under vacuum but if the *outlet* of the system is under vacuum then things get a little trickier because such a design requires a vacuum pump to maintain this artificially low pressure, and that’s undesirable because of the price to purchase, operate, and maintain such pumps.

There can also be times when the calculated absolute pressure approaches 0, which is not possible in a real system. This occurs somewhat commonly during reactor design when Equation (10.15) is used for packed beds: for the right combinations of small particles, large beds, and small void fractions the absolute pressure can sharply decrease to zero. The only thing to conclude from such a result is that at least some of the parameters for that design will need to change in order to produce a physically realizable system.

### Example 10.3: Pressure drop in a packed bed

Plot the pressure drop of liquid through two different 10 cm diameter tubes, one smooth and empty and the other packed with 0.5 cm diameter glass spheres ( $\rho_{\text{cat}} = 2.5 \text{ kg} \cdot \text{L}^{-1}$ ) with a void fraction of 39%. Both tubes are 2 m in length, the inlet pressure is 1.1 bar (absolute), and the total mass flow rate either fluid is  $1 \text{ kg} \cdot \text{s}^{-1}$ . Interpret any unusual features of the resulting plots.

The density and viscosity of water are constant and equal to  $1000 \text{ kg} \cdot \text{m}^{-3}$  and  $10^{-5} \text{ Pa} \cdot \text{s}$ .

### Solution

The Reynolds number is

$$\text{Re} = \frac{4m_T}{\pi d \mu} = \frac{4 \times 1 \text{ kg} \cdot \text{s}^{-1}}{\pi \times 0.1 \text{ m} \times 10^{-5} \text{ Pa} \cdot \text{s}} = 1273,$$

which is a dimensionless number. The friction factor  $f$  for the empty tube (the PFR

case) is

$$f = \frac{0.0791}{\text{Re}^{1/4}} = \frac{0.0791}{1273^{1/4}} = 0.0074$$

and the friction factor  $f_p$  for the packed tube (the PBR case) is

$$f_p = \left[ 160 + 3 \left( \frac{1273}{1 - 0.39} \right)^{0.9} \right] \frac{(1 - 0.39)^2}{0.39^3 \times 1273} = 11.49.$$

Since  $v = v_0 = m_T/\rho = 0.001 \text{ m}^3 \cdot \text{s}^{-1}$  for liquid water, everything on the right-hand sides of Equations (8.11) and (10.15) is constant. The ODE for pressure drop in a smooth tube is

$$\begin{aligned} \frac{dP}{dV} &= -vf \frac{128m_T}{\pi^3 d^7} = -0.001 \times 0.0074 \frac{128 \times 1}{\pi^3 \times 0.1^7} \\ &= -307.4 \text{ Pa} \cdot \text{m}^{-3} \end{aligned}$$

which is integrated to yield

$$P = P_0 - (307.4 \text{ Pa} \cdot \text{m}^{-3})V \quad (10.19)$$

Similarly the ODE for pressure drop in a packed tube is

$$\begin{aligned} \rho_{\text{bed}} \frac{dP}{dW} &= -vf_p \frac{64m_T}{\pi^3 d^6 d_p} = -0.001 \times 11.49 \frac{64 \times 1}{\pi^3 \times 0.1^6 \times 0.005} \\ &= -4.74 \times 10^6 \text{ Pa} \cdot \text{m}^{-3} \end{aligned}$$

which is integrated to yield

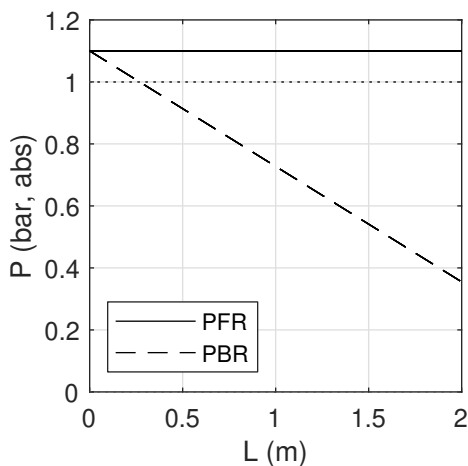
$$P = P_0 - (4.74 \times 10^6 \text{ Pa} \cdot \text{m}^{-3}) \frac{W}{\rho_{\text{bed}}}$$

or, equivalently,

$$P = P_0 - (4.74 \times 10^6 \text{ Pa} \cdot \text{m}^{-3})V \quad (10.20)$$

since  $W = \rho_{\text{bed}} V$ . We choose to use  $V$  instead of  $W$  for this particular problem because it's easier to compare the two pressure drops on the same plot if they're functions of the same independent variables. Better yet, we can calculate Equations (10.19) and (10.20) as functions of  $V$  but plot them as functions of  $L$  by recalling from Section 10.2 that  $L = V/A_c = 4V/\pi d^2$ . Plots as functions of  $L$  can be a little more intuitive to think about

because it's somewhat easier to visualize length down a reactor rather than volume along a reactor:



Focusing on the PFR curve first we see that there is no appreciable drop in pressure across the pipe: at  $V = 2 \text{ m}^3$  the pressure is 1.09995 bar, only fractionally less than the inlet pressure of 1.10000 bar. The outlet of the PFR is also well above atmospheric pressure, so this system is fairly normal.

The packed bed (PBR) curve tells a different story: within the first 0.25 m of the reactor the pressure has fallen below atmospheric pressure, implying that the system will require a vacuum pump (or other vacuum source) in order to operate.

If the bed is extended too much beyond 2 m then the pressure will continue to fall until it reaches 0 bar at

$$0 = P_0 - (4.74 \times 10^6 \text{ Pa} \cdot \text{m}^{-3}) V_{\max} \quad \Rightarrow \quad L_{\max} \approx 2.69 \text{ m}.$$

This value implies that there is no method of operating the bed for lengths greater than 2.69 m because the pressure falls below absolute zero (perfect vacuum). Should bed lengths greater than 2.69 m (or bed volumes greater than  $0.0211 \text{ m}^3$ ) be required, some aspect of the design will have to be changed to maintain  $P > 0$  throughout the bed, and preferably  $P > 1 \text{ atm}$  throughout the bed.

Example 10.3 demonstrates that pressure drop in packed beds can become quite significant even for non-reacting systems, far more so than we saw with plug flow reactors in Section 8.2. The situation is complicated a bit further for gas-phase reacting systems because the volumetric



flow rate  $v$  changes along the length of the reactor according to

$$v = v_0 \frac{n_T}{n_{T0}} \frac{P_0}{P} \frac{T}{T_0}, \quad (5.5)$$

but as we've seen repeatedly throughout this text it's fairly straightforward to calculate  $v$  at every step of the integration using MATLAB. In most cases the viscosity  $\mu$  will be constant as well, meaning that the friction factor  $f_p$  is also constant. In cases of considerable temperature change—more than about 100 °C—this assumption should be revisited, and if the viscosity changes by more than about 10-15% over this temperature range then an expression for  $\mu(T)$  should be used in Equation (8.13).<sup>4</sup>

When applied to a reacting system Equation (10.15) becomes an additional ODE coupled to the usual material and energy balance ODEs, and pressure becomes an additional independent variable. Let's revisit the adiabatic ammonia process to see how Equation (10.15) can be applied to a reacting system.

#### Example 10.4: Industrial production of ammonia, revisited

Repeat Example 9.5, now taking into account pressure drop within the reactor according to the KTA expression for the friction factor  $f_p$ . The reactor diameter is 2.7 m, the viscosity of the process gas stream is about  $3.1 \times 10^{-5} \text{ Pa} \cdot \text{s}$  (constant), and various properties of the catalyst are summarized below.

In addition to plots of all molar flow rates and bed temperature, also plot the pressure as a function of catalyst weight.

$$\rho_{\text{cat}} = 3700 \text{ kg} \cdot \text{m}^{-3}$$

$$\rho_{\text{bed}} = 2200 \text{ kg} \cdot \text{m}^{-3}$$

$$d_p = 2 \text{ cm}$$

#### Solution

The total mass flow rate  $m_T$  is always constant regardless of whether a chemical

<sup>4</sup>An excellent resource for such expressions is B.E. Poling; J.M. Prausnitz; J.P. O'Connell. Properties of gases and liquids, 5th ed. McGraw-Hill, New York (2001).

reaction occurs, and we can calculate it from the total inlet molar flow rate as

$$m_T = n_{T0} \overline{MM} = n_{T0} \sum y_{i0} MM_i,$$

where  $y_i$  and  $MM_i$  are the inlet mole fraction and molar mass of species  $i$ . The 30 000 kmol · h<sup>-1</sup> feed stream from Example 9.5 contains 30 mol% N<sub>2</sub>, 65 mol% H<sub>2</sub>, and 5 mol% NH<sub>3</sub>, and therefore the total mass flow rate  $m_T$  is

$$m_T = 30000(0.30(14) + 0.65(2) + 0.05(17)) = 190\,500 \text{ kg} \cdot \text{h}^{-1} = 52.92 \text{ kg} \cdot \text{s}^{-1}$$

where we converted from kg · h<sup>-1</sup> to kg · s<sup>-1</sup> in order to match units on viscosity. The Reynolds number is

$$\text{Re} = \frac{4(52.92 \text{ kg} \cdot \text{s}^{-1})}{\pi(2.7 \text{ m})(3.1 \times 10^{-5} \text{ Pa} \cdot \text{s})} = 8.05 \times 10^5 \approx 8 \times 10^5$$

which is dimensionless. That's a pretty big Reynolds number as is somewhat larger than the upper bound of Equation (10.17). If we were being critical we'd have to go find another expression for  $f_p$  that's valid up to about 10<sup>6</sup> but we'll use Equation (10.17) anyway but make a mental note that we're probably introducing additional uncertainty into our results by doing so.

We also need the void fraction, which we calculate from Equation (9.6) as

$$\rho_{\text{bed}} = (1 - \phi)\rho_{\text{cat}} \quad \Rightarrow \quad \phi = 1 - \frac{\rho_{\text{bed}}}{\rho_{\text{cat}}} = 1 - \frac{2200}{3700} = 0.405.$$

From Equation (10.17) the particle friction factor  $f_p$  is

$$f_p = \left[ 160 + 3 \left( \frac{8 \times 10^5}{1 - 0.405} \right)^{0.9} \right] \frac{(1 - 0.405)^2}{0.405^3 \times 8 \times 10^5} = 411.8$$

The inlet volumetric flow rate can be calculated from the ideal gas law as

$$v_0 = \frac{n_{T0}RT_0}{P_0} = 1.488 \text{ m}^3 \cdot \text{s}^{-1}$$

where we've chosen base SI units of m<sup>3</sup> · s<sup>-1</sup> in anticipation of using  $v_0$  to calculate  $v$  within the reactor. We could have equally chosen to calculate  $v$  at any point in the reac-

tor by using the ideal gas equation directly instead of Equation (5.5) but this approach was taken to highlight its similarity to previous approaches.

Rearranging Equation (10.15) yields

$$\begin{aligned}\frac{dP}{dW} &= -v \frac{f_p}{\rho_{\text{bed}}} \frac{64m_T}{\pi^3 d^6 d_p} \\ &= -v \frac{411.8}{2200 \text{ kg}_{\text{cat}} \cdot \text{m}^{-3}} \frac{64(52.92 \text{ kg} \cdot \text{s}^{-1})}{\pi^3 (2.7 \text{ m})^6 (0.02 \text{ m})} \\ &= (-2.439 \text{ kg} \cdot \text{kg}_{\text{cat}}^{-1} \cdot \text{s}^{-1} \cdot \text{m}^{-4})v\end{aligned}$$

Notice the unusual units on the coefficient  $-2.439$ : the  $\text{kg}$  and  $\text{kg}_{\text{cat}}$  don't cancel out because one is referring to the process stream and the other is referring to the catalyst. When  $v$  is calculated with units of  $\text{m}^3 \cdot \text{s}^{-1}$  then  $dP/dW$  will have units of  $\text{Pa} \cdot \text{kg}_{\text{cat}}^{-1}$ , which we'll then convert to bar by dividing by  $10^5$  to match the units of pressure used elsewhere in the problem.

All of these calculations are typically more convenient to do within the MATLAB solution script but they were worked out in detail here to demonstrate which numbers go where. Their addition to the solution script from Example 9.5 is provided below; unnecessary elements were retained but commented out to provide a comparison of how the code changes with and without pressure effects.

```

1 % ammoniaPBR2.m
2 T0 = 340+273; % K
3 nT0 = 30000; % kmol/h
4 P0 = 290; % bar % == NEW ==
5 y0 = [0.30; 0.65; 0.05];
6 n0 = nT0*y0;
7 Y0 = [n0; T0; P0]; % [H2 N2 NH3 T bar] % == MODIFIED ==
8 Wspan = [0 33440]; % kgcat
9 [W, Y] = ode45(@fun, Wspan, Y0);
10 subplot 131; plot(W, Y(:, 1:3)); % == MODIFIED ==
11 subplot 132; plot(W, Y(:, 4)); % == MODIFIED ==
12 subplot 133; plot(W, Y(:, 5)); % == NEW ==
13
14 function dYdW = fun(W, Y)
15 n = Y(1:3); % kmol/h, [N2 H2 NH3]
16 T = Y(4); % K
17 Ptot = Y(5); % bar, == NEW ==
18 P0 = 290; % bar
19 P = n/sum(n)*Ptot; % bar, [N2 H2 NH3] % == MODIFIED ==
20 rhobed = 2200; % kgcat/m3

```

```

21 v0 = 1.488; % m3/s % == NEW ==
22 nT0 = 30000; % kmol/h % == NEW ==
23 T0 = 340+273; % K % == NEW ==
24
25 dHf = [0; 0; -45.9]*1e3; % kJ/kmol
26 Cp = [31; 29; 48]; % kJ/kmol-K
27 nu = [-1; -3; 2]; % mol/mol reacted
28 dCp = sum(nu.*Cp); % kJ/kmol-K
29 dH = sum(nu.*dHf) + dCp*(T-298); % kJ/kmol
30 v = v0*(sum(n)/nT0)*(P0/Ptot)*(T/T0); % m3/s, % == NEW ==
31
32 f = 4.75;
33 kf = 1.79e4*exp(-87090/(8.314*T));
34 kr = 2.57e16*exp(-198464/(8.314*T));
35 r = f*(kf*P(1)*(P(2)^1.5)/P(3) - kr*P(3)/(P(2)^1.5)); % kmol/kgcat-h
36
37 dndW = nu.*r./rhobed; % kmol_i/kgcat-h
38 dTdW = -r*dH/(rhobed*sum(n.*Cp)); % K/kgcat
39 dPdW = -2.439*v/1e5; % bar/kgcat % == NEW ==
40 dYdW = [dndW; dTdW; dPdW]; % == MODIFIED ==
41 end

```

Major changes to the code have been marked as either `%== MODIFIED ==` or `%== NEW ==` and are summarized as follows:

**Lines 4, 7, 10-12:** Added pressure terms to the solution portion of the script and modified the plots.

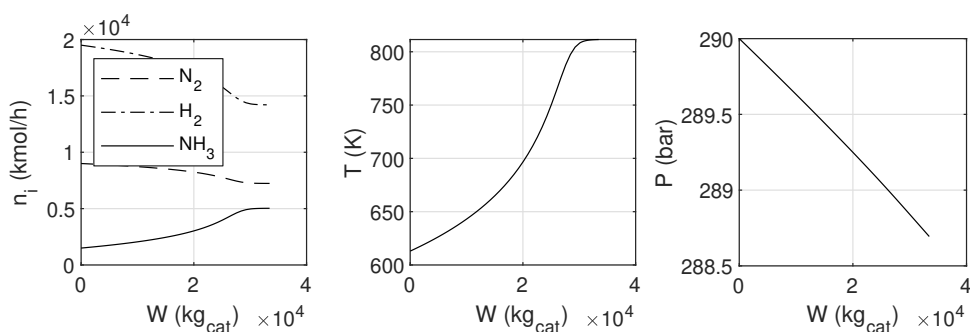
**Line 17:** Pressure  $P$  in the reactor was defined as  $P_{tot}$  to distinguish from the existing variable  $P$  for partial pressures.

**Line 19:** Since  $P$  is no longer constant, partial pressures are calculated based on the pressure anywhere in the reactor,  $P_{tot}$ .

**Lines 21-23, 30:** Terms for volumetric flow rate  $v$  were added.

**Lines 39, 40:** Terms for pressure drop ODE are generated and appended to the  $dYdW$  vector of ODE terms.

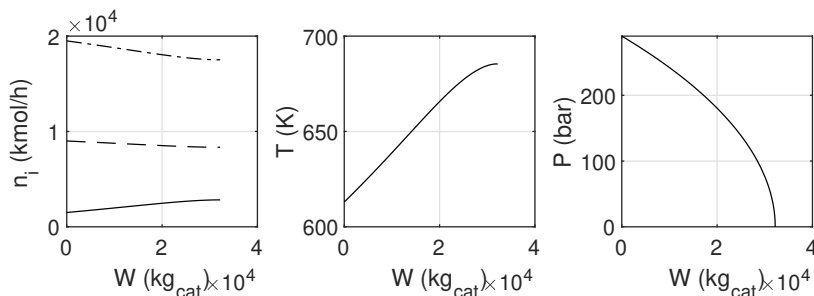
Naturally if we had chosen to do the  $f_p$  calculations within the script then the code would have needed additional lines, and these would have gone within the local function `fun`, anywhere between Lines 18 and 39. Running the script produces the following plots:



There's no appreciable change to any of the molar flow rates or the temperature profile, and the pressure profile reveals a fairly modest drop of a little less than 1.5 bar.

Entering the code to calculate  $f_p$  into MATLAB instead of calculating  $f_p$  by hand would allow us to investigate a number of interesting questions regarding pressure effects, including the remarkable dependence of pressure drop on reactor diameter. The Reynolds number varies linearly with reactor diameter and the friction factor  $f_p$  varies close to linearly with reactor diameter, but look at the denominator of Equation (10.15): the rate of pressure drop scales with reactor diameter as  $d^6$ ! This, coupled with the linear dependence of  $f_p$  on  $d$ , means  $dP/dW$  scales with reactor diameter as  $d^7$ !

For example, reducing the diameter by a factor of two will double the Reynolds number to  $16 \times 10^5$  and increase the friction factor  $f_p$  by nearly double to 768.4, but the coefficient  $-2.439$  in  $dP/dW$  changes by a factor of about  $2^7$ , up to  $-291.3$ : an increase of more than *two orders of magnitude*! This is indeed a remarkable increase, one so large that in fact the reactor can no longer operate as evidenced from the pressure profile of this modified reactor:



The pressure drop is so severe that the calculated pressure falls to zero before the end of the bed, which means we no longer have a viable design! Back to the drawing board!

Can you find the minimum diameter necessary to operate the reactor under these conditions?



# Chapter 11

## Multireactor Systems

“First rule in government spending: why build one when you can have two at twice the price?”

S.R. Hadden, *Contact* (1997)

Every reactor we’ve investigated up to now has operated in isolation as a single-unit process. As should have been evident from your material and energy balances course most processes require more than one *unit*, and some processes also require more than one *reactor*. This chapter introduces two tools that can help you design and evaluate multireactor systems, the Levenspiel plot and the interstage heat exchanger plot.<sup>1</sup> The Levenspiel plot arises from the similarities of *material balances* for different reactors; the interstage heat exchanger plot arises from the similarities of *energy balances* for different reactors.

These two tools are accompanied by several, seemingly severe assumptions designed to allow quick estimates of multireactor process features, and therefore these shortcut tools do not (and should not) replace a thorough reactor design. Instead they are a precursor to the rigorous analyses that we’ve presented in previous chapters, one that allows you to quickly identify promising types, sizes, and arrangements of individual reactors and heat exchangers within a multireactor process.

Our first significant assumption is that *there are no side streams*: the outlet from one reactor is sent directly to the following reactor with no addition or removal of material, although we do allow the temperature to change by means of a heat exchanger.

Our second significant assumption is that only *one* reaction occurs in any reactor in the system, and that it’s the *same* reaction in all reactors. This allows us to leverage the conver-

---

<sup>1</sup>Nobody has come up with a clever name for the “interstage heat exchanger plot.”

sion  $X_A$  as a powerful tool to evaluate multireactor designs in terms of their effect on this single parameter: designs which yield higher  $X_A$  with smaller reactor volume are generally favored compared to designs which yield lower  $X_A$  with larger reactor volume. To increase the usefulness of  $X_A$  we first consider how it's applied to multireactor systems.

## 11.1 Overall conversion is the conversion achieved by the entire system

Let's start with a quick review of conversion from Chapter 2, specifically when we defined  $X_A$  as

$$X_A = \frac{A \text{ in} - A \text{ out}}{A \text{ in}} = \frac{n_{A0} - n_A}{n_{A0}}. \quad (2.33)$$

When we say "A in" and "A out" in Equation (2.33) we're referring specifically to the inlets and outlets of a *single* reactor. This is somewhat similar to the concept of "single-pass" conversion from your material and energy balances, and owing to this similarity we now introduce the "sp" superscript such that Equation (2.33) becomes

$$X_A^{\text{sp}} = \left[ \frac{n_{A0} - n_A}{n_{A0}} \right]_{\text{single reactor}} \quad (11.1)$$

where the specification of "single reactor" is to remind us that the notation of  $n_{A0}$  and  $n_A$  as the inlet and outlet for that *particular* reactor: if the streams are numbered differently from 0 and 1 as the inlet and outlet then we still apply Equation (11.1) but we update the subscripts to represent the appropriate streams. We can also add a numerical subscript to  $X_A^{\text{sp}}$  to indicate which reactor is being referenced, such as

$$X_{A,k}^{\text{sp}} = \left[ \frac{n_{A0} - n_A}{n_{A0}} \right]_{X_A^{\text{sp}} \text{ for reactor } k} \quad (11.2)$$

Let's now recall how to define conversion for a system containing more than one unit: the *overall* conversion,  $X_A^{\text{ov}}$ , is the conversion of A across the entire process,

$$X_A^{\text{ov}} = \left[ \frac{n_{A0} - n_A}{n_{A0}} \right]_{\text{entire process}} \quad (11.3)$$

Here again we're using  $n_{A0}$  and  $n_A$  in a flexible manner: we use  $n_{A0}$  to represent the inlet to the process and we should feel free to change the subscript from 0 to whatever is appropriate based on the drawing, and similarly with  $n_A$  we can add subscripts to make clear our meaning



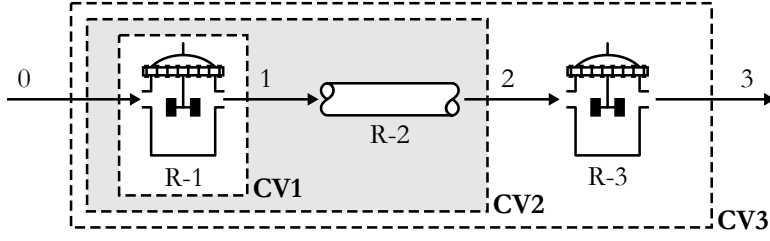


Figure 11.1: Several different control volumes (CVs) are possible for a system of three reactors in series, three of which are shown here.

about which stream is the outlet of the process. Assuming the reactors are in series we can also define the overall conversion after the feed stream has passed through unit  $k$  as

$$X_{A,k}^{\text{ov}} = \left[ \frac{n_{A0} - n_A}{n_{A0}} \right]_{X_A^{\text{ov}} \text{ from feed through reactor } k} \quad (11.4)$$

Consider the first control volume CV1 for the series of three reactors shown in Figure 11.1: for this particular CV the inlet stream is labeled 0 and the outlet stream is labeled 1, so the single-pass and overall conversions are denoted as

$$X_{A,R-1}^{\text{sp}} = \frac{n_{A0} - n_{A1}}{n_{A0}} \quad \text{and} \quad X_{A,R-1}^{\text{ov}} = \frac{n_{A0} - n_{A1}}{n_{A0}}.$$

For a single reactor we see that the single pass conversion  $X_A^{\text{sp}}$  is identical to the overall conversion  $X_A^{\text{ov}}$ . For the second control CV2 we have two single-pass conversions, one for R-1 and another for R-2, but only one overall conversion, this time comparing stream 2 to stream 0 as

$$\begin{aligned} X_{A,R-1}^{\text{sp}} &= \frac{n_{A0} - n_{A1}}{n_{A0}} & \text{and} & & X_{A,R-2}^{\text{ov}} &= \frac{n_{A0} - n_{A2}}{n_{A0}} \\ X_{A,R-2}^{\text{sp}} &= \frac{n_{A1} - n_{A2}}{n_{A1}} \end{aligned}$$

Finally, for CV3 we have three single-pass conversions—one each for R-1, R-2, and R-3—and again only one overall conversion as

$$\begin{aligned} X_{A,R-1}^{\text{sp}} &= \frac{n_{A0} - n_{A1}}{n_{A0}} \\ X_{A,R-2}^{\text{sp}} &= \frac{n_{A1} - n_{A2}}{n_{A1}} & \text{and} & & X_{A,R-3}^{\text{ov}} &= \frac{n_{A0} - n_{A3}}{n_{A0}} \\ X_{A,R-3}^{\text{sp}} &= \frac{n_{A2} - n_{A3}}{n_{A2}} \end{aligned}$$

The relationship between single-pass and overall conversion is not always trivial, but one thing is clear: you should *never* assume that the overall conversion is a simple sum or product of the single-pass conversions. The next example provides a sample derivation for the relationship between single-pass and overall conversion in a serial arrangement of reactors.

### Example 11.1: Relating single-pass to overall conversion for serial reactors

Use Equations (11.2) and (11.4) to derive a relationship between  $X_{A,k}^{\text{sp}}$  and  $X_A^{\text{ov}}$  for the system of serial reactors shown in Figure 11.1. Calculate the overall conversion if the conversions in reactors R-1, R-2, and R-3 were 81%, 45%, and 27%.

### Solution

Let's build up the solution one step at a time. The overall conversion after R-1 is the same as the single-pass conversion across R-1,

$$X_{A,R-1}^{\text{ov}} = \frac{n_{A0} - n_{A1}}{n_{A0}} = X_{A,R-1}^{\text{sp}},$$

and the overall conversion after R-2 is

$$X_{A,R-2}^{\text{ov}} = \frac{n_{A0} - n_{A2}}{n_{A0}},$$

We can now relate  $n_{A2}$  to  $n_{A1}$  by  $X_{A,R-2}^{\text{sp}}$  as

$$X_{A,R-2}^{\text{sp}} = \frac{n_{A1} - n_{A2}}{n_{A1}} \quad \Rightarrow \quad n_{A2} = n_{A1} (1 - X_{A,R-2}^{\text{sp}})$$

and  $n_{A1}$  to  $n_{A0}$  by  $X_{A,R-1}^{\text{sp}}$  as

$$X_{A,R-1}^{\text{sp}} = \frac{n_{A0} - n_{A1}}{n_{A0}} \quad \Rightarrow \quad n_{A1} = n_{A0} (1 - X_{A,R-1}^{\text{sp}}).$$

Substituting these expressions for  $n_{A2}$  and  $n_{A1}$  into  $X_{A,R-2}^{ov}$  yields

$$\begin{aligned}
 X_{A,R-2}^{ov} &= \frac{n_{A0} - n_{A2}}{n_{A0}} \\
 &= \frac{n_{A0} - n_{A1}(1 - X_{A,R-2}^{sp})}{n_{A0}} \\
 &= \frac{n_{A0} - n_{A0}(1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp})}{n_{A0}} \\
 &= 1 - (1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp})
 \end{aligned}$$

Building on this pattern, the overall conversion after R-3 is

$$X_{A,R-3}^{ov} = 1 - (1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp})(1 - X_{A,R-3}^{sp})$$

If the conversions in reactors R-1, R-2, and R-3 were 81%, 45%, and 27% then the overall conversion of A for the entire system would be

$$\begin{aligned}
 X_{A,R-3}^{ov} &= 1 - (1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp})(1 - X_{A,R-3}^{sp}) \\
 &= 1 - (1 - 0.81)(1 - 0.45)(1 - 0.27) \\
 &= 0.9237
 \end{aligned}$$

Notice that the overall conversion cannot exceed 100%, which is intuitively reasonable.

The result in Example 11.1 generally only extends for reactors in series wherein the outlet of one reactor is provided as the inlet to the next reactor with no addition or removal of material. If the system involves a mixture of parallel and series arrangements or recycle streams then the relationship between the overall and single-pass conversions is not so readily identified. Fortunately, we can still use the concept of the overall conversion as a useful tool in multireactor design even in such situations.

## 11.2 Under limited conditions the adiabatic energy balances simplify to one expression

Let's now take a look at how the energy balance simplifies under adiabatic conditions, which we'll use in the next section to help us design series of reactors using the overall conversion. Two assumptions made throughout this text—constant heat capacities and negligible shaft

work—continue to be applied. In addition to these we further assume that the *change* in heat capacity is zero,  $\Delta C_p = 0$ . This has two effects:  $\Delta H$  is constant, and a few terms will drop out of the subsequent equations.

The CSTR energy balance is the most straightforward to deal with so we start there. We saw in Section 5.2 that the energy balance for a CSTR in which only one reaction occurs was

$$X_A = \frac{\sum \Theta_i C_{p,i} (T - T_0) - \frac{Q}{n_{A0}}}{\frac{\Delta H}{\nu_A}}. \quad (5.9)$$

With  $Q = 0$  and  $\Delta H = \text{constant}$  our simplified expression for  $X_A$  becomes

$$X_A = \frac{\nu_A \sum \Theta_i C_{p,i} (T - T_0)}{\Delta H} \quad \begin{array}{l} \text{adiabatic, 1 reaction} \\ C_{p,i} = \text{constant} \\ W_S = Q = \Delta C_p = 0 \end{array} \quad (11.5)$$

The PFR case takes slightly more effort but ends up in the same place. The PFR energy balance for a single reaction in terms of conversion is

$$\frac{dT}{dV} = \frac{\delta Q - r \Delta H}{n_{A0} \left[ \sum \Theta_i C_{p,i} - \frac{X_A}{\nu_A} \Delta C_p \right]}, \quad (8.3)$$

and with  $\delta Q = \Delta C_p = 0$  this becomes

$$\frac{dT}{dV} = \frac{-r \Delta H}{n_{A0} \sum \Theta_i C_{p,i}}. \quad (11.6)$$

With  $\Delta H = \text{constant}$  we can separate and integrate this expression to yield

$$\int_{T_0}^T dT = -\frac{\Delta H}{n_{A0} \sum \Theta_i C_{p,i}} \int_0^V r dV. \quad (11.7)$$

The PFR material balance in terms of conversion is

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}} = -\frac{\nu_A r}{n_{A0}} \quad (8.2)$$

which we can rearrange to find the reaction rate  $r$  to be

$$r = -\frac{n_{A0}}{\nu_A} \frac{dX_A}{dV} \quad (11.8)$$

Substituting Equation (11.8) into the volume integral of Equation (11.7) yields

$$\int_0^V r dV = -\frac{n_{A0}}{\nu_A} \int_0^V \frac{dX_A}{dV} dV = -\frac{n_{A0}}{\nu_A} \int_0^{X_A} dX_A = -\frac{n_{A0}X_A}{\nu_A}, \quad (11.9)$$

and therefore Equation (11.7) becomes

$$T - T_0 = \left( -\frac{\Delta H}{n_{A0} \sum \Theta_i C_{P,i}} \right) \left( -\frac{n_{A0}X_A}{\nu_A} \right), \quad (11.10)$$

Simplifying and rearranging to isolate  $X_A$  yields the familiar equation

$$X_A = \frac{\nu_A \sum \Theta_i C_{P,i} (T - T_0)}{\Delta H} \quad \begin{array}{l} \text{adiabatic, 1 reaction} \\ C_{P,i} = \text{constant} \\ W_S = Q = \Delta C_P = 0 \end{array} \quad (11.11)$$

That's a neat coincidence! The CSTR energy balance—Equation (11.5)—and the PFR energy balance—Equation (11.11)—simplify to *exactly* the same expression under these conditions. Indeed the result is identical even for the PBR, but since the PBR equations are so very similar to those of the PFR the derivation is not repeated a third time here.<sup>2</sup> Therefore for *any* reactor design we have the simplified energy balance as

$$\boxed{X_A^{\text{ad}} = \frac{\nu_A \sum \Theta_i C_{P,i} (T - T_0)}{\Delta H}} \quad \begin{array}{l} \text{CSTR, PFR, or PBR:} \\ \text{adiabatic, 1 reaction} \\ C_{P,i} = \text{constant} \\ W_S = Q = \Delta C_P = 0 \end{array} \quad (11.12)$$

where we've added the superscript “ad” to remind us that this is a special case of the energy balance restricted to any adiabatic reactor. Here we also see that conversion  $X_A^{\text{ad}}$  is linearly proportional to the reactor temperature  $T$ , a feature we'll use shortly. It can also be useful to

<sup>2</sup>If it's not clear why they're the same then you're encouraged to follow these steps again beginning from Equation (10.3).

rearrange Equation (11.12) to isolate  $T$  as

$$T = T_0 + \frac{X_A^{\text{ad}} \Delta H}{\nu_A \sum \Theta_i C_{P,i}} \quad \begin{array}{l} \text{CSTR, PFR, or PBR} \\ \text{adiabatic, 1 reaction} \\ C_{P,i} = \text{constant} \\ W_S = Q = \Delta C_P = 0 \end{array} \quad (11.13)$$

where we can again identify a linear relationship between  $T$  and  $X_A^{\text{ad}}$ .

### 11.3 Interstage heat exchangers can increase conversion of equilibrium-limited processes

We now investigate an interesting application of the overall conversion, Equation (11.4), and the adiabatic energy balance, Equation (11.12). The context is that of an *equilibrium-limited* reactor, which occurs when the rate of reaction is high enough that equilibrium or near-equilibrium conditions are achieved before the feed stream exits the reactor.

The outlet composition of an equilibrium-limited reactor is therefore the same as what we calculated way back in Section 2.7 for an equilibrium reactor, and indeed there's no appreciable difference in how to calculate the *outlet* of an equilibrium-limited reactor compared to an equilibrium reactor. The reason for the distinction at all is simply to indicate the difference between a real reactor—CSTR, PFR, or PBR—operating in such a way that equilibrium is achieved, and an entirely imaginary reactor—the equilibrium or Gibbs reactor—being used as a calculation tool.

Let's briefly recall the steps needed to calculate equilibrium conversion at any temperature  $T$  according to Chapter 2:

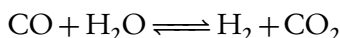
1. Calculate  $K_a(T) \approx K_0 K_1$  using the shortcut van't Hoff equation, Equation (2.29).
2. Equate  $K_a$  to  $\Pi a_i^{\nu_i}$  according to Equation (2.32) after using Table 2.3 to estimate appropriate  $a_i$  expressions.
3. At fixed pressure the expression developed above should be a function of only temperature  $T$  and conversion  $X_A^{\text{eq}}$ . Solve for  $X_A^{\text{eq}}$  at the desired  $T$ , as we did in either Example 2.5 or Example 2.6 in Section 2.8. Keep in mind that we refer to the calculated  $X_A$  value as  $X_A^{\text{eq}}$  to identify it as the conversion achieved at equilibrium.

With these two independent means of calculating conversion— $X_A^{\text{ad}}$  according to Equation (11.12) and  $X_A^{\text{eq}}$  from Section 2.8—we can take our analysis one step further as follows:

the conversion and operating temperature achieved of an adiabatic, equilibrium-limited reactor is given at the point where  $X_A^{\text{ad}}(T) = X_A^{\text{eq}}(T)$ . We can find this intersection point by either a root-finding algorithm like `fsolve` as in Section 1.5, or by plotting the two curves and looking for an intersection. Let's demonstrate the latter approach because we're going to do some crazy stuff with it in a moment!

### Example 11.2: Water-gas shift reaction

The water-gas shift reaction (WGSR) is an equilibrium-limited reaction which converts carbon monoxide and water to hydrogen and carbon dioxide according to the reversible, gas-phase reaction



At industrial scales the reaction is carried out in adiabatic, packed bed reactors over various catalysts such as platinum, copper, or nickel. Assuming an equimolar feed of CO and H<sub>2</sub>O at 1 mol · s<sup>-1</sup> each,

- Derive an expression for the equilibrium conversion  $X_A^{\text{eq}}$  of CO as a function of temperature.
- Derive an expression for the adiabatic conversion  $X_A^{\text{ad}}$  of CO as a function of temperature.
- Assuming the inlet temperature is 150 °C,<sup>a</sup> estimate the conversion of CO and the reactor outlet temperature by plotting the expressions for  $X_A^{\text{eq}}$  and  $X_A^{\text{ad}}$ .
- Evaluate the validity of the  $\Delta C_p = 0$  assumption by calculating the change in  $\Delta H$  over a relevant temperature range.

<sup>a</sup>Usually the inlet temperature is closer to something like 350 °C for this reaction.

### Solution

In anticipation of needing terms as functions of conversion we'll include them in our organizational stoichiometric table to get started. Heat capacities are evaluated at 500 K because that's about the middle of the temperature range over which significant variation occurs in the two  $X_A$  terms, something which would not become apparent until solving the problem at least once.

Sp.	Formula	$\nu_i$	$n_{i0}$	$\Theta_i$	$n_i$	$C_p$ (500 K) (J · mol <sup>-1</sup> · K <sup>-1</sup> )	$\Delta H_f^\circ$ (kJ · mol <sup>-1</sup> )	$\Delta G_f^\circ$ (kJ · mol <sup>-1</sup> )
A	CO	-1	1	1	$n_{A0}(1-X_A)$	30	-110.5	-137.2
B	H <sub>2</sub> O	-1	1	1	$n_{A0}(1-X_A)$	35	-241.8	-228.6
C	CO <sub>2</sub>	1	0	0	$n_{A0}X_A$	45	-393.5	-394.4
D	H <sub>2</sub>	1	0	0	$n_{A0}X_A$	29	0	0
$\Delta M = 9$							-41.2	-28.6

(a) The equilibrium conversion  $X_A^{\text{eq}}$  is related to the equilibrium constant  $K_a$  as

$$\begin{aligned}
 K_a \approx K_{sc} &= K_0 K_1 = \prod_{i=A}^Z P_i^{\nu_i} \\
 &= \frac{P_{\text{H}_2} P_{\text{CO}_2}}{P_{\text{CO}} P_{\text{H}_2\text{O}}} \\
 &= \frac{(\gamma_{\text{H}_2} P)(\gamma_{\text{CO}_2} P)}{(\gamma_{\text{CO}} P)(\gamma_{\text{H}_2\text{O}} P)} \\
 &= \frac{n_{\text{H}_2} n_{\text{CO}_2}}{n_{\text{CO}} n_{\text{H}_2\text{O}}} \frac{n_T n_T}{n_T n_T} \\
 &= \frac{X_A^{\text{eq}}}{(1 - X_A^{\text{eq}})^2},
 \end{aligned}$$

which we can rearrange to isolate  $X_A^{\text{eq}}$  using the quadratic equation<sup>a</sup> to get

$$X_A^{\text{eq}} = \frac{K_{sc} - \sqrt{K_{sc}}}{K_{sc} - 1}.$$

The shortcut van't Hoff equation for  $K_{sc}$ , Equation (2.29), can be used to find  $K_{sc}$  at any temperature followed by  $X_A^{\text{eq}}$  at any temperature using the expression above.

(b) The adiabatic conversion  $X_A^{\text{ad}}$  can be found from Equation (11.12) as

$$X_A^{\text{ad}} = \frac{-(C_{p,\text{CO}} + C_{p,\text{H}_2\text{O}})(T - T_0)}{\Delta H},$$

which we can easily plot as a function of temperature since everything in this expres-



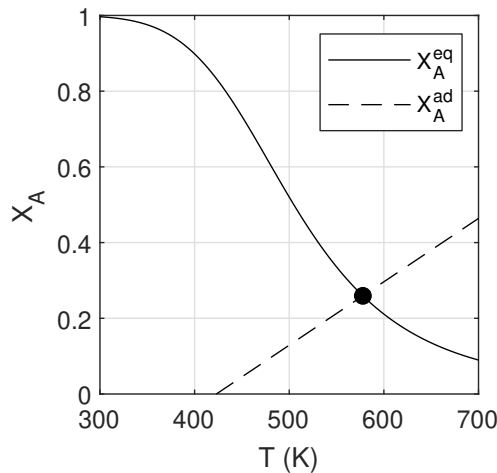
sion is constant except for  $T$  itself.

(c) The script below produces a plot of  $X_A^{\text{eq}}$  and  $X_A^{\text{ad}}$ :

```

1 % wgs.m
2 % vector pattern is [CO H2O CO2 H2]
3 nu = [-1 -1 1 1];
4 dGf = [-137.2 -228.6 -394.4 0]*1e3; % J/mol
5 dHf = [-110.5 -241.8 -393.5 0]*1e3; % J/mol
6 Cp = [31 38 49 29]; % J/mol-K
7
8 T0 = 150+273; % K
9 T = linspace(300, 700); % K
10 K0 = exp(-sum(nu.*dGf)./(8.314*T));
11 K1 = exp(sum(nu.*dHf)/8.314*(1/298-1./T));
12 Ksc = K0.*K1;
13 Xaeq = (Ksc-sqrt(Ksc))./(Ksc-1);
14 Xaad = -(Cp(1)+Cp(2))*(T-T0)/sum(nu.*dHf);
15
16 hold on
17 plot(T, Xaeq)
18 plot(T, Xaad)
19 hold off
20 ylim([0 1])

```



The two curves intersect at about 580 K with a corresponding conversion of about 26%.

(d) The change in heat capacity is

$$\Delta C_P = \sum_{i=A}^Z \nu_i C_{P,i} = 9 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1},$$

which is quite small all by itself, so our original assumption that  $\Delta C_P = 0$  should be fairly good for this system. At 298 K the enthalpy of formation is  $-41.2 \text{ kJ} \cdot \text{mol}^{-1}$ , and at 700 K the enthalpy of formation is

$$\Delta H = \Delta H_R + \Delta C_P (T - T_R) = -37.6 \text{ kJ} \cdot \text{mol}^{-1},$$

a decrease of about 10%. That's enough to be important in a detailed design calculation, but if we're making initial designs and doing things like reading values off figures then this is a perfectly acceptable change to ignore.

---

<sup>a</sup>We keep only the result for which  $0 \leq X_A^{\text{eq}} \leq 1$ .

The relatively low conversion found in Example 11.2 cannot be improved merely by making the reactor bigger or sending the reactor outlet stream into yet another reactor because we've reached thermodynamic equilibrium. However, hope is not lost! We stand on the shoulders of giants, and some of those giants have been particularly clever!

What we will do is send the reactor outlet stream through a heat exchanger to cool it down before sending it through yet another adiabatic, equilibrium reactor. When a heat exchanger is used in this way—whether to cool or to heat—we refer to it as an *interstage heat exchanger*.

The truly remarkable aspect of using an interstage heat exchanger is that we can do the entire analysis of such a scheme graphically in almost precisely the same manner as we did in Example 11.2, with one caveat: we have to use the overall conversion instead of a single-pass conversion. The *reason* for the change from single-pass conversion to overall conversions is that we want to keep using the same form of  $X_A^{\text{ad}}$  in Equation (11.12) with as few changes as possible. To illustrate this reasoning consider the two adiabatic reactor systems shown in Figure 11.2, wherein heat exchanger HX-1 cools the outlet from reactor R-1 back to the fresh feed inlet temperature  $T_0$ . For CV 1 containing only R-1, Equation (11.12) becomes

$$X_{A,R-1}^{\text{ad,ov}} = \frac{\nu_A \sum \Theta_i C_{P,i} (T_1 - T_0)}{\Delta H}, \quad (11.14)$$

where the subscript “R-1” implies that the overall conversion was calculated based on the outlet

of reactor R-1. For CV 2 containing R-1, HX-1, and R-2 Equation (11.12) becomes

$$X_{A,R-2}^{\text{ad,ov}} = X_{A,R-1}^{\text{ad,ov}} + \frac{\nu_A \sum \Theta_i C_{P,i} (T_3 - T_2)}{\Delta H}, \quad (11.15)$$

which, it's worth pointing out, is truly a remarkable result! Equation (11.15) says that the overall conversion after the second reactor is equal to the overall conversion after the first reactor, plus another  $\nu_A \sum \Theta_i (T_3 - T_2) / \Delta H$  term that looks almost exactly the same as the one in Equation (11.14) except that the temperatures have subscripts specific to reactor R-2. The derivation of Equation (11.15) is provided as something of an afterthought in Section 11.3.1 because it's a little bit tedious and we're more interested in how to use Equation (11.15) right now, rather than where it came from.

The key idea is that Equation (11.15) allows the overall conversion of adiabatic reactors combined in series to be determined *graphically* as follows:

**Reactor R-1:** Plot the *overall* equilibrium conversion line,  $X_A^{\text{eq,ov}}$ , and the *overall* adiabatic conversion line of the first reactor according to Equation (11.14). The intersection represents the outlet conditions of the first reactor.

**Heat exchanger HX-1:** To add a heat exchanger draw a horizontal line from the  $T_1$  to  $T_2$ .

**Reactor R-2:** Starting from the end of the previous line, draw the overall adiabatic conversion line *parallel* to the one drawn for Reactor R-1. The intersection of this line with  $X_A^{\text{eq,ov}}$  represents the outlet conditions of the second reactor.

**Additional units:** Continuing adding heat exchangers and reactors in the same manner as above until the desired overall conversion is achieved.

The reason the line drawn for R-2 is *parallel* to that of R-1 is that Equations (11.14) and (11.15) have the same slope! More subtly, the reason both reactor lines terminate on the same equi-

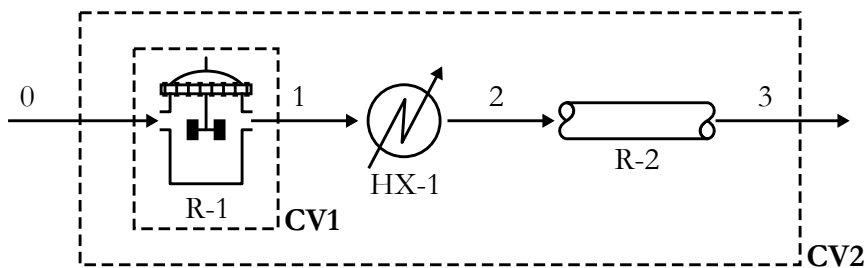


Figure 11.2: Two adiabatic reactors, R-1 and R-2, separated by a heat exchanger, HX-1. The style of the reactors—CSTR, PFR, or PBR—is arbitrary. The first control volume, CV1, encompasses only the first reactor, while the second control volume, CV2, encompasses both reactors and the heat exchanger.

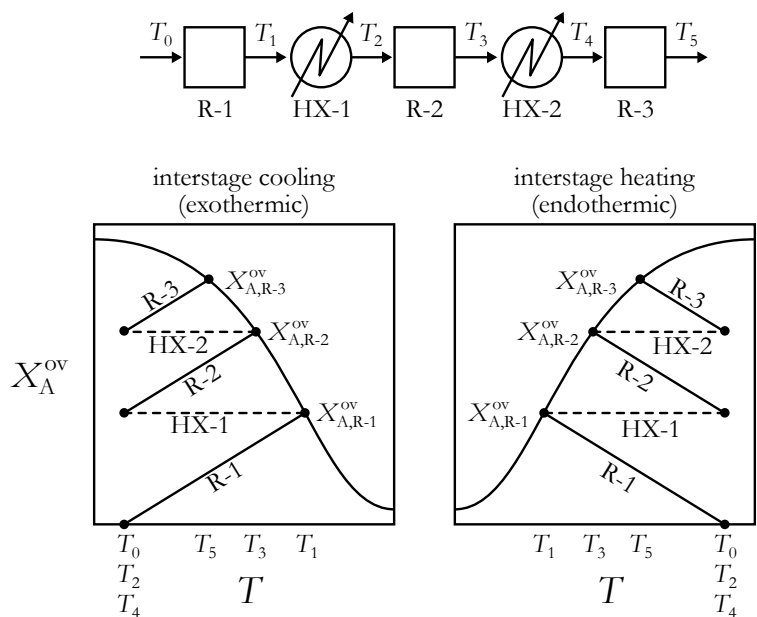


Figure 11.3: Hypothetical system consisting of three reactors and two heat exchangers. If the reaction is exothermic then the system is called *interstage cooling* (left); if the reaction is endothermic then the system is called *interstage heating* (right). Regardless of heating or cooling the general approach is similar.

librium line is because we’re calculating *overall* quantities, all of which are relative to the fresh feed. Certainly the approach need not be graphical—the equations can easily be solved analytically or numerically—but the graphical nature of the solution can be handy for making quick estimates.

Figure 11.3 demonstrates this process for a hypothetical series of reactors and heat exchangers. If the reaction is exothermic then the heat exchangers are used to decrease the process stream temperature between reactors, and the process is called *interstage cooling*. Similarly if the reaction is endothermic then the heat exchangers are used to increase the process stream temperature between reactors, and the process is called *interstage heating*. Notice that the heat exchanger outlet temperatures in Figure 11.3 are all the same ( $T_0 = T_2 = T_4$ ): this choice was simply to make the figure more readable because Equation (11.15) does not specify any relationship between exchanger outlet temperatures.

Let’s see how to use this to extend our water-gas shift reactor from Example 11.2.

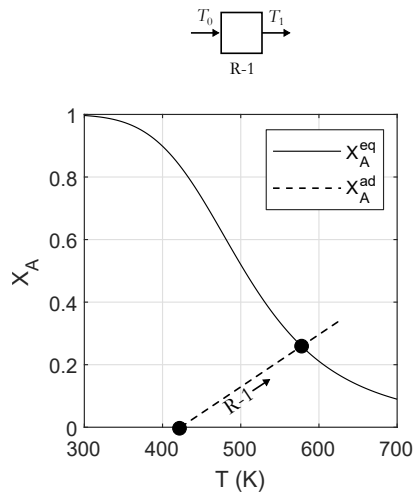
**Example 11.3: Water-gas shift reaction, improved**

As noted in Example 11.2, the water-gas shift reaction is equilibrium-limited. As-

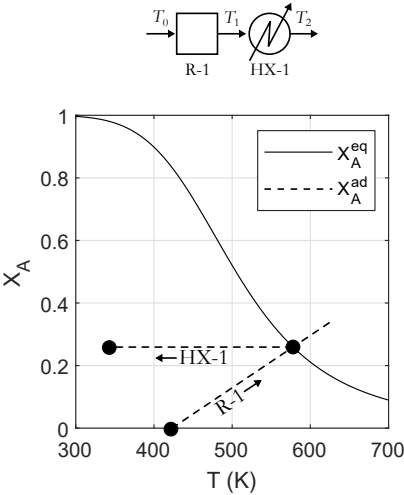
suming the same conditions as described in Example 11.2, design a series of adiabatic reactors with interstage cooling which provides an overall conversion in excess of 60%. Assume each heat exchanger returns the process stream to 350 K before it enters the next reactor. Summarize the inlet and outlet conditions of each equipment piece.

### Solution

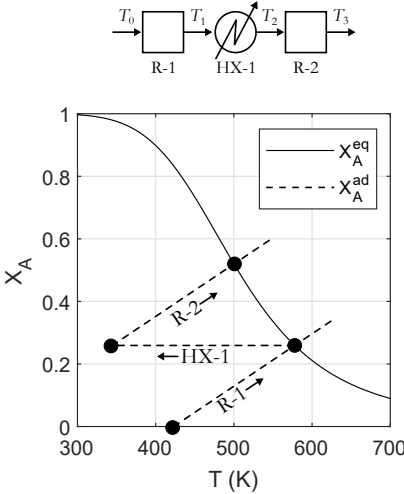
The reaction is exothermic so we expect to build a figure similar to the one shown in the left plot of Figure 11.3. The equilibrium curve  $X_A^{\text{eq}}$  and adiabatic reactor conversion curve  $X_A^{\text{ad}}$  were already constructed in Example 11.2 for the first reactor, so we begin there:



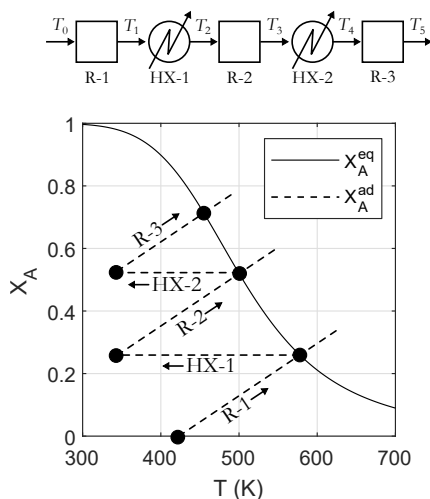
Next, a horizontal line is added for the first heat exchanger, HX-1. The overall conversion doesn't change within the heat exchanger but the outlet temperature is cooled to 350 K:



Third, a line for the second adiabatic reactor, R-2, is added by starting from the outlet of HX-1 and drawing a line at the same slope as the line for R-1 because according to Equation (11.15) both reactors will have the same slope but different inlet and outlet conversions and temperatures:



A second heat exchanger, HX-2, and a third adiabatic reactor, R-3, are added to bring the overall conversion above 60% as desired:



Estimates of the inlet and outlets of each unit can be read directly from the figure. For example, the overall conversion of A after reactor R-2 is a little more than 50% and the outlet temperature of this reactor is about 500 K.

You should keep two other things when designing interstage heating and cooling systems: the overall conversion never changes when the process stream passes through a heat exchanger, and the outlet temperatures of the heat exchangers can be set to different values without changing the general process of drawing lines on the figure.

### 11.3.1 Derivation of Equation (11.15)

Central to the graphical approach developed in Section 11.3 was Equation (11.15), which we now derive to reassure you that it's not magical and that no assumptions beyond those of Equation (11.12) are required. The system we'll use is the one in Figure 11.2, consisting of two adiabatic reactors of any style separated by a heat exchanger.

The overall conversion of A achieved after the second reactor, R-2, is given by Equation (11.4) as

$$X_{A,R-2}^{ov} = \frac{n_{A0} - n_{A3}}{n_{A0}} = 1 - (1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp}), \quad (11.16)$$

where the second equality was obtained by using the pattern we found in Example 11.1. Noting that  $X_{A,R-1}^{sp} = X_{A,R-1}^{ov}$ , Equation (11.16) can be expanded and rearranged to yield

$$X_{A,R-2}^{ov} = X_{A,R-1}^{ov} + X_{A,R-2}^{sp}(1 - X_{A,R-1}^{ov}). \quad (11.17)$$

We now play around with  $X_{A,R-2}^{\text{sp}}$ , and after we're done we'll put the results back into Equation (11.17) and see where it takes us. Equation (11.12) gives  $X_{A,R-2}^{\text{sp}}$  as

$$X_{A,R-2}^{\text{sp}} = \frac{\nu_A \sum \frac{n_{i1}}{n_{A1}} C_{P,i} (T_3 - T_2)}{\Delta H}, \quad (11.18)$$

where the  $\Theta_i$  term was expanded to more clearly indicate that the inlet to R-2 is stream 2 at  $T_2$ , and we replace  $n_{i2}$  with  $n_{i1}$  because streams 2 and 1 have the same composition. Substituting Equation (2.35) for the  $n_i$  terms in Equation (11.18) yields

$$X_{A,R-2}^{\text{sp}} = \frac{\nu_A \sum \frac{(\Theta_i + \frac{\nu_i}{\nu_A} X_{A,R-1}^{\text{ov}})}{1 - X_{A,R-1}^{\text{ov}}} C_{P,i} (T_3 - T_2)}{\Delta H} \quad (11.19)$$

where the  $\Theta_i$  term is now understood to refer to the fresh feed such that  $\Theta_i = n_{i0}/n_{A0}$  as we're accustomed to seeing. Expanding the sum in the numerator we have

$$X_{A,R-2}^{\text{sp}} = \frac{\nu_A \sum \Theta_i C_{P,i} (T_3 - T_2) + \sum \nu_i C_{P,i} (T_3 - T_2) X_{A,R-1}^{\text{ov}}}{\Delta H (1 - X_{A,R-1}^{\text{ov}})}. \quad (11.20)$$

Notice in the second numerator term that the quantity  $(T_3 - T_2) X_{A,R-1}^{\text{ov}}$  is the same for each element in the summation, and therefore it can be pulled in front of the summation to get

$$(T_3 - T_2) X_{A,R-1}^{\text{ov}} \sum \nu_i C_{P,i},$$

which is quite convenient because  $\sum \nu_i C_{P,i} = \Delta C_P = 0$ , and therefore this entire term can be eliminated from Equation (11.20) to yield

$$X_{A,R-2}^{\text{sp}} = \frac{\nu_A \sum \Theta_i C_{P,i} (T_3 - T_2)}{\Delta H (1 - X_{A,R-1}^{\text{ov}})}. \quad (11.21)$$

Substituting Equation (11.21) into Equation (11.17) yields Equation (11.15),

$$X_{A,R-2}^{\text{ad,ov}} = X_{A,R-1}^{\text{ad,ov}} + \frac{\nu_A \sum \Theta_i C_{P,i} (T_3 - T_2)}{\Delta H} \quad (11.15)$$

which is what we were after all along! A similar procedure can be applied for any additional reactors placed in series after R-2.



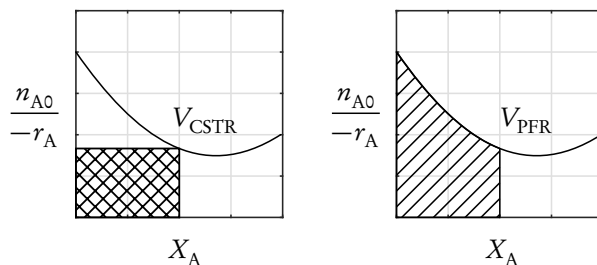


Figure 11.4: Basic concepts of the Levenspiel plot, consisting of a plot of  $(n_{A0}/-r_A)$  as a function of  $X_A$ . (left) The volume of a CSTR can be interpreted as the area of a rectangle with “width”  $X_A$  and “height”  $(n_{A0}/-r_A)$ . (right) The volume of a PFR can be interpreted as the area under the  $(n_{A0}/-r_A)$  curve from  $X_A = 0$  to the desired  $X_A$ .

## 11.4 Levenspiel plots can estimate the type, arrangement, and size of multireactor systems

There’s an interesting link between the material balances of CSTRs and PFRs that becomes apparent with a little rearrangement. The CSTR material balance in terms of conversion is

$$V = \frac{n_{A0}X_A}{-r_A} = \left( \frac{n_{A0}}{-r_A} \right) X_A, \quad (5.8)$$

where the second equality was used simply to regroup terms to make apparent the dependence of  $V$  on conversion  $X_A$  and the lumped term  $(n_{A0}/-r_A)$ . The reason for the lumpiness is because the same term is apparent when the PFR material balance,

$$\frac{dX_A}{dV} = \frac{-r_A}{n_{A0}}, \quad (8.2)$$

is rearranged slightly to

$$V = \int \left( \frac{n_{A0}}{-r_A} \right) dX_A, \quad (11.22)$$

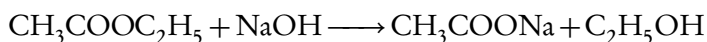
where again the lumped term  $(n_{A0}/-r_A)$  has appeared. Equation (11.22) implies that the volume needed to achieve a certain conversion  $X_A$  within a PFR can be interpreted as the area under the  $(n_{A0}/-r_A)$  curve, while Equation (5.8) implies that the volume needed to achieve a certain conversion  $X_A$  within a CSTR can be interpreted as the area of a rectangle of “width”  $X_A$  and “height”  $(n_{A0}/-r_A)$ . A similar comparison can be made for PBRs and fluidized CSTRs, but for simplicity we restrict our discussion to PFRs and CSTRs.

This graphical interpretation of the material balances for different reactors is shown in Figure 11.4 and is referred to as a *Levenspiel plot*. It can be a useful tool for quickly identifying

which reactor will have the smaller volume to achieve the same conversion and if an appropriate grid system is provided then volumes can be estimated directly from the plot, as we now demonstrate.

#### Example 11.4: Reactor selection for saponification

As we'll see in Chapter 13 the saponification of ethyl acetate ( $\text{CH}_3\text{COOC}_2\text{H}_5$ , A) with sodium hydroxide to produce ethanol according to the reaction



proceeds according to the second-order rate expression

$$r = kC_{A0}^2(1 - X_A)(1.25 - X_A),$$

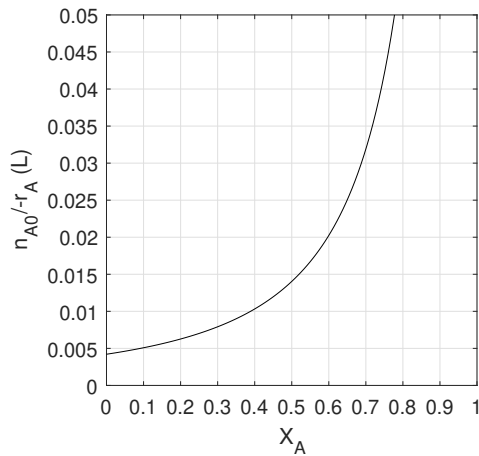
where  $k = 0.38 \text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1}$  at  $70^\circ\text{C}$ , and  $C_{A0} = 10 \text{ mol} \cdot \text{L}^{-1}$ . Create a Levenspiel plot for this reaction assuming the molar flow rate of ethyl acetate is  $12 \text{ mol} \cdot \text{min}^{-1}$ , and decide on the reactor which would require the smallest volume to achieve 70% conversion.

#### Solution

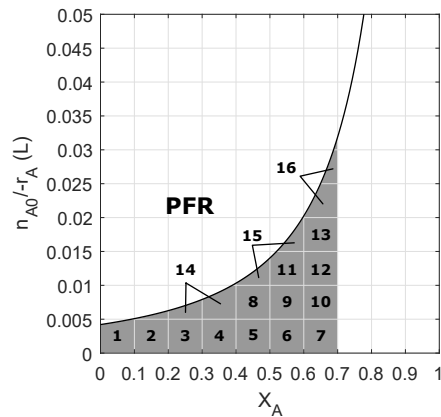
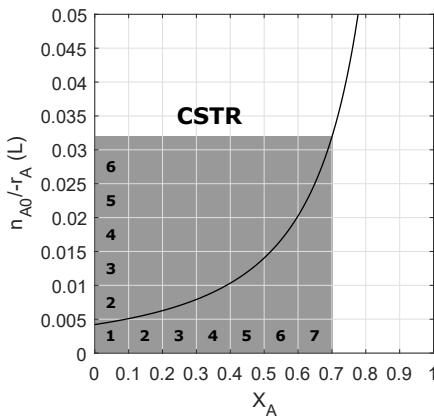
To create a Levenspiel plot requires a plot of  $n_{A0}/-r_A$  as a function of  $X_A$ , which for this reaction is

$$\frac{n_{A0}}{-r_A} = \frac{0.2 \text{ mol} \cdot \text{s}^{-1}}{r} = \frac{0.2 \text{ mol} \cdot \text{s}^{-1}}{38 \text{ mol} \cdot \text{L}^{-1} \cdot \text{s}^{-1}(1 - X_A)(1.25 - X_A)},$$

which we note has units of volume as liters, which is what the resulting units on the Levenspiel plot will be. To create the Levenspiel plot we simply plot this function over a range of  $X_A$  values, from 0 to close to just under 1 so that the denominator doesn't become infinite:



To estimate the CSTR volume we draw a square starting from  $X_A = 0.7$ , extending up to the  $n_{A0}/-r_A$  curve, and back to the y-axis; the area of that square corresponds to the volume of the CSTR in L. To estimate the PFR volume we shade in the area under the  $n_{A0}/-r_A$  curve up to  $X_A = 0.7$ ; the area of that shape corresponds to the volume of the PFR in L:



The plot grid was set to help our calculations because each grid square corresponds to

$$0.1 \times 0.005 \text{ L} = 0.0005 \text{ L} = 0.5 \text{ mL},$$

so if we count up the total number of grid squares in either shaded area and multiply by 0.5 mL then the result will be the volume of that reactor in mL.

For the CSTR the main portion of the shaded area is about 7 squares by 6 squares for a total of 42 squares, plus about 7 half-squares along the top edge for another 3.5 squares.

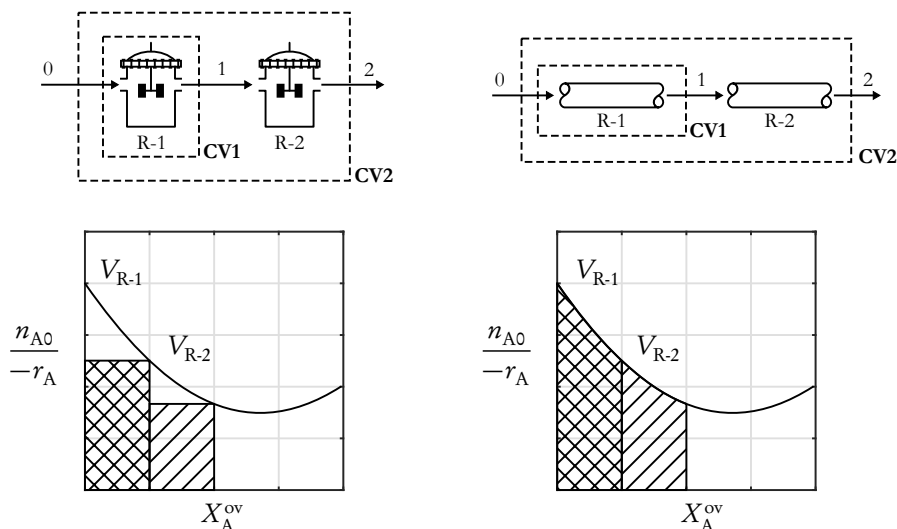


Figure 11.5: Levenspiel plots for CSTRs in series (*left*) and tubular reactors in series (*right*). The choice of overall conversion for individual reactors was arbitrary. Notice from the left plot that many small CSTRs will approximate one large PFR, and from the right plot that it's rather pointless to have two PFRs directly in series because they're identical to one large PFR.

Our estimate of the CSTR volume needed to achieve 70% conversion is therefore about  $(42 + 3.5) \times 0.5 \text{ mL} = 22.75 \text{ mL}$ .

For the PFR the main portion of the shaded area is about 13 squares, with about another 3 full squares (labeled 14, 15, and 16) along the curve itself. Our estimate of the PFR volume needed to achieve 70% conversion is therefore about  $(13 + 3) \times 0.5 \text{ mL} = 8 \text{ mL}$ .

The better choice is the PFR with a volume about 3 times less than the CSTR. Indeed the volume is quite small at just 8 mL, but since the flow is both highly concentrated ( $C_{A0} = 10 \text{ mol} \cdot \text{L}^{-1}$ ) and fairly slow (around  $15 \text{ g} \cdot \text{s}^{-1}$ ) this is not a terribly surprising result.

Example 11.4 illustrates the basic idea of the Levenspiel plot but the technique truly shines when it's applied to a series of reactors by means of the overall conversion,  $X_A^{ov}$ . Referring to the series of CSTRs in Figure 11.5, the volume of CSTR reactor R-1 as  $V_{R-1}$  is related to the overall conversion after R-1 as

$$V_{R-1} = \left( \frac{n_{A0}}{-r_A} \right) X_{A,R-1}^{ov}, \quad (11.23)$$

which is the same as Equation (5.8) with  $X_A$  replaced by  $X_A^{ov}$  as noted in Example 11.1. The

volume of CSTR reactor R-2 is

$$V_{R-2} = \frac{n_{A1}X_{A,R-2}^{sp}}{-r_A} = \frac{n_{A0}(1 - X_{A,R-1}^{sp})X_{A,R-2}^{sp}}{-r_A}, \quad (11.24)$$

and one of the results from Example 11.1 was

$$X_{A,R-2}^{ov} = 1 - (1 - X_{A,R-1}^{sp})(1 - X_{A,R-2}^{sp}) \quad \Rightarrow \quad (1 - X_{A,R-1}^{sp})X_{A,R-2}^{sp} = X_{A,R-2}^{ov} - X_{A,R-1}^{ov}, \quad (11.25)$$

which we substitute into Equation (11.24) to yield

$$V_{R-2} = \left( \frac{n_{A0}}{-r_A} \right) (X_{A,R-2}^{ov} - X_{A,R-1}^{ov}). \quad (11.26)$$

Equations (11.23) and (11.26) are interpreted graphically as shown in the left plot of Figure 11.5: the first CSTR is represented by a rectangle up to  $X_{A,R-1}^{ov}$  and the second CSTR is represented by a rectangle from  $X_{A,R-1}^{ov}$  to  $X_{A,R-2}^{ov}$ , and the areas of these two rectangles are the volumes of the respective CSTRs. A similar analysis for the series of PFRs in Figure 11.5 such that the volume of PFR reactor R-1 is

$$V_{R-1} = \int_0^{X_{A,R-1}^{ov}} \left( \frac{n_{A0}}{-r_A} \right) dX_A^{ov}, \quad (11.27)$$

and the volume of PFR reactor R-2 is

$$V_{R-2} = \int_{X_{A,R-1}^{ov}}^{X_{A,R-2}^{ov}} \left( \frac{n_{A0}}{-r_A} \right) dX_A^{ov}. \quad (11.28)$$

Equations (11.27) and (11.28) are interpreted graphically as shown in the right plot of Figure 11.5: the first PFR volume is represented by the area under the  $(n_{A0}/-r_A)$  curve up to  $X_{A,R-1}^{ov}$ , and the second PFR volume is represented by the area under the  $(n_{A0}/-r_A)$  curve from  $X_{A,R-1}^{ov}$  to  $X_{A,R-2}^{ov}$ .

Even more interestingly, the forms of Equations (11.26) and (11.28) imply that the reactors can be swapped around but the graphical approach retained because the volume and overall conversion of the new reactor depends only on the overall conversion of the previous reactor, not its type as either CSTR or PFR.

Consequently the type, volume, and increase in overall conversion for each reactor in a se-

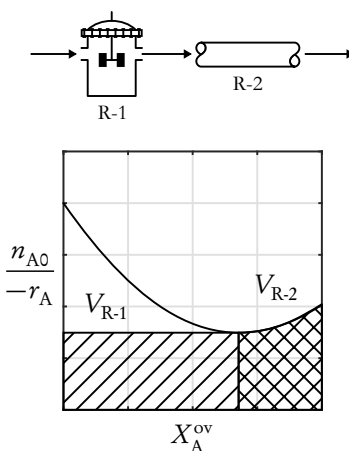


Figure 11.6: The arrangement of reactors which minimizes the total volume needed to maximize overall conversion for the Levenspiel plot of Example 11.4 is a CSTR followed by a PFR.

ries can then be quickly estimated simply from the shape of the  $(n_{A0}/-r_A)$  curve. The same curve from the plots in Figure 11.5 is reproduced in Figure 11.6 with the minimal total volume given by a CSTR followed by a PFR. With knowledge of the actual  $(n_{A0}/-r_A)$  curve, volume estimates of each reactor could quickly be found in the same manner as Example 11.4. Knowing even the type and arrangement of the reactors, if not their volume, can be of considerable help during the design process, as we now demonstrate.

### Example 11.5: Reactor optimization for reversible, exothermic reactions

Reversible, exothermic reactions are quite common and represent an interesting optimization problem: as energy is released the temperature rises, which causes the rate constant  $k$  to increase—favoring products—but also causes the equilibrium constant  $K_a$  to decrease—favoring reactants. Use a Levenspiel plot to describe the optimal reactor series for the generic reaction



which follows the elementary, reversible rate

$$r = k \left( C_A - \frac{C_B}{K_C} \right)$$

within adiabatic reactors with a fresh feed of pure A. Additional information is provided

below.

$$k = 0.1 \text{ s}^{-1} \text{ at } 298 \text{ K with } E_a = 35 \text{ kJ} \cdot \text{mol}^{-1}$$

$$K_C = 4 \text{ at } 298 \text{ K}$$

$$\Delta H = -5 \text{ kJ} \cdot \text{mol}^{-1} \text{ at } 298 \text{ K}$$

$$C_{P,A} = C_{P,B} = 55 \text{ J} \cdot \text{mol}^{-1} \cdot ^\circ\text{C}^{-1}$$

$$T_0 = 300 \text{ K}$$

### Solution

Even though we're not performing a "rigorous" design on this problem a stoichiometric table is nevertheless a helpful organizational tool. In anticipation of working with conversion we also include a column for  $n_i$  as a function of  $X_A$ .

Sp.	$\nu_i$	$n_{i0}$	$\Theta_i$	$r_i$	$n_i$	$C_p$ (J · mol <sup>-1</sup> · K <sup>-1</sup> )
A	-1	$n_{A0}$	1	$-r$	$n_{A0}(1 - X_A)$	55
B	1	0	0	$r$	$n_{A0}X_A$	55

Our goal is to determine  $r$  as a function of  $X_A^{\text{ov}}$  so that we can calculate and plot  $n_{A0}/-r_A$ . Notice that we're not given any information about the inlet molar flow rate or concentration so we won't be able to make a quantitative plot of  $n_{A0}/-r_A$ . Instead, we'll factor out all the unknown constants so that we can at least plot  $1/-r_A$  to get a semi-quantitative understanding of the system. For simplicity we'll refer to  $X_A^{\text{ov}}$  as simply  $X_A$ , and therefore the rate expression can be rewritten as

$$\frac{r}{C_{A0}} = k \left( (1 - X_A) - \frac{X_A}{K_C} \right)$$

where

$$k = 0.1 \exp \left[ \frac{E_a}{R} \left( \frac{1}{298 \text{ K}} - \frac{1}{T} \right) \right]$$

$$K_C = 4 \exp \left[ \frac{\Delta H}{R} \left( \frac{1}{298 \text{ K}} - \frac{1}{T} \right) \right]$$

and we moved the  $C_{A0}$  to the left-hand side because we don't have a numerical value for it (nor will we generate one). Since the reactors are to be operated adiabatically we also know that Equation (11.13) relates the reactor temperature to conversion as

$$T = T_0 + \frac{X_A \Delta H}{\nu_A \sum \Theta_i C_{P,i}} = T_0 - \frac{X_A \Delta H}{C_{P,A}}$$

which implies the following calculation approach:

1. Generate a large list of  $X_A$  values between 0 and 1.
2. For each element in the list of  $X_A$  values,
  - (a) calculate  $T$  using Equation (11.13),
  - (b) use this  $T$  value to calculate  $k$  and  $K_C$  as given above,
  - (c) use these  $k$  and  $K_C$  values to calculate  $r$  using the rate expression.

Remember that we can't calculate  $n_{A0}/-r_A$  quantitatively because we don't know  $n_{A0}$  or  $C_{A0}$ , so we settle for a semi-quantitative description of its behavior by the approximation

$$\frac{n_{A0}}{-r_A} \sim \frac{1}{r},$$

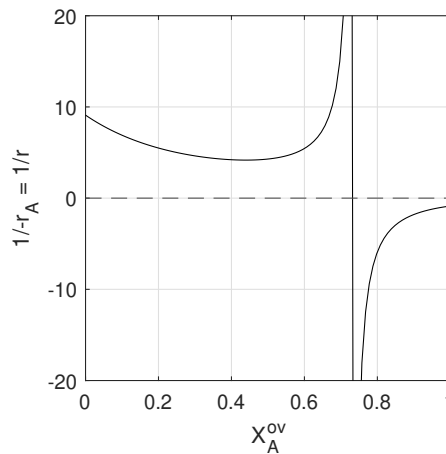
where “ $\sim$ ” in this context means that the two expressions will share the same functional dependence on  $X_A$  but the values will be offset by a constant factor  $n_{A0}/C_{A0}$ . The code below generates the pseudo-Levenspiel plot which we then use to analyze the rate expression:

```

1 % levenAdiabatic.m
2 T0 = 300; % K
3 dH = -5e3; % J/mol
4 Ea = 35e3; % J/mol
5 Cp = 55; % J/mol-K
6
7 Xa = linspace(0, 1);
8 T = T0 - Xa*dH/Cp; % K
9 k = 0.1*exp(Ea/R*(1/298-1./T)); % 1/s
10 Kc = 4*exp(dH/R*(1/298-1./T));
11 r = k.*(1-Xa-Xa./Kc);
12
13 plot(Xa, 1./r)
14 axis([0 1 -20 20])

```



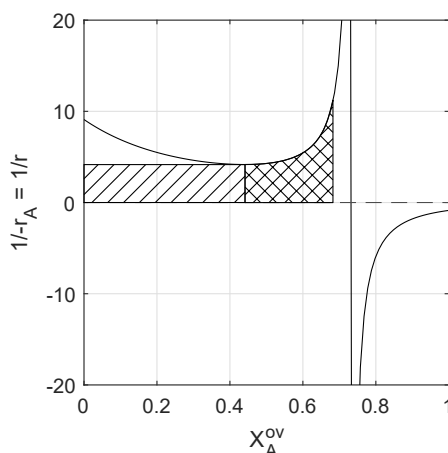


Obviously there's some interesting behavior around  $X_A^{\text{ov}} \approx 0.7$  where the sign of  $1/r$  switches from positive to negative, which happens to be the equilibrium composition. We can find its exact value by setting the net rate to zero, or

$$r = 0 = k(C_A - C_B K_C) \quad \Rightarrow K_C = \frac{X_A^{\text{eq}}}{1 - X_A^{\text{eq}}},$$

where  $K_C$  is a function of temperature and therefore  $X_A$  by Equation (11.13). Solving this equality yields  $X_A^{\text{eq}} = 73.3\%$ . We should immediately discard any conversions higher than 73.3% because this represents the thermodynamic limit for our adiabatic reactors.<sup>a</sup>

Focusing only on  $X_A^{\text{ov}} < 73.3\%$  we also see that there's a minimum around  $X_A^{\text{ov}} \approx 0.45$ , so based on the same reasoning as Figure 11.6 we know the series with minimum volume will be a CSTR accomplishing conversion up to about 45%, followed by a PFR accomplishing as much conversion as we want, up to 73.3%:



Notice that  $1/r$  blows up to infinity as it approaches the equilibrium composition: this is another reminder that equilibrium is a theoretical state, one only *approachable* but which requires infinite volume to actually achieve.

---

“Of course we could employ interstage cooling as we saw in Section 11.3, but then we wouldn’t have a valid Levenspiel plot.

If we had actual values of  $n_{A0}$  and  $C_{A0}$  for Example 11.5 then we’d be able to also estimate the volume required for our CSTR and PFR, but sometimes such information is unavailable because, for example, you’re still in the preliminary design stages and are perhaps comparing several different alternatives.

Even if such information were available, always keep in mind that the Levenspiel plot method is limited to reactors with no side streams, no interstage cooling, and identical reactions. This is clearly a helpful first step in identifying a promising arrangement of reactors but it must always be followed by a rigorous analysis of individual reactors.

## Chapter 12

# MATLAB III: Regression Analysis

“There are three kinds of lies: lies, damned lies, and statistics.”

Attributed to anonymous by Sir Charles Dilke (1891)

Up to now we’ve always been provided information about rate laws, rate constants, and so forth. However, as mentioned as early as Chapter 3 this information *cannot* be obtained from first principles (*i.e.*, it cannot be obtained from theory) and must instead be determined by experiment.<sup>1</sup>

The techniques we’ll develop in this MATLAB chapter will provide tools that we can then use to design and analyze experiments, from which rate laws and rate constants can be determined. The depth of coverage on statistical topics will be limited to the computation and interpretation of results rather than detailed derivations or justifications, which are left to your statistics course.

### 12.1 A model can be fitted to data by minimizing the sum-of-squares

Suppose you think two parameters of some system are related, perhaps by a function such as a material or energy balance that you know to be true, or perhaps because your experience with the system suggests that the two parameters are somehow linked. Whatever the source of the equation, how might we evaluate the suitability of this model for our system? The answer is to perform a series of experiments and analyze the results using a statistical procedure called *regression*.

---

<sup>1</sup>Some rate parameters for simple reactions can be determined from quantum mechanics, but these too must be verified experimentally.

First, a series of experiments are performed wherein the parameter  $x$  is set to several different values, each of which are referred to as  $x_i$ . For each value  $x_i$  another parameter  $y$  is measured resulting in a  $y_i$  value corresponding to each  $x_i$  value. Since  $y$  is expected to vary with  $x$ , we call  $x$  the *predictor variable* and  $y$  the *response variable*.<sup>2</sup>

Next, a *regression model* is proposed, perhaps based on a theoretical foundation such as a material or energy balance, or simply as an educated guess based on familiarity with the system at hand. The model takes as its input a predictor value  $x_i$  and provides as its output a *predicted* value of the response  $y_i$ , which we denote with a hat as  $\hat{y}_i$  to remind us that this is a prediction of theory, not an experimental observation. For flexibility we also provide an adjustable constant  $\beta$  to the model so that we can write

$$\hat{y}_i = f(\beta, x_i) \tag{12.1}$$

Finally, to determine the accuracy of Equation (12.1) and the value of  $\beta$  we compare its predictions (as  $\hat{y}_i$ ) to our experimentally observed values (as  $y_i$ ) by calculating a term called the *sum of squares for error*,

$$\text{SSE} = \sum_{i=1}^n [y_i - \hat{y}_i]^2 = \sum_{i=1}^n [y_i - f(\beta, x_i)]^2 \tag{12.2}$$

In this context the word “error” simply means the difference between the model and obser-

<sup>2</sup>Occasionally  $x$  and  $y$  are also called the *independent* and *dependent* variables.

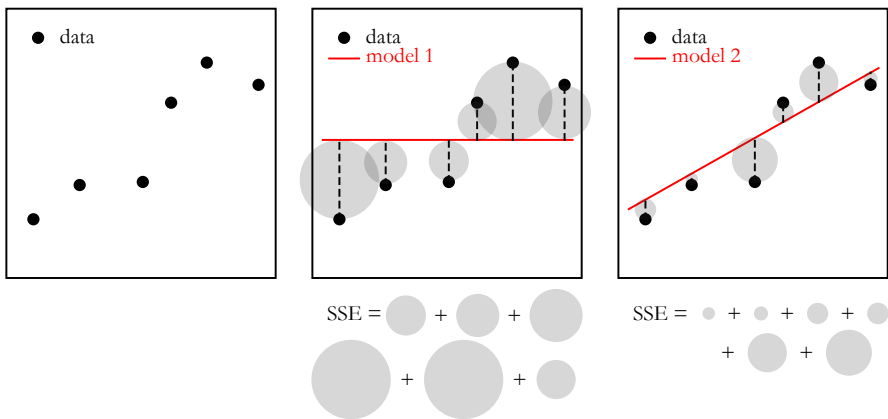


Figure 12.1: An arbitrary set of data (*left*) fitted to two models. A simple horizontal line (model 1, *center*) is a poor representation of the data and has a high SSE, as quantified by the sum of all circular areas. A sloped line (model 2, *right*) provides a better representation of the data and has a lower SSE.

vation, not that a mistake in calculation or experimentation was made. The term inside the parentheses,  $y_i - \hat{y}_i$ , is called the *residual* of the  $i$ -th observation, and for this reason Equation (12.2) is sometimes referred to as the *residual* sum-of-squares (RSS).

When the model is accurate many  $\hat{y}_i$  will be close to their corresponding  $y_i$ , many terms in the sum will be small, and the corresponding SSE will be small. Conversely when the model is poor many  $\hat{y}_i$  will be far from their corresponding  $y_i$ , many terms in the sum will be large, and the SSE will be large. This interpretation of the SSE is illustrated in Figure 12.1, where each  $(y_i - \hat{y}_i)^2$  terms is interpreted as the area of a circle whose diameter is equal to  $|y_i - \hat{y}_i|$ .

### Example 12.1: Calculating the SSE

Suppose the response variable  $y$  is expected to vary with the predictor variable  $x$  according to the regression model

$$\hat{y}_i = \beta x_i$$

where  $\beta = 3$ . Using the data in the table below, calculate the SSE for this value of  $\beta$  and compare it to the SSE if  $\beta = 2$ . Which model is a better representation of the data?

x	0	1	2	3	4	5
y	0.1	2.3	4.2	7.2	8.4	11.9

### Solution

We need to use Equation (12.2) to calculate the SSE for each model, and this calculation is summarized readily in a table with columns for  $x_i$ ,  $y_i$ ,  $\hat{y}_i$ , and  $(y_i - \hat{y}_i)^2$ . For  $\beta = 3$  we have

$x_i$	$y_i$	$\hat{y}_i = 3x_i$	$(y_i - \hat{y}_i)^2$
0	0.1	0	0.01
1	2.3	3	0.49
2	4.2	6	3.24
3	7.2	9	3.24
4	8.4	12	12.96
5	11.9	15	9.61
SSE =			29.55

If  $\beta = 2$  then the table is slightly different:

$x_i$	$y_i$	$\hat{y}_i = 2x_i$	$(y_i - \hat{y}_i)^2$
0	0.1	0	0.01
1	2.3	2	0.09
2	4.2	4	0.04
3	7.2	6	1.44
4	8.4	8	0.16
5	11.9	10	3.61
SSE =			5.35

Since the SSE for  $\beta = 2$  is less than the SSE for  $\beta = 3$ , we conclude that  $\beta = 2$  is a better model for this system than  $\beta = 3$ .

The approach demonstrating in Example 12.1 is the central calculation procedure for regression analysis: the parameter  $\beta$  is varied until the SSE is minimized. Naturally we won't do the actual guessing of  $\beta$  parameters nor the calculation of the SSE by hand, but we'll instead rely on MATLAB to do the guessing in a more systematic and efficient manner. We also won't limit ourselves to a single adjustable  $\beta$ , nor a single predictor  $x_i$ , nor even a single model. Instead, we'll learn a little bit about the statistical tools necessary to choose the best combination of these options, though of course we'll always be guided by our own engineering judgment.

We begin our discussion with the most common regression model, the linear model.

## 12.2 Linear models can be developed using the fitlm function

A *linear* model is one that's linear in the *coefficients* as opposed to linear in the predictor variable. For example,

$$\hat{y} = \beta_0 + \beta_1 \sin x$$

is considered a linear model for regression purposes because it's linear in  $\beta_0$  and  $\beta_1$ ; the non-linearity in  $x$  due to the  $\sin x$  term is irrelevant. However,

$$\hat{y} = \beta_0 + \beta_1 \sin(\beta_2 x)$$

is nonlinear for regression purposes because of the  $\sin(\beta_2 x)$  term. Models are also considered linear even if more than one predictor variable is present, such as

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2,$$

which has two predictor variables— $x_1$  and  $x_2$ —and is linear in all three model coefficients— $\beta_0$ ,  $\beta_1$ , and  $\beta_2$ .

To perform a linear regression analysis we need to organize our data in a particular way, as follows:

1. Place all the response data into an  $n$ -by-1 column vector, where  $n$  is the number of observations. This vector will be called the *response vector* and will usually be given the symbol  $Y$  (or  $\mathbf{Y}$ ), but it can be any convenient name such as  $n_A$ ,  $C_A$ ,  $X_A$ , and so forth.
2. Place all linear predictors into an  $n$ -by- $m$  matrix, where  $n$  is the number of observations and  $m$  is the number of predictors. This vector will be called the *predictor matrix* and will usually be given the symbol  $X$  (or  $\mathbf{X}$ ).
3. Pass  $Y$  and  $X$  to MATLAB's `fitlm` function to perform the regression.
4. Check the output of `fitlm` to evaluate model quality.

The first, third, and fourth steps are fairly straightforward (or will be when we look at a few examples) but the second step deserves a bit of explanation: what do we mean when we say “linear predictors?” A *linear predictor* is a term on the right-hand side of your model which is multiplied by an as-yet undetermined coefficient. For example, the model

$$\hat{y} = \beta_0 + \beta_1 x_1$$

has just one linear predictor,  $x_1$ . Assuming the data for  $x_1$  exist as a vector `x1` then it's simple to create the predictor matrix `X` as

```
X = [x1];
```

which is so simple it's not entirely necessary to repeat every single time. Notice that there's no column for the intercept  $\beta_0$ ; its effects will be calculated automatically such that it's not necessary to add any extra columns to `X` in order to get estimates of  $\beta_0$ . Suppose we had the more complicated model

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 \sin x_1.$$

We have the same linear predictor  $x_1$  as before but now we've also got another linear predictor in the form of the  $\sin x_1$  term. Therefore our predictor matrix `X` requires two columns, the

first for  $x_1$  and the second for  $\sin x_1$ , such as

```
X = [x1, sin(x1)]
```

or equivalently,

```
x2 = sin(x1);
X = [x1, x2];
```

### Example 12.2: Constructing the predictor matrix

Assume the vectors  $x_1$  and  $x_2$  exist in the base workspace. Identify whether each of the models provided below is linear or nonlinear, and if linear provide a code snippet to calculate and assemble the predictor matrix  $X$  using only  $x_1$  and  $x_2$ .

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 \quad (1)$$

$$\hat{y} = \beta_0 + \beta_1 x_1 x_2 \quad (2)$$

$$\hat{y} = \beta_0 + \beta_1 \sin x_1 + \beta_2 \cos x_1 \quad (3)$$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^3 + \beta_4 x_1^{-1} \quad (4)$$

### Solution

(1) This model is linear in all  $\beta$  terms. The predictor matrix  $X$  could be assembled as

```
X = [x1 x2];
```

(2) This model is linear in all  $\beta$  terms; the nonlinear product of  $x_1$  and  $x_2$  is irrelevant. The predictor matrix  $X$  could be assembled as

```
X = [x1.*x2];
```

(3) This model is linear in all  $\beta$  terms. The predictor matrix  $X$  could be assembled as

```
X = [sin(x1), cos(x1)];
```

(4) This model is linear in all  $\beta$  terms. The predictor matrix  $X$  could be assembled as

```
X = [x1, x1.^2, x1.^3, x1.^(-1)];
```

Once you've got  $Y$  and  $X$  it's time to perform the regression analysis itself, which means that it's time to guess values of all  $\beta_i$  until the sum-of-squares, defined as SSE in Equation (12.2), is



minimized. To do so, pass  $X$  and  $Y$  to `fitlm` as

```
mdl = fitlm(X, Y)
```

where `mdl` is a *linear model* datatype containing lots of information about the fitted model, only some of which we're interested in now. In many instances you can learn everything you need to know simply by inspecting the default display of `mdl`, which is why the code above was shown without a semicolon. Another way to display `mdl` after running `fitlm` is to type `mdl` or `disp(mdl)` at the command line,

```
>> mdl
>> disp(mdl) % equivalent to above
```

The estimates of the  $\beta_i$  terms are displayed directly in such outputs in the “Estimate” column, or they can be accessed programatically as

```
>> betaVals = mdl.Coefficients.Estimate;
```

### Example 12.3: Linear regression to a first-order polynomial

Fit the response  $y$  to the predictor  $x$  using the data given below and assuming a first-order polynomial model

$$\hat{y} = \beta_0 + \beta_1 x.$$

Report the values of  $\beta_0$  and  $\beta_1$ .

$x$	0	1	2	3	4	5
$y$	0.1	2.3	4.2	7.2	8.4	11.9

### Solution

The script below defines  $x$  and  $y$ , then passes these to the `fitlm` function. Keep in mind the  $x$  and  $y$  must be *column* vectors, not row vectors.

```
1 % linfit1.m
2 x = [0 1 2 3 4 5]';
3 y = [0.1 2.3 4.2 7.2 8.4 11.9]';
4 mdl = fitlm(x, y)
```

Running this script from the command line produces the following output:

```
>> linfit1
```

```
mdl =

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:
      Estimate SE tStat pValue
-----
(Intercept) -0.052381 0.36775 -0.14243 0.89362
x1 2.2943 0.12146 18.888 4.6269e-05

Number of observations: 6, Error degrees of freedom: 4
Root Mean Squared Error: 0.508
R-squared: 0.989, Adjusted R-Squared 0.986
F-statistic vs. constant model: 357, p-value = 4.63e-05
```

The estimate of  $\beta_0$  is provided as the estimate of (Intercept), or  $-0.052381$ . The estimate of  $\beta_1$  is provided as the estimate of x1, or  $2.2943$ . Rounding each of these to a more reasonable three decimal points and placing the estimates in our model yields

$$\hat{y} = 2.294x - 0.052$$

as our best estimate of a linear relationship between  $x$  and  $y$ .

As noted previously we don't have to do anything special to include the intercept term  $\beta_0$  to our regression model because it's included automatically. Sometimes we might want to exclude the intercept  $\beta_0$ , such as if the model we're trying to fit doesn't include an intercept. For example, light absorbance  $A$  is often related to species concentration  $C_i$  by the Beer-Lambert law,

$$A = \varepsilon \ell C_i, \quad (12.3)$$

where the product  $\varepsilon \ell$  is an experimentally-determined constant which can be found by fitting a set of  $C_i$  and  $A$  data to Equation (12.3). Importantly, there is no intercept in Equation (12.3), so the regression procedure must not include the  $\beta_0$  term.

To exclude the intercept from a fitting procedure with `fitlm`, simply pass the name-value pair '**Intercept**', `false` as an additional input argument. The  $\beta_0$  term will not be calculated and the resulting `mdl` object will no longer have the (Intercept) row.

#### Example 12.4: Linear regression without intercept

Repeat the linear regression of Example 12.3 but set the intercept to zero. Compare

the value of  $\beta_1$  with and without the intercept included in the regression.

### Solution

The script below includes the name-value pair `'Intercept'`, `false` as an additional input to the `fitlm` function, which sets the intercept to zero in the regression procedure:

```
1 % linfit2.m
2 x = [0 1 2 3 4 5]';
3 y = [0.1 2.3 4.2 7.2 8.4 11.9]';
4 mdl = fitlm(x, y, 'Intercept', false)
```

Running this script produces the following output:

```
>> linfit2

mdl =

Linear regression model:
y ~ x1

Estimated Coefficients:
      Estimate SE tStat pValue
      -----
x1 2.28 0.061437 37.111 2.6756e-07

Number of observations: 6, Error degrees of freedom: 5
Root Mean Squared Error: 0.456
```

As expected, the linear regression model is  $y \sim x_1$  (no intercept) as compared to  $y \sim 1 + x_1$  (with intercept, denoted by the 1 term) from Example 12.3.

The impact of the intercept in this example is small but noticeable:

	with intercept	without intercept
$\beta_1$	2.2943	2.28

Since we're usually going to be fitting data to various forms of the material balance, the choice of whether to include or exclude the intercept will usually be dictated by the material balance for that problem: if there's no intercept in the balance then there should be no intercept in the fit model.

However, even when such clear guidance on terms is available we'll still have to make a choice between competing models, and to guide our choice we'll need to investigate a few

more of the output terms from `fitlm`.

## 12.3 Several statistics should be evaluating when deciding model quality

Each time we perform a regression analysis we have to decide two things, usually in this order:

1. Is the regression itself meaningful?
2. If the regression is meaningful, are the coefficients also meaningful?

To answer the first question we have three tools: the overall  $p$ -value, the goodness-of-fit statistic  $R^2$ , and a plot of the residuals. Not all three are present in the output of all regression models, particularly those that lack intercepts, but you should look for them all each time you perform a regression analysis. Each provides a piece of evidence that can help you decide to accept or reject the model.

The overall  $p$ -value provides an answer to the following question:<sup>3</sup> do your predictors provide better explanation of your data than does a horizontal line? If the  $p$ -value is less than some small value—commonly 0.05—then we have one piece of evidence that our model has statistically significant explanatory power.

A related term is the goodness-of-fit parameter  $R^2$ , which always varies between 0 and 1. A low  $R^2$  indicates that the model explains little of the variability of the response data around its mean (a horizontal line), and a high  $R^2$  indicates that the model explains much of the variability around its mean. One form of statistical malpractice is to keep adding coefficients to the model until  $R^2$  becomes close to 1 because  $R^2$  *always* increases as additional terms are added. To avoid falling into this trap you should use the *adjusted*  $R^2$  instead of the plain  $R^2$ , which takes into account the number of parameters.

The third quantity to check are the *residuals*, defined as the difference between each response point and its corresponding model prediction,

$$\text{residual}_i = e_i = y_i - \hat{y}_i, \quad (12.4)$$

where  $e_i$  is the residual of the  $i$ -th observation. Residuals can be quickly plotted after the fit procedure is completed with a call to the `plotResiduals` function as

```
plotResiduals mdl, 'fitted')
```

---

<sup>3</sup>This is a paraphrasing of the formal statements of what are called the “null hypothesis” and “alternative hypothesis” that you’ll learn about in your statistics course.

Residuals are an indicator of random effects and one of the core assumptions of regression is that random effects are truly random. A plot of residuals should look like random noise; the presence of *any* distinguishable pattern in the residuals is problematic because it suggests that your model is missing some explanatory information, such as missing higher-order terms or missing interactions between terms.

If all three of the overall model evaluations—the  $p$ -value, the adjusted  $R^2$  parameter, and a residual plot—all appear to be satisfactory then we can move on to the last step, an investigation of the  $p$ -values of each fitted coefficient. For each coefficient we ask the question, “Is this coefficient equal to zero?” If the  $p$ -value for that coefficient is small (usually  $p < 0.05$ ) then we conclude that no, the coefficient is not equal to zero and based on the available information and within this model the best estimate of the coefficient is the one obtained from regression. If the  $p$ -value is *not* small then we cannot conclude that changes in the predictor are associated with changes in the response, which for our purposes usually means that we reject the proposed rate law and look for a different one.

In summary, ask yourself the following questions after each regression:

- ☐ Is the overall  $p$ -value less than 0.05?
- ☐ Is the adjusted  $R^2$  close to one?
- ☐ Does the residuals plot look like noise?
- ☐ Is the  $p$ -value of each coefficient less than 0.05?

If the answer to each of these questions is “yes” then you’ve probably got a statistically significant model!

### Example 12.5: Choosing the best model

Compare the models from Example 12.3 and 12.4 and identify the model which best describes the data.

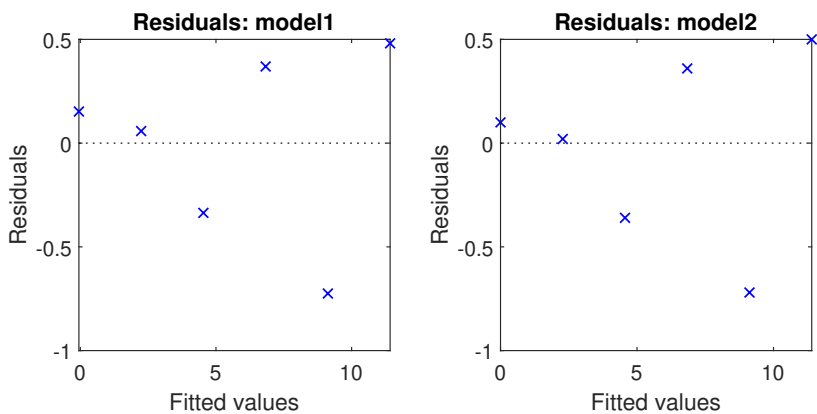
### Solution

The overall  $p$ -value and adjusted  $R^2$  value for Example 12.3 are  $4.63 \times 10^{-5}$  and 0.986, both of which are good. However, we cannot compare these to equivalent values for Example 12.4 because these values are not calculated for models that do not include intercepts. We’ll have to use other comparisons to decide between the models.

We turn next to the plot of residuals, which we can produce with the `plotResiduals` function as follows:

```
1 % fitCheck.m
2 x = [0 1 2 3 4 5]';
3 y = [0.1 2.3 4.2 7.2 8.4 11.9]';
4 model1 = fitlm(x, y);
5 model2 = fitlm(x, y, 'Intercept', false);
6 subplot 121;
7 plotResiduals(model1, 'fitted')
8 title('Residuals: model1')
9 subplot 122;
10 plotResiduals(model2, 'fitted')
11 title('Residuals: model2')
```

Running this script produces the two residual plots shown below, neither of which indicate an obvious pattern or trend (which is good).



The first three statistical checks—the overall  $p$ -value, the adjusted  $R^2$ , and the residual plots—have so far been ambiguous: both models are acceptable but neither is particularly better than the other. We turn finally to the  $p$ -values of individual coefficients, as summarized in the following table:

Model	$p$ -value for	
	intercept ( $\beta_0$ )	x1 ( $\beta_1$ )
Model 1 (with intercept)	0.89362	$4.6269 \times 10^{-5}$
Model 2 (without intercept)	n/a	$2.6756 \times 10^{-7}$

Clearly the  $p$ -value of the intercept ( $\beta_0$ ) for Model 1 far exceeds 0.05, so in this model the intercept is not associated with changes in the response—put another way, this coefficient is not necessary to explain any variation in the response. This is grounds for

rejection of Model 1, and fortunately when we look at the  $p$ -values for the coefficients in Model 2 they're all less than 0.05.

We therefore accept Model 2 as the model which provides better explanatory power compared to Model 1. This is not to say that Model 2 is the *only* possible model to meet our statistical requirements, only that it's better than Model 1.

## 12.4 Nonlinear models can be developed using the `fitnlm` function

If the model function  $f(\beta, x)$  is not linear in the coefficients  $\beta$  then the `fitlm` function won't be able to perform the regression procedure. We must instead turn to the `fitnlm` function to perform such regressions, which has a similar syntax but with two additions:

```
mdl = fitnlm(X, y, @fun, betaGuess)
```

The matrix  $X$  is still an  $n$ -by- $m$  matrix of predictor values as it was for the `fitlm` function, and  $y$  is still an  $n$ -by-1 vector of response variable values. The new input, `@fun`, is a handle to the (local) fit function that should be used by `fitnlm` to fit the model. The fit function syntax is

```
function yhat = fun(beta, X)
```

where  $\beta$  is a vector of adjustable coefficients,  $X$  is the matrix of predictor variables, and  $yhat$  is an  $n$ -by-1 vector of model predictions, one for each row in  $X$ . Notice that the size of  $yhat$  is the same as  $y$ : this is because the same SSE calculation described by Equation (12.2) is performed for non-linear regression as it was for linear regression, so tables similar to those shown in Example 12.1 will be calculated.

The vector `betaGuess` is a vector containing an initial guess for each  $\beta_i$  parameter in the model. Unlike linear regression, non-linear regression is performed by an iterative process not unlike the one used for solving non-linear equations with the `fsolve` function from Section 1.5, and just as `fsolve` did the `fitnlm` requires a starting point for the iteration process. As with `fsolve` the results of `fitnlm` can be sensitive to the initial guesses.

The output `mdl` from `fitnlm` contains information similar to the output from `fitlm` and the checks you should do for  $p$ -values, adjusted  $R^2$ , and residuals should still be performed.

### Example 12.6: Basic usage of `fitnlm`

Repeat Example 12.3 using the `fitnlm` function.

## Solution

The model function is

$$\hat{y} = \beta_1 + \beta_2 x,$$

which indicates that there are two adjustable parameters,  $\beta_1$  and  $\beta_2$ . The input `beta` for the fit function `fun` will therefore be a vector with two elements where `beta(1)` =  $\beta_1$  and `beta(2)` =  $\beta_2$ . Similarly the vector of initial guesses for  $\beta$  will be a two element vector where the first element, `betaGuess(1)`, is a guess for  $\beta_1$  and the second element, `betaGuess(2)`, is a guess for  $\beta_2$ .

The script below assembles these components and performs the regression procedure:

```

1 % fitnlm1.m
2 x = [0 1 2 3 4 5]';
3 y = [0.1 2.3 4.2 7.2 8.4 11.9]';
4 betaGuess = [1; 1];
5 mdl = fitnlm(x, y, @fun, betaGuess);
6 disp(mdl)
7
8 function yhat = fun(beta, x)
9 yhat = beta(1) + beta(2)*x;
10 end

```

Running this script produces the following output:

```

>> fitnlm1

Nonlinear regression model:
y ~ fun(b,X)

Estimated Coefficients:
    Estimate SE tStat pValue
    -----
b1 -0.052381 0.36775 -0.14243 0.89362
b2 2.2943 0.12146 18.888 4.6269e-05

Number of observations: 6, Error degrees of freedom: 4
Root Mean Squared Error: 0.508
R-Squared: 0.989, Adjusted R-Squared 0.986
F-statistic vs. constant model: 357, p-value = 4.63e-05

```

Unsurprisingly, the results from `fitnlm` are identical to those of `fitlm` for this data set.

Example 12.6 demonstrates that for simple models the results of `fitlm` and `fitnlm` will be sufficiently close that it won't matter which method you choose to use. The real utility of



`fitnlm` is when the model is nonlinear, as shown in the next example.

### Example 12.7: Fitting a nonlinear model

Fit the data set provided below to the harmonic model

$$\hat{y} = A \sin(2\pi f t).$$

Determine the amplitude  $A$  and frequency  $f$  of the data set.

$t$ (s)	$y$	$t$ (s)	$y$
0.00	-0.04	0.50	-1.16
0.05	0.57	0.55	-1.15
0.10	0.99	0.60	-1.02
0.15	1.16	0.65	-0.57
0.20	1.20	0.70	-0.04
0.25	0.92	0.75	0.67
0.30	0.48	0.80	0.98
0.35	-0.07	0.85	1.17
0.40	-0.59	0.90	1.34
0.45	-1.13	0.95	0.99

### Solution

First the harmonic model is expressed as a function of  $\beta$ ,

$$\hat{y} = \beta_1 \sin(\beta_2 t),$$

where  $\beta_1 = A$  and  $\beta_2 = 2\pi f$ . Next we need an estimate for  $\beta_1$  and  $\beta_2$ : it appears from the data set that the maximum amplitude is about 1 so we'll guess  $\beta_1^{\text{guess}} = 1$  or `betaGuess(1)= 1`, and it looks like about one period is completed in 0.70 s so that

$$\beta_2^{\text{guess}} = 2\pi f = \frac{2\pi}{P} \approx 9,$$

or `betaGuess(2)= 9`. The following script assembles these elements into appropriate

forms, displays the fit results, and plots the fit and residuals.

```

1 % fitnlm2.m
2 t = (0:0.05:0.95)'; % s
3 y = [-0.04 0.57 0.99 1.16 1.20...
4       0.92 0.48 -0.07 -0.59 -1.13...
5       -1.16 -1.15 -1.02 -0.57 -0.04...
6       0.67 0.98 1.17 1.34 0.99]';
7
8 betaGuess = [1 9];
9 mdl = fitnlm(t, y, @fun, betaGuess);
10 disp(mdl)
11 beta = mdl.Coefficients.Estimate; % [beta1; beta2]
12 subplot 121; hold on
13 plot(t, y, 'ko')
14 plot(linspace(0,1), fun(beta, linspace(0,1)), 'k');
15 hold off
16 subplot 122
17 plotResiduals(mdl, 'fitted')
18
19 function yhat = fun(beta, x)
20     yhat = beta(1)*sin(beta(2)*x);
21 end

```

Running this script produces the output shown below:

```

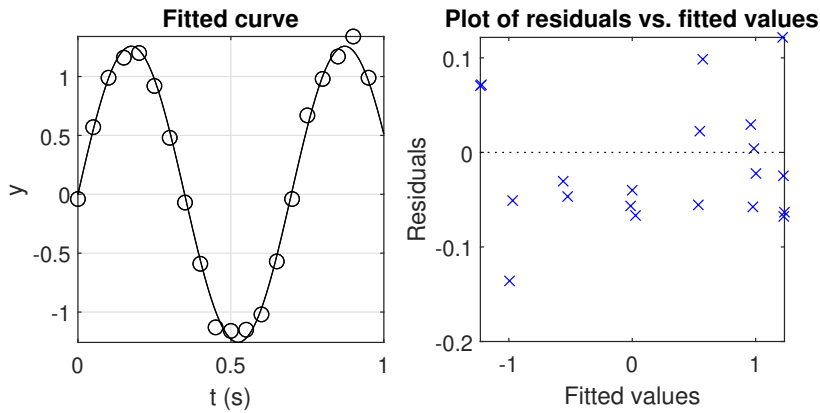
>> fitnlm2

Nonlinear regression model:
y ~ fun(b,X)

Estimated Coefficients:
      Estimate SE tStat pValue
      -----
b1 1.258 0.021487 58.547 5.3848e-22
b2 9.0063 0.034111 264.03 9.4432e-34

Number of observations: 20, Error degrees of freedom: 18
Root Mean Squared Error: 0.0689
R-Squared: 0.994, Adjusted R-Squared 0.994
F-statistic vs. zero model: 1.72e+03, p-value = 2.86e-21

```



Based on the printed output all of the  $p$ -values are virtually zero, and the adjusted  $R^2$  value is nearly equal to one. The plot of residuals shows no particular pattern, so from these observations we conclude that the model provides an accurate fit to the experimental data.

The amplitude is  $A = \beta_1 = 1.258$  and the frequency  $f$  is

$$\beta_2 = 2\pi f \Rightarrow f = \frac{\beta_2}{2\pi} = \frac{9.0063 \text{ s}^{-1}}{2\pi} = 1.4334 \text{ s}^{-1}.$$

The model could have equally well been defined as

$$\hat{y} = \beta_1 \sin(2\pi\beta_2 f),$$

with a corresponding fit function defined as

```
20 yhat = beta(1)*sin(2*pi*beta(2)*x);
```

and the same values would result.



## Chapter 13

# Experimental Determination of Rate Laws

“A month in the laboratory can often save an hour in the library.”

Attributed in various forms to chemist Frank Westheimer (1912-2007)

The topic of this chapter is (obviously) the experimental determination of rates laws, but the quote above is not to be ignored! If you are some day faced with the challenge of designing a reactor and need to come up with a reaction mechanism then your first stop should *invariably* be a library before you even consider going to the laboratory. Experiments are messy, time-consuming, difficult to perform, potentially dangerous, and inevitably include a lengthy data analysis. Reading published articles, on the other hand, can be done quietly and while eating snacks.

All that said, there are still times where you may find it necessary to determine an appropriate rate expression for a system of interest. The material presented here is appropriate for systems which require simple experimental equipment such as light sensors, temperature sensors, pH probes, timers, and the like. In this regard, the material is well-suited for application in the American Institute of Chemical Engineers’ Chem-E-Car competition, but it also serves as an introduction to the general analysis procedures of more advanced methods.<sup>1</sup>

### 13.1 The batch reactor material balances are differential equations

The isothermal, liquid-phase batch reactor will serve as our reactor model because it’s simple and inexpensive to construct: a simple beaker or cuvette with a little bit of insulation is

---

<sup>1</sup>For an introduction to several other methods see H.S. Fogler. Elements of Chemical Reaction Engineering, 4th ed. Pearson: Upper Saddle River, NJ (2006).

sufficient in most instances. The material balance for such a system is

$$\text{Acc} = \text{In} - \text{Out} + \text{Change}, \quad (13.1)$$

and since by definition a batch system has no inlet or outlet streams we simplify the material balance further to

$$\text{Acc} = \text{Change}. \quad (13.2)$$

The concept of “steady state operation” is a batch reactor is somewhat meaningless: if the system is at steady state then  $\text{Acc} = 0$  and therefore  $\text{Change} = 0$ , indicating no changes are taking place in the system! Therefore we will never refer to a batch reactor operating at steady state: the only meaningful operation of a batch reactor is one for which some quantities change with time.

Denoting the moles of any species  $i$  within the reactor as  $N_i$  (capital  $N$  to distinguish from the previous molar flow rate terms as  $n$ ), the change in  $N_i$  between two moments in time is

$$N_i|_{t+\Delta t} - N_i|_t = r_i V \Delta t, \quad (13.3)$$

where  $V$  is the (constant) reactor volume and  $\Delta t$  is time elapsed between the two time points. Rearranging and taking the limit as  $\Delta t \rightarrow 0$  yields

$$\lim_{\Delta t \rightarrow 0} \frac{N_i|_{t+\Delta t} - N_i|_t}{\Delta t} = r_i V, \quad (13.4)$$

or

$$\frac{dN_i}{dt} = r_i V \quad N_i(t=0) = N_{i0} \quad (13.5)$$

which is the general material balance for any species  $i$  in a batch reactor.

Since liquid-phase rate laws are often expressed in terms of concentration  $C_i$  it's convenient to replace  $N_i$  with  $C_i V$  to yield

$$\frac{dN_i}{dt} = \frac{d}{dt}(C_i V) = V \frac{dC_i}{dt}, \quad (13.6)$$

where the second equality assumes  $V$  is constant, which is usually true for liquids. Substituting into Equation (13.5) yields

$$\boxed{\frac{dC_i}{dt} = r_i} \quad C_i(t=0) = C_{i0} \quad (13.7)$$

which is the material balance expressed in terms of molar concentration  $C_i$  for any species  $i$ , assuming constant volume  $V$ . Equation (13.7) will be the working equation for the remainder of the chapter, although in some cases we may need to slightly modify it to work with available data.

## 13.2 The integrated material balance can be used to fit rate expressions

The most common rate expressions are as functions of concentration and temperature, or

$$r = r(C_i, T),$$

where the temperature dependence is often expressed in the Arrhenius form of the rate constant  $k$  that we've used previously. Reversible rate laws can also be expressed as functions of temperature as we saw in Chapter 3 but we'll restrict our attention to irreversible reactions for this discussion.

Given a proposed rate law and a set of experimental data for concentration as a function of time, two common ways to go about fitting the rate law to the data are the *differential* method and the *integral* method. The analysis approach for these two methods are as follows:

**Differential method:** Fit the raw concentration data to whatever form best captures its behavior, such as a third or fourth order polynomial, or an appropriate exponential. The derivative of this function is  $dC_i/dt$ ; use it to produce a table of  $dC_i/dt$  at each experimentally measured value of  $C_i$ . Perform linear or nonlinear regression to fit this tabulated data directly to the proposed rate expression.

**Integral method:** Integrate the material balance to obtain an analytic expression for  $C_i(t)$  as a function of rate parameters. Perform linear or nonlinear regression to the experimental data to the analytic expression.

One advantage of the differential method is that no integrals are evaluated, and when rate expressions become complicated so too does the integration of the material balance become complicated (or even impossible!). A disadvantage of the differential method is that it requires two regression procedures, one to fit the experimental data and another to fit the rate expression to the first fit. This can introduce significant error, and care must be taken with the initial fit to the experimental data to avoid introducing non-physical behavior (e.g., fitting a 6th order polynomial might produce a curve that goes through many experimental data points, but also introduces unwanted oscillations).

When the proposed rate law is not overly complex then the integral method can be used. An advantage of the integral method is that it requires only one regression, but as already indicated its primary limitation is that it can only be performed when an analytic expression of the integrated material balance can be obtained. Many reactions are initially assumed to follow elementary rate laws, and the resulting power law expressions are readily integrated. Even a few non-elementary forms can be used in the integral method without too much headache, as we'll see later in the chapter.

For its simplicity we'll focus only on the integral method for this chapter. Indeed the differential and integral methods are only two of many approaches for experimental determination of rate laws,<sup>2</sup> but the integral method is particularly well-suited for an introductory discussion of the topic.

### 13.3 Classical linearization methods should be avoided

If the proposed rate expression is zero, first, or second order in a single reactant then the material balance can be integrated to yield an expression that can be linearized by performing an appropriate operation. For example, the first order rate expression is

$$r = kC_A$$

and the integrated material balance of Equation (13.7) for this rate expression is

$$C_A = C_{A0}e^{-kt} \quad (13.8)$$

which is routinely linearized by taking the logarithm of both sides to yield

$$\ln C_A = \ln C_{A0} - kt. \quad (13.9)$$

Similarly, if the rate expression is second order as  $r = kC_A^2$  then the integrated material balance is

$$\frac{1}{C_A} = \frac{1}{C_{A0}} + kt \quad (13.10)$$

which is also a linear form. You're probably accustomed to using a form like Equation (13.9) or Equation (13.10) in your chemistry courses to determine rate constants by plotting  $\ln C_A$  versus  $t$  or  $1/C_A$  versus  $t$ , then finding the slope and intercept as the rate constant and something

<sup>2</sup>See Ch. 7 in H.S. Fogler, *Essentials of Chemical Reaction Engineering*, Prentice Hall, NJ (2011) or Ch. 28 in D.A. McQuarrie, J.D. Simon, *Physical Chemistry: A Molecular Approach*, University Science Books, CA (1997).



proportional to  $C_{A0}$ . The advantage of such linearizations is that they're easy to perform, the resulting data are easy to plot, and the rate constants are easily extracted from the plots.

Unfortunately these linearizations have a significant drawback because they skew the uncertainty associated with each measurement.<sup>3</sup> When experimental data are collected they typically have an uncertainty that's constant across the range in which they are collected, which roughly translates into the ability of regression analysis to treat each data point equally in the sum of squares term in Equation (12.2). When the uncertainty on the data are *not* equal then the individual squared terms of Equation (12.2) must be treated differently, a treatment that's beyond the scope of this text.

The alternative is to avoid linearizations which apply transformations to  $C_A$  and instead use nonlinear regression with the `fitnlm` function. Not only does nonlinear regression avoid skewing uncertainties as noted above, it also *encompasses* linear regression: if you learn how to do nonlinear regression for nonlinear models then you'll be able to use exactly the same approach for linear models as well.

Given the drawbacks of linearization and the ease-of-use of nonlinear regression routines, *linearizations of data should be avoided except* as a visual aide for quickly plotting the data to determine if the rate is first or second order, or as a means of estimating initial guesses of  $\beta$  values to be used in a nonlinear regression procedure.

One practical impact of choosing to focus on nonlinear regression with `fitnlm` is that the integrated material balance will be usually be solved to obtain a function explicit in  $C_A$ , and this function will be the one used to create the local function for use with `fitnlm`. Another practical impact is that we can define a template for nonlinear regression—a template similar to the templates for root finding and numerical ODE solutions—as follows:

```

1 % template for fitting rate laws
2
3 % Load data and perform regression
4 t = [ ]; % col vec of data
5 Ca = [ ]; % col vec of data
6 beta0 = [ ]; % initial guesses for beta1, beta2, etc
7 model = fitnlm(t, Ca, @fun, beta0);
8
9 % Compare data to fit
10 disp(model)
11 tFit = linspace(min(t), max(t));
12 CaFit = feval(model, tFit);
13 subplot 121; hold on
14 plot(t, Ca, 'ko')
15 plot(tFit, CaFit, 'k')
```

<sup>3</sup>For a lengthier but easily approachable discussion see C.L. Perrin. Linear or Nonlinear Least-Squares Analysis of Kinetic Data? J. Chem. Edu. 94 (2017) 669-672.

```

16 hold off
17
18 % Examine residuals
19 subplot 122
20 plotResiduals(model, 'fitted')
21
22 % Define local function
23 function Ca = fun(beta, t)
24     % integrated material balance goes here
25 end

```

Unlike previous templates the “analysis” portion of a fitting procedure should *always* include the steps shown in the template above:

**Line 10:** Display the regression results to inspect  $p$ -values and adjusted  $R^2$  values.

**Lines 11-16:** Plot the regression line along with the raw data.

**Lines 19-20:** Plot the residuals and look for patterns.

Keep in mind that even if all of these analysis steps support the model it’s nevertheless inappropriate to make a categorical statement such as, “*the rate expression is first order.*” Instead, we can state that the experimental data *support* the hypothesis that the rate expression is first order, or we can state that the experimental data are *consistent* with a first order rate expression. We never state with absolute certainty exactly what the rate expression is or the value of its coefficients; we can only state how strongly the proposed model is supported by experimental observation.

### 13.4 $C_{A0}$ can be treated as an adjustable parameter even if its value is known

One final note before we start looking at some examples: the initial concentration  $C_{A0}$  often appears in the integrated material balance but we should generally consider it an unknown parameter for the purposes of regression analysis. The reasoning here is that  $C_{A0}$  can play an outsized role in the regression analysis, yet our knowledge of the true value of  $C_{A0}$  is typically no better than our knowledge of any other value of  $C_A(t)$ .

As a simple example, a zero-order rate expression  $r = k$  yields an integrated material balance as

$$C_A = C_{A0} - kt,$$

which is a linear model with slope  $-k$  and intercept  $C_{A0}$ . If the value of  $C_{A0}$  happens to be known then there's a temptation to rearrange the integrated material balance to

$$C_A - C_{A0} = -kt,$$

which is a linear model with slope  $-k$  but a zero intercept. Regression results with and without an intercept will usually be quite different, and making such a large change in the model based on the value of a single quantity  $C_{A0}$  is usually a bad idea. Instead, we can use  $C_{A0}$  as an adjustable parameter in the regression model and then compare its value to the experimental value as an additional check on the validity of the model.

## 13.5 Determination of experimental rate laws

Each of the examples provided here are based on real experimental studies of various systems. The examples were chosen based on their experimental simplicity: dye concentrations are easily measured with photosensors, and salt concentrations are easily measured with conductivity probes.

### Example 13.1: Methyl orange degradation

The degradation of methyl orange ( $C_{14}H_{14}N_3NaO_3S$ , species A), a textile dye, pH indicator, and generally undesirable compound for waste streams, can be broken down by UV light in the presence of titanium dioxide. Use the experimental data below<sup>a</sup> to determine if the degradation process of methyl orange follows zero, first, or second order kinetics. If any of these power law models are reasonable, also provide an estimate of the rate constant  $k$ .

$t$ (s)	0	10	20	30	40	50	60	70
$C_A$ ( $\mu\text{mol} \cdot \text{L}^{-1}$ )	14.6	11.1	7.7	5.3	3.4	2.0	1.2	0.7

<sup>a</sup>Collected from S. Haji, B. Benstaali, N. Al-Bastaki. Degradation of methyl orange by UV/ $H_2O_2$  advanced oxidation process. Chem. Eng. J. 168 (2011) 134-139.

### Solution

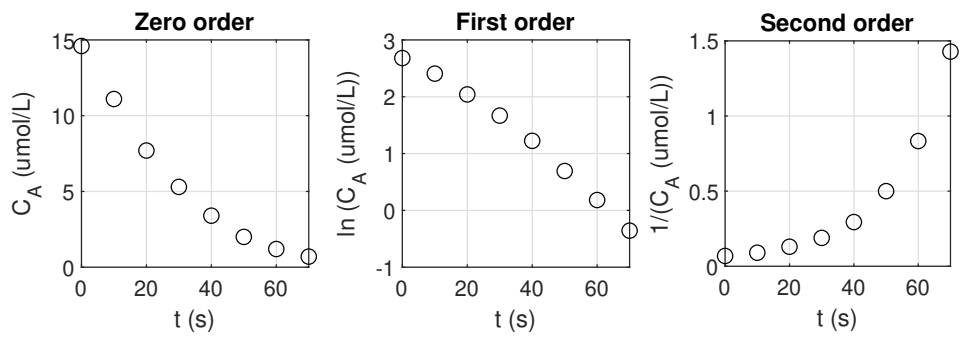
The first thing to do is determine the integrated material balance for each rate expression. In anticipation of using linearized plots to get quick estimates of reasonable

rate expressions and corresponding parameters, both the linearized form and a form explicit in  $C_A$  are summarized in the table below.

Order	Rate	$C_A(t)$	Linearized model
0	$r = k$	$C_A = C_{A0} - kt$	same
1	$r = kC_A$	$C_A = C_{A0}e^{-kt}$	$\ln C_A = \ln C_{A0} - kt$
2	$r = kC_A^2$	$C_A = (C_{A0}^{-1} + kt)^{-1}$	$C_A^{-1} = C_{A0}^{-1} + kt$

As noted in Section 13.3, the linearized models will be used only as initial estimates of possible rate forms and values of their coefficients, but our “best estimates” will be acquired from nonlinear regression of the  $C_A(t)$  function. The linearized models suggest plots of  $C_A$  vs.  $t$ ,  $\ln C_A$  vs.  $t$ , and  $C_A^{-1}$  vs.  $t$  will be linear if the rate follows a zero, first, or second order rate expression, so we plot those three quantities to start our analysis:

```
1 t = (0:10:70)'; % s
2 Ca = [14.6, 11.1, 7.7, 5.3, 3.4, 2.0, 1.2, 0.7]'; % ummol/L
3
4 subplot 131; plot(t, Ca);
5 subplot 132; plot(t, log(Ca));
6 subplot 133; plot(t, 1./Ca);
```



From this figure it looks like the zero and first order plots are fairly linear but the second order plot is definitely not linear, so we exclude second order going forward. We now perform linear regression on the linearized versions of the zero and first order models to get estimates of the regression parameters:

```
1 disp(fitlm(t, Ca));
2 disp(fitlm(t, log(Ca)));
```

The output from these two regression procedures are summarized in the table below, but additional statistics such as  $p$ -values and adjusted  $R^2$  values are irrelevant because we're only using the linear fits to provide a starting point for  $C_{A0}$  and  $k$  in the full, nonlinear regression.

Order	$\beta_1$	$\beta_2$	$C_{A0}$ ( $\mu\text{mol} \cdot \text{L}^{-1}$ )	$k$
0	12.658	-0.197	12.658	$0.197 \mu\text{mol} \cdot \text{L}^{-1} \cdot \text{s}^{-1}$
1	2.854	-0.044	17.357	$0.044 \text{s}^{-1}$

Notice that the units of  $C_{A0}$  are always the same but the units of  $k$  vary depending on the order of the reaction.

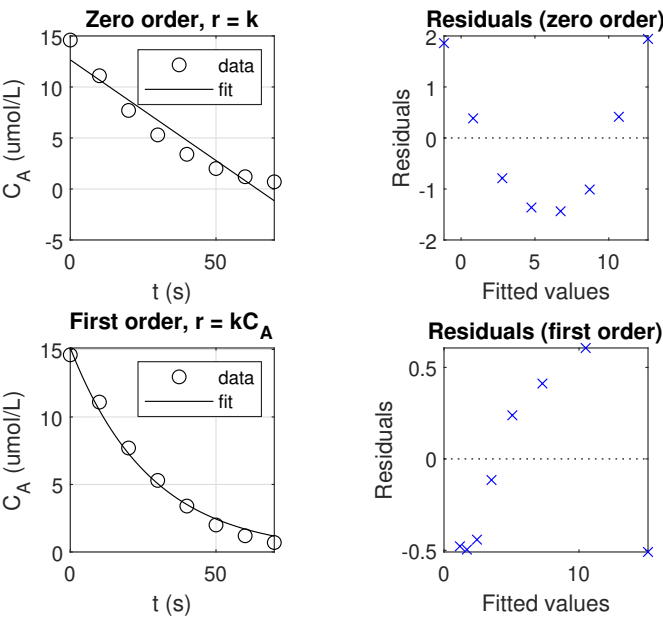
We now proceed with the full, nonlinear regression for the zero and first order models. We can put both local functions for the explicit forms of  $C_A$  into the very same script, and in this script we also examine the statistical quantities and residual plots for each fit.

```

1 % methylOrange.m
2 t = (0:10:70)'; % s
3 Ca = [14.6, 11.1, 7.7, 5.3, 3.4, 2.0, 1.2, 0.7]'; % ummol/L
4
5 % Zero-order model
6 beta0 = [12.658, 0.197]; % [Ca0, k]
7 model = fitnlm(t, Ca, @funZero, beta0);
8 tFit = linspace(min(t), max(t));
9 CaFit = feval(model, tFit);
10 subplot 221; hold on
11 plot(t, Ca, 'ko'); plot(tFit, CaFit, 'k-');
12 hold off;
13 subplot 222
14 plotResiduals(model, 'fitted')
15
16 % First-order model
17 beta0 = [17.357, 0.044]; % [Ca0, k]
18 model = fitnlm(t, Ca, @funFirst, beta0);
19 tFit = linspace(min(t), max(t));
20 CaFit = feval(model, tFit);
21 subplot 223; hold on
22 plot(t, Ca, 'ko'); plot(tFit, CaFit, 'k-');
23 hold off;
24 subplot 224
25 plotResiduals(model, 'fitted')
26
27 % Local functions
28 function Ca = funZero(beta, t)

```

```
29 Ca0 = beta(1);
30 k = beta(2);
31 Ca = Ca0 - k*t;
32 end
33
34 function Ca = funFirst(beta, t)
35 Ca0 = beta(1);
36 k = beta(2);
37 Ca = Ca0*exp(-k*t);
38 end
```



Statistic	Zero order	First order
Overall $p$	$1.31 \times 10^{-4}$	$4.22 \times 10^{-8}$
Adj. $R^2$	0.91	0.99
$C_{A0}$ (fitted)	12.66	15.11
$p$ -val	$1.34 \times 10^{-5}$	$4.54 \times 10^{-8}$
$k$ (fitted)	0.197	0.036
$p$ -val	$1.31 \times 10^{-4}$	$1.30 \times 10^{-6}$

In every regard the zero order model is a worse model than the first order model: all  $p$ -values are larger, as is the adjusted  $R^2$  value, but most importantly is the obvious

pattern to the residuals. Such a pattern—smoothly varying from underestimate to overestimate and back to underestimate—strongly implies that the zero order model fails to capture the appropriate influence of the predictor variable, time, on the response variable, concentration.

Even the first order's residuals seem to exhibit some sort of pattern, albeit less obviously than the zero order's residuals. Given these two choices (and remembering that second order was dismissed based on the linearized plot) we can conclude that the first order rate expression,  $r = kC_A$ , does the best job of these three models at capturing the trends of the experimental data.

From these results the initial concentration is  $C_{A0} = 15.11 \mu\text{mol} \cdot \text{L}^{-1}$  and the first order rate constant is  $0.036 \text{ s}^{-1}$ . Notice that these are slightly different from the predictions of the linearized model, and should generally be considered the more accurate estimates. The value of  $C_{A0}$  estimated from the regression procedure is also similar to the value of  $C_{A0}$  reported experimentally, which was  $14.6 \mu\text{mol} \cdot \text{L}^{-1}$ .

Notice that the conclusion at the end of Example 13.1 was *not* a categorical statement that the methyl orange degradation follows a first order power law model; instead we restricted the statement to an observation that one of the models was supported by the data and the others were not supported by the data.

How do we know that there's not another model out there that's an even better fit of the data? We don't! We're usually guided by our engineering judgment to decide when a model captures enough of the experimentally observed trends to satisfy our needs, but those needs vary from case to case.

Let's look at another example which compares an elementary rate expression to a widely-used non-elementary rate expression, the Langmuir-Hinshelwood rate expression.

### Example 13.2: Methyl blue degradation

Methyl blue, another textile dye, can be broken down by UV light in the presence of titanium dioxide in a reactor similar to the one used to break down methyl orange in Example 13.1. The data below were collected from a 250 mL batch reactor containing  $1 \text{ g} \cdot \text{L}^{-1} \text{ TiO}_2$  and equipped with a 125 W UV lamp. The reactor was stirred and the initial concentration of methyl blue was  $C_{A0} = 0.25 \text{ mmol} \cdot \text{L}^{-1}$ .

Compare a first order model to the Langmuir-Hinshelwood (LH) model,

$$r = \frac{kC_A}{1 + KC_A},$$

where  $k$  and  $K$  are positive constants. The LH model is commonly used for heterogeneous catalysis, *i.e.*, when the catalyst is solid phase and the reactant is either liquid or gas phase.

$t$ (min)	0	10	20	30	40	50	60
$C_A/C_{A0}$	1.00	0.71	0.60	0.50	0.42	0.35	0.28

### Solution

The integrated rate expression for the first order model will be the same as the one we used for Example 13.1,

$$C_A = C_{A0}e^{-kt}.$$

The material balance for the LH model is

$$\frac{dC_A}{dt} = \frac{-kC_A}{1 + KC_A}, \quad C_A(0) = C_{A0}$$

which can be integrated to yield

$$\ln \frac{C_A}{C_{A0}} + K(C_A - C_{A0}) = -kt.$$

This expression is somewhat problematic because it's *implicit* in  $C_A$ : the variable  $C_A$  cannot be isolated. To overcome this problem we instead write an explicit function for *time* as

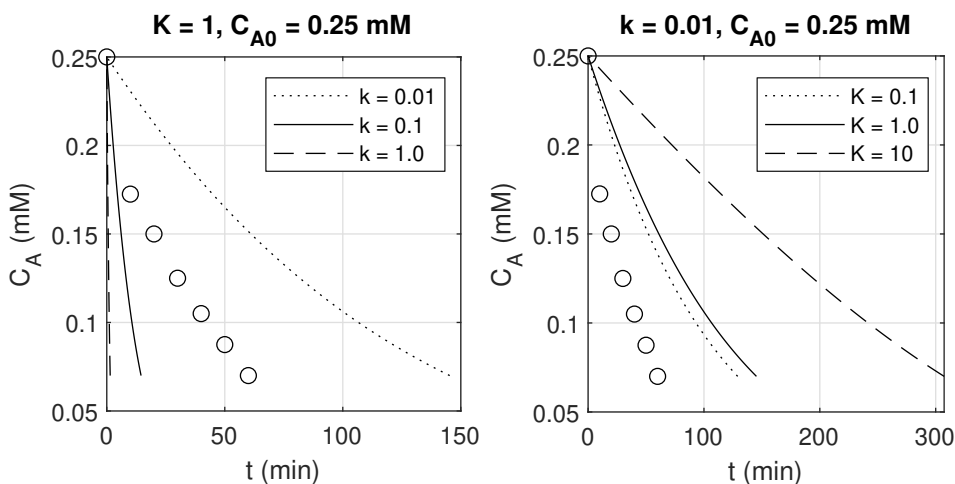
$$t = \frac{1}{k} \left[ \ln \frac{C_{A0}}{C_A} + K(C_{A0} - C_A) \right]$$

and perform nonlinear regression on this expression with  $C_A$  as the independent variable and  $t$  as the dependent variable. That's a little bit backwards compared to previous analyses but the `fminlm` function won't care one bit which way we provide the variables as long as we write the local function correctly.



We also face the problem of developing initial guess for both models. In the first order model we can use  $C_{A0} = 0.25 \text{ mmol} \cdot \text{L}^{-1}$  as an initial guess for  $C_{A0}$  since that's given in the data set, and since the concentration decays to  $e^{-1} = 36\%$  of its initial value in about 45 min, we can use  $k = 45^{-1} = 0.02 \text{ min}^{-1}$  as an initial guess for  $k$ .

For the LH model it's less obvious what to choose as an initial guess. Certainly we'll use the same guess for  $C_{A0}$  as we did for the first order model, and one approach to find appropriate guesses for  $k$  and  $K$  is to guess a few values over a few orders-of-magnitude for each parameter to observe their effect on the fit:



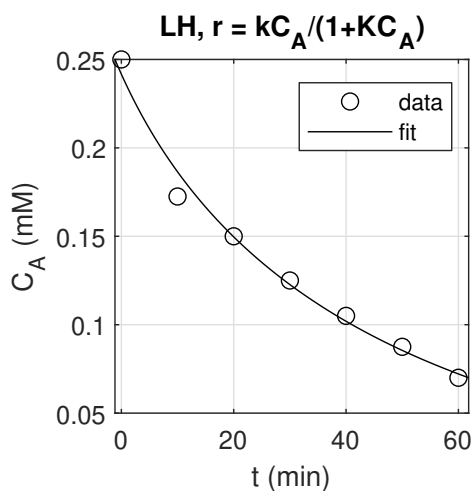
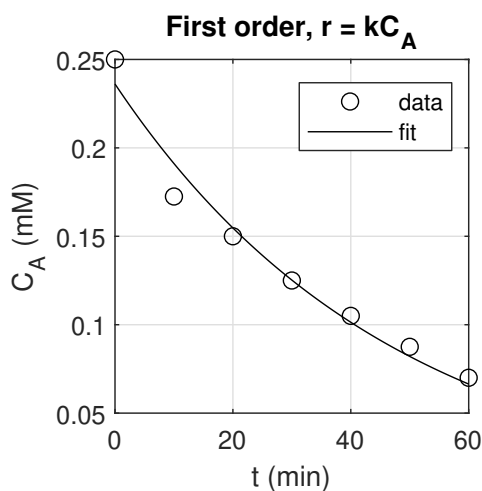
From the left plot it seems that a good estimate for  $k$  will be between 0.01 and 0.1, and from the right plot it looks like an asymptote is approached as  $K \rightarrow 0$ , so a low value such as 0.1 will probably be a good guess. With these as the initial guesses let's take a look at the nonlinear regression for the first order and LH models:

```
1 % methylBlue.m
2 t = (0:10:60)'; % min
3 Ca = [1.00; 0.71; 0.60; 0.50; 0.42; 0.35; 0.28];
4 Ca = Ca*0.25; % mmol/L
5
6 % First order model:
7 beta0 = [0.25; 0.02]; % [Ca0; k]
8 model = fitnlm(t, Ca, @funFirst, beta0);
9 disp(model)
10 tFit = linspace(min(t), max(t));
11 CaFit = feval(model, tFit);
12 subplot 121; hold on
13 plot(t, Ca, 'ko')
14 plot(tFit, CaFit, 'k')
```

```

15 hold off
16
17 % LH model:
18 beta0 = [0.25; 0.01; 0.1]; % [Ca0; k; K]
19 model = fitnlm(Ca, t, @funLH, beta0);
20 disp(model)
21 CaFit = linspace(min(Ca), max(Ca));
22 tFit = feval(model, CaFit);
23 subplot 122; hold on
24 plot(t, Ca, 'ko')
25 plot(tFit, CaFit, 'k')
26 hold off
27
28 % Local functions
29 function Ca = funFirst(beta, t)
30 Ca0 = beta(1);
31 k = beta(2);
32 Ca = Ca0*exp(-k*t);
33 end
34
35 function t = funLH(beta, Ca)
36 Ca0 = beta(1);
37 k = beta(2);
38 K = beta(3);
39 t = (1/k)*(log(Ca0./Ca)+K*(Ca0-Ca));
40 end

```



From the plots and statistics it seems that both models offer a fair approximation of the experimental data, but with one significant problem: the coefficient  $K$  for the LH

model is  $-2.02$ ! According to the problem statement the parameters  $k$  and  $K$  must be positive in the LH model, so we immediately conclude that this model is not appropriate for this reaction.

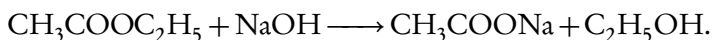
Since our choice was restricted to just these two models, we further conclude that the first order model is a better choice. From the regression output we get  $C_{A0} = 0.238 \text{ mmol} \cdot \text{L}^{-1}$  ( $p\text{-value} = 7.09 \times 10^{-7}$ ) and  $k = 0.021 \text{ min}^{-1}$  ( $p\text{-value} = 2.86 \times 10^{-5}$ ).

Example 13.2 shows that simply because a model has more adjustable parameters compared to another model, it's not necessarily a better fit to an experimental data set compared to a model with fewer adjustable parameters. Even if all the usual statistical indicators are acceptable we must always evaluate the physical reasonableness of the coefficients before accepting the model.

Our final example looks at a more complicated mechanism, the bimolecular reaction between ethyl acetate and sodium hydroxide.

### Example 13.3: Saponification of ethyl acetate

The saponification of ethyl acetate ( $\text{CH}_3\text{COOC}_2\text{H}_5$ , EA) with sodium hydroxide (NaOH) proceeds according to the elementary reaction



The conductivity of NaOH is significantly higher than that of  $\text{CH}_3\text{COONa}$ , which allows a simple conductivity probe to easily monitor the progress of the reaction. Using a novel microreactor, Schneider and Stoessel<sup>a</sup> measured the conversion of ethyl acetate at two different temperatures as reported below.

Estimate the rate constant  $k$  at  $25^\circ\text{C}$  and the activation energy  $E_a$  ( $\text{kJ} \cdot \text{mol}^{-1}$ ) for this saponification reaction. The initial concentrations of EA and sodium hydroxide are known to high accuracy to be  $0.8 \text{ mol} \cdot \text{L}^{-1}$  and  $1.0 \text{ mol} \cdot \text{L}^{-1}$ , respectively, and should *not* be treated as unknown parameters for this problem.

$t$ (s)	$X_{EA}$	
	40 °C	70 °C
1	0.09	0.32
2	0.14	0.43
4	0.27	0.63
5	0.32	0.69
7	0.39	0.77
10	0.50	0.85
20	0.69	0.95

<sup>a</sup>M.A. Schneider, F. Stoessel. Determination of the kinetic parameters of fast exothermal reactions using a novel microreactor-based calorimeter. Chem. Eng. J. 115 (2005) 73-83.

### Solution

The first thing to do is to integrate the material balance,

$$\frac{dC_A}{dt} = r_A = -kC_A C_B, \quad C_A(0) = C_{A0},$$

where species A is ethyl acetate ( $X_A = X_{EA}$ ) and species B is sodium hydroxide.

Regression can only be performed on one equation so it's not feasible to write another ODE for  $C_B$  and then somehow ask `fminlm` to fit expressions for *both*  $C_A$  and  $C_B$ . Instead we need to replace  $C_B$  with an expression for  $C_A$ , which we can do by using the conversion from Equation (2.35) as

$$\frac{n_B}{V} = C_B = C_{A0}(\Theta_B - X_A)$$

where  $\Theta_B = n_{B0}/n_{A0} = C_{B0}/C_{A0}$ . Similarly  $C_A = C_{A0}(1 - X_A)$ , and substituting these two expressions into the material balance yields

$$\frac{dC_A}{dt} = -kC_{A0}^2(1 - X_A)(\Theta_B - X_A).$$

Rather than transform all of the  $X_A$  terms into  $C_A$ , we can replace the differential on

the left-hand side with one in terms of  $X_A$  as

$$\frac{dC_A}{dt} = \frac{d}{dt}(C_{A0}(1 - X_A)) = -C_{A0} \frac{dX_A}{dt},$$

and therefore the material balance can be expressed entirely in terms of  $X_A$  as

$$\frac{dX_A}{dt} = k C_{A0} (1 - X_A)(\Theta_B - X_A), \quad X_A(0) = 0.$$

Rearranging this separable ODE yields

$$\int_0^{X_A} \frac{dX_A}{(1 - X_A)(\Theta_B - X_A)} = \int_0^t k C_{A0} dt.$$

The integral on the right is straightforward to evaluate; the integral on the left can be found in Appendix A as Equation (A.12). Evaluating these integrals yields

$$\frac{1}{\Theta_B - 1} \ln \frac{\Theta_B - X_A}{\Theta_B (1 - X_A)} = k C_{A0} t. \quad (\text{E-1})$$

This expression is a bit cumbersome to rearrange for an explicit function of  $X_A$ , but after a bit of algebra the result is

$$X_A = \frac{\Theta_B (g(t) - 1)}{\Theta_B g(t) - 1} \quad (\text{E-2})$$

where

$$g(t) = \exp[(\Theta_B - 1)(k C_{A0} t)]. \quad (\text{E-3})$$

We can make two choices at this stage: use *only* Equation (E-1) but with  $t$  as the *response* variable and  $X_A$  as the *predictor* variable—which is what we did in Example 13.2—or we can use Equations (E-2) and (E-3) and perform the nonlinear regression as usual with  $t$  as the predictor and  $X_A$  as the response. We'll opt to use Equations (E-2) and (E-3) because it preserves the expected predictor-response relationship.

We have another choice to make as well: the problem is asking us to find an activation energy and a rate constant, which implies that the rate constant follows an

Arrhenius form as

$$k(T) = A \exp \left[ \frac{-E_a}{RT} \right] \quad (3.9)$$

which can also be written as the ratio of two rate constants at two different temperatures as

$$\frac{k(T_2)}{k(T_1)} = \exp \left[ \frac{E_a}{R} \left( \frac{1}{T_1} - \frac{1}{T_2} \right) \right] \quad (3.10)$$

Our choice is as follows:

**Option 1:** Perform nonlinear regression using Equations (E-2) and (E-3) for the data at 40 °C to get  $k(40^\circ\text{C})$ , then perform nonlinear regression again for the data at 70 °C to get  $k(70^\circ\text{C})$ . Use Equation (3.10) twice, once to get  $E_a$  and again to get  $k(25^\circ\text{C})$ .

**Option 2:** Perform nonlinear regression using Equations (3.9), (E-2) and (E-3) *once* for the entire data set to get  $A$  and  $E_a$ , then use Equation (3.9) once to get  $k(25^\circ\text{C})$ .

Option 1 is a straightforward application of standard nonlinear regression techniques but requires a bit of extra math when we're done, whereas Option 2 is a somewhat less routine application of nonlinear regression that simplifies the post-regression analysis. We'll show both approaches here and leave it to you to decide which seems preferable.

**Option 1.** The script below performs nonlinear regression twice, once for the 40 °C data and again for the 70 °C data. Note that the regression function is the same for both data sets, and that the code to produce the plots shown below was omitted to keep the code shorter. The semicolons were intentionally left off Lines 7 and 8 to display the results from `fitnlm`.

```

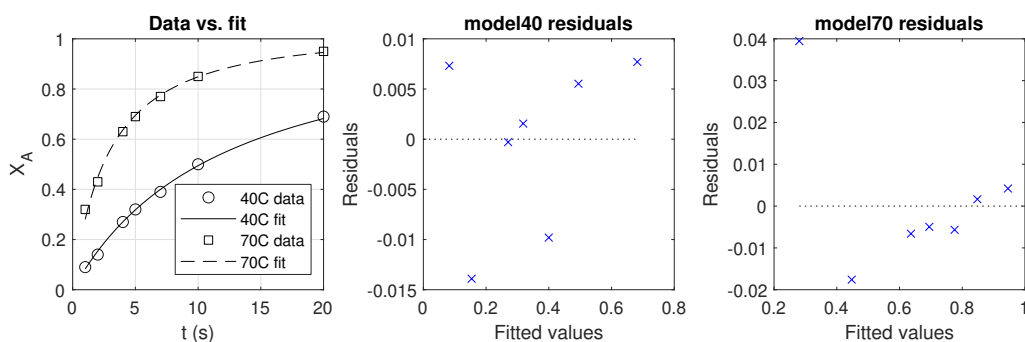
1 % option1.m
2 t = [1; 2; 4; 5; 7; 10; 20]; % s
3 Xa40 = [0.09; 0.14; 0.27; 0.32; 0.39; 0.50; 0.69];
4 Xa70 = [0.32; 0.43; 0.63; 0.69; 0.77; 0.85; 0.95];
5
6 beta0 = 0.01; % k
7 model40 = fitnlm(t, Xa40, @fun, beta0)
8 model70 = fitnlm(t, Xa70, @fun, beta0)
9
10 k40 = model40.Coefficients.Estimate; % L/mol-s
11 k70 = model70.Coefficients.Estimate; % L/mol-s
12 R = 8.314; % J/mol-K
13 T40 = 40+273; % K

```

```

14 T70 = 70+273; % K
15 Ea = R*log(k70/k40)/(1/T40-1/T70); % J/mol-K
16 k25 = k40*exp(Ea/R*(1/T40-1/(25+273))); % L/mol-s
17
18 function Xa = fun(beta, t)
19 k = beta(1); % L/mol-s
20 Ca0 = 0.8; % mol/L
21 Cb0 = 1.0; % mol/L
22 ThetaB = Cb0/Ca0;
23 g = exp((ThetaB-1)*(k*Ca0*t));
24 Xa = ThetaB*(g-1)./(ThetaB*g - 1);
25 end

```



All the regular descriptive statistics for both fits are acceptable: all  $p$ -values are less than  $10^{-7}$ , and all adjusted  $R^2$  values are above 0.99. The residual plots have no apparent pattern, so we conclude that the model is supported by the data. From Lines 10 through 16 we find that  $E_a = 42.7 \text{ kJ} \cdot \text{mol}^{-1}$  and  $k(25^\circ\text{C}) = 0.039 \text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1}$ .

**Option 2.** This approach requires a more clever use of the local function passed to `fitnlm`: from an input of  $t$  and  $\beta$  we have to produce a single vector that contains estimates of *all*  $X_A$  at every temperature, which will be a vector of size 14-by-1 instead of size 7-by-1 as in Option 1. The syntax requirements of `fitnlm` also require the input matrix for predictor variables to have the same number of rows as the response variable, so our  $t$  vector will also need to be 14-by-1.

The first 7 rows, accessed by MATLAB's indexing notation, will be used to calculate  $X_A$  at  $T = 40^\circ\text{C}$  and the next 7 rows will be used to calculate  $X_A$  at  $T = 70^\circ\text{C}$ . Another approach could be to use an `if` statement within the local function, but indexing is equally effective.

```

1 % option2.m

```

```

2 | t = [1; 2; 4; 5; 7; 10; 20]; % s
3 | Xa40 = [0.09; 0.14; 0.27; 0.32; 0.39; 0.50; 0.69];
4 | Xa70 = [0.32; 0.43; 0.63; 0.69; 0.77; 0.85; 0.95];
5 | t = [t; t]; % 14-by-1
6 | Xa = [Xa40; Xa70]; % 14-by-1
7 |
8 | beta0 = [1e6; 25]; % [A, Ea]
9 | model = fitnlm(t, Xa, @fun, beta0)
10 | A = model.Coefficients.Estimate(1); % L/mol-s
11 | Ea = model.Coefficients.Estimate(2); % kJ/mol
12 | k25 = A*exp(-Ea*1e3/(8.314*(25+273))); % L/mol-s
13 |
14 | function Xa = fun(beta, t)
15 | A = beta(1); % L/mol-s
16 | Ea = beta(2)*1e3; % J/mol
17 | R = 8.314; % J/mol-K
18 | Ca0 = 0.8; % mol/L
19 | Cb0 = 1.0; % mol/L
20 | ThetaB = Cb0/Ca0;
21 |
22 | k40 = A*exp(-Ea/(R*(40+273)));
23 | k70 = A*exp(-Ea/(R*(70+273)));
24 |
25 | g40 = exp((ThetaB-1)*(k40*Ca0*t(1:7)));
26 | g70 = exp((ThetaB-1)*(k70*Ca0*t(8:14)));
27 |
28 | Xa = zeros(14, 1);
29 | Xa(1:7) = ThetaB*(g40-1)./(ThetaB*g40 - 1);
30 | Xa(8:14) = ThetaB*(g70-1)./(ThetaB*g70 - 1);
31 | end

```

The output plots and statistics from this fit procedure are nearly identical to those of Option 1 except that the fitted parameters are  $A = 1.203 \times 10^6 \text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1}$  and  $E_a = 42.7 \text{ kJ} \cdot \text{mol}^{-1}$ . From Line 12 we can also calculate  $k(25^\circ\text{C}) = 0.039 \text{ L} \cdot \text{mol}^{-1} \cdot \text{s}^{-1}$ , again identical to the approach used in Option 1.

As a final comment to Example 13.3, Option 2 is slightly preferable to Option 1 because it finds the best estimates of  $A$  and  $E_a$  for the entire data set simultaneously, rather than treating each data set independently. Option 1 may also require additional regression analyses to get proper estimates from Equation (3.9) if more than two temperature sets were collected.



# Appendix A

## Useful Math Formulae

### The Quadratic Formula

Given

$$ax^2 + bx + c = 0$$

the variable  $x$  is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (\text{A.1})$$

where the quantity  $b^2 - 4ac$  is sometimes called the *discriminant*.

### Indefinite Integrals

These forms can be useful with PFRs and PBRs when the full material balance is used because the bounds of the integral are often from something like  $n_{i0}$  to  $n_i$ .

$$\int a \, dx = ax \quad (\text{A.2})$$

$$\int \frac{1}{x} \, dx = \ln|x| \quad (\text{A.3})$$

$$\int x^n \, dx = \frac{1}{n+1} x^{n+1}, \quad n \neq -1 \quad (\text{A.4})$$

$$\int \frac{1}{ax+b} \, dx = \frac{1}{a} \ln|ax+b| \quad (\text{A.5})$$

$$\int \frac{1}{(x+a)(x+b)} \, dx = \frac{1}{b-a} \ln \frac{a+x}{b+x}, \quad a \neq b \quad (\text{A.6})$$

$$\int \frac{1}{ax^2 + bx + c} dx = \frac{2}{\sqrt{4ac - b^2}} \tan^{-1} \frac{2ax + b}{\sqrt{4ac - b^2}} \quad (\text{A.7})$$

### Definite Integrals from 0 to $X_A$

These forms can be useful with PFRs and PBRs when the material balance is written in terms of conversion  $X_A$  because the lower bound is typically zero.

$$\int_0^{X_A} \frac{1}{1-x} dx = \ln \frac{1}{1-X_A} \quad (\text{A.8})$$

$$\int_0^{X_A} \frac{1}{(1-x)^2} dx = \frac{X_A}{1-X_A} \quad (\text{A.9})$$

$$\int_0^{X_A} \frac{1}{1+ax} dx = \frac{1}{a} \ln(1+aX_A) \quad (\text{A.10})$$

$$\int_0^{X_A} \frac{1+ax}{1-x} dx = (1+a) \ln \frac{1}{1-X_A} - ax \quad (\text{A.11})$$

$$\int_0^{X_A} \frac{1}{(1-x)(a-x)} dx = \frac{1}{a-1} \ln \frac{a-X_A}{a(1-X_A)}, \quad a \neq 1 \quad (\text{A.12})$$

## Appendix B

### Vector Formulations

In highly technical terms, MATLAB is *super good* at vector and matrix operations! Most of the examples in this text make use of at least vector math, such as when you see something like

```
1 A = [0.019; 0.013; 0.021];  
2 Ea = [26.8; 27.4; 27.1]*1e3;  
3 k = A.*exp(-Ea/(R*T));
```

This code snippet defined the variables  $A$  and  $E_a$  as vectors instead of scalars, then used MATLAB's elementwise operators  $.*$  and  $\exp$  to calculate  $k = A \exp[-E_a/RT]$  for all three reactions with just a single line. This is obviously more convenient than defining three separate variables for  $A$ , three separate variables for  $E_a$ , and calculating  $E_a$  three different times.

Before we dive in, keep one thing in mind: *the shape and order of operations generally cannot be switched!* In Line 3 of the code snippet above it wouldn't matter if we reversed the order to something like

```
3 k = exp(-Ea/(T*R)) .* A;
```

The reason the order doesn't matter is because the elementwise operator has a property called the *commutative property*, which just means that the order of operations can be reversed without changing the outcome.<sup>1</sup> However, the operations we'll make use of in the next few subsections are *not* commutative, which—as described in greater detail in your linear algebra course—is a hallmark of matrix algebra.

Let's make the following definitions: the number of species  $i$  runs from A to Z and the number of reactions  $j$  runs from 1 to  $q$ . The stoichiometry of species  $i$  in reaction  $j$  is  $\nu_{ij}$ , the rate of reaction  $j$  is  $r_j$ , the molar flow rate of species  $i$  is  $n_i$ , the heat capacity of species  $i$  is  $C_{p,i}$ , and the enthalpy of reaction for reaction  $j$  is  $\Delta H_j$ .

---

<sup>1</sup>Granted, that's a little bit of circular reasoning.

## B.1 Variables can be expressed as vectors and matrices

The most useful shape of a vector for our purposes is a column vector, mostly because MATLAB's `ode45` and `fsolve` functions require column vectors as outputs. The column vector of all species molar flow rates ( $n_i$ ) is  $\vec{n}$ , or

$$\vec{n} = \begin{bmatrix} n_A \\ n_B \\ \vdots \\ n_Z \end{bmatrix}, \quad \text{size} = [Z \times 1]. \quad (\text{B.1})$$

All other species properties—namely the inlet molar flow rates  $\vec{n}_0$  and heat capacities  $\vec{C}_{P_i}$ —are defined in an analogous way and are size  $Z \times 1$ . The column vector of all reaction rates ( $r_j$ ) is  $\vec{r}$ , or

$$\vec{r} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_q \end{bmatrix}, \quad \text{size} = [q \times 1]. \quad (\text{B.2})$$

All other properties of the reactions—the enthalpies of reaction  $\Delta\vec{H}$ , Gibbs energies of reaction  $\Delta\vec{G}$ , and heat capacity changes  $\Delta\vec{C}_p$ —are defined in an analogous way and are size  $q \times 1$ . The matrix of stoichiometric coefficients of all species  $i$  in reaction  $j$  ( $v_{ij}$ ) is  $\vec{v}$ , or

$$\vec{v} = \begin{bmatrix} v_{A1} & v_{A2} & \cdots & v_{Aq} \\ v_{B1} & v_{B2} & \cdots & v_{Bq} \\ \vdots & \vdots & \ddots & \vdots \\ v_{Z1} & v_{Z2} & \cdots & v_{Zq} \end{bmatrix}, \quad \text{size} = [Z \times q] \quad (\text{B.3})$$

## B.2 Vector and matrix operations greatly simplify many common calculations

With these definitions we can replace most of the common operations needed for reactor analysis with simple vector and matrix operations in MATLAB. One of the most common operations in reaction engineering is to sum the products of two quantities over an index, such as

$$\sum_{i=A}^Z n_i C_{P_i}$$

which shows up all over the place and can be easily expressed in MATLAB<sup>2</sup> as

$$\sum_{i=A}^Z n_i C_{Pi} = \text{sum}(n.*Cp)$$

Similarly the sum of rates multiplied by enthalpies of reaction is simply

$$\sum_{j=1}^q r_j \Delta H_j = \text{sum}(r.*dH)$$

One of the most significant savings in terms of both time and likelihood of making mistakes is in the calculation of net rates. The vector of net rates of reaction for each species requires a bit of matrix algebra that we usually take care to avoid, the operation of *matrix multiplication*. Let's take a simple example of calculating the net rates of two species, A and B, participating in three reactions with reaction rates  $r_1, r_2, r_3$ . We're going to perform matrix multiplication between the matrix of stoichiometric coefficients  $\vec{v}$  and the column vector of reaction rates  $\vec{r}$ , or

$$\vec{v} \times \vec{r},$$

where we use the  $\times$  symbol to denote matrix multiplication.<sup>3</sup> Expanding each term to our example yields

$$\begin{bmatrix} v_{A1} & v_{A2} & v_{A3} \\ v_{B1} & v_{B2} & v_{B3} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

Notice that the number of *columns* of  $\vec{v}$  equals the number of *rows* of  $\vec{r}$ : this must *always* be the case for matrix multiplication, and another way of stating this requirement is to say that the inner dimensions must agree.<sup>4</sup> The result of matrix multiplication will be a matrix with the same number of *rows* as  $\vec{v}$  and the same number of *columns* as  $\vec{r}$ . Let's call this result  $y$  for now as we work out its details, so that

$$\vec{v} \times \vec{r} = \vec{y}.$$

<sup>2</sup>There are other ways to do exactly the same operation, such as using the dot product as  $\vec{n} \cdot \vec{C}_{Pi}$ . The savings in computation time is basically zero, and other formats tend to make the code less similar to the original equations and therefore harder to troubleshoot.

<sup>3</sup>This symbol is also commonly used to denote something called a "cross product" in linear algebra, but we never use cross products in this text.

<sup>4</sup>That's the most common kind of error MATLAB will complain about if you mess up the sizes when you try to do matrix multiplication.

We're always going to have a matrix multiplied by a column vector, and therefore the result  $\vec{y}$  will always be a column vector with the same number of rows as  $\vec{v}$ , or

$$\begin{bmatrix} v_{A1} & v_{A2} & v_{A3} \\ v_{B1} & v_{B2} & v_{B3} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

To calculate each element  $y_i$  of  $\vec{y}$ , we choose a row of  $\vec{v}$ , multiply the first row element by the first element of  $\vec{r}$ , add to this the product of the second row element by the second element of  $\vec{r}$ , and so on until all such products are formed and summed. This sum is the first element of  $\vec{y}$ . For example, the calculation of  $y_1$  is

$$\begin{bmatrix} v_{A1} & v_{A2} & v_{A3} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} y_1 \end{bmatrix} = \begin{bmatrix} v_{A1} r_1 + v_{A2} r_2 + v_{A3} r_3 \end{bmatrix}$$

The second element of  $\vec{y}$  is calculated using the second row of  $\vec{v}$  as

$$\begin{bmatrix} v_{B1} & v_{B2} & v_{B3} \end{bmatrix} \times \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} y_2 \end{bmatrix} = \begin{bmatrix} v_{B1} r_1 + v_{B2} r_2 + v_{B3} r_3 \end{bmatrix}$$

We can identify both of the sums we just wrote as the net rates of each species, which suggests that the vector of net rates for any system can be calculated by using matrix multiplication,

$$\text{vector of net rates} = \begin{bmatrix} r_A \\ r_B \\ \vdots \\ r_Z \end{bmatrix} = \vec{v} \times \vec{r} = \text{nu} * \text{r}$$

where you need to be careful *not* to use the elementwise multiplier in MATLAB and to order them in *exactly* the way shown here (matrix multiplication is not commutative!).

We can also use matrix multiplication to calculate many of the “ $\Delta$ ” quantities for reactions

such as  $\Delta H$ ,  $\Delta G$ , and  $\Delta C_P$ . Denoting the material property generally as  $M_i$  we can write

$$\vec{v}^T \times \vec{M} = \begin{bmatrix} \nu_{A1} & \nu_{B1} & \cdots & \nu_{Z1} \\ \nu_{A2} & \nu_{B2} & \cdots & \nu_{Z2} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{Aq} & \nu_{Bq} & \cdots & \nu_{Zq} \end{bmatrix} \times \begin{bmatrix} M_A \\ M_B \\ \vdots \\ M_Z \end{bmatrix}$$

where  $\vec{v}^T$  is the *transpose* of  $\vec{v}$ , which means that we've swapped the rows and columns. The reason for using the transpose is that when we perform matrix multiplication we get the quantities of interest, namely

$$\vec{v}^T \times \vec{M} = \begin{bmatrix} \nu_{A1} & \nu_{B1} & \cdots & \nu_{Z1} \\ \nu_{A2} & \nu_{B2} & \cdots & \nu_{Z2} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{Aq} & \nu_{Bq} & \cdots & \nu_{Zq} \end{bmatrix} \times \begin{bmatrix} M_A \\ M_B \\ \vdots \\ M_Z \end{bmatrix} = \begin{bmatrix} \nu_{A1}M_A + \nu_{B1}M_B + \cdots + \nu_{Z1}M_Z \\ \nu_{A2}M_A + \nu_{B2}M_B + \cdots + \nu_{Z2}M_Z \\ \vdots \\ \nu_{Aq}M_A + \nu_{Bq}M_B + \cdots + \nu_{Zq}M_Z \end{bmatrix}$$

Each of the sum terms on the right are a  $\Delta M$  term, so we have

$$\text{vector of } \Delta M \text{ terms} = \begin{bmatrix} \Delta M_1 \\ \Delta M_2 \\ \vdots \\ \Delta M_q \end{bmatrix} = \vec{v}^T \times \vec{M} = \text{nu}' * \text{M}$$

where  $\Delta M_j$  is the  $\Delta M$  quantity of reaction  $j$ ,  $\text{nu}'$  is the *transpose* of the stoichiometric coefficient matrix and  $\text{M}$  is the column vector of material properties such as  $\Delta H_{f,i}^\circ$ ,  $C_{P,i}$ , etc.

### B.3 CSTR equations can be written quickly and easily using vectors and matrices

The operations developed above greatly simplify the CSTR material balance,

$$0 = n_i - n_{i0} - r_i V \quad (4.1)$$

We can rewrite Equation (4.1) in vector form as

$$\vec{0} = \vec{n} - \vec{n}_0 - (\vec{v} \times \vec{r}) V = \text{n} - \text{n0} - \text{nu} * \text{r} * V$$

Once again, that's *all* of the species material balances, all at once, including the calculation of net rates, and you have to be careful *not* to use the elementwise multiplication between  $\mathbf{n_u}$  and  $\mathbf{r}$  (since  $V$  is a constant it can be before or after the matrix multiplication). For example, if you're dealing with a problem containing species A, B, C, and D then with this one command you're creating the following vector of material balances all at once:

$$\mathbf{n} - \mathbf{n_0} - \mathbf{n_u} * \mathbf{r} * V = \begin{bmatrix} n_A - n_{A0} - V \sum v_{Aj} r_j \\ n_B - n_{B0} - V \sum v_{Bj} r_j \\ n_C - n_{C0} - V \sum v_{Cj} r_j \\ n_D - n_{D0} - V \sum v_{Dj} r_j \end{bmatrix}$$

In most applications we don't actually write the vector of zeros because the material balance is typically written within the local function which is then used by `fsolve` to find values of  $\mathbf{n}$  (or whatever is unknown) to satisfy the material balances. Instead, the material balances are typically assigned to the output of the local function as something like

```
1 function out = fun(n)
2 % define n0, nu, r, V, etc
3 out = n - n0 - nu*r*V;
4 end
```

Similarly the general energy balance for the CSTR,

$$0 = \sum_{i=A}^Z n_{i0} C_{P,i} (T - T_0) + V \sum_{j=1}^q r_j \Delta H_j - Q - W_s, \quad (4.2)$$

can be written in vector form as

$$0 = \text{sum}(\mathbf{n_0}.*\mathbf{C_p})*(T-T_0) + V*\text{sum}(\mathbf{r}*\mathbf{dH}) - Q,$$

which is again more commonly assigned as one of the output variables for a local function to be used within `fsolve` as

```
1 function out = fun(Y)
2 n = Y(1:end-1);
3 T = Y(end);
4 % define n0, nu, r, V, etc
5 out(1:end-1, 1) = n - n0 - nu*r*V;
6 out(end, 1) = sum(n0.*Cp)*(T-T0) + V*sum(r*dH) - Q;
7 end
```

Notice that unlike the material balance, the energy balance is only a single value, not a vector of values. That's something helpful to keep in mind when debugging your code: if the size



of whatever variable you're using to store your energy balance calculation isn't 1-by-1 then there's a typo somewhere.

## B.4 PFR and PBR equations can also be written using vectors and matrices

The material and energy balances for PFRs and PBRs—collectively referred to as tubular reactors here—can be vectorized in a manner similar to those of the CSTR. Aside from the equations themselves being slightly different, the biggest difference is that we don't expect the equations for tubular reactors to equal zero as we do for the CSTR. Instead the material balances are differential equations which are then used as part of a call to the ode45 function to numerically integrate the ODEs.

The material balances for all species in a PFR,

$$\frac{dn_i}{dV} = r_i \quad (7.1)$$

can be written as

$$\frac{d\vec{n}}{dV} = \begin{bmatrix} dn_A/dV \\ dn_B/dV \\ dn_C/dV \\ dn_D/dV \end{bmatrix} = \mathbf{n} \times \mathbf{r}$$

and the energy balance for a PFR,

$$\frac{dT}{dV} = \frac{\delta Q - \sum_{j=1}^q r_j \Delta H_j}{\sum_{i=A}^Z n_i C_{P,i}} \quad (7.2)$$

can be written as

$$dTdV = (dQ - \text{sum}(r.*dH))/\text{sum}(n.*Cp)$$

Both dndV and dTdV are usually defined within the ODE function file used with ode45, and if both are to be solved simultaneously, as in the case of most nonisothermal reactors, then they're bundled into a single vector dYdV as

$$dYdV = [dndV; dTdV];$$

This formulation was demonstrated in Example 7.6 for PFRs, and an analogous usage was demonstrated in Example 9.5 for PBRs.



## Appendix C

### The Gibbs Reactor

Recall from Chapter 2 that one of the jewels of thermodynamics is to recognize that the Gibbs energy decreases as equilibrium is approached as stated by the inequality

$$dG \leq 0 \quad \text{at constant } T \text{ and } P. \quad (2.7)$$

such that equilibrium is the point where the Gibbs energy is minimized. One method of identifying equilibrium conditions is to find the derivative of Equation (2.7) with respect to the variable(s) of interest and set the result equal to zero. Indeed this was the approach started in Section 2.4 and subsequently used throughout the text whenever we needed to determine equilibrium compositions.

One requirement of that approach is the ability to write a chemical reaction for the components of interest. Certainly this shouldn't be an issue when only a handful of components are present but when many are present then finding an appropriate set of reactions may pose quite a challenge indeed. Fortunately there is an alternative: The Gibbs Reactor.

A Gibbs Reactor is another type of equilibrium reactor in the sense that (i) it's a theoretical tool and (ii) the outlet stream from a Gibbs Reactor has achieved its equilibrium composition, both of which are the same considerations afforded to the Equilibrium Reactor in Section 2.7. Although limited to gas-phase systems the advantages of the Gibbs Reactor over the Equilibrium Reactor are two:

1. No chemical reactions must be written, and
2. The solution approach scales well to systems with many components.

The Gibbs Reactor is uniquely well-suited to initial design efforts because it can provide upper limits for conversion as well as identification of potentially substantial production of unwanted

by-products. Both can provide useful information for the design of down-stream separation equipment and cost analyses.

## C.1 $G$ can be minimized directly for ideal gases (Gibbs Reactor)

By definition, the total Gibbs energy  $G$  (an extensive quantity with units of energy, usually kJ) is related to the partial molar Gibbs energies of each species ( $\bar{G}_i$ ) as

$$G \equiv \sum_{i=A}^Z n_i \bar{G}_i = \sum_{i=A}^Z n_i \mu_i, \quad (\text{C.1})$$

where  $n_i$  is the molar flow rate (or moles) of species  $i$ ,  $\mu_i$  is the chemical potential of species  $i$ , and both sums are performed over all  $Z$  components. The second equality was made because the partial molar Gibbs energy of species  $i$  is defined as the chemical potential of species  $i$ , and we happen to have a convenient expression for  $\mu_i$  for ideal gases as

$$\mu_i = G_i + RT \ln y_i \quad (\text{ideal gas}) \quad (\text{C.2})$$

where  $y_i$  is the mole fraction of species  $i$  and  $G_i$  is the Gibbs energy of species  $i$  *at the temperature of the reaction*. If we pick the reference state to be the elements in their natural forms at the standard state then  $G_i = \Delta G_{f,i}^\circ$ , where  $\Delta G_{f,i}^\circ$  is the Gibbs energy of formation and is tabulated in Appendix D for many species at 298.15 K and  $P^\circ = 101.325$  kPa.

Most reactions do not occur at 298.15 K and 101.325 kPa, so we need a way to adjust  $\mu_i$  for temperature and pressure. The change in Gibbs energy at constant temperature is

$$dG = V dP,$$

which can be integrated for an ideal gas from the reference pressure to the actual pressure to yield

$$G_i = \Delta G_{f,i}^\circ + RT \ln \frac{P}{P^\circ} \quad (\text{C.3})$$

Equation (C.3) can be substituted into Equation (C.2), and the result then substituted into Equation (C.1) to yield the working equation for a *Gibbs reactor*,

$$G = \sum_{i=A}^Z n_i \left[ \Delta G_{f,i}^\circ + RT \ln \left( \frac{y_i P}{P^\circ} \right) \right] \quad (\text{ideal gas}) \quad (\text{C.4})$$

The dependence of  $G$  on temperature—other than through the  $T$  that appears in Equation (C.4)—

appears in the  $\Delta G_{f,i}^\circ$  term, which is a function of temperature. This adjustment is developed in Appendix C.3 but since it requires a little more space to work out we're just going to assume at this time that each  $\Delta G_{f,i}^\circ$  is available at the temperature of interest so that we can take a look at some examples.

Before we get to those examples, take a moment to review this section and write down the chemical reaction we assumed when we first started. Go ahead, we'll wait for you.

Didn't find it? We didn't think so.<sup>1</sup> This lack of a reaction is the key point to keep in mind: *no reactions are defined for the Gibbs Reactor!* The idea is that if you send an assortment of molecules into an exceptionally large reactor—one large enough to allow the system to approach equilibrium—and specify which molecules are allowed to form, then equilibrium will ensure that those molecules are made in such a way as to minimize Equation (C.4) *by whatever reactions are necessary*.

The only piece of missing information is conservation of mass, for even the Gibbs Reactor must obey conservation of mass. Since we haven't specified any reactions we need a form of conservation of mass that doesn't need to know what those reactions are, and that form is the *atomic species balance*. If you didn't learn the atomic species balance in your material and energy balance course, don't despair: you probably already know it from basic chemistry.

The atomic species balance arises because atoms cannot be destroyed or created<sup>2</sup> and therefore the number of atoms (or moles) of each element *entering* the system must be the same as the number of atoms (or moles) of each element *exiting* the system. This is entirely analogous to the process of balancing chemical reactions that you've been doing since basic chemistry, except in that case you were balancing the number of atoms on either side of the reaction whereas here you'll be balancing the number of atoms on either side of the reactor. Just as when balancing reactions, you get one atomic species balance per element (e.g., one for hydrogen as H, one for oxygen as O, one for carbon as C, etc).

How then to combine minimization of  $G$  via Equation (C.4) and the atomic species balances? For small, gas-phase systems you can apply the following method:

1. Write down the atomic species balances.
2. Solve the atomic species balances for a large number of compositions.
3. For each composition, calculate  $G$  by Equation (C.4).
4. Find the minimum of all  $G$ . The corresponding composition is the equilibrium composition.

<sup>1</sup>This is what passes for a practical joke in a textbook.

<sup>2</sup>At least not in the absence of nuclear reactions, and we don't consider those in this textbook.

### Example C.1: Plotting the Gibbs free energy

Suppose  $1.5 \text{ mol} \cdot \text{s}^{-1}$  ethylene ( $\text{C}_2\text{H}_4$ ) enter a Gibbs Reactor at 1000 K and 4 bar. Assuming only ethylene, acetylene ( $\text{C}_2\text{H}_2$ ), and hydrogen ( $\text{H}_2$ ) can form, create a plot of  $G$  for many values of the outlet molar flow rate of ethylene and use this plot to determine the outlet composition. The Gibbs energies of formation of each compound at 1000 K are provided in the table below.

	$\text{C}_2\text{H}_4(\text{g})$	$\text{C}_2\text{H}_2(\text{g})$	$\text{H}_2(\text{g})$
$\Delta G_{f,i}^\circ \text{ (kJ} \cdot \text{mol}^{-1}\text{)}$	118.6	169.9	0

### Solution

Since this is a Gibbs Reactor the outlet composition will be the equilibrium composition predicted by Equation (C.4). There are two elements, C and H, so there are two atomic species balances, both of which take the form of “in = out.” There is only ethylene in the feed and we have to assume an unknown, non-zero amount of *each species* in the outlet. The atomic species balances are

$$\begin{aligned} \text{C:} \quad & 2(1.5) = 2n_{\text{C}_2\text{H}_4} + 2n_{\text{C}_2\text{H}_2} + 0n_{\text{H}_2} \\ \text{H:} \quad & 4(1.5) = 4n_{\text{C}_2\text{H}_4} + 2n_{\text{C}_2\text{H}_2} + 2n_{\text{H}_2} \end{aligned}$$

where  $n_{\text{C}_2\text{H}_4}$ ,  $n_{\text{C}_2\text{H}_2}$ , and  $n_{\text{H}_2}$  are the unknown outlet molar flow rates of  $\text{C}_2\text{H}_4$ ,  $\text{C}_2\text{H}_2$ , and  $\text{H}_2$ . If we guess a value of any one of these three unknowns then we’ll be able to use the two atomic species balances to solve for the other two, and with all molar flow rates known then we’ll be able to calculate  $G$  using Equation (C.4).

The problem statement asked us to create a plot of  $G$  as a function of  $n_{\text{C}_2\text{H}_4}$ , so our approach will be to

1. guess a value of  $n_{\text{C}_2\text{H}_4}$ ,
2. calculate  $n_{\text{C}_2\text{H}_2}$  using the carbon balance and  $n_{\text{H}_2}$  using the hydrogen balance,
3. calculate  $G$  using Equation (C.4), and
4. repeat Steps 1-3 until a minimum in  $G$  can be identified.

To illustrate the calculation, suppose we guess the outlet molar flow rate of  $\text{C}_2\text{H}_4$  to be  $n_{\text{C}_2\text{H}_4} = 0.25 \text{ mol} \cdot \text{s}^{-1}$  (we could guess anything between 0 and  $1.5 \text{ mol} \cdot \text{s}^{-1}$ ). According to the atomic species balance on C we have

$$2(1.5) = 2(0.25) + 2n_{\text{C}_2\text{H}_2} \quad \Rightarrow \quad n_{\text{C}_2\text{H}_2} = 1.25 \text{ mol} \cdot \text{s}^{-1}$$

and from the atomic species balance on H we have

$$4(1.5) = 2(0.25) + 2(1.25) + 2n_{\text{H}_2} \quad \Rightarrow \quad n_{\text{H}_2} = 1.25 \text{ mol} \cdot \text{s}^{-1}.$$

The total number of moles is  $n = 2.75 \text{ mol} \cdot \text{s}^{-1}$  and the mole fractions  $y_{\text{C}_2\text{H}_4}$ ,  $y_{\text{C}_2\text{H}_2}$ , and  $y_{\text{H}_2}$  are 0.0909, 0.4545, and 0.4545. Substituting values into Equation (C.4) yields

$$\begin{aligned} G &= \sum_{i=A}^C n_i \left[ \Delta G_{f,i}^\circ + RT \ln \left( \frac{y_i P}{P^\circ} \right) \right] \\ &= 0.25 \left[ 14.27 + RT \ln \left( \frac{(0.0909)(4)}{1} \right) \right] \\ &\quad + 1.25 \left[ 20.43 + RT \ln \left( \frac{(0.454)(4)}{1} \right) \right] \\ &\quad + 1.25 \left[ 0 + RT \ln \left( \frac{(0.4545)(4)}{1} \right) \right] \\ &= 252.3 \text{ kJ} \cdot \text{mol}^{-1} \end{aligned}$$

We now repeat this calculation until we have enough  $G$  values to make a plot and identify the minimum:

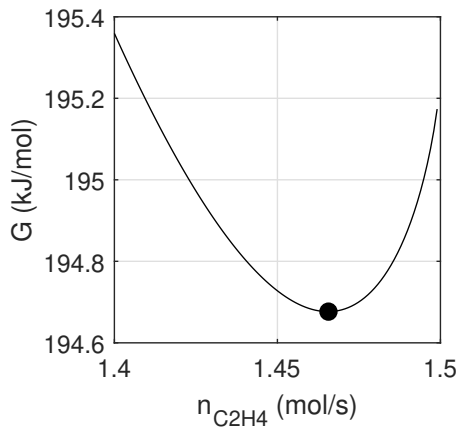


Figure C.1: Gibbs energy  $G$  as a function of  $n_{\text{C}_2\text{H}_4}$  for the ethylene–acetylene–hydrogen system in Example C.1. The minimum Gibbs energy (●) is  $194.678 \text{ kJ} \cdot \text{mol}^{-1}$ .

At the minimum of  $G$  (corresponding to equilibrium) the outlet molar flow rates are  $n_{\text{C}_2\text{H}_4} = 1.466 \text{ mol} \cdot \text{s}^{-1}$ , and  $n_{\text{C}_2\text{H}_2} = n_{\text{H}_2} = 0.034 \text{ mol} \cdot \text{s}^{-1}$ .

Example C.1 highlights the general principle of the Gibbs Reactor: calculate the Gibbs energy at several compositions, and whichever composition results in the lowest Gibbs energy is the equilibrium composition. The approach can also be extended to slightly more complex systems as we'll see in the next example, but further generalization will require us to change the way we make our guesses.

### Example C.2: A simple Gibbs Reactor

Repeat Example C.1 but also allow methane ( $\text{CH}_4$ ) to form, for which  $\Delta G_{f,i}^\circ$  is  $20.5 \text{ kJ} \cdot \text{mol}^{-1}$  at  $1000 \text{ K}$ .

### Solution

With the addition of  $\text{CH}_4$  we have to modify the atomic species balances to

$$\text{C:} \quad 2(1.5) = 2n_{\text{C}_2\text{H}_4} + 2n_{\text{C}_2\text{H}_2} + 0n_{\text{H}_2} + 1n_{\text{CH}_4}$$

$$\text{H:} \quad 4(1.5) = 4n_{\text{C}_2\text{H}_4} + 2n_{\text{C}_2\text{H}_2} + 2n_{\text{H}_2} + 4n_{\text{CH}_4}$$

and now we see a potential problem: we have increased the number of unknowns by one but we still have only two atomic species balances. This means that we'll have to guess a range of values for two species instead of just one species as we did in Example C.1, but otherwise the approach to calculate  $G$  using Equation (C.4) will be unchanged. We can then produce a surface plot of  $G$  on the  $z$ -axis as a function of our two guesses, say  $n_{\text{C}_2\text{H}_4}$  on the  $x$ -axis and  $n_{\text{C}_2\text{H}_2}$  on the  $y$ -axis.

As a sample calculation, suppose we guess  $n_{\text{C}_2\text{H}_4}$  to be  $0.25 \text{ mol} \cdot \text{s}^{-1}$  and  $n_{\text{C}_2\text{H}_2}$  to be  $1.0 \text{ mol} \cdot \text{s}^{-1}$ . From the carbon balance we find  $n_{\text{CH}_4}$  to be  $0.5 \text{ mol} \cdot \text{s}^{-1}$  and from the hydrogen balance we find  $n_{\text{H}_2}$  to be  $0.5 \text{ mol} \cdot \text{s}^{-1}$ . Using the same approach as in Example C.1 we find  $G = 211.86 \text{ kJ} \cdot \text{mol}^{-1}$ . Repeating this calculation for many guesses of  $n_{\text{C}_2\text{H}_4}$  and  $n_{\text{C}_2\text{H}_2}$  produces the plot shown below.



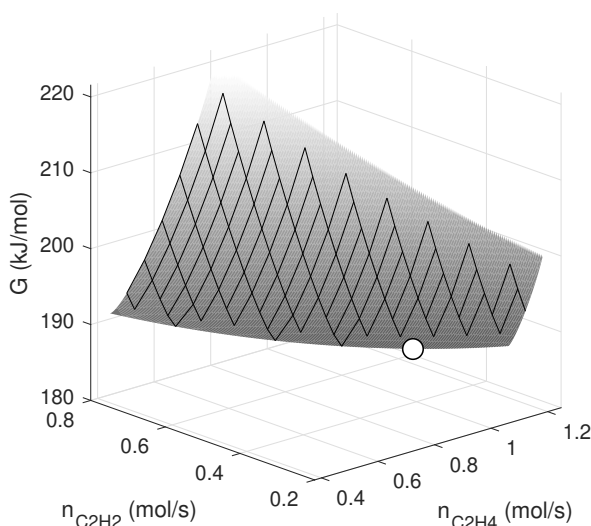


Figure C.2: Gibbs energy  $G$  as a function of  $n_{\text{C}_2\text{H}_4}$  and  $n_{\text{C}_2\text{H}_2}$  for Example C.2. The minimum Gibbs energy (o) is  $188.061 \text{ kJ} \cdot \text{mol}^{-1}$ .

At the minimum of  $G$  (corresponding to equilibrium) the outlet molar flow rates and mole fractions are

	$\text{C}_2\text{H}_4$	$\text{C}_2\text{H}_2$	$\text{H}_2$	$\text{CH}_4$
$n_i \text{ (mol} \cdot \text{s}^{-1}\text{)}$	0.947	0.369	0.002	0.368
$y_i$	0.562	0.219	0.001	0.218

Notice that there are certain combinations of  $n_{\text{C}_2\text{H}_4}$  and  $n_{\text{C}_2\text{H}_2}$  which do not have corresponding  $G$  values; this is because we require all molar flow rates to be positive.

Based on Example C.2 you can probably imagine what happens as more and more species are added: the number of unknowns generally exceeds the number of atomic species balances, so more and more values must be guessed. Conceptually this process is no different from the illustrated here, but the actual process of finding the minimum Gibbs energy for any general system is performed more conveniently by using a minimization algorithm.

## C.2 A general Gibbs Reactor uses the fmincon function to minimize $G$

Although the “blind” guessing we used in Example C.1 and Example C.2 is perfectly suitable to small problems, it scales poorly once the degrees of freedom—the number of outlet molar

flow rates minus the number of atomic species balances—grows much beyond two. Instead of blind guessing we’re going to use MATLAB’s `fmincon` function, which is designed to find a function’s minimum subject to certain constraints. In an attempt to minimize  $G$  this function will efficiently guess lots of values of all  $n_i$ , with all guesses always satisfying conservation of mass.

Let’s take a look at the syntax of `fmincon` according to MATLAB’s documentation:

```
x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub)
```

A few of those inputs are not needed for our purposes and we can change some of the variable names to make it look a little more like what we need:

```
n = fmincon(@gibbs, nGuess, [], [], Aeq, beq, lb, [])
```

The way to interpret this function is as follows: MATLAB will use `fmincon` to find the minimum of the function `@gibbs` by guessing values of  $n$ . The `fmincon` function should start guessing values of  $n$  at `nGuess`, the guessed values should never be lower than the lower bounds specified in `lb`, and guesses should always satisfy the matrix equality  $Aeq*n = beq$ .

The `gibbs` function should be a function that accepts a column vector of  $n$  values and returns a scalar value for  $G$  according to Equation (C.4). The general setup looks something like this:

```
1 function G = gibbs(n)
2 % Function to calculate G for ideal gases
3 % Define deltaG_f, R, T, P, and Pcirc for problem of interest
4 y = n/sum(n); % mole fractions
5 G = sum(n.*(deltaGf + R*T*log(y*P/Pcirc)));
6 end
```

When writing the `gibbs` function be careful to define  $\Delta G_{f,i}^\circ$  as `deltaG_f` for each component *at the temperature  $T$*  (as  $T$ ) of the reactor, not at the tabulated value of 298.15 K (unless of course your reactor happens to operate at 298.15 K).

The `nGuess` column vector contains your initial guesses for the outlet molar flow rates (or final moles if you’re working with a batch reactor) of all species. A fair initial guess is the inlet molar flow rates but be aware that your final answer can be dependent on this initial guess, so you should try several different initial guesses to assure yourself that the solution is stable. If your answer is sensitive to the initial guess then (i) check your code to make sure you haven’t made a mistake and (ii) take the one with the lowest  $G$  value.

The `lb` column vector is an easy one: it contains lower bounds for each  $n_i$  value. Since the moles of any component cannot be negative, `lb` is simply a column vector of zeros.

The equality  $Aeq*n=beq$  is how we’ll maintain conservation of mass by means of an atom balance. In this context, `beq` is a column vector of the total number of atoms of each element

present in the system (one for each atom), and  $A_{eq}$  is a matrix containing the number of atoms of each element per molecule. The matrix  $A_{eq}$  has one row per atom and one column per molecule.

### Example C.3: Writing $A_{eq}$ and $b_{eq}$

Suppose  $100 \text{ mol} \cdot \text{s}^{-1}$  of an ideal gas mixture containing 5 mol% cyclohexane ( $\text{C}_6\text{H}_{12}$ ), 14 mol% benzene ( $\text{C}_6\text{H}_6$ ), 57 mol% methane ( $\text{CH}_4$ ), and 24 mol% hydrogen ( $\text{H}_2$ ) is fed to a Gibbs Reactor operating at 750 K and 15 bar. In addition to the components in the feed, toluene ( $\text{C}_7\text{H}_8$ ), propane ( $\text{C}_3\text{H}_8$ ), and ethane ( $\text{C}_2\text{H}_6$ ) are also expected to form. Determine  $A_{eq}$  and  $b_{eq}$ .

### Solution

There are two elements to balance, C and H. The moles of atomic carbon (C) entering the reactor per second is

$$6(5) + 6(14) + 1(57) + 0(24) = 171$$

and the moles of atomic hydrogen (H) entering the reactor per second is

$$12(5) + 6(14) + 4(57) + 2(24) = 420$$

therefore  $b_{eq} = [171; 420]$ . The number of elements per molecule are as follows:

	$\text{C}_7\text{H}_8$	$\text{C}_6\text{H}_{12}$	$\text{C}_6\text{H}_6$	$\text{C}_3\text{H}_8$	$\text{C}_2\text{H}_6$	$\text{CH}_4$	$\text{H}_2$
C:	7	6	6	3	2	1	0
H:	8	12	6	8	6	4	2

The matrix  $A_{eq}$  is essentially this table:

```
1 Aeq = [7 6 6 3 2 1 0; % C
2       8 12 6 8 6 4 2]; % H
```

If you happen to have some experience with linear algebra or have reviewed Appendix B then you can probably show for yourself exactly why the matrix equality  $A_{eq} \cdot n = b_{eq}$  implies conservation of mass, but if not then simply use  $A_{eq}$  and  $b_{eq}$  *exactly* as shown in this example and your code should function just fine.

We now have all necessary parts to form a fully functional Gibbs Reactor, one that lets MATLAB do all the guessing of  $n_i$  values through the `fmincon` function.

**Example C.4: Using a full Gibbs Reactor**

Find the equilibrium composition for the system described in Example C.3. The Gibbs energies of formation for each species have been calculated at 750 K using the shortcut van't Hoff equation and are provided in the table below.

Table C.1:  $\Delta G_{f,i}^\circ$  (kJ · mol<sup>-1</sup>) at 750 K by the shortcut van't Hoff equation.

C <sub>7</sub> H <sub>8</sub>	C <sub>6</sub> H <sub>12</sub>	C <sub>6</sub> H <sub>6</sub>	C <sub>3</sub> H <sub>8</sub>	C <sub>2</sub> H <sub>6</sub>	CH <sub>4</sub>	H <sub>2</sub>
231.6	266.8	200.8	97.5	46.9	-13.4	0

**Solution**

```
1 Aeq = [7 6 6 3 2 1 0; % C
2       8 12 6 8 6 4 2]; % H
3 beq = [171; 420]; % [C; H]
4 nGuess = [0 5 14 0 0 57 24];
5 lb = [0; 0; 0; 0; 0; 0; 0];
6 n = fmincon(@gibbs, nGuess, [], [], Aeq, beq, lb, []);
7
8 function G = gibbs(n)
9 % Function to calculate G for ideal gases
10 dGf = [231.6 266.8 200.8 97.5 46.9 -13.4 0]; % kJ/mol, 750 K
11 R = 8.314/1000; % kJ/mol-K
12 T= 750; % K
13 Pcirc = 1; % bar
14 P = 15; % bar
15
16 y = n/sum(n); % mole fractions
17 G = sum(n.*(dGf + R*T*log(y*P/Pcirc)));
18 end
```

Running this script produces the outlet molar flow rates shown in the table below. Also included are the outlet mole fractions, calculated after the script ran as  $y = n/\text{sum}(n)$ .

Table C.2: Outlet molar flow rates (mol · s<sup>-1</sup>) and mole fractions for Example C.4.

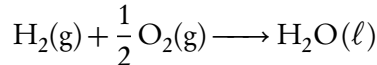
	C <sub>7</sub> H <sub>8</sub>	C <sub>6</sub> H <sub>12</sub>	C <sub>6</sub> H <sub>6</sub>	C <sub>3</sub> H <sub>8</sub>	C <sub>2</sub> H <sub>6</sub>	CH <sub>4</sub>	H <sub>2</sub>
$n_i =$	3.381	0.000	10.91	0.003	0.223	81.42	0.219
$y_i =$	0.035	0.000	0.113	0.000	0.002	0.847	0.002

You can also use the molar masses of each species to verify that the total inlet *mass* flow rate is  $1446 \text{ g} \cdot \text{s}^{-1}$ , as is the total outlet mass flow rate.

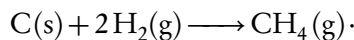
### C.3 The full van't Hoff expression can adjust $\Delta G_{f,i}^\circ$ for temperature

In each of the Gibbs Reactor examples  $\Delta G_{f,i}^\circ$  has been provided at the temperature of interest, but of course this usually isn't the case and you'll need to be able to calculate  $\Delta G_{f,i}^\circ$  on your own. Conceptually this process is straightforward:

1. Write the *formation reaction* for one mole of the desired species. The formation reaction is the reaction to form the species from its elemental building blocks in their naturally occurring state. For example, the formation reaction of liquid water is



and the formation reaction of gaseous methane is



The aggregation state of all species must be specified.

2. Use the *full* van't Hoff equation,

$$K_a = K_0 K_1 K_2 \quad (2.25)$$

to calculate  $K_a$  at the temperature of interest. This calculation must include  $K_2$ , something we've omitted up until now.

3. Use the calculated value of  $K_a$  and the definition of the equilibrium constant,

$$K_a \equiv \exp \left[ \frac{-\Delta G}{RT} \right], \quad (2.24)$$

to calculate  $\Delta G$  at the temperature of interest. This  $\Delta G$  is  $\Delta G_{f,i}^\circ$  for the desired species.

Writing the formation reaction is a little new but not terribly complicated, nor is using Equation (2.24) to calculate  $\Delta G$ . The part that's new and challenging is the evaluation of  $K_2$  in Equation (2.25): here, unlike everywhere else in this book, our usual assumption of constant heat capacity isn't good enough and we have to take into account the temperature dependence of all  $C_p$ .

If  $C_p$  is provided to you as a polynomial such as  $C_p = a + bT + cT^2 + dT^3$  then there are several ways to work out the integrals of Equation (2.28) to get  $K_2(T)$  in analytic form,<sup>3</sup> and those are the forms you should use if you ever lack a platform like MATLAB to perform numerical integration, or if speed is of interest.

More recently the trend in reporting heat capacities of substances in the ideal gas state has been to represent them in a hyperbolic form that more closely resembles their origins in quantum and statistical mechanics, namely

$$C_p^\circ = c_1 + c_2 \left[ \frac{c_3/T}{\sinh(c_3/T)} \right]^2 + c_4 \left[ \frac{c_5/T}{\cosh(c_5/T)} \right]^2, \quad (\text{C.5})$$

where the  $^\circ$  symbol implies a species in its ideal gas state, and  $c_1$  through  $c_5$  are constants. The hyperbolic form of Equation (C.5) doesn't easily lend itself to manual evaluation of the integrals required for  $K_2$  in Equation (2.28) but if you've got a computer with MATLAB around then it's just as easy to use Equation (C.5) as it is the polynomial form for the heat capacities.

The reason for this ease-of-use is because we can ask MATLAB to numerically approximate the integrals in Equation (2.28) and the accuracy of this approximation is quite high—not unlike the accuracy of the numerical ODE solution used by ode45. Let's see how to use MATLAB to adjust the Gibbs energy of formation for methane to any desired temperature.

#### Example C.5: $\Delta G_{f,i}^\circ$ for methane at 600 K

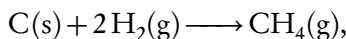
Estimate the Gibbs energy of formation for methane ( $\text{CH}_4$ ) in its ideal gas state at 600 K.

#### Solution

According to the steps above, the first thing we need is the formation reaction for

<sup>3</sup>See, for example, Equation (13.24) in J.M. Smith; H.C. Van Ness; M.M. Abbot. Introduction to Chemical Engineering Thermodynamics, 6th ed. McGraw-Hill, MA (2001).

methane from the elements, which is



where carbon is solid and hydrogen is gaseous because those are the way these elements are found in their naturally occurring states. The choice to make methane in its gaseous state was because the problem statement asked for the Gibbs formation energy of methane in its ideal gas state.

Next, we need to calculate  $K_a$  using the full van't Hoff equation. From Appendix D and the usual routines from Chapter 2 we can calculate  $K_0 = 6.9016 \times 10^8$  and  $K_1 = 2.5048 \times 10^{-7}$ . For  $K_2$  we need to find an expression for the heat capacities for each of the components are written in the formation reaction above. From Elliott and Lira<sup>a</sup> the heat capacity of solid carbon is given by the polynomial

$$C_{P,C}(\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}) = c_1 + c_2 T + c_3 T^2 + c_4 T^3,$$

and from Perry's Chemical Engineers' Handbook<sup>b</sup> the heat capacities of hydrogen and methane are given by the hyperbolic form in Equation (C.5), which provides  $C_p^\circ$  with units of  $\text{J} \cdot \text{kmol}^{-1} \cdot \text{K}^{-1}$  (note the kmol instead of mol). The coefficients for each species' heat capacity equations are as follows:

	C(s)	H <sub>2</sub> (g)	CH <sub>4</sub> (g)
$c_1$	-3.958	$0.27617 \times 10^5$	$0.33298 \times 10^5$
$c_2$	0.05586	$0.0956 \times 10^5$	$0.79933 \times 10^5$
$c_3$	$-4.55 \times 10^{-5}$	$2.466 \times 10^3$	$2.0869 \times 10^3$
$c_4$	$1.517 \times 10^{-8}$	$0.0376 \times 10^5$	$0.41602 \times 10^5$
$c_5$	-	567.6	991.96

Clearly Equation (2.28) would be a nightmare to evaluate analytically because we have two different kinds of heat capacity expressions, one polynomial and two hyperbolics. Numerical integration, however, has no qualms at all with such differences.

To calculate the integrals of Equation (2.28) with MATLAB requires two steps:

1. Create functions for each of the heat capacities such that provided any  $T$  the corresponding heat capacity is returned with units of  $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ .
2. Use the integral function in MATLAB to evaluate the integrals of Equation (2.28) over the desired range, in this case from 298.15 K to 600 K.

The fastest way to create  $C_p$  functions is to use anonymous functions, which are slightly different from local functions but are straightforward enough that seeing them written as they are below should be enough for you to pick up on the syntax. An anonymous function for the heat capacity of solid carbon is

```
1 c = [-3.958 0.05586 -4.55e-5 1.517E-08];
2 Cs = @(T) c(1) + c(2).*T + c(3).*T.^2 + c(4).*T.^3;
```

for hydrogen is

```
3 c = [0.27617*1e5 0.0956e5 2.466e3 0.0376e5 567.6 250 0.2843];
4 H2 = @(T) (c(1)+c(2)*(c(3)./T./sinh(c(3)./T)).^2 ...
5           + c(4)*(c(5)./T./cosh(c(5)./T)).^2)/1000;
```

and for methane is

```
6 c = [0.33298e5 0.79933e5 2.0869e3 0.41602e5 991.96 50 0.3330];
7 CH4 = @(T) (c(1)+c(2)*(c(3)./T./sinh(c(3)./T)).^2 ...
8           + c(4)*(c(5)./T./cosh(c(5)./T)).^2)/1000;
```

Notice the factor of 1000 for  $H_2$  and  $CH_4$ , which is because the hyperbolic form had units of kmol instead of mol. To evaluate the integrals within Equation (2.28) we create anonymous functions within the integral function, and pass as second and third arguments the lower and upper bounds of the integral:

```
9 R = 8.314; % J/mol-K
10 T0 = 298.15; % K
11 T = 600; % K
12 K2 = exp(-1/T*integral(@(T)(CH4(T)-2*H2(T)-Cs(T))/R, T0, T)
13       +integral(@(T)(CH4(T)-2*H2(T)-Cs(T))./(R*T), ...
14       T0, T));
```

Running this script estimates  $K_2$  as  $K_2 = 0.4963$ , and therefore

$$K_a = K_0 K_1 K_2 = (6.9016 \times 10^8)(2.5048 \times 10^{-7})(0.4963) = 85.8016.$$

Finally, Equation (2.24) is used to calculate  $\Delta G_{f,i}^\circ$  for methane at 600 K as

$$\Delta G_{f,i}^\circ(600\text{ K}) = -RT \log K_a = -22.208 \text{ kJ} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$$

As you can see, we're letting MATLAB do all the heavy lifting of evaluating  $\int \Delta C_p$  and  $\int \Delta C_p/T$ , but we still have to make an effort to set up the problem in the first place. That's also one of the reasons why, throughout this text, we've only ever used the *shortcut* van't Hoff Equation as opposed to the full van't Hoff equation.



<sup>a</sup>J.R. Elliott; C.T. Lira. Introductory Chemical Engineering Thermodynamics, 2nd ed. Prentice Hall: NJ, 2012.

<sup>b</sup>B.E. Poling *et. al.* Physical and Chemical Data. In *Chemical Engineers' Handbook*, R. Perry; C. Chilton, eds. 8th ed, McGraw-Hill: New York, 2008.

## C.4 The Gibbs energy at 298.15 K can be estimated from $\Delta H_f^\circ$ and $S_f^\circ$

Although the procedure is not needed in this text, should you ever need to calculate the standard Gibbs energy of formation for a species at 298.15 K you can do so if you've got access to the standard enthalpy of formation and standard entropy of that compound (two values which are often available, for example through the NIST Chemistry Webbook). The relationship is

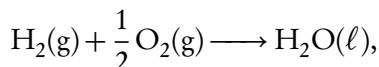
$$\Delta G_f^\circ = \Delta H_f^\circ - T \Delta S_f^\circ,$$

where the standard entropy of formation  $\Delta S_f^\circ$  is

$$\Delta S_f^\circ = \sum_{i=A}^Z \nu_i S_i^\circ,$$

and  $S_i^\circ$  is the standard entropy. The standard entropy is tabulated for many compounds and is an absolute quantity, so don't expect it to be zero. The sum and  $\nu_i$  values are for the formation reaction to make 1 mol of whatever compound you're interested in.

For example, let's confirm the standard Gibbs energy of formation given in Table D.3 for liquid water. The formation reaction to make one mole of liquid water is



and if you look up the standard entropies for each of these compounds you'll find  $S_{\text{H}_2(\text{g})} = 130.6 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ ,  $S_{\text{O}_2(\text{g})} = 205.0 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ , and  $S_{\text{H}_2\text{O}(\ell)} = 69.9 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ . The standard entropy of formation for liquid water is therefore

$$\Delta S_f^\circ = (1)(69.9) + (-1)(130.6) + (-1/2)(205.0) = -163.2 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}.$$

According to Table D.3 the standard enthalpy of formation of liquid water is  $-285.8 \text{ kJ} \cdot \text{mol}^{-1}$ ,

and therefore the calculated value for the standard Gibbs energy of formation is

$$\Delta G_f^\circ = -285.8 \text{ kJ} \cdot \text{mol}^{-1} - (298.15 \text{ K})(-0.1632 \text{ kJ} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}) = -237.14 \text{ kJ} \cdot \text{mol}^{-1},$$

which compares favorably to the value provided in Table D.3 of  $-237.1 \text{ kJ} \cdot \text{mol}^{-1}$  (in fact they're identical but for rounding).

Again, this method is primarily useful when the Gibbs energy of formation is not available from any of the usual sources such as the ones provided in Appendix D.

# Appendix D

## Physical Properties

This appendix contains the formulae, molar masses, heat capacities, and energies of formation ( $\Delta H_{f,i}^\circ$  and  $\Delta G_{f,i}^\circ$ ) for a variety of compounds in the gas phase, liquid phase, and as an aqueous solution. All data were rounded to the precision shown for uniformity, not as a representation of any underlying uncertainties.

All data for gas-phase compounds were calculated using Tables 2-156 and 2-179 of Perry's Chemical Engineers' Handbook, 8th edition (2008). The data sets therein were developed from R.L. Rowley, W.V. Wilding, J.L. Oscarson, Y. Yang, N.A. Zundel, T.E. Daubert, R.P. Danner, DIPPR Data Compilation of Pure Chemical Properties, Design Institute for Physical Properties, AIChE, New York (2007).

The data for liquid and aqueous compounds were assembled from a variety of sources as indicated by the following reference numbers in the table at the corresponding entry:

1. J.M. Smith, H.C. Van Ness, M.M. Abbot. "Introduction to Chemical Engineering Thermodynamics" 6th ed. McGraw-Hill, New York, NY (2001).
2. E.S. Domalski, E.D. Hearing. "Condensed phase heat capacity data" in NIST Chemistry WebBook, NIST Standard Reference Database Number 69, Eds. P.J. Linstrom, W.G. Mallard. NIST, Gaithersburg, MD.
3. S.I. Sandler. "Chemical, Biochemical, and Engineering Thermodynamics," 4th ed. Wiley, Hoboken, NJ (2006).
4. "Standard thermodynamic properties of chemical substances," in CRC Handbook of Chemistry and Physics, Internet Version 2005, D.R. Lide, ed., CRC Press, Boca Raton, FL (2005)

Approximate values of the gas constant  $R$  in commonly encountered units. Values taken from the “Gas constant” page in Wikipedia.<sup>1</sup>

Table D.1: Values of the gas constant  $R$ .

Numerator	Value
energy	$8.314 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
	$1.987 \text{ cal} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
	$1.987 \times 10^{-3} \text{ kcal} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
pressure $\times$ volume	$8.314 \text{ m}^3 \cdot \text{Pa} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
	$8.314 \text{ L} \cdot \text{kPa} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
	$8.314 \times 10^{-2} \text{ L} \cdot \text{bar} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$
	$0.082 \text{ L} \cdot \text{atm} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$

---

<sup>1</sup>Enough of this “Wikipedia isn’t a good source” garbage; it’s fine. Accessed 14 Aug 2019, [wikipedia.org/wiki/Gas\\_constant](https://wikipedia.org/wiki/Gas_constant).

Table D.2: Physical properties of gas-phase compounds. Enthalpies and Gibbs energies of formation are given at 298.15 K, have units of  $\text{kJ} \cdot \text{mol}^{-1}$  and their reference states are the ideal gas state as 101.325 kPa and 298.15 K. Heat capacity is given at several temperatures and has units of  $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ .

Name	Formula	MM	$\Delta H_f^\circ$	$\Delta G_f^\circ$	$C_p^\circ$			
					300 K	500 K	700 K	900 K
acetaldehyde	$\text{C}_2\text{H}_4\text{O}$	44	-166.4	-133.3	55	77	94	107
acetamide	$\text{C}_2\text{H}_5\text{NO}$	59	-238.3	-159.6	63	94	117	131
acetic acid	$\text{C}_2\text{H}_4\text{O}_2$	60	-461.1	-403.0	64	94	117	133
acetic anhydride	$\text{C}_4\text{H}_6\text{O}_3$	102	-572.5	-473.4	100	154	191	216
acetone	$\text{C}_3\text{H}_6\text{O}$	58	-215.7	-151.3	75	108	134	154
acetonitrile	$\text{C}_2\text{H}_3\text{N}$	41	74.0	91.9	52	70	83	94
acetylene	$\text{C}_2\text{H}_2$	26	228.2	210.7	44	55	61	66
acrylic acid	$\text{C}_3\text{H}_4\text{O}_2$	72	-355.9	-306.0	78	111	134	149
acrylonitrile	$\text{C}_3\text{H}_3\text{N}$	53	183.7	193.7	64	88	104	116
ammonia	$\text{H}_3\text{N}$	17	-45.9	-16.4	36	42	48	54
anisole	$\text{C}_7\text{H}_8\text{O}$	108	-67.9	22.7	113	184	234	269
benzene	$\text{C}_6\text{H}_6$	78	82.9	129.6	83	139	176	202
benzoic acid	$\text{C}_7\text{H}_6\text{O}_2$	122	-294.1	-214.2	104	171	218	249
benzonitrile	$\text{C}_7\text{H}_5\text{N}$	103	215.7	257.8	110	167	204	229
benzyl alcohol	$\text{C}_9\text{H}_8\text{O}$	108	-90.3	-2.5	113	185	234	268
bromine	$\text{Br}_2$	160	30.9	3.1	36	37	37	38
bromomethane	$\text{CH}_3\text{Br}$	95	-37.7	-28.2	43	57	68	76
1,2-butadiene	$\text{C}_4\text{H}_6$	54	162.3	198.6	80	114	139	159
1,3-butadiene	$\text{C}_4\text{H}_6$	54	109.2	149.7	80	122	149	167
butane	$\text{C}_4\text{H}_{10}$	58	-125.8	-16.7	99	149	187	216
1-butanol	$\text{C}_4\text{H}_{10}\text{O}$	74	-275.1	-150.7	108	165	204	234
2-butanol	$\text{C}_4\text{H}_{10}\text{O}$	74	-292.9	-167.0	113	167	206	235
1-butene	$\text{C}_4\text{H}_8$	56	-0.5	70.4	86	129	162	187
cis-2-butene	$\text{C}_4\text{H}_8$	56	-7.4	65.4	80	124	157	183
trans-2-butene	$\text{C}_4\text{H}_8$	56	-11.0	63.2	88	129	160	185
1-butyne	$\text{C}_4\text{H}_6$	54	165.2	202.3	82	116	140	159
butyraldehyde	$\text{C}_4\text{H}_8\text{O}$	72	-207.0	-116.3	104	148	186	215
butyric acid	$\text{C}_4\text{H}_8\text{O}_2$	88	-475.8	-360.0	116	165	201	226
carbon dioxide	$\text{CO}_2$	44	-393.5	-394.4	37	45	49	53
carbon monoxide	$\text{CO}$	28	-110.5	-137.2	29	30	31	33
carbon tetrachloride	$\text{CCl}_4$	154	-95.8	-53.5	84	97	102	104
carbon tetrafluoride	$\text{CF}_4$	88	-922.1	-877.6	61	85	94	99
chlorine	$\text{Cl}_2$	71	0.0	0.0	34	36	37	37
chlorobenzene	$\text{C}_6\text{H}_5\text{Cl}$	113	51.1	98.3	98	152	188	211
chloroethane	$\text{C}_2\text{H}_5\text{Cl}$	65	-112.3	-60.5	63	91	111	126
chloroform	$\text{CHCl}_3$	119	-102.9	-70.1	66	80	89	94
chloromethane	$\text{CH}_3\text{Cl}$	50	-82.0	-58.4	41	55	67	75
1-chloropropane	$\text{C}_3\text{H}_7\text{Cl}$	79	-133.2	-52.6	85	125	153	175
2-chloropropane	$\text{C}_3\text{H}_7\text{Cl}$	79	-144.8	-61.4	88	127	155	176
m-cresol	$\text{C}_7\text{H}_8\text{O}$	108	-132.3	-40.2	125	196	240	271
o-cresol	$\text{C}_7\text{H}_8\text{O}$	108	-128.6	-35.4	128	195	240	272

Table D.2: Physical properties of gas-phase compounds (*continued*).

Name	Formula	MM	$\Delta H_f^\circ$	$\Delta G_f^\circ$	$C_p^\circ$			
					300 K	500 K	700 K	900 K
p-cresol	C <sub>7</sub> H <sub>8</sub> O	108	-125.4	-31.7	126	194	239	271
cumene	C <sub>9</sub> H <sub>12</sub>	120	4.0	137.9	150	241	304	347
cyanogen	C <sub>2</sub> N <sub>2</sub>	52	309.1	297.6	57	65	71	75
cyclobutane	C <sub>4</sub> H <sub>8</sub>	56	28.5	112.2	71	122	160	187
cyclohexane	C <sub>6</sub> H <sub>12</sub>	84	-123.3	31.9	108	187	252	299
cyclohexanol	C <sub>6</sub> H <sub>12</sub> O	100	-286.2	-109.5	128	213	277	323
cyclohexene	C <sub>6</sub> H <sub>10</sub>	82	-4.6	107.7	102	173	224	261
cyclopentane	C <sub>5</sub> H <sub>10</sub>	70	-77.0	38.9	84	150	199	235
cyclopentene	C <sub>5</sub> H <sub>8</sub>	68	32.3	110.5	82	139	180	209
cyclopropane	C <sub>3</sub> H <sub>6</sub>	42	53.3	104.4	56	94	120	140
deuterium	D <sub>2</sub>	4	0.0	0.0	29	29	30	31
m-dichlorobenzene	C <sub>6</sub> H <sub>4</sub> Cl <sub>2</sub>	147	25.7	77.9	114	167	199	220
o-dichlorobenzene	C <sub>6</sub> H <sub>4</sub> Cl <sub>2</sub>	147	30.2	82.9	114	166	198	220
p-dichlorobenzene	C <sub>6</sub> H <sub>4</sub> Cl <sub>2</sub>	147	22.5	76.7	114	166	199	220
1,1-dichloroethane	C <sub>2</sub> H <sub>4</sub> Cl <sub>2</sub>	99	-129.4	-72.6	76	104	122	135
1,2-dichloroethane	C <sub>2</sub> H <sub>4</sub> Cl <sub>2</sub>	99	-129.8	-73.9	78	105	123	136
dichloromethane	CH <sub>2</sub> Cl <sub>2</sub>	85	-95.5	-69.0	51	66	77	84
1,1-difluoroethane	C <sub>2</sub> H <sub>4</sub> F <sub>2</sub>	66	-497.0	-439.5	68	97	117	131
1,2-difluoroethane	C <sub>2</sub> H <sub>4</sub> F <sub>2</sub>	66	-447.7	-391.9	68	95	116	131
difluoromethane	CH <sub>2</sub> F <sub>2</sub>	52	-452.3	-424.7	43	59	71	80
di-isopropyl amine	C <sub>6</sub> H <sub>15</sub> N	101	-143.8	64.2	160	246	307	351
dimethyl acetylene	C <sub>4</sub> H <sub>6</sub>	54	145.7	184.9	78	110	136	156
dimethyl amine	C <sub>2</sub> H <sub>7</sub> N	45	-18.5	68.4	71	106	132	152
dimethyl ether	C <sub>2</sub> H <sub>6</sub> O	46	-184.1	-112.8	66	93	116	134
dimethylsilane	C <sub>2</sub> H <sub>8</sub> Si	60	-94.7	-19.3	93	129	157	178
dimethyl sulfide	C <sub>2</sub> H <sub>6</sub> S	62	-37.2	7.3	74	101	123	140
dipropyl amine	C <sub>6</sub> H <sub>15</sub> N	101	-116.0	86.8	159	243	303	348
ethane	C <sub>2</sub> H <sub>6</sub>	30	-83.8	-31.9	53	78	99	116
ethanol	C <sub>2</sub> H <sub>6</sub> O	46	-235.0	-167.9	65	96	119	136
ethyl acetate	C <sub>4</sub> H <sub>8</sub> O <sub>2</sub>	88	-444.5	-328.0	114	162	200	225
ethyl amine	C <sub>2</sub> H <sub>7</sub> N	45	-47.2	36.2	73	106	131	151
ethylbenzene	C <sub>8</sub> H <sub>10</sub>	106	29.9	130.7	128	207	261	299
ethylcyclohexane	C <sub>8</sub> H <sub>16</sub>	112	-171.5	39.6	160	267	346	402
ethylene	C <sub>2</sub> H <sub>4</sub>	28	52.5	68.4	43	63	78	90
ethylene glycol	C <sub>2</sub> H <sub>6</sub> O <sub>2</sub>	62	-392.2	-301.8	78	110	134	152
ethyleneimine	C <sub>2</sub> H <sub>5</sub> N	43	123.4	178.0	52	85	108	125
ethylene oxide	C <sub>2</sub> H <sub>4</sub> O	44	-52.6	-13.2	48	75	95	110
ethyl formate	C <sub>3</sub> H <sub>6</sub> O <sub>2</sub>	74	-388.3	-303.1	90	126	157	180
fluorine	F <sub>2</sub>	38	0.0	0.0	31	34	36	37
fluorobenzene	C <sub>6</sub> H <sub>5</sub> F	96	-116.6	-69.0	95	151	187	211
fluoroethane	C <sub>2</sub> H <sub>5</sub> F	48	-264.4	-212.3	59	87	108	124
fluoromethane	CH <sub>3</sub> F	34	-234.3	-210.4	38	51	64	73
formaldehyde	CH <sub>2</sub> O	30	-108.6	-102.6	35	44	52	59
formamide	CH <sub>3</sub> NO	45	-192.2	-147.1	46	66	82	93
formic acid	CH <sub>2</sub> O <sub>2</sub>	46	-405.5	-377.8	46	63	76	86

Table D.2: Physical properties of gas-phase compounds (*continued*).

Name	Formula	MM	$\Delta H_f^\circ$	$\Delta G_f^\circ$	$C_p^\circ$			
					300 K	500 K	700 K	900 K
furan	C <sub>4</sub> H <sub>4</sub> O	68	-34.8	0.8	66	108	134	152
heptane	C <sub>7</sub> H <sub>16</sub>	100	-187.7	8.2	165	253	316	363
1-heptene	C <sub>7</sub> H <sub>14</sub>	98	-62.9	94.8	155	234	291	334
hexane	C <sub>6</sub> H <sub>14</sub>	86	-166.9	-0.1	143	218	273	315
1-hexene	C <sub>6</sub> H <sub>12</sub>	84	-41.7	87.0	134	199	248	285
3-hexyne	C <sub>6</sub> H <sub>10</sub>	82	106.0	199.0	119	177	220	253
1-hexyne	C <sub>6</sub> H <sub>10</sub>	82	123.7	218.5	129	187	227	256
2-hexyne	C <sub>6</sub> H <sub>10</sub>	82	105.0	199.0	122	179	222	254
hydrazine	H <sub>4</sub> N <sub>2</sub>	32	95.4	159.2	49	66	78	88
hydrogen	H <sub>2</sub>	2	0.0	0.0	29	29	29	30
hydrogen bromide	HBr	81	-36.3	-53.3	29	29	30	32
hydrogen chloride	HCl	36	-92.3	-95.3	29	29	30	31
hydrogen cyanide	CHN	27	135.1	124.7	36	42	46	49
hydrogen fluoride	HF	20	-273.3	-275.4	29	29	29	30
hydrogen sulfide	H <sub>2</sub> S	34	-20.6	-33.4	34	37	41	44
isobutyric acid	C <sub>4</sub> H <sub>8</sub> O <sub>2</sub>	88	-484.1	-362.1	105	161	199	226
isopropyl amine	C <sub>3</sub> H <sub>9</sub> N	59	-83.8	31.9	98	143	176	202
malonic acid	C <sub>3</sub> H <sub>4</sub> O <sub>4</sub>	104	-766.8	-670.0	98	143	170	188
methane	CH <sub>4</sub>	16	-74.5	-50.5	36	47	58	68
methanol	CH <sub>4</sub> O	32	-200.9	-162.3	44	60	74	85
methyl acetate	C <sub>3</sub> H <sub>6</sub> O <sub>2</sub>	74	-411.9	-324.2	85	122	153	175
methyl acetylene	C <sub>3</sub> H <sub>4</sub>	40	184.9	193.8	61	83	98	111
methyl acrylate	C <sub>4</sub> H <sub>6</sub> O <sub>2</sub>	86	-333.0	-257.0	100	144	176	197
methyl amine	CH <sub>5</sub> N	31	-23.0	32.1	53	75	90	102
methyl benzoate	C <sub>8</sub> H <sub>8</sub> O <sub>2</sub>	136	-287.9	-181.0	126	205	260	296
3-methyl-1-butanol	C <sub>5</sub> H <sub>12</sub> O	88	-303.0	-141.0	132	203	248	282
methylbutyl ether	C <sub>5</sub> H <sub>12</sub> O	88	-258.1	-101.7	133	198	247	284
methyl butyrate	C <sub>5</sub> H <sub>10</sub> O <sub>2</sub>	102	-450.7	-305.3	135	200	243	274
methylcyclohexane	C <sub>7</sub> H <sub>14</sub>	98	-154.8	27.3	136	231	302	352
methylethyl ether	C <sub>3</sub> H <sub>8</sub> O	60	-216.4	-117.1	92	132	163	187
methylethyl ketone	C <sub>4</sub> H <sub>8</sub> O	72	-239.0	-147.0	104	145	178	204
methyl formate	C <sub>2</sub> H <sub>4</sub> O <sub>2</sub>	60	-352.4	-295.0	66	95	114	128
methyl isocyanate	C <sub>2</sub> H <sub>3</sub> NO	57	-62.4	0.2	52	73	93	108
methyl mercaptan	CH <sub>4</sub> S	48	-22.9	-9.8	50	67	80	90
2-methylpropane	C <sub>4</sub> H <sub>10</sub>	58	-135.0	-21.4	97	149	188	217
2-methyl propene	C <sub>4</sub> H <sub>8</sub>	56	-17.1	58.1	88	129	161	186
methylsilane	CH <sub>6</sub> Si	46	-29.1	18.5	66	94	113	128
methyl vinyl ether	C <sub>3</sub> H <sub>6</sub> O	58	-108.0	-47.3	77	110	135	155
naphthalene	C <sub>10</sub> H <sub>8</sub>	128	150.6	224.1	133	220	276	315
nitrogen	N <sub>2</sub>	28	0.0	0.0	29	30	31	32
nitromethane	CH <sub>3</sub> NO <sub>2</sub>	61	-74.7	-6.9	57	82	99	112
nitrous oxide	N <sub>2</sub> O	44	82.1	104.2	39	46	51	54
octane	C <sub>8</sub> H <sub>18</sub>	114	-208.8	16.0	188	288	359	411
oxygen	O <sub>2</sub>	32	0.0	0.0	29	31	33	34
ozone	O <sub>3</sub>	48	142.7	163.2	39	48	52	55

Table D.2: Physical properties of gas-phase compounds (*continued*).

Name	Formula	MM	$\Delta H_f^\circ$	$\Delta G_f^\circ$	$C_p^\circ$			
					300 K	500 K	700 K	900 K
pentane	C <sub>5</sub> H <sub>12</sub>	72	-146.8	-8.8	120	183	231	267
1-pentyne	C <sub>5</sub> H <sub>8</sub>	68	144.4	210.3	107	152	184	208
2-pentyne	C <sub>5</sub> H <sub>8</sub>	68	128.9	194.5	99	144	178	205
phenol	C <sub>6</sub> H <sub>6</sub> O	94	-96.4	-32.6	105	160	199	224
propadiene	C <sub>3</sub> H <sub>4</sub>	40	190.5	200.8	59	83	100	112
propane	C <sub>3</sub> H <sub>8</sub>	44	-104.7	-24.4	74	113	142	165
1-propanol	C <sub>3</sub> H <sub>8</sub> O	60	-254.6	-159.9	86	128	160	184
2-propanol	C <sub>3</sub> H <sub>8</sub> O	60	-272.1	-175.2	90	132	163	185
propionic acid	C <sub>3</sub> H <sub>6</sub> O <sub>2</sub>	74	-479.9	-385.0	90	129	157	177
propionitrile	C <sub>3</sub> H <sub>5</sub> N	55	51.8	97.5	74	104	126	143
propyl amine	C <sub>3</sub> H <sub>9</sub> N	59	-70.5	41.7	96	141	174	200
propylene	C <sub>3</sub> H <sub>6</sub>	42	20.2	62.6	65	94	119	138
quinone	C <sub>6</sub> H <sub>4</sub> O <sub>2</sub>	108	-122.9	-69.2	108	160	193	216
styrene	C <sub>8</sub> H <sub>8</sub>	104	147.4	213.9	123	192	239	271
sulfur dioxide	O <sub>2</sub> S	64	-296.8	-300.1	40	47	51	54
sulfur hexafluoride	F <sub>6</sub> S	146	-1220.5	-1116.5	97	128	141	147
tetrahydrofuran	C <sub>4</sub> H <sub>8</sub> O	72	-184.2	-79.7	77	134	176	206
toluene	C <sub>7</sub> H <sub>8</sub>	92	50.2	122.2	105	171	216	249
triethyl amine	C <sub>6</sub> H <sub>15</sub> N	101	-95.8	114.1	162	243	305	349
trimethyl amine	C <sub>3</sub> H <sub>9</sub> N	59	-24.3	99.0	92	141	177	203
vinyl acetate	C <sub>4</sub> H <sub>6</sub> O <sub>2</sub>	86	-314.9	-227.9	100	143	177	202
vinyl acetylene	C <sub>4</sub> H <sub>4</sub>	52	304.6	306.0	74	101	120	133
vinyl chloride	C <sub>2</sub> H <sub>3</sub> Cl	62	28.5	42.0	54	74	88	98
water	H <sub>2</sub> O	18	-241.8	-228.6	34	35	38	40
m-xylene	C <sub>8</sub> H <sub>10</sub>	106	17.3	118.8	127	203	257	296
o-xylene	C <sub>8</sub> H <sub>10</sub>	106	19.1	122.0	133	206	259	298
p-xylene	C <sub>8</sub> H <sub>10</sub>	106	18.0	121.4	126	203	257	296



Table D.3: Physical properties of liquid-phase compounds. Enthalpies and Gibbs energies of formation are given at 298.15 K, have units of  $\text{kJ} \cdot \text{mol}^{-1}$ , and the standard state is the pure substance at 298.15 K and 1 bar. Heat capacity is given at 298.15 K and has units of  $\text{J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ .

Name	Formula	MM	$\Delta H_f^\circ$	$\Delta G_f^\circ$	Ref	$C_p$	Ref
acetaldehyde	$\text{C}_2\text{H}_4\text{O}$	44	-192.2	-127.6	4	89	4
acetic acid	$\text{C}_2\text{H}_4\text{O}_2$	60	-484.5	-389.9	1	123	2
ammonium hydroxide	$\text{NH}_5\text{O}$	35	-361.2	-254.0	4	155	4
benzene	$\text{C}_6\text{H}_6$	78	49.1	142.5	1	134	1
biodiesel	<i>see methyl lineolate</i>						
bromoethane	$\text{C}_2\text{H}_5\text{Cl}$	109	-90.5	-25.8	4	101	4
carbon disulfide	$\text{CS}_2$	76	89.7	65.3	3	79	2
carbon tetrachloride	$\text{CCl}_4$	154	-135.4	-65.2	3	131	2
chloroethane	$\text{C}_2\text{H}_5\text{Cl}$	65	-136.8	-59.3	4	104	4
cyclohexane	$\text{C}_6\text{H}_{12}$	84	-156.2	26.9	1	156	1
cyclohexene	$\text{C}_6\text{H}_{10}$	82	-38.5	106.9	3	149	2
ethanol	$\text{C}_2\text{H}_6\text{O}_2$	46	-277.7	-174.8	1	112	1
1,2-ethanediol	$\text{C}_2\text{H}_6\text{O}_2$	62	-454.8	-323.1	1	150	2
ethylacetate	$\text{C}_4\text{H}_8\text{O}_2$	88	-463.3	-318.4	3	169	2
formic acid	$\text{CH}_2\text{O}_2$	46	-425.0	-361.4	4	99	4
glyceryl trilinoleate	$\text{C}_{57}\text{H}_{98}\text{O}_6$	879	-2431	-	var.	1782	var.
glyceryl dilinoleate	$\text{C}_{39}\text{H}_{68}\text{O}_5$	617	-1850	-	var.	1377	var.
glyceryl monolinoleate	$\text{C}_{21}\text{H}_{38}\text{O}_4$	355	-1269	-	var.	972	var.
glycerol	$\text{C}_3\text{H}_8\text{O}$	92	-670.0	-	var.	222	var.
hydrazine	$\text{N}_2\text{H}_4$	32	50.6	149.3	4	99	4
hydrogen cyanide	$\text{HCN}$	27	108.9	125.0	3	71	2
hydrogen peroxide	$\text{H}_2\text{O}_2$	34	-187.8	-120.4	3	89	4
iodoethane	$\text{C}_2\text{H}_5\text{I}$	156	-40.0	14.7	4	115	4
methanol	$\text{CH}_4\text{O}$	32	-238.7	-166.3	1	81	1
methyl lineolate	$\text{C}_{19}\text{H}_{34}\text{O}_2$	295.0	-605.0	-	var.	580	var.
methylcyclohexane	$\text{C}_7\text{H}_{14}$	98	-190.2	20.6	1	184	2
nitric acid	$\text{HNO}_3$	63	-174.1	-80.7	1	110	4
n-pentane	$\text{C}_5\text{H}_{12}$	72	-173.1	-9.2	3	169	2
phosphoric acid	$\text{H}_3\text{PO}_4$	98	-1271.7	-1123.6	4	145	4
soybean oil	<i>see glyceryl trilinoleate</i>						
sulfuric acid	$\text{H}_2\text{SO}_4$	98	-814.0	-690.0	1	139	4
toluene	$\text{C}_7\text{H}_8$	92	12.2	113.6	1	155	1
water	$\text{H}_2\text{O}$	18	-285.8	-237.1	1	75	1
meta-xylene	<i>m</i> - $\text{C}_8\text{H}_{10}$	106	-25.4	107.6	2	185	2
para-xylene	<i>p</i> - $\text{C}_8\text{H}_{10}$	106	-24.4	110.6	2	183	2
ortho-xylene	<i>o</i> - $\text{C}_8\text{H}_{10}$	106	-24.4	111.0	2	187	2

Table D.4: Physical properties of aqueous compounds. Gibbs energies of formation are given at 298.15 K, have units of  $\text{kJ} \cdot \text{mol}^{-1}$ , and the standard state is a hypothetical, ideal 1-molal solution of solute in water at 298.15 K and 1 bar.

Name	Formula	MM	$\Delta G_f^\circ$	Ref
ammonia	$\text{NH}_3$	17	-26.5	1
acetic acid	$\text{CH}_3\text{COOH}$	60	-396.5	3
calcium chloride	$\text{CaCl}_2$	111	-8101.9	1
calcium hydroxide	$\text{Ca}(\text{OH})_2$	74	-868.1	1
carbon dioxide	$\text{CO}_2$	44	-386.0	3
carbonic acid	$\text{H}_2\text{CO}_3$	62	-623.1	3
hydrochloric acid	$\text{HCl}$	36	-131.2	3
hydrogen sulfide	$\text{H}_2\text{S}$	34	-27.8	3
nitric acid	$\text{HNO}_3$	63	-111.3	1
sodium chloride	$\text{NaCl}$	58	-293.1	1
sodium hydroxide	$\text{NaOH}$	40	-419.2	1
sulfuric acid	$\text{H}_2\text{SO}_4$	98	-744.5	1

# Index

## MATLAB functions

- fmincon, 354
- ode45, 153
  - deval function, 156
  - solution object, 156
  - solution points, 155

## Activities, $a_i$ , 50

## Arrhenius rate constant

- activation energy, 69
- pre-exponential factor, 69

## Batch reactor

- material balance,  $C_i$ , 319
- material balance,  $N_i$ , 318

## Continuously stirred tank reactor (CSTR)

- energy balance
  - adiabatic, 277
  - general, 85
  - in terms of  $X_A$ , 130
- fluidized bed reactor, 123
- material balance
  - general, 85
  - in terms of  $X_A$ , 129

## Conversion, 58

- overall,  $X_A^{\text{ov}}$ , 272
- single reactor,  $X_A^{\text{sp}}$ , 272

## Enthalpy of reaction

- adjusted for  $T$ , 93

## standard state, 93

## Equilibrium constant

- $K_C$ , 79
- $K_a$ , 45
- $K_{sc}$ , 47
- $K_p$ , 82
- summary of relations, 83

## Equilibrium conversion $X_A^{\text{eq}}$ , 58

## Equilibrium reactor, 45, 53

## Experimental rate laws

- differential method, 319
- integral method, 319

## Extent of reaction, 40

## Fanning friction factor, 204

## Fluidized bed reactor, 123

## Gibbs reactor

- atomic species balance, 349
- general, 347
- simple, 348

## Heat exchanger

- co-flow, 210
- coil, 95
- counter-flow, 215
- double pipe, 170
- interstage, 282
- jacketed, 95
- shell and tube, 170

## Heat transfer coefficient

- individual, 96
  - overall, 95
- Interstage heating, cooling, 278
- Levenspiel plot, 289
- MATLAB functions
- `fsolve`, 25
  - hold off, 15
  - hold on, 15
  - length, 22
  - `linspace`, 15
  - `numel`, 22
  - `plotResiduals`, 308
  - size, 22
  - subplot, 15
  - local, 19
  - user-defined, 18
- MATLAB programming
- debugger, 18
  - function handle, 26
  - local functions, 19
  - nested functions, 19
  - plotting, 15
  - preallocation, 22
  - scripts, 18
  - user-defined functions, 18
- Membrane reactor, 219
- diffusion coefficient,  $k_c$ , 220
- Multiple steady states
- extinction, 145
  - extinction temperature, 145
  - ignition, 145
  - ignition temperature, 145
  - metastable, 141
  - stable, 141
  - unstable, 141
- Ordinary differential equations
- `ode45`, 153
  - coupled, 160
  - systems of, 160
- Packed bed reactor (PBR), 225
- bed density, 227
  - energy balance
    - adiabatic, 277
    - general, 225
    - in terms of  $X_A$ , 252
  - heat exchangers, 232
  - material balance
    - general, 225
    - in terms of  $X_A$ , 251
  - pressure drop
    - Ergun equation, 260
    - KTA equation, 260
  - void fraction, 228
- Plug flow reactor (PFR)
- energy balance
    - adiabatic, 277
    - general, 165
    - in terms of  $X_A$ , 198
  - heat exchanger
    - $\delta Q$  term, 171
    - co-flow, 210
    - counter-flow, 215
    - double pipe, 170
    - shell and tube, 170
  - material balance
    - general, 165
    - in terms of  $X_A$ , 197
  - membrane reactor, 219
  - pressure drop, 205

- static mixers, 170
- Pressure
  - absolute, 261
  - atmospheric, 261
  - drop
    - PBR, 260
    - PFR, 205
  - gauge, 261
- Rate law
  - elementary, 72
  - mechanisms, 71
  - net rate, 75
  - non-elementary, 72
  - observed, 71
  - order, 65
  - reversible reaction, 79
- Regression, 299
  - plotResiduals, 308
  - linear model, 302
  - model, 300
  - predictor matrix, 303
  - predictor variable, 300
  - residual, 301
  - residual sum of squares, 301
  - response variable, 300
  - response vector, 303
  - sum of squares for error, 300
- Reynolds number, 206
- Root finding, 24
- Shooting method, 216
- Stoichiometric
  - coefficients, 40
  - table, 42
- van't Hoff equation
  - full, 46
  - shortcut, 47
- Vector formulations, 339