Linear Algebra, Statistics, and Vector Calculus for Engineers

T. Scofield and M. Stob

January 26, 2009

Foreward

These notes are for use in the course MATH 232, a 2nd mathematics course for engineering students following the Calculus sequence at Calvin College. Their suitability for other courses in other settings is by no means assured.

The linear algebra part (Part I) of these notes have undergone significant revision after my first attempt to use them. My thanks go out to the students in that first semester who provided helpful input. I am also grateful for the input of my colleagues, and for certain articles written by Carlson, Lay, and other members of the Linear Algebra Curriculum Study Group formed in 1990, in so far as they have assisted in the choice of topics from linear algebra, whose content comprises about (5/13)ths of this course, to include.

I am greatly indebted to Michael Stob for writing (except for very minor changes on my part) Part II of these notes, giving an introductory treatment of statistics. He not only wrote the notes, but also volunteered, above and beyond his assigned duties as a full-time faculty member at Calvin, to teach the statistics part of the course during its first run using his notes. Furthermore, both of us are indebted to Randall Pruim, another colleague in the Department of Mathematics and Statistics at Calvin, for his pre-existing organizational structure which greatly streamlined the typesetting of these notes, and for examples he had prepared already in the writing his own text for probability and statistics.

Despite all this help, it is I who is responsible for putting together these notes in their current form. The errors are all mine.

Thomas L. Scofield
Aug. 13, 2008

A Note to Students

The three main topics in this course—linear algebra, statistics and vector calculus (particularly integration in vector fields)—are taught in separate courses at a majority of institutions around the country. At Calvin College we offer this introduction to the three subjects, taught in 3- to 5-week modules, in a single course. We do so *not* because we wish to be unique, but rather because of the number of credits already required of engineering students in their major program and by the Core Curriculum.

The material in this course has been chosen so as to provide literacy in these three topics. With just 3–5 weeks to spend on each, we will not have much time for delving into applications. The kinds of applications of current interest to engineers are often quite advanced, putting them out of reach for a one-semester course such as this one. Our goals, then, are to gain the ability to do simple problems (often with the aid of software) and, just as importantly, the ability to talk intelligently in these three mathematical subjects, at least at the foundational level. Another goal, still important but less so than those just stated, is to develop facility with several industry-standard software packages (the ability both to use them and to interpret results from them). To evaluate your progress towards these goals, test questions will be given fitting the following types:

- **procedural**: You must be able to identify reliable procedures to use in a given scenario, and to apply them correctly to obtain an answer. Often, both for the purpose of saving time during the test period and to test further the depth of your understanding, the problem scenario will provide you with a 'start' you may not be accustomed to in homework (like the difference between asking a young pianist to start in the middle of a piece instead of at the beginning).

- **conceptual**: "Is this an example of Concept A, Concept B, or something else?" "Give two different examples of a ______." "Explain why ______ does not fit into the category of ______." These questions require you to learn and use the vocabulary (terms and symbols) of the three subjects.

- **factual**: "Is it true that every system of linear algebraic equations has a unique solution?" "What types of random variables can be assumed to be *normal* (i.e., have a normal distribution)?" The facts you need to know sometimes are stated as theorems, at other times appear just in the reading and/or class lectures; homework problems alone generally do *not* prepare students for these questions.

- **reflective**: "Problems A and B look similar, yet require different methods for solving them. What key feature(s) make this necessary?" "What assumptions must one make to consider the answer you just obtained (in the previous problem) reliable? How might one check that these assumptions hold? Just how unreliable is the answer if they do not?"

In short, students who do well in the course are those who demonstrate a certain amount of *knowledge* (albeit at the rudimentary level) in these three subject areas, which includes, but is not limited to, the ability to carry out various procedures to obtain a correct answer. While it is no guarantee of success, the course has been designed to reward students who

- do the homework themselves.[1]
- read and ponder every piece of information thrown at them, trying to formulate a coherent understanding of the subject at hand into which all of the lecture material, reading, examples, and homework fit nicely.
- are willing to experiment with the tools they've seen, inventing different scenarios and seeing (most likely with the use of software) the outcomes. Such students approach exercises with an open mind, an attitude of experimentation that means their very conceptions of one or more mathematical ideas/processes is on the line, and ready to surrender up and replace these conceptions with new formulations when results conflict with prior notions.
- attempt (successfully, at times) to draw broader conclusions that serve to guide them through settings not necessarily encountered in the examples and homework.
- have assimilated the concepts and vocabulary used by professionals to express the underlying ideas.

These are not the kinds of activities one can carry out at the last minute; you must commit to them as daily habits. Learning is not a passive activity, but a *life discipline*, sometimes pleasurable, but always demanding your attention. A class full of students that take this discipline seriously, bringing to class (or to outside meetings with the professor) questions of all types that arise out of discussions of the previous day (week, month, etc.), greatly contributes to the overall experience and knowledge of everyone involved. (Indeed, in such an environment the contribution of the professor greatly surpasses what he can do on his own, apart from this 'community-of-learners' atmosphere.) The opposite extreme, a class in which students expect to be force-fed material from which they remain detached/unengaged—where real knowledge is not the goal so much as the ability to *appear* knowledgeable—generally leaves a bad taste and minimizes the possibility of lasting value.

[1] It is possible to seek out and employ hints from the professor or classmates without compromising this. To borrow another's paper and copy down her answer, however, does *not* contribute in any way to one's knowledge, and will be treated as a case of *academic dishonesty* (on the part of both borrower and borrowee).

And what is the value we seek to acquire from this course? Is it the many useful formulas which have been produced through the work of people doing mathematics? Not just this, certainly. Indeed, not even primarily this, as a college education should comprise much more than training in the use of formulas. The discipline of mathematics is first and foremost a manner of thinking—formulating abstract concepts from specific examples, adapting them when necessary to new settings, drawing a clear line of reasoning from assumptions to conclusions—that tends to produce good problem-solvers of immense worth to society. To improve one's ability to think in this fashion is of paramount importance, the most valuable thing a mathematics course has to offer. At the same time it is a difficult thing to teach—mathematics professors generally do it either by 1) providing open-ended questions that force students to formulate their own concepts, while giving feedback and guidance along the way (a technique that requires *lots* of time and a great deal of flexibility concerning how much material will get covered in the course), or 2) modeling this type of thinking for the class (i.e., explaining the various ties between concepts, explaining the thoughts in the teacher's head as he approaches the problem, giving the reasons for taking one approach as opposed to another, etc.). Time constraints require we employ this second approach. Know, then, that the heart and soul of the course is wrapped up not primarily in the examples (though these are an important part of the dialogue), but rather in the commentary that accompanies them (the commentary during lectures, the paragraphs in between examples in the text, etc.). Accordingly, give significant hearing to these between-example commentaries.

Contents

Part I Linear Algebra

Part II Statistics

Part III Vector Calculus

Part I

Linear Algebra: Theory and Computation

Foreward to Part I

Linear algebra is likely the most useful mathematical subject that an applied mathematician or engineer can learn. Most practical problems are both multidimensional and nonlinear, making them too difficult to solve in any general way. The first attempt to solving such a problem usually involves approximating it locally by a linearized version, hence bringing it into the purview of linear algebra.

While the kind of linear algebra problems addressed in the homework for this course will generally be small enough in size that calculations can be done on paper or with a modern calculator, many applied problems are several orders of magnitude larger, requiring calculations to be done using software on a computer. One package often used for such calculations is MATLAB, and an aim in this course is to familiarize you with the syntax and use of this language. But, while the college has MATLAB installed in certain of its computer laboratories, it is somewhat expensive to purchase your own copy. There is a GNU-public license package called OCTAVE which has much of the same command structure and feel of MATLAB. As it is free, powerful in its own right, and available for the three main platforms (Windows, Macintosh OS-X, and Linux), a number of book examples will be provided in OCTAVE. When this is the case, commands will usually be portable to MATLAB with few changes required.

While software may free us, in practice, from some technical calculations, it is necessary that we be aware of the underlying structure. Thus, we spend a good deal of time away from the computer and calculations, focusing on new concepts and their interconnections (as provided by theorems).

1 Solving Linear Systems of Equations

1.1 Matrix Algebra

Definition 1.1.1. An m-by-n **real matrix** is a table of m rows and n columns of real numbers. We say that the matrix has **dimensions** m-by-n.

The plural of *matrix* is **matrices**.

Remarks:

1. Often we write a matrix $\mathbf{A} = (a_{ij})$, indicating that the matrix under consideration may be referred to as a single unit by the name $\mathbf{A}$, but that one may also refer to the entry in the i^{th} row, j^{th} column as a_{ij}.

2. If one of the matrix dimensions m or n is equal to 1, it is common to call the table a **vector** (or **column vector**, if $n = 1$; a **row vector** if $m = 1$). Though column vectors are just special matrices, it is common to use lowercase boldface letters for them (like $\mathbf{u}$, $\mathbf{v}$, $\mathbf{x}$, etc.), reserving uppercase boldface letters for other types of matrices. When $\mathbf{x}$ is an n-by-1 vector, we often denote its components with singly-subscripted non-bold letters—x_1 for the first component, x_2 for the 2^{nd}, and so on.

Practitioners carry out large-scale linear algebraic computations using software, and in this section we will alternate between discussions of concepts, and demonstrations of corresponding implementations in the software package Octave. To create a matrix (or vector) in Octave, you enclose elements in square brackets ([and]). Elements on the same row should be separated only by a space (or a comma). When you wish to start a new row, you indicate this with a semicolon (;). So, to enter the matrices

$$\begin{bmatrix} 1 & 5 & -2 \end{bmatrix}, \quad \begin{bmatrix} 4 \\ -1 \\ 3 \\ 7 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 3 & 0 \\ -1 & 5 \\ 2 & 1 \end{bmatrix},$$

you can type

```
octave-3.0.0:1> [1 5 -2]
ans =
```

```
   1   5  -2

octave-3.0.0:2> [4; -1; 3; 7]
ans =
   4
  -1
   3
   7

octave-3.0.0:3> A = [3 0; -1 5; 2 1];
```

In all but the third of these commands, Octave echoed back to us its understanding of what we typed. It did not do so for the last command because we ended that command with a final semicolon. Also, since we preceded our final matrix with "`A =`", the resulting matrix may now be referred to by the letter (variable) `A`.

Just as writing $\mathbf{A} = (a_{ij})$ gives us license to refer to the element in the 2nd row, 1st column as a_{21}, storing a matrix in a variable in Octave gives us an immediate way to refer to its elements. The entry in the 2nd row, 1st column of the matrix `A` defined above can be obtained immediately by

```
octave-3.0.0:4> A(2,1)
ans = -1
```

That is, you can pick and choose an element from `A` by indicating its location in parentheses.

One can easily extract whole **submatrices** from `A` as well. Suppose you want the entire first row of entries. This you do by specifying the row, but using a colon (:) in place of specifying the column.

```
octave-3.0.0:5> A(1,:)
ans =
   3   0
```

Next, suppose we want to get the first and third rows of `A`. Since we want full rows here, we continue to use the colon where a column can be specified. We use a vector whose entries are 1 and 3 to specify which rows.

```
octave-3.0.0:6> A([1 3],:)
ans =
   3   0
   2   1
```

There are some shortcuts in Octave when creating matrices or vectors with particular kinds of structure. The colon may be used between numbers as a quick way to create row vectors whose entries are evenly spaced. For instance, a row vector containing the first five positive integers is produced by the command

```
octave-3.0.0:7> 1:5
  ans =
    1   2   3   4   5
```

You can also specify a "step" or "meshsize" along the way, as in

```
octave-3.0.0:8> 1:.5:3
ans =
   1.0000   1.5000   2.0000   2.5000   3.0000
```

One implication of the ability to create vectors in this fashion is that, if we desire the first two rows of the matrix `A` above, either of the commands

```
octave-3.0.0:9> A(1:2, :)
```

or

```
octave-3.0.0:10> A([1 2], :)
```

will do the trick.

Among **square matrices** (i.e., ones having equal numbers of rows and columns) are those classified as **diagonal** matrices. Such a matrix $\mathbf{A} = (a_{ij})$ is one whose entries a_{ij} are zero whenever $i \neq j$. The `diag()` command makes it easy to construct such a matrix in Octave, even providing the ability to place specified entries on a **super-** or **subdiagonal** (i.e., a diagonal that lies above or below the **main diagonal**). We give here two examples of the use of `diag()`. In the first case, the only argument is a vector, whose entries are then placed on the main diagonal of an appropriately-sized diagonal matrix; in the 2^{nd} case, the additional argument of (-1) is used to request that the vector of entries be placed on the first subdiagonal.

```
octave-3.0.0:7> diag([1 3 -1])
ans =
   1   0   0
   0   3   0
   0   0  -1

octave-3.0.0:8> diag([1 3 -1], -1)
ans =
   0   0   0   0
   1   0   0   0
   0   3   0   0
   0   0  -1   0
```

Other Octave commands that are helpful in producing certain types of matrices are `zeros()`, `ones()`, `eye()`, and `rand()`. You can read the help pages to learn the purpose and required syntax of these and other Octave commands by typing

```
octave-3.0.0:1> help <command name>
```

at the Octave prompt. It is perhaps relevant to note that numbers (scalars) themselves are considered by Octave to be 1-by-1 matrices.

The title of this section is “Matrix Algebra”. In any algebraic system we must know what one means by the word *equality*.

Definition 1.1.2. Two matrices $\mathbf{A} = (a_{ij})$ and $\mathbf{B} = (b_{ij})$ are said to be *equal* if their dimensions are equal, and if the entries in every location are equal.

Example 1.1.1

The two vectors

$$\begin{bmatrix} 3 & -1 & 2 & 5 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 3 \\ -1 \\ 2 \\ 5 \end{bmatrix}$$

cannot be considered equal, since they have different dimensions. While the entries are the same, the former is a *row* vector and the latter a *column* vector. ■

Octave has many built-in functions. Users can augment these built-in functions with functions of their own. Let’s write a function which accepts two matrices and checks whether they are equal. (Such a function has no practical application, but gives us the opportunity to illustrate the writing of a function in Octave, as well as some other language constructs.)

```
octave-3.0.0:30> function equalMat(AA, BB)
>   sizeAA = size(AA);
>   sizeBB = size(BB);
>   if (sizeAA(1) ~= sizeBB(1) || sizeAA(2) ~= sizeBB(2))
>     disp('These matrices do not have the same dimension.')
>     disp('They cannot possibly be equal.')
>   elseif (sum(sum(AA ~= BB)) == 0)
>     disp('The matrices are equal.')
>   else
>     disp('The matrices have the same dimensions, but are not equal.')
>   end
> end
```

Some notes about what you see in the function above.

- Octave is case-sensitive.

- The function above, as written, requires two inputs, both matrices. It, however, has no return value. Appendix A contains examples of functions that *do* return a value, but for what this function does, there is no need of one.

- Certain keywords herald the start of a larger construct in Octave; `function` and `if` are such keywords in the code above. When a larger construct opens, Octave holds off on evaluating anything, as exhibited by the lack of another prompt beginning with the word `octave-3.0.0`. Only once all constructs have been completed with appropriate ending keywords will Octave really do something—in this case, store the sequence of commands to execute at some later time, when someone types something like

```
A = [1 3; 1 1; 0 1];
equalMat([1 5; 2 7; 8 -1], A)
```

- A single equals sign makes an assignment of a value to a variable. For instance, in the code above, `sizeAA` is the name of a variable which holds the value (a vector) returned by the call to the function `size()`.

  When one wishes to test whether two quantities are equal, a double equals (==) must be used. If, on the other hand, one wants to test whether two values are not equal, one uses the pair of symbols ~= (or the pair !=).

- One can store these exact same commands in a text file named `equalMat.m`. So long as such a file is in your path (in your working directory, or in some other directory Octave knows to look in for commands you've written), a call to the `equalMat()` function will put this function into action.

In this course, the term **vector** will be synonomous with *column vector*. The set of vectors having n components, all of which are real numbers, will be called $\mathbb{R}^n$, or sometimes **Euclidean n-space**. The elements of $\mathbb{R}^n$ are n-by-1 matrices, sometimes called **n-vectors**. However, as it takes less room out of a page to list the contents of a vector horizontally rather than vertically, we will often specify an n-vector horizontally using parentheses, as in

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) .$$

The most fundamental algebraic operations on matrices are as follows:

1. **Addition of Two Matrices**.

   Given two m-by-n matrices $\mathbf{A} = (a_{ij})$ and $\mathbf{B} = (b_{ij})$, we define their sum $\mathbf{A} + \mathbf{B}$ to be the m-by-n matrix whose entries are $(a_{ij} + b_{ij})$. That is,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mn} \end{bmatrix} := \begin{bmatrix} a_{11}+b_{11} & a_{12}+b_{12} & \cdots & a_{1n}+b_{1n} \\ a_{21}+b_{21} & a_{22}+b_{22} & \cdots & a_{2n}+b_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1}+b_{m1} & a_{m2}+b_{m2} & \cdots & a_{mn}+b_{mn} \end{bmatrix}.$$

In order to add two matrices, they must have the same number of rows and columns (i.e., be matrices with *the same dimensions*). Note that this is not the same as saying they must be square matrices!

It is simple to add two matrices in Octave. One possibility is code like

```
octave-3.0.0:39> A = [3 1 6; 1 2 -1];
octave-3.0.0:40> A + ones(2,3)
ans =
    4   2   7
    2   3   0
```

which creates a 2-by-3 matrix $\mathbf{A}$, and then adds to it another 2-by-3 matrix whose entries are all ones.

2. **Multiplication of a Matrix by a Scalar**.

   Given an m-by-n matrix $\mathbf{A} = (a_{ij})$ and a scalar c, we define the scalar multiple $c\mathbf{A}$ to be the m-by-n matrix whose entries are (ca_{ij}). That is,

$$c \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} := \begin{bmatrix} ca_{11} & ca_{12} & \cdots & ca_{1n} \\ ca_{21} & ca_{22} & \cdots & ca_{2n} \\ \vdots & \vdots & & \vdots \\ ca_{m1} & ca_{m2} & \cdots & ca_{mn} \end{bmatrix}.$$

Our definitions for matrix addition and scalar multiplication have numerous implications. They include the following:

a) Matrix *subtraction* is merely a combination of matrix addition and scalar multiplication by (-1): $\mathbf{A} - \mathbf{B} := \mathbf{A} + (-1)\mathbf{B}$.

b) Distributive laws between matrix addition and scalar multiplication hold:

   i. $c(\mathbf{A} + \mathbf{B}) = c\mathbf{A} + c\mathbf{B}$.

   ii. $(c + d)\mathbf{A} = c\mathbf{A} + d\mathbf{A}$.

c) An appopriately-sized matrix whose entries are all zeros serves as an **additive identity** (or **zero matrix**, denoted in boldface by $\mathbf{0}$). That is, $\mathbf{A} + \mathbf{0} = \mathbf{A}$.

d) Scalar multiplication by 0 produces the *zero matrix* $\mathbf{0}$. That is, $(0)\mathbf{A} = \mathbf{0}$.

Scalar multiplication in Octave is as easy as in the following sample code

```
octave-3.0.0:47> 3*randn(3,2)
ans =
  -4.03239   3.04860
   1.67442   2.60456
   0.33131   2.31099
```

which produces a 3-by-2 matrix whose entries are sampled from a normal distribution with mean 0 and standard deviation 1, and then multiplies it by the scalar 3.

3. **Multiplication of Two Matrices**

When we multiply two matrices, the product is a matrix whose elements arise from **dot products**[1] between the rows of the first (matrix) factor and columns of the second. An immediate consequence of this: if $\mathbf{A}$ and $\mathbf{B}$ are matrices, the product $\mathbf{AB}$ makes sense precisely when the number of columns in $\mathbf{A}$ is equal to the number of rows in $\mathbf{B}$. To be clearer about how such a matrix product is achieved, suppose $\mathbf{A}$ is an m-by-n matrix while $\mathbf{B}$ is an $n \times p$ matrix. If we write

$$\mathbf{A} = \begin{bmatrix} \mathbf{r_1} & \rightarrow \\ \mathbf{r_2} & \rightarrow \\ \vdots & \\ \mathbf{r_m} & \rightarrow \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_p} \\ \downarrow & \downarrow & & \downarrow \end{bmatrix},$$

with each of the rows $\mathbf{r_i}$ of $\mathbf{A}$ having n components and likewise each of the columns $\mathbf{c_j}$ of $\mathbf{B}$, then their product is an $m \times p$ matrix whose entry in the i^{th}-row, j^{th}-column is obtained by taking the dot product of $\mathbf{r_i}$ with $\mathbf{c_j}$. Thus if

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 0 & 3 \\ -5 & 1 \\ 7 & -4 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 3 & 1 & 0 \\ -2 & 4 & 10 \end{bmatrix},$$

[1]The *dot product* of two vectors is a concept from MATH 162, studied primarily in the case where those vectors have just two components. It appears as well in elementary physics courses.

then the product **AB** will be the 4×3 matrix

$$\mathbf{AB} = \begin{bmatrix} (2,-1)\cdot\begin{bmatrix}3\\-2\end{bmatrix} & (2,-1)\cdot\begin{bmatrix}1\\4\end{bmatrix} & (2,-1)\cdot\begin{bmatrix}0\\10\end{bmatrix} \\ (0,3)\cdot\begin{bmatrix}3\\-2\end{bmatrix} & (0,3)\cdot\begin{bmatrix}1\\4\end{bmatrix} & (0,3)\cdot\begin{bmatrix}0\\10\end{bmatrix} \\ (-5,1)\cdot\begin{bmatrix}3\\-2\end{bmatrix} & (-5,1)\cdot\begin{bmatrix}1\\4\end{bmatrix} & (-5,1)\cdot\begin{bmatrix}0\\10\end{bmatrix} \\ (7,-4)\cdot\begin{bmatrix}3\\-2\end{bmatrix} & (7,-4)\cdot\begin{bmatrix}1\\4\end{bmatrix} & (7,-4)\cdot\begin{bmatrix}0\\10\end{bmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} 8 & -2 & -10 \\ -6 & 12 & 30 \\ -17 & -1 & 10 \\ 29 & -9 & -40 \end{bmatrix}.$$

Remarks:

- When we write **AB**, where **A**, **B** are appropriately-sized matrices, we will mean the product of these two matrices using multiplication as defined above. In Octave, you must be careful to include the multiplication symbol (since `AB` is a valid variable name), as in

```
octave-3.0.0:53> A = [1 2 3; 4 5 6];
octave-3.0.0:54> B = [2; 3; 1];
octave-3.0.0:55> A*B
ans =
   11
   29
```

  The manner in which we defined matrix multiplication is the standard (and most useful) one. Nevertheless, there are times one simply wants to multiply a bunch of numbers together quickly and efficiently. If those numbers are all stored in matrices having the same dimensions, Octave offers a way to carry this out. It is called **componentwise multiplication**, and to get Octave to do it, we must precede the multiplication symbol with a period (.). That is,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} .* \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix} = \begin{bmatrix} a\alpha & b\beta \\ c\gamma & d\delta \end{bmatrix}.$$

  In fact, the operations of multiplication, division and exponentiation can be performed pointwise in Octave (note that addition and subtraction already are), when the occasion merits it, by preceding them with a period. The difference between regular and pointwise exponentiation is illustrated below.

```
octave-3.0.0:59> A = [3 1; 4 3]
A =
   3   1
   4   3

octave-3.0.0:60> A^2
ans =
   13    6
   24   13

octave-3.0.0:61> A.^2
ans =
    9    1
   16    9
```

- Notice that, if $\mathbf{A}$ is 2-by-4 and $\mathbf{B}$ is 4-by-3, then the product $\mathbf{AB}$ is defined, but the product $\mathbf{BA}$ is not. This is because the number of columns in $\mathbf{B}$ is unequal to the number of rows in $\mathbf{A}$. Thus, for it to be possible to multiply two matrices, one of which is m-by-n, in either order, it is necessary that the other be n-by-m. Even when both products $\mathbf{AB}$ and $\mathbf{BA}$ are possible, however, *matrix multiplication is not commutative*. That is, $\mathbf{AB} \neq \mathbf{BA}$, in general.
- We *do* have a distributive law for matrix multiplication and addition. In particular, $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB} + \mathbf{AC}$, for all appropriately-sized matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$.
- When an m-by-n matrix $\mathbf{A}$ is multiplied by an n-by-1 (column) vector (an n-vector, for short), the result is an m-vector. That is, for each n-vector $\mathbf{v}$, $\mathbf{Av}$ is an m-vector. It is natural to think of left-multiplication by $\mathbf{A}$ as a *mapping* (or function) which takes n-vectors $\mathbf{v}$ as inputs and produces m-vectors $\mathbf{Av}$ as outputs. Of course, if $\mathbf{B}$ is an ℓ-by-m matrix, then one can left-multiply the product $\mathbf{Av}$ by $\mathbf{B}$ to get $\mathbf{B}(\mathbf{Av})$. The manner in which we defined matrix products ensures that things can be grouped differently with no change in the answer—that is, so
$$(\mathbf{BA})\mathbf{v} = \mathbf{B}(\mathbf{Av}) .$$
- Notice that the n-by-n matrix
$$\mathbf{I}_n := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$
has the property that, whenever $\mathbf{C}$ is an n-by-p matrix (so that the product $\mathbf{I}_n\mathbf{C}$ makes sense), it is the case that $\mathbf{I}_n\mathbf{C} = \mathbf{C}$. Moreover, if $\mathbf{B}$ is an m-by-n matrix, then

$\mathbf{BI}_n = \mathbf{B}$. Since multiplication by $\mathbf{I}_n$ does not change the matrix (or vector) with which you started, $\mathbf{I}_n$ is called the ***n*-by-*n* identity matrix**. In most instances, we will write $\mathbf{I}$ instead of $\mathbf{I}_n$, as the dimensions of $\mathbf{I}$ should be clear from context.

In Octave, the function that returns the n-by-n identity matrix is `eye(n)`. This explains the result of the commands

```
octave-3.0.0:73> A = [1 2 3; 2 3 -1]
A =
   1   2   3
   2   3  -1

octave-3.0.0:74> A*eye(3)
ans =
   1   2   3
   2   3  -1
```

- For a square (n-by-n) matrix $\mathbf{A}$, there may be a corresponding n-by-n matrix $\mathbf{B}$ having the property that

$$\mathbf{AB} = \mathbf{BA} = \mathbf{I}_n .$$

If so, the matrix $\mathbf{A}$ is said to be **nonsingular** or **invertible**, with inverse matrix $\mathbf{B}$. Usually the **inverse of A**, when it exists, is denoted by $\mathbf{A}^{-1}$. This relationship is symmetric, so if $\mathbf{B}$ is the inverse of $\mathbf{A}$, then $\mathbf{A}$ is the inverse of $\mathbf{B}$ as well. If $\mathbf{A}$ is not invertible, it is said to be **singular**.

The following fact about the product of invertible matrices is easily proved.

Theorem 1.1.3. Suppose $\mathbf{A}$, $\mathbf{B}$ are both n-by-n invertible matrices. Then their product $\mathbf{AB}$ is invertible as well, having inverse $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.

When $\mathbf{A}$ is invertible, it is not so easy to find $\mathbf{A}^{-1}$ as one might think. With rounding (and sometimes instability) in the calculations, one cannot, in general, get a perfect representation of the inverse using a calculator or computer, though the representation one gets is often good enough. In Octave one uses the `inv()` command.

```
octave-3.0.0:66> A = [1 2 3; 2 3 -1; 1 0 -2]
A =
   1   2   3
   2   3  -1
   1   0  -2

octave-3.0.0:67> B = inv(A)
B =
   0.66667  -0.44444   1.22222
  -0.33333   0.55556  -0.77778
```

```
   0.33333  -0.22222   0.11111

octave-3.0.0:68> B*A
ans =
   1.00000  -0.00000   0.00000
   0.00000   1.00000   0.00000
   0.00000   0.00000   1.00000
```

4. **Transposition of a Matrix**

   Look closely at the two matrices

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 0 & -1 \\ -3 & -1 & 1 & -1 \\ 2 & -2 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & -3 & 2 \\ 2 & -1 & -2 \\ 0 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}$$

   for a connection between the two. The matrix **B** has been formed from **A** so that the first column of **A** became the first row of **B**, the second column of **A** became the 2^{nd} row of **B**, and so on. (One might say with equal accuracy that the rows of **A** became the columns of **B**, or that the rows/columns of **B** are the columns/rows of **A**.) The operation that produces this matrix **B** from (given) matrix **A** is called **transposition**, and matrix **B** is called the **transpose of A**, denoted as $\mathbf{B} = \mathbf{A}^T$. (Note: In some texts the *prime* symbol is used in place of the ^{T}, as in $\mathbf{B} = \mathbf{A}'$.)

   When you already have a matrix **A** defined in Octave, there is a simple command that produces its transpose. Strictly speaking that command is `transpose()`. However, placing an apostrophe (a *prime*) after the name of the matrix produces the tranpose as well, so long as the entries in the matrix are all *real* numbers (i.e., having zero *imaginary parts*). That is why the result of the two commands below is the same for the matrix **A** on which we use them.

```
octave-3.0.0:77> A = [1 2 3; 2 3 -1]
A =
   1   2   3
   2   3  -1

octave-3.0.0:78> transpose(A)
ans =
   1   2
   2   3
   3  -1

octave-3.0.0:79> A'
ans =
```

```
1   2
2   3
3  -1
```

Remarks:

- If $\mathbf{A}$ is an m-by-n matrix, then $\mathbf{A}^T$ is n-by-m.
- Some facts which are easy to prove about matrix transposition are the following:
  (i) For all matrices $\mathbf{A}$ it is the case that $(\mathbf{A}^T)^T = \mathbf{A}$.
  (ii) Whenever two matrices $\mathbf{A}$ and $\mathbf{B}$ can be added, it is the case that $(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$.
  (iii) Whenever the product $\mathbf{AB}$ of two matrices $\mathbf{A}$ and $\mathbf{B}$ is defined, it is the case that $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$.
  (Compare this result to Theorem 1.1.3, a similar-looking fact about the inverse of the product of two invertible matrices.)
  (iv) For each invertible matrix $\mathbf{A}$, $\mathbf{A}^T$ is invertible as well, with $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$.
- There are some matrices $\mathbf{A}$ for which $\mathbf{A}^T = \mathbf{A}$. Such matrices are said to be **symmetric**.

1.2 Matrix Multiplication and Systems of Linear Equations

1.2.1 Several interpretations of matrix multiplication

In the previous section we saw what is required (in terms of matrix dimensions) in order to be able to produce the product $\mathbf{AB}$ of two matrices $\mathbf{A}$ and $\mathbf{B}$, and we saw how to produce this product. There are several useful ways to conceptualize this product, and in this first sub-section we will investigate them. We first make a definition.

Definition 1.2.1. Let $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k$ be matrices all having the same dimensions. For each choice of real numbers $c_1, \ldots, c_k$, we call

$$c_1\mathbf{A}_1 + c_2\mathbf{A}_2 + \cdots + c_k\mathbf{A}_k$$

a **linear combination** of the matrices $\mathbf{A}_1, \ldots, \mathbf{A}_k$. The set of all such linear combinations

$$S := \{c_1\mathbf{A}_1 + c_2\mathbf{A}_2 + \cdots + c_k\mathbf{A}_k \mid c_1, \ldots, c_k \in \mathbb{R}\}$$

is called the **linear span** (or simply **span**) of the matrices $\mathbf{A}_1, \ldots, \mathbf{A}_k$. We sometimes write $S = \text{span}(\{\mathbf{A}_1, \ldots, \mathbf{A}_k\})$.

Here, now, are several different ways to think about product **AB** of two appropriately sized matrices **A** and **B**.

1. **Block multiplication**. This is the first of four descriptions of matrix multiplication, and it is the most general. In fact, each of the three that follow is a special case of this one.

   Any matrix (table) may be separated into **blocks** (or *submatrices*) via horizontal and vertical lines. We first investigate the meaning of matrix multiplication at the block level when the left-hand factor of the matrix product **AB** has been subdivided using only vertical lines, while the right-hand factor has correspondingly been blocked using only horizontal lines.

Example 1.2.1

Suppose

$$\mathbf{A} = \left[\begin{array}{cc|c|cc} 8 & 8 & 3 & -4 & 5 \\ 6 & -6 & 1 & -8 & 6 \\ 5 & 3 & 4 & 2 & 7 \end{array}\right] = \left[\begin{array}{c|c|c} \mathbf{A}_1 & \mathbf{A}_2 & \mathbf{A}_3 \end{array}\right]$$

(Note how we have named the three blocks found in **A**!), and

$$\mathbf{B} = \left[\begin{array}{cccc} -3 & 5 & -5 & -2 \\ 2 & -2 & 2 & -7 \\ \hline -6 & 6 & 0 & 3 \\ \hline -3 & 2 & -5 & 0 \\ 0 & -1 & -1 & 4 \end{array}\right] = \left[\begin{array}{c} \mathbf{B}_1 \\ \hline \mathbf{B}_2 \\ \hline \mathbf{B}_3 \end{array}\right].$$

Then

$$\begin{aligned} \mathbf{AB} &= \mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_2 + \mathbf{A}_3\mathbf{B}_3 \\ &= \begin{bmatrix} 8 & 8 \\ 6 & -6 \\ 5 & 3 \end{bmatrix}\begin{bmatrix} -3 & 5 & -5 & -2 \\ 2 & -2 & 2 & -7 \end{bmatrix} + \begin{bmatrix} 3 \\ 1 \\ 4 \end{bmatrix}\begin{bmatrix} -6 & 6 & 0 & 3 \end{bmatrix} + \begin{bmatrix} -4 & 5 \\ -8 & 6 \\ 2 & 7 \end{bmatrix}\begin{bmatrix} -3 & 2 & -5 & 0 \\ 0 & -1 & -1 & 4 \end{bmatrix} \\ &= \begin{bmatrix} -8 & 24 & -24 & -72 \\ -30 & 42 & -42 & 30 \\ -9 & 19 & -19 & -31 \end{bmatrix} + \begin{bmatrix} -18 & 18 & 0 & 9 \\ -6 & 6 & 0 & 3 \\ -24 & 24 & 0 & 12 \end{bmatrix} + \begin{bmatrix} 12 & -13 & 15 & 20 \\ 24 & -22 & 34 & 24 \\ -6 & -3 & -17 & 28 \end{bmatrix} \\ &= \begin{bmatrix} -14 & 29 & -9 & -43 \\ -12 & 26 & -8 & 57 \\ -39 & 40 & -36 & 9 \end{bmatrix}. \end{aligned}$$

∎

While we were trying to keep things simple in the previous example by drawing only vertical lines in **A**, the number and locations of those vertical lines was somewhat arbitrary. Once we chose how to subdivide **A**, however, the horizontal lines in **B** had to be drawn to create blocks with rows as numerous as the columns in the blocks of **A**.

Now, suppose we subdivide the left factor with *both* horizontal and vertical lines. Say that

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \\ \hline \mathbf{A}_{31} & \mathbf{A}_{32} \end{array}\right].$$

Where the vertical line is drawn in **A** continues to dictate where a horizontal line must be drawn in the right-hand factor **B**. On the other hand, if we draw any vertical lines in to create blocks in the right-hand factor **B**, they can go anywhere, paying no heed to where the horizontal lines appear in **A**. Say that

$$\mathbf{B} = \left[\begin{array}{c|c|c|c} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} & \mathbf{B}_{14} \\ \hline \mathbf{B}_{21} & \mathbf{B}_{22} & \mathbf{B}_{23} & \mathbf{B}_{24} \end{array}\right].$$

Then

$$\begin{aligned} \mathbf{AB} &= \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \\ \hline \mathbf{A}_{31} & \mathbf{A}_{32} \end{array}\right] \left[\begin{array}{c|c|c|c} \mathbf{B}_{11} & \mathbf{B}_{12} & \mathbf{B}_{13} & \mathbf{B}_{14} \\ \hline \mathbf{B}_{21} & \mathbf{B}_{22} & \mathbf{B}_{23} & \mathbf{B}_{24} \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} & \mathbf{A}_{11}\mathbf{B}_{12} + \mathbf{A}_{12}\mathbf{B}_{22} & \mathbf{A}_{11}\mathbf{B}_{13} + \mathbf{A}_{12}\mathbf{B}_{23} & \mathbf{A}_{11}\mathbf{B}_{14} + \mathbf{A}_{12}\mathbf{B}_{24} \\ \hline \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} & \mathbf{A}_{21}\mathbf{B}_{12} + \mathbf{A}_{22}\mathbf{B}_{22} & \mathbf{A}_{21}\mathbf{B}_{13} + \mathbf{A}_{22}\mathbf{B}_{23} & \mathbf{A}_{21}\mathbf{B}_{14} + \mathbf{A}_{22}\mathbf{B}_{24} \\ \hline \mathbf{A}_{31}\mathbf{B}_{11} + \mathbf{A}_{32}\mathbf{B}_{21} & \mathbf{A}_{31}\mathbf{B}_{12} + \mathbf{A}_{32}\mathbf{B}_{22} & \mathbf{A}_{31}\mathbf{B}_{13} + \mathbf{A}_{32}\mathbf{B}_{23} & \mathbf{A}_{31}\mathbf{B}_{14} + \mathbf{A}_{32}\mathbf{B}_{24} \end{array}\right]. \end{aligned}$$

Example 1.2.2

Suppose **A**, **B** are the same as in Example 1.2.1. Let's subdivide **A** in the following (arbitrarily chosen) fashion:

$$\mathbf{A} = \left[\begin{array}{cccc|c} 8 & 8 & 3 & -4 & 5 \\ \hline 6 & -6 & 1 & -8 & 6 \\ 5 & 3 & 4 & 2 & 7 \end{array}\right] = \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right].$$

Given the position of the vertical divider in **A**, we must place a horizontal divider in **B** as shown below. Without any requirements on where vertical dividers appear, we choose (again arbitrarily) not to have any.

$$\mathbf{B} = \left[\begin{array}{cccc} -3 & 5 & -5 & -2 \\ 2 & -2 & 2 & -7 \\ -6 & 6 & 0 & 3 \\ -3 & 2 & -5 & 0 \\ \hline 0 & -1 & -1 & 4 \end{array}\right] = \left[\begin{array}{c} \mathbf{B}_1 \\ \hline \mathbf{B}_2 \end{array}\right].$$

Then

$$
\begin{aligned}
\mathbf{AB} &= \left[\begin{array}{c} \mathbf{A}_{11}\mathbf{B}_1 + \mathbf{A}_{12}\mathbf{B}_2 \\ \hline \mathbf{A}_{21}\mathbf{B}_1 + \mathbf{A}_{22}\mathbf{B}_2 \end{array}\right] \\
&= \left[\begin{array}{c} \begin{bmatrix} 8 & 8 & 3 & -4 \end{bmatrix}\begin{bmatrix} -3 & 5 & -5 & -2 \\ 2 & -2 & 2 & -7 \\ -6 & 6 & 0 & 3 \\ -3 & 2 & -5 & 0 \end{bmatrix} + 5\begin{bmatrix} 0 & -1 & -1 & 4 \end{bmatrix} \\ \hline \begin{bmatrix} 6 & -6 & 1 & -8 \\ 5 & 3 & 4 & 2 \end{bmatrix}\begin{bmatrix} -3 & 5 & -5 & -2 \\ 2 & -2 & 2 & -7 \\ -6 & 6 & 0 & 3 \\ -3 & 2 & -5 & 0 \end{bmatrix} + \begin{bmatrix} 6 \\ 7 \end{bmatrix}\begin{bmatrix} 0 & -1 & -1 & 4 \end{bmatrix} \end{array}\right] \\
&= \left[\begin{array}{cccc} -14 & 29 & -9 & -43 \\ \hline -12 & 26 & -8 & 57 \\ -39 & 40 & -36 & 9 \end{array}\right].
\end{aligned}
$$

■

2. **Sums of rank-one matrices**. Now let us suppose that $\mathbf{A}$ has n columns and $\mathbf{B}$ has n rows. Suppose also that we block (as described in the simpler case above) $\mathbf{A}$ by column—one column per block—and correspondingly $\mathbf{B}$ by row:

$$
\mathbf{A} = \left[\begin{array}{c|c|c|c} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \end{array}\right] \qquad \text{and} \qquad \mathbf{B} = \left[\begin{array}{c} \mathbf{B}_1 \\ \hline \mathbf{B}_2 \\ \hline \vdots \\ \hline \mathbf{B}_n \end{array}\right].
$$

Following Example 1.2.1, we get

$$
\mathbf{AB} = \mathbf{A}_1\mathbf{B}_1 + \mathbf{A}_2\mathbf{B}_2 + \cdots + \mathbf{A}_n\mathbf{B}_n = \sum_{j=1}^{n} \mathbf{A}_j\mathbf{B}_j \,. \tag{1.1}
$$

The only thing new here to say concerns the individual products $\mathbf{A}_j\mathbf{B}_j$ themselves, in which the first factor $\mathbf{A}_j$ is a vector in $\mathbb{R}^m$ and the 2$^{\text{nd}}$ $\mathbf{B}_j$ is the *transpose* of a vector in $\mathbb{R}^p$ (for some m and p).

So, take $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^p$. Since $\mathbf{u}$ is m-by-1 and $\mathbf{v}^T$ is 1-by-p, the product $\mathbf{u}\mathbf{v}^T$, called the **outer product** of $\mathbf{u}$ and $\mathbf{v}$, makes sense, yielding an m-by-p matrix.

Example 1.2.3

Given $\mathbf{u} = (-1, 2, 1)$ and $\mathbf{v} = (3, 1, -1, 4)$, their vector outer product is

$$\mathbf{u}\mathbf{v}^T = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & -1 & 4 \end{bmatrix} = \begin{bmatrix} -3 & -1 & 1 & -4 \\ 6 & 2 & -2 & 8 \\ 3 & 1 & -1 & 4 \end{bmatrix}.$$

If you look carefully at the resulting outer product in the previous example, you will notice it has relatively simple structure—its 2^{nd} through 4^{th} columns are simply scalar multiples of the first, and the same may be said about the 2^{nd} and 3^{rd} rows in relation to the 1^{st} row. Later in these notes, we will define the concept of the **rank of a matrix**. Vector outer products are always matrices of rank 1 and thus, by (1.1), every matrix product can be broken into the sum of rank-one matrices.

3. **Linear combinations of columns of A**. Suppose $\mathbf{B}$ has p columns, and we partition it in this fashion (Notice that $\mathbf{B}_j$ represents the j^{th} *column* of $\mathbf{B}$ instead of the j^{th} row, as it meant above!):
$$\mathbf{B} = \left[\begin{array}{c|c|c|c} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_p \end{array} \right].$$
This partitioning by *vertical* lines of the right-hand factor in the matrix product $\mathbf{AB}$ does not place any constraints on how $\mathbf{A}$ is partitioned, and so we may write
$$\mathbf{AB} = \mathbf{A}\left[\begin{array}{c|c|c|c} \mathbf{B}_1 & \mathbf{B}_2 & \cdots & \mathbf{B}_p \end{array} \right] = \left[\begin{array}{c|c|c|c} \mathbf{AB}_1 & \mathbf{AB}_2 & \cdots & \mathbf{AB}_p \end{array} \right].$$
That is, for each $j = 1, 2, \ldots, p$, the j^{th} column of $\mathbf{AB}$ is obtained by left-multiplying the j^{th} column of $\mathbf{B}$ by $\mathbf{A}$.

Having made that observation, let us consider more carefully what happens when $\mathbf{A}$—suppose it has n columns $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$—multiplies a vector $\mathbf{v} \in \mathbb{R}^n$. (Note that each $\mathbf{B}_j$ is just such a vector.) Blocking $\mathbf{A}$ by columns, we have
$$\mathbf{Av} = \left[\begin{array}{c|c|c|c} \mathbf{A}_1 & \mathbf{A}_2 & \cdots & \mathbf{A}_n \end{array} \right] \left[\begin{array}{c} v_1 \\ \hline v_2 \\ \hline \vdots \\ \hline v_n \end{array} \right] = v_1\mathbf{A}_1 + v_2\mathbf{A}_2 + \cdots + v_n\mathbf{A}_n.$$
That is, the matrix-vector product $\mathbf{Av}$ is simply a linear combination of the columns of $\mathbf{A}$, with the scalars multiplying these columns taken (in order, from top to bottom) from $\mathbf{v}$. The implication for the matrix product $\mathbf{AB}$ is that each of its columns $\mathbf{AB}_j$ is a linear combination of the columns of $\mathbf{A}$, with coefficients taken from the j^{th} column of $\mathbf{B}$.

4. **Linear combinations of rows of B**. In the previous interpretation of matrix multiplication, we begin with a partitioning of **B** via vertical lines. If, instead, we begin with a partitioning of **A**, a matrix with m rows, via horizontal lines, we get

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_1 \\ \hline \mathbf{A}_2 \\ \hline \vdots \\ \hline \mathbf{A}_m \end{bmatrix} \mathbf{B} \; = \; \begin{bmatrix} \mathbf{A}_1\mathbf{B} \\ \hline \mathbf{A}_2\mathbf{B} \\ \hline \vdots \\ \hline \mathbf{A}_m\mathbf{B} \end{bmatrix}.$$

That is, the j^{th} row of the matrix product **AB** is obtained from left-multiplying the entire matrix **B** by the j^{th} *row* (considered as a submatrix) of **A**.

If **A** has n columns, then each $\mathbf{A}_j$ is a 1-by-n matrix. The effect of multiplying a 1-by-n matrix **V** by an n-by-p matrix **B**, using a blocking-by-row scheme for **B**, is

$$\mathbf{VB} \; = \; \left[\; v_1 \; \middle| \; v_2 \; \middle| \; \cdots \; \middle| \; v_n \; \right] \begin{bmatrix} \mathbf{B}_1 \\ \hline \mathbf{B}_2 \\ \hline \cdots \\ \hline \mathbf{B}_n \end{bmatrix} \; = \; v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + \cdots + v_n\mathbf{B}_n \; ,$$

a linear combination of the rows of **B**. Thus, for each $j = 1, \ldots, m$, the j^{th} row $\mathbf{A}_j\mathbf{B}$ of the matrix product **AB** is a linear combination of the rows of **B**, with coefficients taken from the j^{th} row of **A**.

1.2.2 Systems of linear equations

Motivated by **Viewpoint 3** concerning matrix multiplication—in particular, that

$$\mathbf{Ax} \; = \; x_1\mathbf{A}_1 + x_2\mathbf{A}_2 + \cdots + x_n\mathbf{A}_n \; ,$$

where $\mathbf{A}_1, \ldots, \mathbf{A}_n$ are the columns of a matrix **A** and $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$—we make the following definition.

Definition 1.2.2. Suppose $\mathbf{A} = \left[\; \mathbf{A}_1 \; \middle| \; \mathbf{A}_2 \; \middle| \; \cdots \; \middle| \; \mathbf{A}_n \; \right]$, where each submatrix $\mathbf{A}_j$ consists of a single column (so **A** has n columns in all). The set of all possible linear combinations of these columns (also known as $\text{span}(\{\mathbf{A}_1, \ldots, \mathbf{A}_n\})$)

$$\{c_1\mathbf{A}_1 + c_2\mathbf{A}_2 + \cdots + c_n\mathbf{A}_n \,|\, c_1, c_2, \ldots, c_n \in \mathbb{R}\} \; ,$$

is called the **column space** of **A**. We use the symbol $\text{ran}(\mathbf{A})$ to denote the column space.

The most common problem in linear algebra (and the one we seek in this course to understand most completely) is the one of solving m linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\ \vdots \qquad\qquad &\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &= b_m \end{aligned} \tag{1.2}$$

in the n unknowns $x_1, \ldots, x_n$. If one uses the coefficients and unknowns to build a **coefficient matrix** and vectors

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

then by our definitions of matrix equality and multiplication, the system (1.2) may be expressed more concisely as the matrix equation

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,, \tag{1.3}$$

where the vector **b** is known and **x** is to be found. Given **Viewpoint 3** for conceptualizing matrix multiplication above, problem (1.3) really presents two questions to be answered:

(I) Is **b** in the column space of **A** (i.e., is (1.3) solvable)?

(II) If it is, then what are the possible n-tuples $\mathbf{x} = (x_1, \ldots, x_n)$ of coefficients so that the linear combination

$$x_1\mathbf{A}_1 + x_2\mathbf{A}_2 + \cdots + x_n\mathbf{A}_n$$

of the columns of **A** equals **b**?

When the number m of equations and the number n of unknowns in system (1.2) are equal, it is often the case that there is one unique answer for each of the variables x_i (or, equivalently, one unique vector **x** satisfying (1.3)). Our main goal in the linear algebra component of this course is to understand completely when (1.3) is and is not solvable, how to characterize solutions when it is, and what to do when it is not.

One special instance of the case $m = n$ is when **A** is nonsingular. In this case, if $\mathbf{A}^{-1}$ is known, then the answer to question (I) is an immediate "yes". Moreover, one may obtain the (unique) solution of (1.3) (thus answering question (II)) via left-multiplication by $\mathbf{A}^{-1}$:

$$\begin{aligned} \mathbf{A}\mathbf{x} = \mathbf{b} \quad &\Rightarrow \quad \mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \\ &\Rightarrow \quad \mathbf{I}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \\ &\Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}\,. \end{aligned}$$

Important Note: It is *never* valid to talk about *dividing by a matrix* (so *not by a vector either*)! One speaks, instead, of multiplying by the inverse matrix, when that exists. It is, moreover, extremely important to pay attention to which side of an expression you wish to multiply by that inverse. Often placing it on the wrong side yields a nonsensical mathematical expression!

In practical settings, however, $\mathbf{A}^{-1}$ must first be found (if, indeed, it exists!) before we can use it to solve the matrix problem. Despite the availability of the Octave function `inv()`, finding the inverse of a matrix is a very inefficient thing to do computationally, and quite impossible when $\mathbf{A}^{-1}$ does not exist. In the Section 1.4 we will look at *Gaussian elimination* as a procedure for solving linear systems of equations. Gaussian elimination serves as a foundation for the *LU*-factorization, which supplies us with a comprehensive method for solving $\mathbf{Ax} = \mathbf{b}$ whenever the matrix problem *can* be solved (even in cases where $\mathbf{A}^{-1}$ does not exist).

1.3 Affine transformations of $\mathbb{R}^2$

Suppose $\mathbf{A}$ is an m-by-n matrix. When we left-multiply a vector $\mathbf{v} \in \mathbb{R}^n$ by such a matrix $\mathbf{A}$, the result is a vector $\mathbf{Av} \in \mathbb{R}^m$. In this section we will focus upon functions which take inputs $\mathbf{v} \in \mathbb{R}^n$ and produces outputs $\mathbf{Av} \in \mathbb{R}^m$. A function such as this could be given a name, but we will generally avoid doing so, referring to it as "the function $\mathbf{v} \mapsto \mathbf{Av}$". When we wish to be explicit about the type of objects the input and output are, we might write "$(\mathbf{v} \mapsto \mathbf{Av})\colon \mathbb{R}^n \to \mathbb{R}^m$", which points out that the function $\mathbf{v} \mapsto \mathbf{Av}$ maps objects from $\mathbb{R}^n$ (inputs) to objects from $\mathbb{R}^m$ (outputs). But if the reader is informed that $\mathbf{A}$ is an m-by-n matrix, he should already be aware that inputs/outputs to and from the function $\mathbf{v} \mapsto \mathbf{Av}$ are in $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively.

In this subsection $\mathbf{A}$ will be understood to be a 2-by-2 matrix. Assuming this, it is the case that $(\mathbf{v} \mapsto \mathbf{Av})\colon \mathbb{R}^2 \to \mathbb{R}^2$. We wish to focus our attention on the action of such a function on the entire plane of vectors for various types of 2-by-2 matrices $\mathbf{A}$.

1. **Rotations of the plane**. Our first special family of matrices are those of the form

$$\mathbf{A} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \tag{1.4}$$

for $\alpha \in \mathbb{R}$. We know that points in the plane may be specified using polar coordinates, so any vector $\mathbf{v} \in \mathbb{R}^2$ may be expressed as $\mathbf{v} = (r\cos\theta, r\sin\theta)$, where (r, θ) is a polar representation of the terminal point of $\mathbf{v}$. To the see the action of $\mathbf{A}$ on a typical $\mathbf{v}$, note that

$$\mathbf{A}\mathbf{v} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} = r \begin{bmatrix} \cos\alpha\cos\theta - \sin\alpha\sin\theta \\ \sin\alpha\cos\theta + \cos\alpha\sin\theta \end{bmatrix}$$
$$= \begin{bmatrix} r\cos(\alpha+\theta) \\ r\sin(\alpha+\theta) \end{bmatrix}.$$

where we have employed several **angle sum formulas**[2] in the last equality. That is, for an input vector $\mathbf{v}$ with terminal point (r, θ), the output $\mathbf{A}\mathbf{v}$ is a vector with terminal point $(r, \alpha + \theta)$. The output is the same distance r from the origin as the input, but has been rotated about the origin through an angle α. Thus, for matrices of the form (1.4), the function $\mathbf{v} \mapsto \mathbf{A}\mathbf{v}$ rotates the entire plane counterclockwise (for positive α) about the origin through an angle α. Of course, the inverse matrix would reverse this process, and hence it must be

$$\mathbf{A}^{-1} = \begin{bmatrix} \cos(-\alpha) & -\sin(-\alpha) \\ \sin(-\alpha) & \cos(-\alpha) \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}.$$

2. **Reflections across a line containing the origin**.

   First notice that, when

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \text{then} \qquad \mathbf{A}\mathbf{v} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ -v_2 \end{bmatrix}.$$

   Thus, for this special matrix $\mathbf{A}$, $\mathbf{v} \mapsto \mathbf{A}\mathbf{v}$ maps points in the plane to their reflections through the x-axis.

   Now let $\mathbf{u} = (\cos\theta, \sin\theta)$ (i.e., $\mathbf{u}$ is a unit vector). Every line in the plane containing the origin may be expressed as a one-parameter family $L = \{t\mathbf{u} \mid t \in \mathbb{R}\}$ of multiples of $\mathbf{u}$ where θ has been chosen (fixed, hence fixing $\mathbf{u}$ as well) to be an angle the line makes with the positive x-axis. (Said another way, each line in $\mathbb{R}^2$ containing $\mathbf{0}$ is the *linear span* of some unit vector.) We can see reflections across the line L as a series of three transformations:

   i) rotation of the entire plane through an angle $(-\theta)$, so as to make the line L correspond to the x-axis,
   ii) reflection across the x-axis, and then
   iii) rotation of the plane through an angle θ, so that the x-axis is returned back to its original position as the line L.

[2] These trigonometric identities appear, for instance, in the box marked equation (4) on p. 26 of **University Calculus**, by Hass, Weir and Thomas.

These three steps may be affected through successive multiplications of matrices (the ones on the left below) which can be combined into one:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix}, \tag{1.5}$$

where $\alpha = 2\theta$. That is, a matrix of the form

$$\mathbf{A} = \begin{bmatrix} \cos\alpha & \sin\alpha \\ \sin\alpha & -\cos\alpha \end{bmatrix} \tag{1.6}$$

will map points in the plane to their mirror images across a line that makes an angle $(\alpha/2)$ with the positive x-axis.

3. **Scaling relative to the origin: perpendicular lines case**. Suppose we wish to rescale vectors so that the x-coordinate of terminal points is multiplied by the quantity s, while the y-coordinates are multiplied by t. It is easy to see that multiplication by the matrix
$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \tag{1.7}$$
would achieve this, since
$$\begin{bmatrix} s & 0 \\ 0 & t \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} sv_1 \\ tv_2 \end{bmatrix}.$$
It is, in fact, only slightly more complicated to do this in directions specified by any pair of perpendicular lines, not just the x- and y-axes. It is left as an exercise to figure out how.

4. **Translations of the plane**. What we have in mind here is, for some given vector $\mathbf{w}$, to translate every vector $\mathbf{v} \in \mathbb{R}^2$ to the new location $\mathbf{v} + \mathbf{w}$. It is an easy enough mapping, described simply in symbols by $(\mathbf{v} \mapsto \mathbf{v} + \mathbf{w})$. Yet, perhaps surprisingly, it is the one type of **affine transformation** (most *affine transformations* of the plane are either of the type 1–4 described here, or combinations of these) which cannot be achieved through left-multiplication by a 2-by-2 matrix. That is, for a given $\mathbf{w} \neq \mathbf{0}$ in $\mathbb{R}^2$, there is no 2-by-2 matrix $\mathbf{A}$ such that $\mathbf{A}\mathbf{v} = \mathbf{v} + \mathbf{w}$.

   When this observation became apparent to computer programmers writing routines for motion in computer graphics, mathematicians working in the area of *projective geometry* had a ready answer: **homogeneous coordinates**. The idea is to embed vectors from $\mathbb{R}^2$ into $\mathbb{R}^3$. A vector $\mathbf{v} = (v_1, v_2) \in \mathbb{R}^2$ is associated with the vector $\tilde{\mathbf{v}} = (v_1, v_2, 1)$ which lies on the plane $z = 1$ in $\mathbb{R}^3$. Say we want to translate all vectors $\mathbf{v} \in \mathbb{R}^2$ by the (given) vector $\mathbf{w} = (a, b)$. We can form the 3-by-3 matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix}$$

and multiply $\tilde{\mathbf{v}}$ (not $\mathbf{v}$ itself) by $\mathbf{A}$. Then the translated vector $\mathbf{v} + \mathbf{w}$ is obtained from $\mathbf{A}\tilde{\mathbf{v}}$ by keeping just the first two coordinates.

We finish this section with two comments. First, we note that even though we needed homogeneous coordinates only for the translations described in 4 above, it is possible to carry out the transformations of 1–3 while in homogeneous coordinates as well. This is possible because we may achieve the appropriate analog of any of the transformations 1–3 by multiplying $\tilde{\mathbf{v}}$ by a 3-by-3 block matrix

$$\mathbf{B} = \left[\begin{array}{c|c} \mathbf{A} & 0 \\ \hline 0 & 1 \end{array}\right],$$

where $\mathbf{A}$ is a 2-by-2 block as described in 1–3 above. You should convince yourself, reviewing what we have said about multiplying matrices in blocks (**Viewpoint 1** in Section 1.2) as needed, that, if $\tilde{\mathbf{v}}$ is the homogeneous coordinates version of $\mathbf{v}$, then $\mathbf{B}\tilde{\mathbf{v}}$ is the homogeneous coordinates version of $\mathbf{A}\mathbf{v}$.

The other note to mention is that, while our discussion has been entirely about affine transformations on $\mathbb{R}^2$, all of the types we have discussed in 1–4 have counterpart transformations on $\mathbb{R}^n$, when $n > 2$. For instance, if you take a plane in $\mathbb{R}^3$ containing the origin and affix to it an axis of rotation passing perpendicularly to that plane through the origin then, for a given angle α, there is a 3-by-3 matrix $\mathbf{A}$ such that rotations of points in $\mathbb{R}^3$ through the angle α about this axis are achieved via the function $(\mathbf{v} \mapsto \mathbf{A}\mathbf{v})\colon \mathbb{R}^3 \to \mathbb{R}^3$. The 3D analogs of transformations 2–3 may be similarly achieved via multiplication by an appropriate 3-by-3 matrix. Only transformation 4 cannot, requiring, as before, that we pass into one higher dimension (homogeneous coordinates for $\mathbb{R}^3$) and multiply by an appropriate 4-by-4 matrix.

1.4 Gaussian Elimination

We have noted that linear systems of (algebraic) equations are representable in matrix form. We now investigate the solution of such systems. You have, now doubt, spent a fair amount of time in previous courses (MATH 231, most recently) learning how to do this, and it is hoped that at least some portion of this section is review. We begin with a definition.

Definition 1.4.1. An m-by-n matrix $\mathbf{A} = (a_{ij})$ is said to be **upper triangular** if $a_{ij} = 0$ whenever $i > j$—that is, when all entries below the main diagonal are zero. When all entries above the main diagonal are zero (i.e., $a_{ij} = 0$ whenever $i < j$), the $\mathbf{A}$ is said to be **lower triangular**. A *square* matrix that is both upper and lower triangular is called a **diagonal matrix**.

The system of linear equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \end{aligned}$$

has two equations in two unknowns. The variables x_1 and x_2 represent *potential* **degrees of freedom**, while the equations represent **contraints**. The solution(s) are points of intersection between two lines, and as such there may be none, precisely one, or infinitely many.

In real settings, there are usually far more than two variables and equations, and the apparent number of constraints need not be the same as the number of variables. We would like a general algorithm which finds solutions to such systems of equations when solutions exist. We will develop a method called **Gaussian elimination** and, in the process, look at examples of various types of scenarios which may arise.

1.4.1 Examples of the method

Example 1.4.1

We begin simply, with a system of 2 equations in 2 unknowns. Suppose we wish to solve

$$\begin{aligned} 7x + 3y &= 1 \\ 3y &= -6 \end{aligned} \qquad \text{or} \quad \mathbf{Ax} = \mathbf{b}\,, \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} 7 & 3 \\ 0 & 3 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 1 \\ -6 \end{bmatrix}.$$

The problem is very easy to solve using **backward substitution**—that is, solving the equation in y alone,

$$3y = -6 \qquad \Rightarrow \qquad y = -2\,,$$

which makes the appearence of y in the other equation no problem:

$$7x + 3(-2) = 1 \qquad \Rightarrow \qquad x = \frac{1}{7}(1+6) = 1\,.$$

We have the unique solution $(1, -2)$. Notice that we can solve by backward substitution because the **coefficient matrix A** is *upper triangular*.

Example 1.4.2

The system

$$\begin{aligned} 2x - 3y &= 7 \\ 3x + 5y &= 1 \end{aligned} \qquad \text{or, in matrix form} \qquad \begin{bmatrix} 2 & -3 \\ 3 & 5 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 1 \end{bmatrix},$$

is only mildly more difficult, though we cannot immediately resort to backward substitution as in the last example. Let us proceed by making this problem like that from the previous example. Perhaps we might leave the top equation alone, but alter the bottom one by adding $(-2/3)$ multiples of the top equation to it. In what follows, we will employ the approach of listing the algebraic equations on the left along with a matrix form of them on the right. Instead of repeating the full matrix equation, we will abbreviate it with a matrix called an **augmented matrix** that lists only the constants in the problem. By adding $(-3/2)$ copies of the top equation to the bottom, our original system

$$\begin{aligned} 2x - 3y &= 7 \\ 3x + 5y &= 1 \end{aligned} \qquad \text{or} \qquad \left[\begin{array}{cc|c} 2 & -3 & 7 \\ 3 & 5 & 1 \end{array}\right],$$

becomes

$$\begin{aligned} 2x - 3y &= 7 \\ (19/2)y &= -19/2 \end{aligned} \qquad \text{or} \qquad \left[\begin{array}{cc|c} 2 & -3 & 7 \\ 0 & 19/2 & -19/2 \end{array}\right].$$

Now, as the (new) coefficient matrix (the part of the matrix lying to the left of the dividing line) is upper triangular, we may finish solving our system using backward substitution:

$$\frac{19}{2}y = -\frac{19}{2} \qquad \Rightarrow \qquad y = -1\,,$$

so

$$2x - 3(-1) = 7 \qquad \Rightarrow \qquad x = 7 + 3 = 10\,.$$

Again, we have a unique solution, the point $(10, -1)$.

Let's pause for some observations. Hopefully it is clear that an upper triangular system is desirable so that backward substitution may be employed to find appropriate values for the variables. When we did not immediately have that in Example 1.4.2, we added a multiple of the first equation to the second to make it so. This is listed below as number 3 of the elementary operations which are allowed when carrying out **Gaussian elimination**, the formal name given to the process of reducing a system of linear equations to a special form which is then easily solved by substitution. Your intuition about solving equations should readily confirm the validity of the other two elementary operations.

Elementary Operations of Gaussian Elimination

1. Multiply a row by a nonzero constant.
2. Exchange two rows.
3. Add a multiple of one row to another.

And what is this special form at which Gaussian elimination aims? It is an upper triangular form, yet not merely that. It is is a special form known as **echelon form** where the first nonzero entries in each row, below depicted by 'p's and asterisks, have a stair-step appearance to them:

$$\begin{bmatrix} p & * & * & * & * & * & * & * & * & \cdots \\ 0 & p & * & * & * & * & * & * & * & \cdots \\ 0 & 0 & 0 & p & * & * & * & * & * & \cdots \\ 0 & 0 & 0 & 0 & p & * & * & * & * & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & p & * & \cdots \\ \vdots & & & & & & & & & \end{bmatrix}$$

In fact, there may also be zeros where the asterisks appear. The 'p's, however, called **pivots**, play a special role in the backward substitution part of the solution process, a role that requires them to be nonzero. If you look back at the pivots in our first two examples (the numbers 7 and 3 in Example 1.4.1; 2 and (19/2) in Example 1.4.2), you will see why they must be nonzero—when we get to the backward substitution stage, we divide through by these pivots. But, as the echelon form above depicts, the number of rows and columns of a matrix does not tell you just how many pivots you will have. The pivot in one row may be followed by a pivot in the next row (progressing downward) which is just one column to the right; but, that next pivot down may also skip *several* columns to the right. The final pivot may not even be in the right-most column. One thing for sure is that the pivots must progress to the right as we move down the rows; all entries below each pivot must be zero.

It is usually necessary to perform a sequence of elementary row operations on a given matrix **A** before arriving at an echelon form **R** (another m-by-n matrix). It would violate our definition of *matrix equality* to call **A** and **R** "equal". Instead, we might say that **R** is an echelon form for **A** (not "the" echelon form for **A**, as there is more than one), or that **A** and **R** are **row equivalent**.

We turn now to examples of the process for larger systems, illustrating some different scenarios in the process, and some different types of problems we might solve using it. After stating the original problem, we will carry out the steps depicting only augmented matrices. Since the various augmented matrices are not equal to their predecessors (in the sense of matrix equality), but do represent equivalent systems of equations (i.e., systems of equations which have precisely the same solutions), we will separate them with the ~ symbol.

Example 1.4.3

Find all solutions to the linear system of equations

$$\begin{aligned} 2x + y - z &= 3\,, \\ 4x + 2y + z &= 9\,. \end{aligned}$$

As these two equations both represent planes in 3-dimensional space, one imagines that there may either be no solutions, or infinitely many. We perform Gaussian elimination:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 3 \\ 4 & 2 & 1 & 9 \end{array}\right] \quad \begin{array}{c} -2\mathbf{r}_1 + \mathbf{r}_2 \to \mathbf{r}_2 \\ \sim \end{array} \quad \left[\begin{array}{ccc|c} 2 & 1 & -1 & 3 \\ 0 & 0 & 3 & 3 \end{array}\right]$$

The latter matrix is in echelon form. It has pivots, 2 in the 1st column and 3 in the 3rd column. Unlike previous examples, these pivots are separated by a column which has no pivot. This 2nd column continues to correspond to y-terms in the system, and the absence of a pivot in this column means that y is a **free variable**. It has no special value, providing a **degree of freedom** within solutions of the system. The **pivot columns** (i.e., the ones with pivots), correspond to the x- and z-terms in the system—the **pivot variables**; their values are either fixed, or contingent on the value(s) chosen for the free variable(s). The echelon form corresponds to the system of equations (equivalent to our original system)

$$\begin{aligned} 2x + y - z &= 3 , \\ 3z &= 3 . \end{aligned}$$

Clearly, the latter of these equations implies $z = 1$. Since y is free, we do not expect to be able to solve for it. Nevertheless, if we plug in our value for z, we may solve for x in terms of the free variable y:

$$x = \frac{1}{2}(3 + 1 - y) = 2 - \frac{1}{2}y .$$

Thus, our solutions (there are infinitely many) are

$$(x, y, z) = (2 - y/2, y, 1) = (2, 0, 1) + t(-1, 2, 0) ,$$

where $t = y/2$ may be any real number (since y may be any real number). Note that this set S of solutions traces out a line in 3D space.

■

Before the next example, we make another definition.

Definition 1.4.2. The *nullspace* of an m-by-n matrix $\mathbf{A}$ consists of those vectors $\mathbf{x} \in \mathbb{R}^n$ for which $\mathbf{A}\mathbf{x} = \mathbf{0}$. That is,

$$\text{null}(\mathbf{A}) := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{0}\} .$$

A related problem to the one in the last example is the following one.

Example 1.4.4

Find the nullspace of the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & -1 \\ 4 & 2 & 1 \end{bmatrix}.$$

That is, we are asked to find those vectors $\mathbf{v} \in \mathbb{R}^3$ for which $\mathbf{Av} = \mathbf{0}$ or, to put it in a way students in a high school algebra class might understand, to solve

$$\begin{aligned} 2x + y - z &= 0 \,, \\ 4x + 2y + z &= 0 \,. \end{aligned}$$

Mimicking our work above, we have

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 0 \\ 4 & 2 & 1 & 0 \end{array}\right] \quad \begin{array}{c} -2\mathbf{r}_1 + \mathbf{r}_2 \to \mathbf{r}_2 \\ \sim \end{array} \quad \left[\begin{array}{ccc|c} 2 & 1 & -1 & 0 \\ 0 & 0 & 3 & 0 \end{array}\right],$$

which corresponds to the system of equations

$$\begin{aligned} 2x + y - z &= 0 \,, \\ 3z &= 0 \,. \end{aligned}$$

Now, we have $z = 0$; y is again a free variable, so $x = -\frac{1}{2}y$. Thus, our solutions (again infinitely many) are

$$(x, y, z) \;=\; (-y/2, y, 0) \;=\; t(-1, 2, 0) \,,$$

where $t = y/2$ may be any real number (since y may be any real number). Note that, like the solutions in Example 1.4.3, this set of solutions—all scalar multiples of the vector $(-1, 2, 0)$—traces out a line. This line is parallel to that of the previous example, but unlike the other, it passes through origin (or zero vector).

Compare the original systems of equations and corresponding solutions of Examples 1.4.3 and 1.4.4. Employing language introduced in MATH 231, the system of equations in Example 1.4.4 is said to be **homogeneous** as its right-hand side is the zero vector. Its solutions form a line through the origin, a line parametrized by t. Since the vector on the right-hand side of Example 1.4.3 is $(9, 3)$ (*not* the zero vector), that system is **nonhomogeneous**. Its solutions form a line as well, parallel to the line for the corresponding homogeneous system of Example 1.4.4, but translated away from the origin by the vector $(2, 0, 1)$ which, itself, is a solution of the nonhomogeneous system of Example 1.4.3. This same thing happens in the solution of linear ODEs: when faced with a nonhomogeneous n^{th}-order ODE (just an ODE to solve, not an initial-value problem), one finds the general solution of the corresponding homogeneous problem, an n-parameter family of solutions, and then adds to it a particular solution of the nonhomogenous problem.

We finish with an example of describing the column space of a matrix.

Example 1.4.5

Find the column (or range) space of the matrix

$$\mathbf{A} \;=\; \begin{bmatrix} 2 & 3 & 0 & -1 \\ 1 & 0 & 3 & 1 \\ -3 & -5 & 1 & 2 \\ 1 & 0 & 3 & 1 \end{bmatrix}.$$

A quick description of the column space of $\mathbf{A}$ is to say it is

$$\text{span}(\{(2,1,-3,1),(3,0,-5,0),(0,3,1,3),(-1,1,2,1)\}) .$$

Since that is so easy, let's see if we can give a more minimal answer. After all, there may be redundancy in these columns.

Our plan of attack will be to assume that $\mathbf{b} = (b_1, b_2, b_3, b_4)$ is in ran($\mathbf{A}$) and solve $\mathbf{Ax} = \mathbf{b}$ via elimination as before. We have

$$\left[\begin{array}{cccc|c} 2 & 3 & 0 & -1 & b_1 \\ 1 & 0 & 3 & 1 & b_2 \\ -3 & -5 & 1 & 2 & b_3 \\ 1 & 0 & 3 & 1 & b_4 \end{array}\right] \quad \begin{array}{c} \mathbf{r}_1 \leftrightarrow \mathbf{r}_2 \\ \sim \end{array} \quad \left[\begin{array}{cccc|c} 1 & 0 & 3 & 1 & b_2 \\ 2 & 3 & 0 & -1 & b_1 \\ -3 & -5 & 1 & 2 & b_3 \\ 1 & 0 & 3 & 1 & b_4 \end{array}\right]$$

$$\begin{array}{c} \mathbf{r}_2 - 2\mathbf{r}_1 \to \mathbf{r}_2 \\ 3\mathbf{r}_1 + \mathbf{r}_3 \to \mathbf{r}_3 \\ \sim \\ \mathbf{r}_4 - \mathbf{r}_1 \to \mathbf{r}_4 \end{array} \quad \left[\begin{array}{cccc|c} 1 & 0 & 3 & 1 & b_2 \\ 0 & 3 & -6 & -3 & b_1 - 2b_2 \\ 0 & -5 & 10 & 5 & 3b_2 + b_3 \\ 0 & 0 & 0 & 0 & b_4 - b_2 \end{array}\right]$$

$$\begin{array}{c} (5/3)\mathbf{r}_2 + \mathbf{r}_3 \to \mathbf{r}_3 \\ \sim \end{array} \quad \left[\begin{array}{cccc|c} 1 & 0 & 3 & 1 & b_2 \\ 0 & 3 & -6 & -3 & b_1 - 2b_2 \\ 0 & 0 & 0 & 0 & (5/3)b_1 - (1/3)b_2 + b_3 \\ 0 & 0 & 0 & 0 & b_4 - b_2 \end{array}\right] .$$

For determining the range space, we focus on the last two rows which say

$$0x_1 + 0x_2 + 0x_3 + 0x_4 \;=\; \frac{5}{3}b_1 - \frac{1}{3}b_2 + b_3 \qquad \text{or} \qquad 0 \;=\; 5b_1 - b_2 + 3b_3 ,$$

and

$$0 \;=\; b_4 - b_2 .$$

These are the constraints which must be met by the components of $\mathbf{b}$ in order to be in ran($\mathbf{A}$). There are two constraints on four components, so two of those components are "free". We choose $b_4 = t$, which means $b_2 = t$ as well. Thus

$$b_1 \;=\; \frac{1}{5}(t - 3b_3) .$$

If we take $b_3 = -5s - 3t$, then $b_1 = 2t + 3s$. (Admittedly, this is a strange choice for b_3. However, even if t is fixed on some value, the appearance of the new parameter s makes it possible for b_3 to take on any value.) So, we have that any $\mathbf{b} \in$ ran($\mathbf{A}$) must take the form

$$\mathbf{b} \;=\; \begin{bmatrix} 2t + 3s \\ t \\ -3t - 5s \\ t \end{bmatrix} \;=\; t\begin{bmatrix} 2 \\ 1 \\ -3 \\ 1 \end{bmatrix} + s\begin{bmatrix} 3 \\ 0 \\ -5 \\ 0 \end{bmatrix},$$

where s, t are arbitrary real numbers. That is,

$$\operatorname{ran}(\mathbf{A}) = \operatorname{span}(\{(2,1,-3,1),(3,0,-5,0)\}) .$$

When we look back at the original matrix, these two vectors in the spanning set for ran(**A**) are precisely the first two columns of **A**. Thus, while we knew before we started that ran(**A**) was spanned by the columns of **A**, we now know just the first two columns suffice.

We will return to this problem of finding ran(**A**) in the next chapter, in which we will see there is an easier way to determine a set of vectors that spans the column space.

1.4.2 Finding an inverse matrix

What would you do if you had to solve

$$\mathbf{A}\mathbf{x} = \mathbf{b}_1 \qquad \text{and} \qquad \mathbf{A}\mathbf{x} = \mathbf{b}_2 ,$$

where the matrix **A** is the same but $\mathbf{b}_1 \neq \mathbf{b}_2$? Of course, one answer is to augment the matrix **A** with the first right-hand side vector $\mathbf{b}_1$ and solve using Gaussian elimination. Then, repeat the process with $\mathbf{b}_2$ in place of $\mathbf{b}_1$. But a close inspection of the process shows that the row operations you perform on the augmented matrix to reach row echelon form are dictated by the entries of **A**, independent of the right-hand side. Thus, one could carry out the two-step process we described more efficiently if one augmented **A** with two extra columns, $\mathbf{b}_1$ and $\mathbf{b}_2$. That is, working with the augmented matrix $[\mathbf{A}|\mathbf{b}_1\, \mathbf{b}_2]$, use elementary (row) operations 1.–3. until the part to the left of the augmentation bar is in row echelon form. This reduced augmented matrix would take the form $[\mathbf{R}|\mathbf{c}_1\, \mathbf{c}_2]$, where **R** is a row echelon form. Then we could use backward substitution separately on $[\mathbf{R}|\mathbf{c}_1]$ and $[\mathbf{R}|\mathbf{c}_2]$ to find solutions to the two matrix problems. Of course, if you have more than 2 matrix problems (with the same coefficient matrix), you tack on more than 2 columns.

This idea is key to findin the inverse of an n-by-n matrix **A**, when it exists. Let us denote the standard vectors in $\mathbb{R}^n$ by $\mathbf{e}_1 = (1, 0, \ldots, 0)$, $\mathbf{e}_2 = (0, 1, \ldots, 0), \ldots, \mathbf{e}_n = (0, 0, \ldots, 1)$. These are the columns of the identity matrix. We know the inverse **B** satisfies

$$\mathbf{A}\mathbf{B} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \cdots & \mathbf{e}_n \\ \downarrow & \downarrow & \downarrow & & \downarrow \end{bmatrix}.$$

Let $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n$ denote the columns of **B**. Then an equivalent problem to finding the matrix **B** is to solve the n problems

$$\mathbf{A}\mathbf{b}_1 = \mathbf{e}_1 , \quad \mathbf{A}\mathbf{b}_2 = \mathbf{e}_2 , \quad \ldots \quad \mathbf{A}\mathbf{b}_n = \mathbf{e}_n ,$$

for the unknown vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$. By the method above, we would augment $\mathbf{A}$ with the full n-by-n identity matrix and perform elementary operations until the part to the left of the augmentation line was in row echelon form. That is, we would reduce $[\mathbf{A}|\mathbf{I}]$ to $[\mathbf{R}|\mathbf{C}]$, where $\mathbf{C}$ is an n-by-n matrix. (Note that if $\mathbf{R}$ does not have n pivots, then $\mathbf{A}$ is singular.) We can then solve each of the problems

$$\mathbf{R}\mathbf{b}_1 = \mathbf{c}_1, \quad \mathbf{R}\mathbf{b}_2 = \mathbf{c}_2, \quad \ldots \quad \mathbf{R}\mathbf{b}_n = \mathbf{c}_n$$

using backward substitution, and arrive at the inverse matrix $\mathbf{B}$ putting the solutions together:

$$\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}.$$

That's a clever method, and it's pretty much along the lines of how an inverse matrix is found when it is really desired. However, in most cases, $\mathbf{A}^{-1}$ is just an intermediate find on the way to solving a matrix problem $\mathbf{A}\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$. If there is a more efficient way to find $\mathbf{x}$, one requiring fewer calculations, we would employ it instead. That is the content of the next section.

1.5 *LU* Factorization of a Matrix

We have three 'legal' elementary operations when using Gaussian elimination to solve the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$. We seek to put the matrix $\mathbf{A}$ in *echelon form* via a sequence of operations consisting of

1. multiplying a row by a nonzero constant.
2. exchanging two rows.
3. adding a multiple of one row to another.

You may have noticed that, at least in theory, reduction to echelon form may be accomplished without ever employing operation 1. Let us focus on operation 3 for the moment. In practice the multiplier is always some nonzero constant β. Moreover, in Gaussian elimination we are always adding a multiple of a row to some other row which is *below* it. For a fixed β, let $\mathbf{E}_{ij}$ be the matrix that only differs from the m-by-m identity matrix in that its $(i, j)^{\text{th}}$ entry is β. We call $\mathbf{E}_{ij}$ an **elementary matrix**. A user-defined function written in Octave code that returns such a matrix might look like the following:

```
function emat = elementary(m, i, j, val)
  emat = eye(m,m);
  emat(i,j) = val;
endfn
```

In Exercise 1.35 you are asked to show that

$$\mathbf{E}_{ij}\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{i-1,1} & a_{i-1,2} & \cdots & a_{i-1,n} \\ a_{i,1} + \beta a_{j,1} & a_{i,2} + \beta a_{j,2} & \cdots & a_{i,n} + \beta a_{j,n} \\ a_{i+1,1} & a_{i+1,2} & \cdots & a_{i+1,n} \\ \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}.$$

In other words, pre-multiplication by $\mathbf{E}_{ij}$ performs an instance of operation 3 on the matrix $\mathbf{A}$, replacing row i with (row i) $+\beta$ (row j). Now, suppose $a_{1,1} \neq 0$. If we now denote $\mathbf{E}_{i1}$ by $\mathbf{E}_{i1}(\beta)$ (in order to make explicit the value being inserted at the $(i,1)^{\text{th}}$ position), then $\mathbf{E}_{21}(-a_{2,1}/a_{1,1})\mathbf{A}$ is a matrix whose entry in its $(2,1)^{\text{th}}$ position has been made to be zero. More generally,

$$\mathbf{E}_{n1}\left(\frac{-a_{n,1}}{a_{1,1}}\right)\cdots\mathbf{E}_{3,1}\left(\frac{-a_{3,1}}{a_{1,1}}\right)\mathbf{E}_{2,1}\left(\frac{-a_{2,1}}{a_{1,1}}\right)\mathbf{A}$$

is the matrix that results from retaining the first pivot of $\mathbf{A}$ and eliminating all entries below it. If our matrix $\mathbf{A}$ is such that no row exchanges occur during reduction to echelon form, then by a sequence of pre-multiplications by elementary matrices, we arrive at an upper-triangular matrix $\mathbf{U}$. We make the following observations:

- Each time we perform elementary operation 3 via pre-multiplication by an elementary matrix $\mathbf{E}_{ij}$, it is the case that $i > j$. Thus, the elementary matrices we use are lower-triangular.
- Each elementary matrix is invertible, and $\mathbf{E}_{ij}^{-1}$ is lower triangular when $\mathbf{E}_{ij}$ is. See Exercise 1.35.
- The product of lower triangular matrices is again lower triangular. See Exercise 1.37.

By these observations, when no row exchanges take place in the reduction of $\mathbf{A}$ to echelon form, we may amass the sequence of elementary matrices which achieve this reduction into a single matrix $\mathbf{M}$ which is lower-triangular. Let us denote the inverse of $\mathbf{M}$ by $\mathbf{L}$, also a lower-triangular matrix. Then

$$\mathbf{LM} = \mathbf{I}, \qquad \text{while} \qquad \mathbf{MA} = \mathbf{U},$$

where $\mathbf{U}$ is an upper-triangular matrix, an echelon form for $\mathbf{A}$. Thus,

$$\mathbf{A} = (\mathbf{LM})\mathbf{A} = \mathbf{L}(\mathbf{MA}) = \mathbf{LU},$$

which is called the *LU* **factorization of A**, and

$$\mathbf{Ax} = \mathbf{b} \qquad \Leftrightarrow \qquad \mathbf{LUx} = \mathbf{b}\,.$$

Let $\mathbf{y} = \mathbf{Ux}$, so that $\mathbf{Ly} = \mathbf{b}$. Since $\mathbf{L}$ is lower-triangular, we may solve for $\mathbf{y}$ by a process known as **forward substitution**. Once we have $\mathbf{y}$, we may solve for $\mathbf{Ux} = \mathbf{y}$ via backward substitution as in the previous section.

But let us not forget that the previous discussion was premised on the idea that no row exchanges take place in order to reduce $\mathbf{A}$ to echelon form. We are aware that, in some instances, row exchanges are absolutely necessary to bring a pivot into position. As it turns out, numerical considerations sometimes call for row exchanges even when a pivot would be in place without such an exchange. How does this affect the above discussion?

Suppose we can know in advance just which row exchanges will take place in reducing $\mathbf{A}$ to echelon form. With such knowledge, we can quickly write down an m-by-m matrix $\mathbf{P}$, called a **permutation matrix**, such that $\mathbf{PA}$ is precisely the matrix $\mathbf{A}$ except that all of those row exchanges have been carried out. For instance, if we ultimately want the 1st row of $\mathbf{A}$ to wind up as row 5, we make the the 5th row of $\mathbf{P}$ be $(1, 0, 0, \ldots, 0)$. More generally, if we want the i^{th} row of $\mathbf{A}$ to wind up as the j^{th} row, we make the j^{th} row of $\mathbf{P}$ have a 1 in the i^{th} column and zeros everywhere else. To illustrate this, suppose

$$\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Then, for any 4-by-n matrix $\mathbf{A}$, $\mathbf{PA}$ will be another 4-by-n whose 1^{st} row is equal to the 2^{nd} row of $\mathbf{A}$, whose 2^{nd} row equals the 4^{th} row of $\mathbf{A}$, whose 3^{rd} row equals the 3^{rd} row of $\mathbf{A}$, and whose 4^{th} row equals the 1^{st} row of $\mathbf{A}$.

Now, the full story about the *LU* decomposition can be told. There is a permutation matrix $\mathbf{P}$ such that $\mathbf{PA}$ will not need any row exchanges to be put into echelon form. It is this $\mathbf{PA}$ which has an *LU* decomposition. That is, $\mathbf{PA} = \mathbf{LU}$.

Example 1.5.1

In Octave, the following commands were entered with accompanying output:

```
octave:1> A = [0 -1 3 1; 2 -1 1 4; 1 3 1 -1];
octave:2> [L,U,P] = lu(A)
L =
   1.00000   0.00000   0.00000
   0.50000   1.00000   0.00000
   0.00000  -0.28571   1.00000

U =
   2.00000  -1.00000   1.00000   4.00000
   0.00000   3.50000   0.50000  -3.00000
   0.00000   0.00000   3.14286   0.14286
```

```
P =
   0   1   0
   0   0   1
   1   0   0
```

We will use it to solve the matrix equation $\mathbf{Ax} = \mathbf{b}$, with

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 3 & 1 \\ 2 & -1 & 1 & 4 \\ 1 & 3 & 1 & -1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} -1 \\ 14 \\ 1 \end{bmatrix}.$$

Since we have been given the *LU* decomposition for **PA**, we will use it to solve $\mathbf{PAx} = \mathbf{Pb}$—that is, solve

$$\begin{bmatrix} 2 & -1 & 1 & 4 \\ 1 & 3 & 1 & -1 \\ 0 & -1 & 3 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 14 \\ 1 \\ -1 \end{bmatrix}.$$

We first solve $\mathbf{Ly} = \mathbf{Pb}$, or[3]

$$\begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & -2/7 & 1 \end{bmatrix} \mathbf{y} = \begin{bmatrix} 14 \\ 1 \\ -1 \end{bmatrix}.$$

We call our manner of solving for **y forward substitution** because we find the components of **y** in forward order, y_1 then y_2 then y_3.

$$\begin{aligned} y_1 &= 14\,, \\ \frac{1}{2}y_1 + y_2 = 1 \qquad &\Rightarrow \qquad y_2 = -6\,, \\ -\frac{2}{7}y_2 + y_3 = -1 \qquad &\Rightarrow \qquad y_3 = -\frac{19}{7}\,, \end{aligned}$$

so $\mathbf{y} = (14, -6, -19/7)$. Now we solve $\mathbf{Ux} = \mathbf{y}$, or

$$\begin{bmatrix} 2 & -1 & 1 & 4 \\ 0 & 7/2 & 1/2 & -3 \\ 0 & 0 & 22/7 & 1/7 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 14 \\ -6 \\ -19/7 \end{bmatrix},$$

via backward substitution. The result is infinitely many solutions, all with the form

$$\mathbf{x} = \begin{bmatrix} 73/11 \\ -35/22 \\ -19/22 \\ 0 \end{bmatrix} + t \begin{bmatrix} -17/11 \\ 19/22 \\ -1/22 \\ 1 \end{bmatrix}, \qquad t \in \mathbb{R}\,.$$

[3] Asking Octave to display $7 * \mathbf{L}$ shows that this is an exact representation of **L**.

Of course, it is possible to automate the entire process—not just the part of finding the *LU*-factorization of **A**, but also the forward and backward substitution steps. And there are situations in which, for a given coefficient matrix **A**, a different kind of solution process for the matrix equation $\mathbf{Ax} = \mathbf{b}$ may, indeed, be more efficient than using the factorization $\mathbf{LU} = \mathbf{PA}$. The Octave command

```
octave-3.0.0:57> A \ b
ans =
   2.30569
   0.82917
  -0.99101
   2.80220
```

(with `A` and `b` defined as in the Example 1.5.1) is sophisticated enough to look over the matrix `A` and choose a suitable solution technique, producing a result. In fact, the solution generated by the command is one that lies along the line of solutions

$$\mathbf{x} = \left(\frac{73}{11}, -\frac{35}{22}, -\frac{19}{22}, 0\right) + t\left(-\frac{17}{11}, \frac{19}{22}, -\frac{1}{22}, 1\right), \qquad t \in \mathbb{R},$$

found in Example 1.5.1, one occurring when $t \doteq 2.8022$. This, however, reveals a shortcoming of the '`A \ b`' command. It can find a particular solution, but when multiple solutions exist, it cannot find them all.

1.6 Determinants and Eigenpairs

1.6.1 Determinants

In MATH 231 you were exposed, at least rudimentarily, to the idea of a **determinant of a matrix**. Determinants of n-by-n matrices can be calculated fairly easily by hand for n small. For instance, when $n = 1$, the value of the determinant is equal to the (single) entry in the matrix. For $n = 2$, the determinant of **A** (denoted either by $\det(\mathbf{A})$ or $|\mathbf{A}|$) is given by

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc .$$

It is possible even to memorize a direct process yielding the formula for the determinant of a 3-by-3 matrix

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{31}a_{22}a_{13} - a_{32}a_{23}a_{11} - a_{33}a_{21}a_{12} ,$$

though by the time $n \geq 4$ (and perhaps even when $n = 3$) most people will employ the iterative process known as *expansion of the determinant in cofactors* if they are going to calculate a determinant by hand.

We will not describe how to do cofactor expansion in these notes, as we will rely on software to calculate determinants for any square matrix with $n \geq 4$ columns. (In Octave the command that calculates the determinant of a matrix `A` is `det(A)`.) Instead, in this subsection we will provide some useful facts, some of which may be familiar to you from MATH 231, about determinants. They appear as a numbered list below, in no particular order of precedence.

Facts about determinants:

A. The idea of the determinant of a matrix does not extend to matrices which are non-square. We only talk about the determinant of square matrices.

B. If $\mathbf{A}$, $\mathbf{B}$ are square matrices having the same dimensions, then $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$.

C. The determinant of an upper- (or lower-) triangular matrix $\mathbf{A} = (a_{ij})$ is the product of its diagonal elements. That is, $\det(\mathbf{A}) = a_{11}a_{22}\cdots a_{nn}$.

D. Suppose $\mathbf{A}$, $\mathbf{B}$ are square matrices having the same dimensions and, in addition, $\mathbf{B}$ has been obtained from $\mathbf{A}$ via one of the elementary row operations described in Section 1.4. If $\mathbf{B}$ was obtained from $\mathbf{A}$ via
   - **row operation 1** (multiplication of a row by a constant c), then $\det(\mathbf{B}) = c\det(\mathbf{A})$.
   - **row operation 2** (exchanging two rows), then $\det(\mathbf{B}) = -\det(\mathbf{A})$.
   - **row operation 3** (adding a multiple of one row to another), then $\det(\mathbf{B}) = \det(\mathbf{A})$.

E. If any of the rows or columns of $\mathbf{A}$ contain all zeros, then $|\mathbf{A}| = 0$.

F. The matrix $\mathbf{A}$ is nonsingular (i.e., $\mathbf{A}^{-1}$ exists) if and only if $|\mathbf{A}| \neq 0$.

1.6.2 Eigenpairs

The main reason determinants were discussed in MATH 231 was as a device for finding eigenvalue-eigenvector pairs (or **eigenpairs**). A complex number λ is considered an **eigenvalue** of the n-by-n matrix $\mathbf{A}$ if there exists a nonzero vector $\mathbf{v} \in \mathbb{R}^n$ for which

$$\mathbf{Av} = \lambda\mathbf{v}\,, \qquad \text{or, equivalently, if} \qquad \det(\mathbf{A} - \lambda\mathbf{I}) = 0\,. \tag{1.8}$$

For a given (fixed) eigenvalue λ of $\mathbf{A}$, any *nonzero* vector $\mathbf{v} \in \mathbb{R}^n$ that satisfies $\mathbf{Av} = \lambda\mathbf{v}$ is said to be an **eigenvector of A associated with** λ.

Some remarks:

- The definition of *eigenvalue* given above is not so easy to employ in practice. There is a fact, the truth of which will become clear later, which says "λ is an eigenvalue of $\mathbf{A}$ if and only if $\det(\mathbf{A} - \lambda\mathbf{I}) \neq 0$. We find eigenvalues of $\mathbf{A}$ by finding multiples of the identity matrix which, when subtracted from $\mathbf{A}$, produce a *singular matrix*.

- If n is the number of rows/columns in $\mathbf{A}$, then the quantity $\det(\mathbf{A} - \lambda\mathbf{I})$ is (always) an n^{th}-degree polynomial. Hence it has, counting multiplicities, exactly n roots which are the eigenvalues of $\mathbf{A}$.
- If $\mathbf{A}$ is upper or lower triangular, its eigenvalues are precisely the elements found on its main diagonal.
- Even though $\mathbf{A}$ has only real-number entries, it can have non-real (complex) eigenvalues. However, such eigenvalues always come in conjugate pairs—if $(a + bi)$ is an eigenvalue of $\mathbf{A}$, then so is $(a - bi)$.
- Once we know the eigenvalues, the search for eigenvectors is essentially the same as Example 1.4.4. For each eigenvalue λ, we find the nullspace of a certain matrix, namely $(\mathbf{A} - \lambda\mathbf{I})$. In each instance, when you reduce $(\mathbf{A} - \lambda\mathbf{I})$ to echelon form, there will be at least one free column, and there can be no more free columns than the multiplicity of λ as a zero of $\det(\mathbf{A} - \lambda\mathbf{I})$. There is one special case, however, the case in which $\mathbf{A}$ is a *symmetric* matrix, when you are *assured* that
  - the eigenvalues will all be *real* numbers, and
  - for each eigenvalue λ, an echelon form that is row equivalent to $(\mathbf{A} - \lambda\mathbf{I})$ will have free columns in precisely the same number as the multiplicity of λ.

As this is not your first encounter with the problem of finding eigenpairs, we will give just a few examples of the process. Keep in mind that, in practice, one finds approximate eigenpairs using software. (The command in Octave is `eig()`.) After these examples, we will focus on what the eigenpairs tell us of the geometry of the transformation $(\mathbf{v} \mapsto \mathbf{A}\mathbf{v})\colon \mathbb{R}^n \to \mathbb{R}^n$.

Example 1.6.1

Find the eigenvalues and associated eigenvectors of the matrix

$$\mathbf{A} = \begin{bmatrix} 7 & 0 & -3 \\ -9 & -2 & 3 \\ 18 & 0 & -8 \end{bmatrix}.$$

We first find the eigenvalues, doing so by getting an expression for $\det(\mathbf{A} - \lambda\mathbf{I})$, setting it equal to zero and solving:

$$\begin{aligned} \begin{vmatrix} 7-\lambda & 0 & -3 \\ -9 & -2-\lambda & 3 \\ 18 & 0 & -8-\lambda \end{vmatrix} &= -(2+\lambda)[(7-\lambda)(-8-\lambda)+54] \\ &= -(\lambda+2)(\lambda^2+\lambda-2) \\ &= -(\lambda+2)^2(\lambda-1)\,. \end{aligned}$$

Thus **A** has two distinct eigenvalues, $\lambda_1 = -2$ (its **algebraic multiplicity**, as a *zero* of $\det(\mathbf{A} - \lambda\mathbf{I})$ is 2), and $\lambda_3 = 1$ (algebraic multiplicity 1).

To find eigenvectors associated with $\lambda_3 = 1$, we solve the matrix equation $(\mathbf{A}-\mathbf{I})\mathbf{v} = \mathbf{0}$ (that is, we find the nullspace of $(\mathbf{A} - \mathbf{I})$). Our augmented matrix appears on the left, and an equivalent echelon form on the right:

$$\left[\begin{array}{ccc|c} 6 & 0 & -3 & 0 \\ -9 & -3 & 3 & 0 \\ 18 & 0 & -9 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 2 & 0 & -1 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Since the algebraic multiplicity of λ_3 is 1, the final bullet point on the previous page indicates we should expect precisely one free column in the echelon form and, indeed, the 3rd column is the free one. Writing $x_3 = 2t$, we have $x_1 = t$ and $x_2 = -t$, giving that

$$\text{null}(\mathbf{A} - \mathbf{I}) = \{t(1,-1,2) \,|\, t \in \mathbb{R}\} = \text{span}(\{(1,-1,2)\}) .$$

That is, the eigenvectors associated with $\lambda_3 = 1$ form a line in $\mathbb{R}^3$ characterized by $(1,-1,2)$.

Now, to find eigenvectors associated with $\lambda_1 = -2$ we solve $(\mathbf{A}+2\mathbf{I})\mathbf{v} = \mathbf{0}$. We know going in that λ_1 has algebraic multiplicity 2, so we should arrive at an echelon form with either 1 or 2 free columns. We find that the augmented matrix

$$\left[\begin{array}{ccc|c} 9 & 0 & -3 & 0 \\ -9 & 0 & 3 & 0 \\ 18 & 0 & -6 & 0 \end{array}\right] \sim \left[\begin{array}{ccc|c} 3 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right].$$

Columns 2 and 3 are free, and we set $x_2 = s$, $x_3 = 3t$. This means $x_1 = t$, and hence

$$\text{null}(\mathbf{A} + 2\mathbf{I}) = \{s(0,1,0) + t(1,0,3) \,|\, s,t \in \mathbb{R}\} = \text{span}(\{(0,1,0),(1,0,3)\}) .$$

So, the eigenvectors associated with $\lambda_1 = -2$ form a plane in $\mathbb{R}^3$, with each of these eigenvectors obtainable as a linear combination of $(0,1,0)$ and $(1,0,3)$.

We finish this example by showing the commands and related output from Octave that duplicate the analysis we have done by hand.

```
octave-3.0.0:106> A = [7 0 -3; -9 -2 3; 18 0 -8]
A =
    7    0   -3
   -9   -2    3
   18    0   -8

octave-3.0.0:107> [V, lam] = eig(A)
V =
   0.00000   0.40825   0.31623
```

```
   1.00000  -0.40825   0.00000
   0.00000   0.81650   0.94868

lam =
  -2   0   0
   0   1   0
   0   0  -2
```

Compare these results with our work.

Example 1.6.2

Find the eigenvalues and associated eigenvectors of the matrix

$$\mathbf{A} = \begin{bmatrix} -1 & 2 \\ 0 & -1 \end{bmatrix}.$$

First, we have

$$\det(\mathbf{A} - \lambda\mathbf{I}) \;=\; \begin{vmatrix} -1-\lambda & 2 \\ 0 & -1-\lambda \end{vmatrix} \;=\; (\lambda+1)^2\,,$$

showing $\lambda = -1$ is an eigenvalue (the only one) with algebraic multiplicity 2. Reducing an augmented matrix for $(\mathbf{A} - (-1)\mathbf{I})$, we should have either one or two free columns. In fact, the augmented matrix is

$$\left[\begin{array}{cc|c} 0 & 2 & 0 \\ 0 & 0 & 0 \end{array}\right],$$

and does not need to be reduced, as it is already an echelon form. Only its first column is free, so we set $x_1 = t$. This augmented matrix also tells us that $x_2 = 0$, so

$$\text{null}(\mathbf{A} + \mathbf{I}) \;=\; \{t(1,0)\,|\,t \in \mathbb{R}\} \;=\; \text{span}(\{\mathbf{i}\})\,.$$

Note that, though the eigenvalue has algebraic multiplicity 2, the set of eigenvectors in this example requires only a single vector to characterize them (in contrast to the previous example).

Compare this work to the output of this (related) Octave command.

```
octave-3.0.0:108> [V, lam] = eig([-1 2; 0 -1])
V =
   1.00000  -1.00000
   0.00000   0.00000

lam =
  -1   0
   0  -1
```

Next we give an example where the matrix has non-real eigenvalues.

Example 1.6.3

Find the eigenvalues and associated eigenvectors of the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}.$$

We compute

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} 2-\lambda & -1 \\ 1 & 2-\lambda \end{vmatrix} = (\lambda - 2)^2 + 1 = \lambda^2 - 4\lambda + 5\,.$$

The roots of this polynomial (found using the quadratic formula) are $\lambda_1 = 2 + i$ and $\lambda_2 = 2 - i$; that is, the eigenvalues are not real numbers. This is a common occurrence, and we can press on to find the eigenvectors just as we have in the past with real eigenvalues. To find eigenvectors associated with $\lambda_1 = 2 + i$, we look for $\mathbf{x}$ satisfying

$$\begin{aligned}(\mathbf{A} - (2+i)\mathbf{I})\mathbf{x} = \mathbf{0} \quad &\Rightarrow \quad \begin{bmatrix} -i & -1 \\ 1 & -i \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &\Rightarrow \quad \begin{bmatrix} -ix_1 - x_2 \\ x_1 - ix_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &\Rightarrow \quad x_1 = ix_2\,.\end{aligned}$$

Thus all eigenvectors associated with $\lambda_1 = 2 + i$ are scalar multiples of $\mathbf{u}_1 = (i, 1)$. Proceeding with $\lambda_2 = 2 - i$, we have

$$\begin{aligned}(\mathbf{A} - (2-i)\mathbf{I})\mathbf{x} = \mathbf{0} \quad &\Rightarrow \quad \begin{bmatrix} i & -1 \\ 1 & i \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &\Rightarrow \quad \begin{bmatrix} ix_1 - x_2 \\ x_1 + ix_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &\Rightarrow \quad x_1 = -ix_2\,,\end{aligned}$$

which shows all eigenvectors associated with $\lambda_2 = 2 - i$ to be scalar multiples of $\mathbf{u}_2 = (-i, 1)$.

Notice that $\mathbf{u}_2$, the eigenvector associated with the eigenvalue $\lambda_2 = 2 - i$ in the last example, is the complex conjugate of $\mathbf{u}_1$, the eigenvector associated with the eigenvalue $\lambda_1 = 2 + i$. It is indeed a fact that, if the n-by-n *real* (i.e., entries all real numbers) matrix $\mathbf{A}$ has a nonreal eigenvalue $\lambda_1 = \lambda + i\mu$ with corresponding eigenvector ξ_1, then it also has eigenvalue $\lambda_2 = \lambda - i\mu$ with corresponding eigenvector $\xi_2 = \bar{\xi}_1$.

Here is the relevant work in Octave.

```
octave-3.0.0:109> [V, lam] = eig([2 -1; 1 2])
V =
   0.70711 + 0.00000i   0.70711 - 0.00000i
   0.00000 - 0.70711i   0.00000 + 0.70711i

lam =
   2 + 1i   0 + 0i
   0 + 0i   2 - 1i
```

In Section 1.3, we investigated the underlying geometry associated with matrix multiplication. We saw that certain kinds of 2-by-2 matrices transformed the plane $\mathbb{R}^2$ by rotating it about the origin; others produced reflections across a line. Of particular interest here is Case 3 from that section, where the matrices involved caused rescalings that were (possibly) different along two perpendicular axes. Now, using our knowledge of eigenpairs, we can discuss the general case where these axes may not be perpendicular.

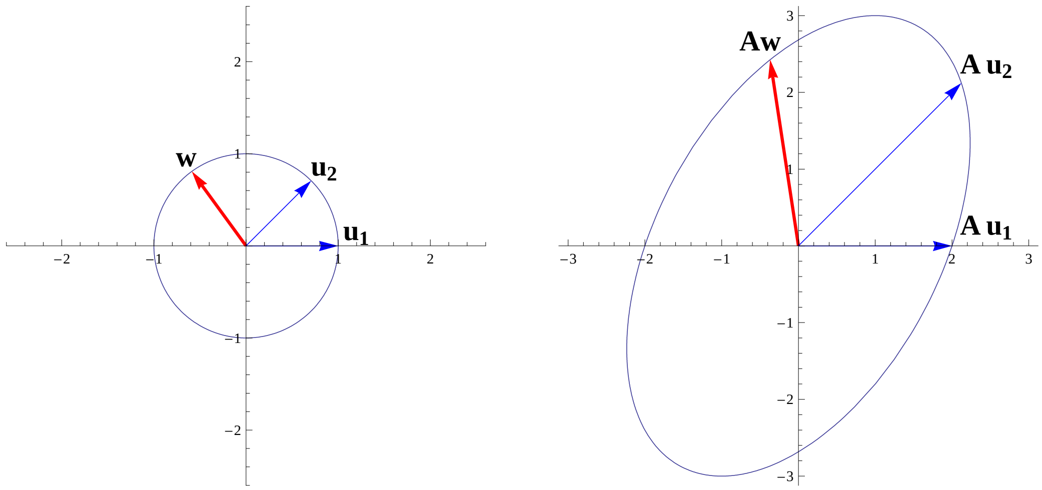
Recall that an eigenpair $(\lambda, \mathbf{v})$ of $\mathbf{A}$ satisfies the relationship $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. This says that the output $\mathbf{A}\mathbf{v}$ (from the function $(\mathbf{x} \mapsto \mathbf{A}\mathbf{x})$) corresponding to input $\mathbf{v}$ is a vector that lies in the "same direction" as $\mathbf{v}$ itself and, in fact, is a predictable rescaling of $\mathbf{v}$ (i.e., it is λ times $\mathbf{v}$).

Example 1.6.4

Suppose $\mathbf{A}$ is a 2-by-2 matrix that has eigenvalues $\lambda_1 = 2$, $\lambda_2 = 3$ with corresponding eigenvectors $\mathbf{u}_1 = (1, 0)$, $\mathbf{u}_2 = (1/\sqrt{2}, 1/\sqrt{2})$. The matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 0 & 3 \end{bmatrix}$$

is just such a matrix, and the associated function $(\mathbf{x} \mapsto \mathbf{A}\mathbf{x})$ rescales vectors in the direction of $(1, 0)$ by a factor of 2 relative to the origin, while vectors in the direction of $(1, 1)$ will be similarly rescaled but by a factor of 3. (See the figure below.) The affect of multiplication by $\mathbf{A}$ on all other vectors in the plane is more complicated to describe, but will nevertheless conform to these two facts. The figure shows (on the left) the unit circle and eigenvectors $\mathbf{u}_1$, $\mathbf{u}_2$ of $\mathbf{A}$. On the right is displayed how this circle is transformed via multiplication by $\mathbf{A}$. Notice that $\mathbf{A}\mathbf{u}_1$ faces the same direction as $\mathbf{u}_1$, but is twice as long; the same is true of $\mathbf{A}\mathbf{u}_2$ in relation to $\mathbf{u}_2$, except it is 3 times as long. The figure displays one more unit vector $\mathbf{w}$ along with its image $\mathbf{A}\mathbf{w}$ under matrix multiplication by $\mathbf{A}$.



We leave it to the exercises to discover what may be said about the eigenvalues of a 2-by-2 matrix **A** when the associated function ($\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$) rotates the plane about the origin. We also investigate similar ideas when **A** is a 3-by-3 matrix.

1.7 Linear Independence and Matrix Rank

We have defined the nullspace of an m-by-n matrix **A** as the set of vectors $\mathbf{v} \in \mathbb{R}^n$ satisfying the equation $\mathbf{A}\mathbf{v} = \mathbf{0}$. In light of the discussion in Section 1.2, the components of any $\mathbf{v} \in \text{null}(\mathbf{A})$ offer up a way to write $\mathbf{0}$ as a linear combination of the columns of **A**:

$$\mathbf{0} \;=\; \mathbf{A}\mathbf{v} \;=\; \left[\; \mathbf{A}_1 \;\middle|\; \mathbf{A}_2 \;\middle|\; \cdots \;\middle|\; \mathbf{A}_n \;\right] \begin{bmatrix} v_1 \\ \hline v_2 \\ \hline \vdots \\ \hline v_n \end{bmatrix} \;=\; v_1\mathbf{A}_1 + v_2\mathbf{A}_2 + \cdots + v_n\mathbf{A}_n \,.$$

For some matrices **A**, the nullspace consists of just one vector, the zero vector **0**. We make a definition that helps us characterize this situation.

Definition 1.7.1. Let $S = \{\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_k\}$ be a set of vectors in $\mathbb{R}^m$. If the zero vector $\mathbf{0} \in \mathbb{R}^n$ can be written as a linear combination

$$c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \cdots + c_k\mathbf{u}_k \;=\; \mathbf{0}\,,$$

with at least one of the coefficients $c_1, \ldots, c_k$ nonzero, then the set S of vectors is said to be **linearly dependent**. If, however, the *only* linear combination of the vectors in S that yields **0** is the one with $c_1 = c_2 = \cdots = c_k = 0$, then set S is **linearly independent**.

Employing this terminology, when $\text{null}(\mathbf{A}) = \{\mathbf{0}\}$ the set of columns of $\mathbf{A}$ are linearly independent. Otherwise, this set is linearly dependent.

Suppose, as in Example 1.4.4, we set out to find the nullspace of $\mathbf{A}$ using Gaussian elimination. The result of elementary row operations is the row equivalence of augmented matrices

$$\left[\ \mathbf{A} \mid \mathbf{0}\ \right] \sim \left[\ \mathbf{R} \mid \mathbf{0}\ \right],$$

where $\mathbf{R}$ is an echelon form for $\mathbf{A}$. We know that $\mathbf{v} \in \text{null}(\mathbf{A})$ if and only if $\mathbf{v} \in \text{null}(\mathbf{R})$. Let's look at several possible cases:

1. Case: $\mathbf{R}$ has no free columns.
   Several possible appearances of $\mathbf{R}$ are

$$\mathbf{R} = \begin{bmatrix} p & * & * & \cdots & * \\ 0 & p & * & \cdots & * \\ 0 & 0 & p & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p \end{bmatrix}, \tag{1.9}$$

   and

$$\mathbf{R} = \begin{bmatrix} p & * & * & \cdots & * \\ 0 & p & * & \cdots & * \\ 0 & 0 & p & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & p \\ \hline 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}. \tag{1.10}$$

   Regardless of whether $\mathbf{R}$ has form (1.9) or form (1.10), the elements of $\mathbf{v}$ are uniquely determined—there is no other solution to $\mathbf{R}\mathbf{v} = \mathbf{0}$ but the one with each component $v_j = 0$ for $j = 1, \ldots, n$. This means that $\text{null}(\mathbf{R}) = \text{null}(\mathbf{A}) = \{\mathbf{0}\}$ and, correspondingly, that the columns of $\mathbf{A}$ are linearly independent.

2. Case: **R** has free columns.
   A possible appearance of **R** is

$$\mathbf{R} = \left[\begin{array}{cccccccc} p & * & * & * & * & \cdots & * & * \\ 0 & 0 & 0 & p & * & \cdots & * & * \\ 0 & 0 & 0 & 0 & p & \cdots & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & \\ 0 & 0 & 0 & 0 & 0 & \cdots & p & * \\ \hline 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right] \tag{1.11}$$

   The matrix **R** pictured here (as a 'for instance') has at least 3 free columns (the 2$^{\text{nd}}$, 3$^{\text{rd}}$ and last ones), each providing a degree of freedom to the solution of $\mathbf{Av} = \mathbf{0}$. If the solution of $\mathbf{Av} = \mathbf{0}$ has even one degree of freedom (one free column in an echelon form of **A**), then the columns of **A** are linearly dependent.

   It should be evident that the set of *pivot columns* of **R** are linearly independent. That is, if we throw out the free columns to get a smaller matrix $\tilde{\mathbf{R}}$ of form (1.9) or (1.10), then the columns of $\tilde{\mathbf{R}}$ (and correspondingly, those of **A** from which these pivot columns originated) are linearly independent.

The number of linearly independent columns in **A** is a quantity that deserves a name.

Definition 1.7.2. The **rank** of an m-by-n matrix **A**, denoted by rank(**A**), is the *number* of pivots (equivalently, the number of pivot columns) in an echelon form **R** for **A**.
The number of free columns in **R** is called the **nullity**, of **A**, denoted by nullity(**A**).

Note that, for an m-by-n matrix, $\text{rank}(\mathbf{A}) + \text{nullity}(\mathbf{A}) = n$.

Now, suppose some vector $\mathbf{b} \in \text{span}(S)$, where $S = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ is some collection of vectors. That is,

$$\mathbf{b} = a_1\mathbf{u}_1 + \cdots + a_k\mathbf{u}_k \,,$$

for some choice of coefficients $a_1, \ldots a_k$. If the vectors in S are linearly dependent, then there is a choice of coefficients $c_1, \ldots c_k$, not all of which are zero, such that

$$c_1\mathbf{u}_1 + c_2\mathbf{u}_k + \cdots + c_k\mathbf{u}_k = \mathbf{0} \,.$$

Let us assume that $c_k \neq 0$. Solving this equation for $\mathbf{u}_k$, we get

$$\mathbf{u}_k = -\frac{1}{c_k}(c_1\mathbf{u}_1 + \cdots + c_{k-1}\mathbf{u}_{k-1}) \,,$$

which we may then plug back into our equation for **b**:

$$\begin{aligned}
\mathbf{b} &= a_1\mathbf{u}_1 + \cdots + a_{k-1}\mathbf{u}_{k-1} + a_k\mathbf{u}_k \\
&= a_1\mathbf{u}_1 + \cdots + a_{k-1}\mathbf{u}_{k-1} - \frac{a_k}{c_k}(c_1\mathbf{u}_1 + \cdots + c_{k-1}\mathbf{u}_{k-1}) \\
&= \left(a_1 - \frac{a_k c_1}{c_k}\right)\mathbf{u}_1 + \left(a_2 - \frac{a_k c_2}{c_k}\right)\mathbf{u}_2 + \cdots + \left(a_{k-1} - \frac{a_k c_{k-1}}{c_k}\right)\mathbf{u}_{k-1} ,
\end{aligned}$$

which shows that, by taking $d_j = a_j - a_k c_j/c_k$ for $j = 1, \ldots, k-1$, this **b** which was already known to be a linear combination of $\mathbf{u}_1, \ldots, \mathbf{u}_k$ may be rewritten as a linear combination $d_1\mathbf{u}_1 + \cdots + d_{k-1}\mathbf{u}_{k-1}$ of the reduced collection $\{\mathbf{u}_1, \ldots, \mathbf{u}_{k-1}\}$. Of course, if this reduced set of vectors is linearly dependent, we may remove another vector—let us assume $\mathbf{u}_{k-1}$ would suit our purposes—to arrive at an even smaller set $\{\mathbf{u}_1, \ldots, \mathbf{u}_{k-2}\}$ which has the same span as the original set S, and continue in this fashion until we arrive at a subcollection of S which is linearly independent. We have demonstrated the truth of the following result.

Theorem 1.7.3. Suppose S is a collection of vectors in $\mathbb{R}^n$. Then some **subset** B of S (that is, every vector in B comes from S, but there *may* be vectors in S excluded from B) has the property that span(B) = span(S) and B is linearly independent.

The collection **B** is called a **basis** (a term we will define more carefully in a later section) for span(S).

The previous theorem is one of those "existence" theorems you see in mathematics, guaranteeing something exists, but not telling how to *find* it. Yet, to find such a set B we may use a strategy adapted from our comments in Point 2 of this section. That is, we may build a matrix whose columns are the vectors in S

$$\mathbf{A} = \left[\ \mathbf{u}_1 \mid \mathbf{u}_2 \mid \cdots \mid \mathbf{u}_k\ \right].$$

We may then reduce **A** to echelon form **R** (another matrix whose dimensions are the same as **A**—there is no need to augment **A** with an extra column for this task), and take B to be the set of columns of **A**—it may be all of them—which correspond to *pivot* columns in **R**. The number of elements in B will be rank(**A**).

There is an important relationship between the value of nullity(**A**) and the number of solutions one finds when solving $\mathbf{A}\mathbf{x} = \mathbf{b}$. You may already have suspected this, given the similarity of results in Examples 1.4.3 and 1.4.4, both of which involved the same matrix **A**. In Example 1.4.4, we found null(**A**) to be the line of vectors passing through the origin in $\mathbb{R}^3$

$$t(-1, 2, 0), \qquad t \in \mathbb{R} .$$

In Example 1.4.3, we solved $\mathbf{A}\mathbf{x} = (3, 9)$, getting solution

$$(2, 0, 1) + t(-1, 2, 0), \qquad t \in \mathbb{R} ,$$

another line of vectors in $\mathbb{R}^3$, parallel to the first line, but offset from the origin by the vector $(2, 0, 1)$. One could describe this latter solution as being the sum of the nullspace and a *particular solution* of $\mathbf{Ax} = \mathbf{b}$.[4] Observe that, if $\mathbf{x}_p$ satisfies the equation $\mathbf{Ax} = \mathbf{b}$ and $\mathbf{x}_n \in \text{null}(\mathbf{A})$, then for $\mathbf{v} = \mathbf{x}_p + \mathbf{x}_n$,

$$\mathbf{Av} \;=\; \mathbf{A}(\mathbf{x}_p + \mathbf{x}_n) \;=\; \mathbf{Ax}_p + \mathbf{Ax}_n \;=\; \mathbf{b} + \mathbf{0} \;=\; \mathbf{b}\,.$$

Thus, when $\mathbf{Ax} = \mathbf{b}$ has a solution, the number of solutions is at least as numerous as the number of vectors in $\text{null}(\mathbf{A})$. In fact, they are precisely as numerous, as stated in the next theorem.

Theorem 1.7.4. Suppose the m-by-n matrix $\mathbf{A}$ and vector $\mathbf{b} \in \mathbb{R}^m$ (both fixed) are such that the matrix equation $\mathbf{Ax} = \mathbf{b}$ is **consistent** (i.e., the equation *has* a solution). Then the solutions are in one-to-one correspondence with the elements in $\text{null}(\mathbf{A})$. Said another way, if $\text{null}(\mathbf{A})$ has just the zero vector, then $\mathbf{Ax} = \mathbf{b}$ has just one solution. If $\text{null}(\mathbf{A})$ is a line (plane, etc.) of vectors, then so is the set of solutions to $\mathbf{Ax} = \mathbf{b}$.

If you review Examples 1.4.3 and 1.4.4 you will see that the appearance of the free variable t is due to a free column in the echelon form we got for $\mathbf{A}$. The rank of $\mathbf{A}$—its number of linearly independent columns it has, is 2, not 3.

We finish this section with an important theorem. Some of these results have been stated (in some form or other) elsewhere, but the theorem provides a nice overview of facts about *square* matrices.

Theorem 1.7.5. Suppose $\mathbf{A}$ is an n-by-n matrix. The following are equivalent (that is, if you know one of them is true, then you know all of them are).

(i) The matrix $\mathbf{A}$ is nonsingular.

(ii) The matrix equation $\mathbf{Ax} = \mathbf{b}$ has a unique solution for each possible n-vector $\mathbf{b}$.

(iii) The determinant $\det(\mathbf{A}) \neq 0$.

(iv) The nullspace $\text{null}(\mathbf{A}) = \{\mathbf{0}\}$.

(v) The columns of $\mathbf{A}$ are linearly independent

(vi) $\text{rank}(\mathbf{A}) = n$.

(vii) $\text{nullity}(\mathbf{A}) = 0$.

[4] That ought to sound like language from MATH 231, when comparisons are made between solutions of nonhomogeneous linear ODEs and their homogeneous counterparts. Indeed, the use of the word *linear* when describing such ODEs suggests certain principles from linear algebra undergird the theory of linear ODEs.

Exercises

1.1 Give a particularly simple command in Octave (one which does not require you to type in every entry) which will produce the matrix

a) $\begin{bmatrix} 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

b) $\begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 0 & 7 & 0 & 0 & -4 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 3 & -2 \end{bmatrix}$

1.2 Suppose **A** is a 5-by-3 matrix.

a) If **B** is another matrix and the matrix product **AB** makes sense, what must be true about the dimensions of **B**?

b) If the matrix product **BA** makes sense, what must be true about the dimensions of **B**?

1.3 Suppose **A**, **B** are matrices for which the products **AB** and **BA** are both possible (both defined).

a) For there to be any chance that **AB** = **BA**, what must be true about the dimensions of **A**? Explain.

b) When we say that **AB** ≠ **BA** in general, we do not mean that it never happens, but rather that you cannot count on their equality. Write a function in Octave which, when called, generates two random 3-by-3 matrices **A** and **B**, finds the products **AB** and **BA**, and checks whether they are equal. Run this code 20 times, and record how many of those times it happens that **AB** = **BA**. Hand in a printout of your function.

c) Of course, when both **A** and **B** are square matrices with one of them equal to the identity matrix, it will be the case that **AB** = **BA**. What other instances can you think of in which **AB** = **BA** is guaranteed to hold?

1.4 Suppose $\mathbf{A}$ commutes with every 2-by-2 matrix (i.e., $\mathbf{AB} = \mathbf{BA}$) and, in particular,

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \text{commutes with} \quad \mathbf{B}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{B}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Show that $a = d$ and $b = c = 0$—that is, if $\mathbf{AB} = \mathbf{BA}$ for all matrices $\mathbf{B}$, then $\mathbf{A}$ is a multiple of the identity matrix.

1.5 Which of the following matrices are guaranteed to equal $(\mathbf{A} + \mathbf{B})^2$?

$$(\mathbf{B}+\mathbf{A})^2\,, \quad \mathbf{A}^2+2\mathbf{AB}+\mathbf{B}^2\,, \quad \mathbf{A}(\mathbf{A}+\mathbf{B})+\mathbf{B}(\mathbf{A}+\mathbf{B})\,, \quad (\mathbf{A}+\mathbf{B})(\mathbf{B}+\mathbf{A})\,, \quad \mathbf{A}^2+\mathbf{AB}+\mathbf{BA}+\mathbf{B}^2\,.$$

1.6

a) If $\mathbf{A}$ is invertible and $\mathbf{AB} = \mathbf{AC}$, prove quickly that $\mathbf{B} = \mathbf{C}$.

b) If $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, find an example with $\mathbf{AB} = \mathbf{AC}$ but $\mathbf{B} \neq \mathbf{C}$.

1.7 If the inverse of $\mathbf{A}^2$ is $\mathbf{B}$, show that the inverse of $\mathbf{A}$ is $\mathbf{AB}$. (Thus, $\mathbf{A}$ is invertible whenever $\mathbf{A}^2$ is.)

1.8 Verify, via direct calculation, Theorem 1.1.3. That is, use the knowledge that $\mathbf{A}$, $\mathbf{B}$ are n-by-n nonsingular matrices to show that $\mathbf{AB}$ is nonsingular as well, having inverse $\mathbf{B}^{-1}\mathbf{A}^{-1}$.

1.9 We have learned several properties of the operations of inversion and transposition of a matrix. The table below summarizes these, with counterparts appearing on the same row.

	matrix transposition	matrix inversion
i.	$(\mathbf{A}^T)^T = \mathbf{A}$	$(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
ii.	$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$	$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
iii.	$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$	

Show that property iii. has no counterpart in the "matrix inversion" column. That is, in general it is not the case that $(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} + \mathbf{B}^{-1}$.

1.10 The previous two problems asked you to "prove" or "show" (basically synonymous words in mathematics) something. Yet there is something fundamentally different about what is required in the two problems. In one problem, all you need to do is come up with a specific instance—matrices $\mathbf{A}$, $\mathbf{B}$ whose entries are concrete numbers—to prove the assertion. In the other problem, if you resort to specific matrices then all you succeed in doing is showing the assertion is true in one particular instance. In which problem is it that you cannot get specific about the entries in **A**, **B**? What is it in the wording of these problems that helps you determine the level of generality required?

1.11

a) Explain why a it is necessary that a symmetric matrix be square.

b) Suppose $\mathbf{A} = (a_{ij})$ is an n-by-n matrix. Prove that **A** is symmetric if and only if $a_{ij} = a_{ji}$ for each $1 \leq i, j \leq n$.

1.12 Suppose there is a town which perenially follows these rules:

- The number of households always stays fixed at 10000.
- Every year 30 percent of households currently subscribing to the local newspaper cancel their subscriptions.
- Every year 20 percent of households not receiving the local newspaper subscribe to it.

a) Suppose one year, there are 8000 households taking the paper. According to the data above, these numbers will change the next year. The total of subscribers will be

$$(0.7)(8000) + (0.2)(2000) = 6000 ,$$

and the total of nonsubscribers will be

$$(0.3)(8000) + (0.8)(2000) = 4000 .$$

If we create a 2-vector whose first component is the number of subscribers and whose 2nd component is the number of nonsubscribers, then the initial vector is $(8000, 2000)$, and the vector one year later is

$$\begin{bmatrix} 0.7 & 0.2 \\ 0.3 & 0.8 \end{bmatrix} \begin{bmatrix} 8000 \\ 2000 \end{bmatrix} = \begin{bmatrix} 6000 \\ 4000 \end{bmatrix}.$$

What is the long-term outlook for newspaper subscription numbers?

b) Does your answer above change if the initial subscription numbers are changed to 9000 subscribing households? Explain.

1.13 In Octave, generate 50 random 4-by-4 matrices. Determine how many of these matrices are singular. (You may find the command `det()` helpful. It's a simple command to use, and like most commands in Octave, you can find out about its use by typing `help det`. You may also wish to surround the work you do on one matrix with the commands `for i = 1:50` and `end`.) Based upon your counts, how prevalent among all 4-by-4 matrices would you say that singular matrices are? What if you conduct the same experiment on 5-by-5 matrices? 10-by-10? (Along with your answers to the questions, hand in the code you used to conduct one of these experiments.)

1.14 Consider a matrix $\mathbf{A}$ that has been blocked in the following manner:

$$\mathbf{A} = \left[\begin{array}{c|c|c} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \end{array}\right],$$

where $\mathbf{A}_{11}$ is 2-by-3, $\mathbf{A}_{23}$ is 4-by-2, and the original matrix $\mathbf{A}$ has 7 columns.

a) How many rows does $\mathbf{A}$ have?

b) Determine the dimensions of the submatrices $\mathbf{A}_{12}$, $\mathbf{A}_{13}$, $\mathbf{A}_{21}$, and $\mathbf{A}_{22}$.

c) Give at least three different ways to partition a matrix $\mathbf{B}$ that has 5 columns so that a block-multiplication of the matrix product $\mathbf{AB}$ makes sense. For each of your answers, specify the block structure of $\mathbf{B}$ using $\mathbf{B}_{ij}$ notation just as we originally gave the block structure of $\mathbf{A}$, and indicate the dimensions of each block.

d) For each of your answers to part (c), write out the corresponding block structure of the product $\mathbf{AB}$, indicating how the individual blocks are computed from the blocks of $\mathbf{A}$ and $\mathbf{B}$ (as was done in the notes immediately preceding Example 1.2.2).

1.15 The first row of a matrix product $\mathbf{AB}$ is a linear combination of all the rows of $\mathbf{B}$. What are the coefficients in this combination, and what is the first row of $\mathbf{AB}$, if

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 4 \\ 0 & -1 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{B} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}?$$

1.16 Describe the rows of $\mathbf{EA}$ and the *columns* of $\mathbf{AE}$ if $\mathbf{E} = \begin{bmatrix} 1 & 5 \\ 0 & 1 \end{bmatrix}$. (Note that, for both products to make sense, $\mathbf{A}$ must be 2-by-2.)

1.17

a) Suppose $\mathbf{A}$ is a 4-by-n matrix. Find a matrix $\mathbf{P}$ (you should determine appropriate dimensions for $\mathbf{P}$, as well as specify its entries) so that $\mathbf{PA}$ has the same entries as $\mathbf{A}$ but the 1st, 2nd, 3rd and 4th rows of $\mathbf{PA}$ are the 2nd, 4th, 3rd and 1st rows of $\mathbf{A}$ respectively. Such a matrix $\mathbf{P}$ is called a **permutation matrix**.

b) Suppose $\mathbf{A}$ is an m-by-4 matrix. Find a matrix $\mathbf{P}$ so that $\mathbf{AP}$ has the same entries as $\mathbf{A}$ but the 1^{st}, 2^{nd}, 3^{rd} and 4^{th} *columns* of $\mathbf{AP}$ are the 2^{nd}, 4^{th}, 3^{rd} and 1^{st} columns of $\mathbf{A}$ respectively.

c) Suppose $\mathbf{A}$ is an m-by-3 matrix. Find a matrix $\mathbf{B}$ so that $\mathbf{AB}$ again has 3 columns, the first of which is the sum of all three columns of $\mathbf{A}$, the 2^{nd} is the difference of the 1^{st} and 3^{rd} columns of $\mathbf{A}$ (column 1 - column 3), and the 3^{rd} column is 3 times the 1^{st} column of $\mathbf{A}$.

1.18 We have given two alternate ways of achieving translations of the plane by a vector $\mathbf{w} = (a, b)$:

(i) $(\mathbf{v} \mapsto \mathbf{v} + \mathbf{w})$, and

(ii) $(\mathbf{v} \mapsto \tilde{\mathbf{v}} \mapsto \mathbf{A}\tilde{\mathbf{v}})$, where $\mathbf{A} = \left[\begin{array}{ccc} 1 & 0 & a \\ 0 & 1 & b \\ \hline 0 & 0 & 1 \end{array}\right]$.

If $\mathbf{v} \in \mathbb{R}^2$ has homogeneous coordinates $\tilde{\mathbf{v}} \in \mathbb{R}^3$, use the indicated blocking on $\mathbf{A}$ in (ii) and what you know about block multiplication to show that the upper block of $\mathbf{A}\tilde{\mathbf{v}}$ gives the same result as the mapping in (i).

1.19

a) Multiply out the matrices on the left-hand side of (1.5) to show that, indeed, they are equal to the matrix on the right-hand side for $\alpha = 2\theta$.

b) Show that a matrix in the form (1.6) may be expressed in an alternate form

$$\begin{bmatrix} a^2 - b^2 & 2ab \\ 2ab & b^2 - a^2 \end{bmatrix},$$

for some choice of constants a, b such that $a^2 + b^2 = 1$.

1.20 Give 3-by-3 examples (not simply $\mathbf{A} = \mathbf{0}$ nor $\mathbf{A} = \mathbf{I}_n$) of

a) a diagonal matrix: $a_{ij} = 0$ when $i \neq j$.

b) a symmetric matrix: $a_{ij} = a_{ji}$ for all i and j.

c) an upper triangular matrix: $a_{ij} = 0$ if $i > j$.

d) a lower triangular matrix: $a_{ij} = 0$ if $i < j$.

e) a skew-symmetric matrix: $a_{ij} = -a_{ji}$ for all i and j.

Obviously there is no real need for the requirement that these examples have 3 rows and 3 columns. But which of these matrix types are *only possible* when the matrix is square (i.e., n-by-n for some n)?

1.21

a) How many entries can be chosen independently in a symmetric n-by-n matrix?

b) How many entries can be chosen independently in a skew-symmetric n-by-n matrix?

1.22 Determine which of the following is an echelon form. For those that are, indicate what are the pivot columns and the pivots.

a) $\begin{bmatrix} 0 & 2 & 1 & 6 & 5 & -1 \\ 0 & 0 & 0 & 3 & 2 & 7 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix}$

b) $\begin{bmatrix} 2 & 7 & 3 & -1 & -5 \\ -1 & 1 & 1 & 4 & 2 \\ 0 & 2 & 3 & 5 & 1 \\ 0 & 0 & -1 & -1 & 7 \\ 0 & 0 & 0 & -3 & 5 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

c) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$

d) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

e) $\begin{bmatrix} 1 & 4 & 2 & 8 \end{bmatrix}$

f) $\begin{bmatrix} 1 & 0 & -2 & 3 \\ 0 & 2 & 3 & -1 \\ 0 & 0 & 1 & 2 \\ 0 & -1 & 0 & 5 \end{bmatrix}$

1.23 Use backward substitution to solve the following systems of equations.

a) $\begin{aligned} x_1 - 3x_2 &= 2 \\ 2x_2 &= 6 \end{aligned}$

b)
$$\begin{aligned} x_1 + x_2 + x_3 &= 8 \\ 2x_2 + x_3 &= 5 \\ 3x_3 &= 9 \end{aligned}$$

c)
$$\begin{aligned} x_1 + 2x_2 + 2x_3 + x_4 &= 5 \\ 3x_2 + x_3 - 2x_4 &= 1 \\ -x_3 + 2x_4 &= -1 \\ 4x_4 &= 4 \end{aligned}$$

1.24 Write out the system of equations that corresponds to each of the following augmented matrices.

a) $\left[\begin{array}{cc|c} 3 & 2 & 8 \\ 1 & 5 & 7 \end{array}\right]$

b) $\left[\begin{array}{ccc|c} 5 & -2 & 1 & 3 \\ 2 & 3 & -4 & 0 \end{array}\right]$

c) $\left[\begin{array}{ccc|c} 2 & 1 & 4 & -1 \\ 4 & -2 & 3 & 4 \\ 5 & 2 & 6 & -1 \end{array}\right]$

1.25 Suppose we wish to perform elementary operation 3 on some matrix $\mathbf{A}$. That is, we wish to produce a matrix $\mathbf{B}$ which has the same dimensions as $\mathbf{A}$, and in most respects is identical to $\mathbf{A}$ except that

$$(\text{row } i \text{ of } \mathbf{B}) = (\text{row } i \text{ of } \mathbf{A}) + \beta(\text{row } j \text{ of } \mathbf{A}) .$$

If $\mathbf{A}$ is m-by-n, then $\mathbf{B} = \mathbf{E}_{ij}\mathbf{A}$, where $\mathbf{E}_{ij}$ is the m-by-m **elementary matrix**

$$\mathbf{E}_{ij} := \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \ddots & & & \\ & & \beta & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix}$$

which looks just like the m-square identity matrix $\mathbf{I}_m$ except for the entry β appearing in its i^{th} row, j^{th} column. Octave code for producing such a matrix might look like

```
function elementaryMatrix = emat(m, i, j, val)
  elementaryMatrix = eye(m,m);
  elementaryMatrix(i,j) = val;
end
```

a) Create a text file containing the code above and called `emat.m`. Place this file in your working directory, the one you are in when running Octave (or Matlab).

b) Another list of commands to put into a file, call it `simpleGE.m`, is the following:

```
B = A;
numRows = size(B)(1);
numCols = size(B)(2);
currRow = 1;
currCol = 1;
while ((currRow < numRows) && (currCol < numCols))
  while ((abs(B(currRow, currCol)) < 10^(-10)) && (currCol < numCols))
    B(currRow, currCol) = 0;
    currCol = currCol + 1;
  end

  if (currCol < numCols)
    pivot = B(currRow, currCol);
    for ii = (currRow + 1):numRows
      B = emat(numRows, ii, currRow, -B(ii,currCol)/pivot) * B;
      % Remove the final semicolon in the previous line
      % if you would like to see the progression of matrices
      % from the original one (A) to the final one in echelon form.
    end
  end
  currRow = currRow + 1;
  currCol = currCol + 1;
end
B
```

One would run `simpleGE` after first storing the appropriate coefficients of the linear system in an augmented matrix `A`.

Save the commands above under the filename `simpleGE.m` in your working directory. Then test it out on the matrix (assumed to be already augmented)

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 2 & 1 & 5 \\ 3 & 2 & 6 & 3 & 1 \\ 6 & 2 & 12 & 4 & 3 \end{bmatrix}.$$

If you have arranged and entered everything correctly, the result will be the matrix in echelon form

$$\begin{bmatrix} 1 & 3 & 2 & 1 & 5 \\ 0 & -7 & 0 & 0 & -14 \\ 0 & 0 & 0 & -2 & 5 \end{bmatrix}.$$

1.26 Using `simpleGE` (see Exercise 1.25) as appropriate, find all solutions to the following linear systems of equations:

a)
$$\begin{aligned} 2x - z &= -4 \\ -4x - 2y + z &= 11 \\ 2x + 2y + 5z &= 3 \end{aligned}$$

b)
$$\begin{aligned} x_1 + 3x_2 + 2x_3 + x_4 &= 5 \\ 3x_1 + 2x_2 + 6x_3 + 3x_4 &= 1 \\ 6x_1 + 2x_2 + 12x_3 + 4x_4 &= 3 \end{aligned}$$

c)
$$\begin{aligned} x + 3y &= 1 \\ -x - y + z &= 5 \\ 2x + 4y - z &= -7 \end{aligned}$$

1.27 Using `simpleGE` (see Exercise 1.25) as appropriate, find all solutions to the following linear systems of equations:

$$\begin{aligned} x + y &= 5 \\ x - 7y - 12z &= 1 \\ 3x - y - 5z &= 15 \\ 2x + 4y + 3z &= 11 \\ x - y - 3z &= 4\,. \end{aligned}$$

1.28 Find choices of constants c_1, c_2 and c_3 such that $\mathbf{b} = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3$. That is, write $\mathbf{b}$ as a linear combination of $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. If there is only one such linear combination, state how you know this is so. Otherwise, your answer should include all possible choices of the constants c_1, c_2 and c_3.

a) $\mathbf{v}_1 = (1, 3, -4)$, $\mathbf{v}_2 = (0, 1, 2)$, $\mathbf{v}_3 = (-1, -5, 1)$, $\mathbf{b} = (0, -5, -6)$.

b) $\mathbf{v}_1 = (1, 2, 0)$, $\mathbf{v}_2 = (2, 3, 3)$, $\mathbf{v}_3 = (-1, 1, -8)$, $\mathbf{b} = (5, 9, 4)$.

c) $\mathbf{v}_1 = (1, 0, 3, -2)$, $\mathbf{v}_2 = (0, 1, 2, -1)$, $\mathbf{v}_3 = (3, -4, 1, -2)$, $\mathbf{b} = (1, -5, -7, 3)$.

1.29

a) For each given set of matrices, show that they commute (i.e., can be multiplied in any order and give the same answer; find an easy way if you can), and find the product of all matrices in the set. (A missing entry should be interpreted as a zero.)

(i) $\begin{bmatrix} 1 & & \\ b_{21} & 1 & \\ & & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & & \\ & 1 & \\ b_{31} & & 1 \end{bmatrix}$

(ii) $\begin{bmatrix} 1 & & & \\ b_{21} & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & & & \\ & 1 & & \\ b_{31} & & 1 & \\ & & & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ b_{41} & & & 1 \end{bmatrix}$

(iii) $\begin{bmatrix} 1 & & & \\ & 1 & & \\ & b_{32} & 1 & \\ & & & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & b_{42} & & 1 \end{bmatrix}$

b) Describe as precisely as you can what characterizes the sets of matrices in (i)–(iii) of part (a). (Each is the set of all matrices which . . .)

c) State and prove a general result for n-by-n matrices, of which (i)–(iii) above are special cases.

1.30 Find the nullspace of the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 & 3 \\ 3 & 6 & 18 & 9 & 9 \\ 2 & 4 & 6 & 2 & 6 \\ 4 & 8 & 12 & 10 & 12 \\ 5 & 10 & 24 & 11 & 15 \end{bmatrix}.$$

1.31 Consider the system of linear equations

$$\begin{aligned} x_1 + 3x_2 + 2x_3 - x_4 &= 4 \\ -x_1 - x_2 - 3x_3 + 2x_4 &= -1 \\ 2x_1 + 8x_2 + 3x_3 + 2x_4 &= 16 \\ x_1 + x_2 + 4x_3 + x_4 &= 8\,. \end{aligned}$$

a) Determine the associated augmented matrix for this system. Run `simpleGE` on this matrix to see that the algorithm fails to put this augmented matrix into echelon form. Explain why the algorithm fails to do so.

b) Though this would not normally be the case, the output from `simpleGE` for this system may be used to find all solutions to the system anyway. Do so.

1.32 Solve the two linear systems

$$\begin{aligned} x_1 + 2x_2 - 2x_3 &= 1 \\ 2x_1 + 5x_2 + x_3 &= 9 \\ x_1 + 3x_2 + 4x_3 &= 9 \end{aligned} \qquad \text{and} \qquad \begin{aligned} x_1 + 2x_2 - 2x_3 &= 9 \\ 2x_1 + 5x_2 + x_3 &= 9 \\ x_1 + 3x_2 + 4x_3 &= -2 \end{aligned}$$

by doing elimination on a 3-by-5 augmented matrix and then performing two back substitutions.

1.33 A well-known formula for the inverse of a 2-by-2 matrix

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \text{is} \qquad \mathbf{A}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Use Gaussian elimination (do it *by hand*) on the matrix $\mathbf{A}$ above to derive this formula for the inverse matrix $\mathbf{A}^{-1}$. Handle separately the following cases: I) $a \neq 0$, II) $a = 0$ but $c \neq 0$, and III) both $a, c = 0$. What does a nonzero determinant for $\mathbf{A}$ have to do with nonsingularity in this case?

1.34 Let $\mathbf{A} = (a_{ij})$, and suppose that $a_{11} \neq 0$ (i.e., it has a pivot in the first row, first column). Use subscript notation to write down

- **a)** the multiplier l_{i1} to be subtracted from row i during Gaussian elimination in order to zero out the entry in row i below this pivot.
- **b)** the new entry that replaces a_{ij} (j any column number) after that subtraction.
- **c)** the second pivot.

1.35 Show that the elementary matrix $\mathbf{E}_{ij}$ of Exercise 1.25 is invertible, and find the form of its inverse. You may assume, as is always the case when such elementary matrices are used in Gaussian elimination, that $i > j$.

1.36

- **a)** Suppose

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ a_{21} & 1 & 0 \\ a_{31} & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & a_{32} & 1 \end{bmatrix}.$$

Find a general expression (give the entries) for each of

(i) $\mathbf{A}_1\mathbf{A}_2$ (iii) $\mathbf{A}_2\mathbf{A}_1$

(ii) $(\mathbf{A}_1\mathbf{A}_2)^{-1}$ (iv) $(\mathbf{A}_2\mathbf{A}_1)^{-1}$

b) Suppose

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_{21} & 1 & 0 & 0 \\ a_{31} & 0 & 1 & 0 \\ a_{41} & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & a_{32} & 1 & 0 \\ 0 & a_{42} & 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & a_{43} & 1 \end{bmatrix}.$$

Find a general expression (give the entries) for each of

(i) $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$ (iii) $\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1$

(ii) $(\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3)^{-1}$ (iv) $(\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1)^{-1}$

c) What special feature does the calculation of $\mathbf{A}_1\mathbf{A}_2$ and $(\mathbf{A}_2\mathbf{A}_1)^{-1}$ (in part (a)) and $\mathbf{A}_1\mathbf{A}_2\mathbf{A}_3$ and $(\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1)^{-1}$ (in part (b)) have? State the corresponding result for arbitrary $n \geq 2$.

1.37 Prove that the product of lower triangular matrices is again lower triangular.

1.38 Use experimentation in OCTAVEto determine which properties of a nonsingular matrix **A** seem to hold also for its inverse: (1) **A** is triangular, (2) **A** is symmetric, (3) all entries in **A** are whole numbers, (4) all entries in **A** are fractions (rational numbers) or whole numbers.

1.39

a) We know that two points in a plane determine a unique line. When those points are not located at the same x-coordinate, the line will take the form

$$p(x) = mx + b\,,$$

a polynomial of degree at most one. Under what conditions on the points would this really be a polynomial of degree zero?

b) If you remember anything about your study of Simpson's Rule in MATH 162, you may suspect that, when three points no two of which share the same x-coordinate are specified in the plane, there is a unique polynomial

$$p(x) \;=\; ax^2 + bx + c\,,$$

having degree at most 2, that passes through the three points. This statement is, indeed, true. One might say the polynomial p **interpolates** the given points, in that it passes through them filling in the gaps between.

i. Write the similar statement that applies to a set of n points in the plane, no two of which share the same x-coordinate.

ii. Consider the problem of finding the smallest degree polynomial that interpolates the n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ in the plane. Once the coefficients $a_0, a_1, \ldots, a_{n-1}$ of

$$p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$$

are found, we are done. The information we have at our disposal to find these coefficients is that

$$p(x_1) = y_1 , \quad p(x_2) = y_2 , \quad \ldots , \quad p(x_n) = y_n .$$

That is, we have n equations to determine the n unknowns. Find the matrix $\mathbf{B}$ so that the problem of finding the coefficients of p is equivalent to solving the matrix problem

$$\mathbf{B} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} .$$

c) Use OCTAVE and your answer to the previous part to find the coefficients of the polynomial that interpolates the six points $(-2, -63)$, $(-1, 3)$, $(0, 1)$, $(1, -3)$, $(2, 33)$, and $(3, 367)$.

1.40 Some texts talk not only about an *LU*-factorization of a matrix, but of an *LDU*-factorization, where $\mathbf{D}$ is a diagonal matrix.

a) Under what conditions is $\mathbf{A}$ nonsingular, if $\mathbf{A}$ is the product

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} d_1 & & \\ & d_2 & \\ & & d_3 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} ?$$

b) Using $\mathbf{A}$ as factored above, solve the system $\mathbf{Ax} = \mathbf{b}$ starting with $\mathbf{Lc} = \mathbf{b}$:

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{b} .$$

1.41 Solve $\mathbf{LUx} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$ without multiplying $\mathbf{L}$ and $\mathbf{U}$ to find $\mathbf{A}$.

1.42 Write down all six of the 3-by-3 permutation matrices, including $\mathbf{P} = \mathbf{I}$. Identify their inverses—permutation matrices themselves which satisfy $\mathbf{P}\mathbf{P}^{-1} = \mathbf{I}$.

1.43 Below we have the output from Octave's `lu()` command for a particular matrix.

```
octave-3.0.0:177> A = [6 -4 5; -4 3 1; 2 -1 1];
octave-3.0.0:178> [L, U, P] = lu(A)
L =
   1.00000   0.00000   0.00000
  -0.66667   1.00000   0.00000
   0.33333   1.00000   1.00000

U =
   6.00000  -4.00000   5.00000
   0.00000   0.33333   4.33333
   0.00000   0.00000  -5.00000

P =
   1   0   0
   0   1   0
   0   0   1
```

Use it (and *not* some other means) to find all solutions to the linear system of equations

$$\begin{aligned} 6x - 4y + 5z &= -10 \\ -4x + 3y + z &= -1 \\ 2x - y + z &= -1\,. \end{aligned}$$

1.44 Below we have the output from Octave's `lu()` command for a particular matrix.

```
octave-3.0.0:157> [L, U, P] = lu([1 -2 3; 1 -4 -7; 2 -5 1])
L =
   1.00000   0.00000   0.00000
   0.50000   1.00000   0.00000
   0.50000  -0.33333   1.00000

U =
   2.00000  -5.00000   1.00000
   0.00000  -1.50000  -7.50000
   0.00000   0.00000   0.00000

P =
   0   0   1
```

```
     0   1   0
     1   0   0
```

Use it (and *not* some other means) to find all solutions to the linear system of equations

$$\begin{aligned} x - 2y + 3z &= -13 \\ x - 4y - 7z &= 1 \\ 2x - 5y + z &= -19\,. \end{aligned}$$

1.45 Here is a quick tutorial of how one might use Octave to produce the circle and oval in the figure from Example 1.6.4. First, to get the circle, we create a row vector of parameter values (angles, in radians) running from 0 to 2π in small increments, like 0.05. Then we create a 2-row matrix whose 1st row holds the x-coordinates around the unit circle (corresponding to the parameter values) and whose 2nd row contains the corresponding y-coordinates. We then plot the list of x-coordinates against the list of y-coordinates.

```
octave-3.0.0:121> t = 0:.05:2*pi;
octave-3.0.0:122> inPts = [cos(t); sin(t)];
octave-3.0.0:123> plot(inPts(1,:),inPts(2,:))
octave-3.0.0:124> axis(‘‘square’’)
```

The last command in the group above makes sure that spacing between points on the x- and y-axes look the same. (Try the same set of commands omitting the last one.) At this stage, if you wish to draw in some lines connecting the origin to individual points on this circle, you can do so. For instance, given that I chose spacing 0.05 between my parameter values, the "circle" drawn above really consists of 126 individual points (pixels), as evidenced by the commands

```
octave-3.0.0:125> length(t)
ans = 126

octave-3.0.0:126> size(inPts)
ans =
     2   126
```

So, choosing (in a somewhat haphazard fashion) to draw in vectors from the origin to the 32nd (green), 55th (red) and 111th (black) of these points, we can use the following commands (assuming that you have not closed the window containing the plot of the circle):

```
octave-3.0.0:147> hold on
octave-3.0.0:148> plot([0 inPts(1,32)], [0 inPts(2,32)], 'g')
octave-3.0.0:149> plot([0 inPts(1,55)], [0 inPts(2,55)], 'r')
octave-3.0.0:150> plot([0 inPts(1,111)], [0 inPts(2,111)], 'k')
octave-3.0.0:151> hold off
```

To get the corresponding oval, we need to multiply the vectors that correspond to the points on the circle (drawn using the commands above) by the **A** in Example 1.6.4.

```
octave-3.0.0:152> A = [2 1; 0 3];
octave-3.0.0:153> outPts = A*inPts;
octave-3.0.0:154> plot(outPts(1,:),outPts(2,:))
octave-3.0.0:155> axis("square")
```

Of course, if you want to know what point corresponds to any individual vector, you can explicitly ask for it. For instance, you can get the point **Av** on the oval corresponding to $\mathbf{v} = (-1/\sqrt{2}, 1/\sqrt{2})$ quite easily using the commands

```
octave-3.0.0:156> v = [-1/sqrt(2); 1/sqrt(2)]
v =
  -0.70711
   0.70711

octave-3.0.0:157> A*v
ans =
  -0.70711
   2.12132
```

To see **Av** for the three (colored) vectors we added to our circle's plot, you can use the commands (assuming the window containing the oval is the last plot your produced)

```
octave-3.0.0:157> subInPts = inPts(:,[32 55 111]);
octave-3.0.0:158> subOutPts = A*subInPts;
octave-3.0.0:159> hold on
octave-3.0.0:160> plot([0 subOutPts(1,1)], [0 subOutPts(2,1)], 'g')
octave-3.0.0:161> plot([0 subOutPts(1,2)], [0 subOutPts(2,2)], 'r')
octave-3.0.0:162> plot([0 subOutPts(1,3)], [0 subOutPts(2,3)], 'k')
octave-3.0.0:163> hold off
```

Use commands like these to help you answer the following questions.

a) Choose an angle $\alpha \in [0, 2\pi)$ and form the corresponding matrix **A** of the form (1.4). In Item 1 of Section 1.3 we established that multiplication by **A** achieves a rotation of the plane. Find the eigenvalues of **A**.

b) Consider the matrix **A** from Example 1.6.3. What are its eigenvalues? Describe as accurately as you can the way the plane $\mathbb{R}^2$ is transformed when vectors are multiplied by this **A**.

c) Still working with the matrix **A** from Example 1.6.3, write it as a product **A** = **BC**, where both matrices on the right side are 2-by-2, one of which has the form (1.4) and the other has the form (1.7). (Hint: If $(a + bi)$ is one of the eigenvalues of $\mathbf{A}$, then the quantity $\sqrt{a^2 + b^2}$ should come into play somewhere.)

d) Make your best effort to accurately finish this statement:

If $\mathbf{A}$ is a 2-by-2 matrix with complex eigenvalues $(a + bi)$ and $(a - bi)$ (with $b \neq 0$), then multiplication by $\mathbf{A}$ transforms the plane $\mathbb{R}^2$ by . . . .

1.46 Suppose $\mathbf{A}$ is a 2-by-2 matrix with real-number entries and having at least one eigenvalue that is real.

a) Explain how you know $\mathbf{A}$ has at most one other eigenvalue.

b) Can $\mathbf{A}$ have a non-real eigenvalue along with the real one? Explain.

c) Consider the mapping $(\mathbf{x} \mapsto \mathbf{Ax})\colon \mathbb{R}^2 \to \mathbb{R}^2$. Is it possible that, given the matrix $\mathbf{A}$, this function brings about a (rigid) rotation of the plane? Explain.

1.47 Write a matrix $\mathbf{A}$ such that, for each $\mathbf{v} \in \mathbb{R}^2$, $\mathbf{Av}$ is the reflection of $\mathbf{v}$

a) across the y-axis. Then use Octave to find the eigenpairs of $\mathbf{A}$.

b) across the line $y = x$. Use Octave to find the eigenpairs of $\mathbf{A}$.

c) across the line $y = (-3/4)x$. Use Octave to find the eigenpairs of $\mathbf{A}$.

d) across the line $y = (a/b)x$, where a, b are arbitrary real numbers with $b \neq 0$.

1.48

a) Write a 3-by-3 matrix $\mathbf{A}$ whose action on $\mathbb{R}^3$ is to reflect across the plane $x = 0$. That is, for each $\mathbf{v} \in \mathbb{R}^3$, $\mathbf{Av}$ is the reflection of $\mathbf{v}$ across $x = 0$. Use Octave to find the eigenpairs of $\mathbf{A}$.

b) Write a 3-by-3 matrix $\mathbf{A}$ whose action on $\mathbb{R}^3$ is to reflect across the plane $y = x$. (Hint: Your answer should be somehow related to your answer to part (b) of Exercise 1.47.) Use Octave to find the eigenpairs of $\mathbf{A}$.

c) Suppose P is a plane in 3D space containing the origin, and $\mathbf{n}$ is a normal vector to P. What, in general, can you say about the eigenpairs of a matrix $\mathbf{A}$ whose action on $\mathbb{R}^3$ is to reflect points across the plane P?

1.49

a) Consider a coordinate axes system whose origin is always fixed at the Earth's center, and whose positive z-axis always passes through the North Pole. While the positive x- and y- axes always pass through the Equator, the rotation of the Earth causes the points of intersection to change, cycling back every 24 hours. Determine a 3-by-3 matrix $\mathbf{A}$ so that, given any $\mathbf{v} \in \mathbb{R}^3$ that specifies the current location of a point on (or in) the Earth relative to this coordinate system, $\mathbf{Av}$ is the location of this same point in 3 hours.

b) Repeat the exerise, but now assuming that, in every 3-hour period, the poles are 1% farther from the origin than they were before.

1.50 When connected in the order given, the points $(0,0)$, $(0.5,0)$, $(0.5,4.5)$, $(4,4.5)$, $(4,5)$, $(0.5,5)$, $(0.5,7.5)$, $(5.5,7.5)$, $(5.5,8)$, $(0,8)$ and $(0,0)$ form the letter 'F', lying in Quadrant I with the bottom of the stem located at the origin.

a) Give Octave commands that produce a plot of the letter with the proper aspect. (Include among them the command you use to store the points, doing so not storing the points themselves, but their corresponding homogeneous coordinates, storing them as `hPts`.)

b) What 3-by-3 matrix would suffice, via matrix multiplication, to translate the letter to Quadrant III, with its top rightmost point at the origin? Give Octave commands that carry out this transformation on `hPts` and produce the plot of the letter in its new position.

c) What 3-by-3 matrix would suffice, via matrix multiplication, to rotate the letter about its effective center (the point $(2.75,4)$), so that it still lies entirely in Quadrant I, but is now upside down? Give Octave commands that carry out this transformation on `hPts` and produce the plot of the letter in its new position.

d) Extract the original points from their homogeneous coordinates with the command

```
octave-3.0.0:31> pts = hPts(1:2,:);
```

Now consider 2-by-2 matrices of the form

$$\mathbf{A} = \begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}.$$

Choose several different values of c, run the command

```
octave-3.0.0:32> chpts = A*pts;
```

and observe the effect by plotting the altered points (found in `chpts`). These matrices are called **shear matrices**. For each $\mathbf{A}$ you try, find the eigenpairs of $\mathbf{A}$. Summarize your observations about the effect of shear matrices on the letter, and what you note about the eigenpairs.

1.51

a) Suppose $\mathbf{D}$ is a diagonal matrix with entries along the main diagonal $d_1, \ldots, d_n$. Suppose also that $\mathbf{A}$, $\mathbf{S}$ are n-by-n matrices with $\mathbf{S}$ nonsingular, such that the equation $\mathbf{AS} = \mathbf{SD}$ is satisfied. If $\mathbf{S}_j$ denotes the j^{th} column (a vector in $\mathbb{R}^n$) of $\mathbf{S}$, show that each $(d_j, \mathbf{S}_j)$ is an eigenpair of $\mathbf{A}$.

b) Find a matrix $\mathbf{A}$ for which $(4, 1, 0, -1)$ is an eigenvector corresponding to eigenvalue (-1), $(1, 2, 1, 1)$ is an eigenvector corresponding to eigenvalue 2, and both $(1, -1, 3, 3)$ and $(2, -1, 1, 2)$ are eigenvectors corresponding to eigenvalue 1. (You may use Octave for this part, supplying your code and using commentary in identifying the result.)

c) Show that, under the conditions of part (a), $\det(A) = \prod_{j=1}^{n} d_j$. That is, $\det(A)$ is equal to the product of the eigenvalues of $\mathbf{A}$. (This result is, in fact, true even for square matrices $\mathbf{A}$ which do not have this form.)

d) Two square matrices $\mathbf{A}$, $\mathbf{B}$ are said to be **similar** if there is an invertible matrix $\mathbf{P}$ for which $\mathbf{B} = \mathbf{P}^{-1}\mathbf{AP}$. Show that, if λ is an eigenvalue of $\mathbf{B}$, then it is also an eigenvalue of $\mathbf{A}$.

1.52 Prove that if any diagonal element (i.e., a, d and f) of

$$\mathbf{A} = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix}$$

is zero, then the rows are linearly dependent.

1.53 Is it true that if $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$ are linearly independent, then also the vectors $\mathbf{w}_1 = \mathbf{v}_1 + \mathbf{v}_2$, $\mathbf{w}_2 = \mathbf{v}_1 + \mathbf{v}_3$, $\mathbf{w}_3 = \mathbf{v}_2 + \mathbf{v}_3$ are linearly independent? Hint: Start with some combination $c_1\mathbf{w}_1 + c_2\mathbf{w}_2 + c_3\mathbf{w}_3 = \mathbf{0}$, and determine which c_j are possible.

1.54 Suppose $\mathbf{A}$ is an m-by-n matrix. Explain why it is not possible for rank($\mathbf{A}$) to exceed m. Deduce that rank($\mathbf{A}$) cannot exceed the minimum value of m and n.

1.55 Give an example of an m-by-n matrix $\mathbf{A}$ for which you can tell at a glance $\mathbf{Ax} = \mathbf{b}$ is not always consistent—that is, there are right-hand side vectors $\mathbf{b} \in \mathbb{R}^m$ for which no solution exists.

1.56 Let

$$\mathbf{A} := \begin{bmatrix} 1 & -2 & 1 & 1 & 2 \\ -1 & 3 & 0 & 2 & -2 \\ 0 & 1 & 1 & 3 & 4 \\ 1 & 2 & 5 & 13 & 5 \end{bmatrix}.$$

a) Use Gaussian elimination (you may use `simpleGE`, or the (better) alternative called `rref()`) to find the rank and nullity of **A**.

b) Find a basis for the column space of **A**.

c) State another way to phrase the question of part (b) that employs the words "linear independent" and "span".

1.57 Suppose **A** is an m-by-n matrix, with $m > n$ and $\text{null}(\mathbf{A}) = \{\mathbf{0}\}$.

a) Are the column vectors of **A** linearly independent? How do you know?

b) How many solutions are there to the matrix equation $\mathbf{Ax} = \mathbf{b}$ if $\mathbf{b} \in \text{ran}(\mathbf{A})$?

c) How many solutions are there to the matrix equation $\mathbf{Ax} = \mathbf{b}$ if $\mathbf{b} \notin \text{ran}(\mathbf{A})$?

1.58 Can a nonzero matrix (i.e., one not completely full of zero entries) be of rank 0? Explain.

1.59 We know that, for an m-by-1 vector **u** and 1-by-n matrix (row vector) **v**, the matrix product **uv** is defined, yielding an m-by-n matrix sometimes referred to as the **outer product** of **u** and **v**. In Section 1.2 we called this product a **rank-one matrix**. Explain why this term is appropriate.

1.60 Determine whether the given set of vectors is linearly independent.

a) $S = \{(3, 2, 5, 1, -2), (5, 5, -2, 0, 1), (2, 2, 6, -1, -1), (0, 1, 4, 1, 2)\}$

b) $S = \{(3, 6, 4, 1), (-1, -1, 2, 5), (2, 1, 3, 0), (6, 13, 0, -8)\}$

1.61 For the matrix **A**, find its nullspace: $\mathbf{A} = \begin{bmatrix} 3 & 6 & 4 & 1 \\ -1 & -1 & 2 & 5 \\ 2 & 1 & 3 & 0 \\ 6 & 13 & 0 & -8 \end{bmatrix}$

1.62 Octave has a command `rank()` which returns a number it thinks equals the rank of a matrix. (Type `help rank` to see how to use it.) The command can be used on square and non-square matrices alike. Use Octave commands to find both the rank and determinant of the following *square* matrices:

(i) $\begin{bmatrix} 2 & 5 & -5 \\ 7 & 0 & 7 \\ -4 & 7 & 0 \end{bmatrix}$

(ii) $\begin{bmatrix} -10 & 0 & 5 & 7 \\ -5 & 3 & -3 & 9 \\ 7 & 7 & -1 & 6 \\ 0 & -9 & -3 & 1 \end{bmatrix}$

(iii) $\begin{bmatrix} 3 & 3 & 0 & -1 & 3 \\ -4 & -2 & 1 & 1 & 5 \\ 1 & -1 & -4 & -5 & -2 \\ 4 & 3 & -3 & 1 & 0 \\ 1 & 5 & 5 & 4 & 1 \end{bmatrix}$

(iv) $\begin{bmatrix} -6 & 5 & 6 & -7 & -2 \\ 4 & -7 & -2 & 5 & 5 \\ -2 & -2 & 4 & -2 & 3 \\ -10 & 12 & 8 & -12 & -7 \\ -8 & 3 & 10 & -9 & 1 \end{bmatrix}$

Using these results, write a statement that describes, for *square* matrices, what knowledge of one of these numbers (the rank or determinant) tells you about the other.

1.63 Theorem 1.7.5 tells of many things one can know about a square matrix when is has **full rank**—that is, rank(**A**) = n for a matrix **A** with n columns. Look back through that theorem, and determine which of the conditions (i)–(vii) still hold true when **A** is non-square but has full rank.