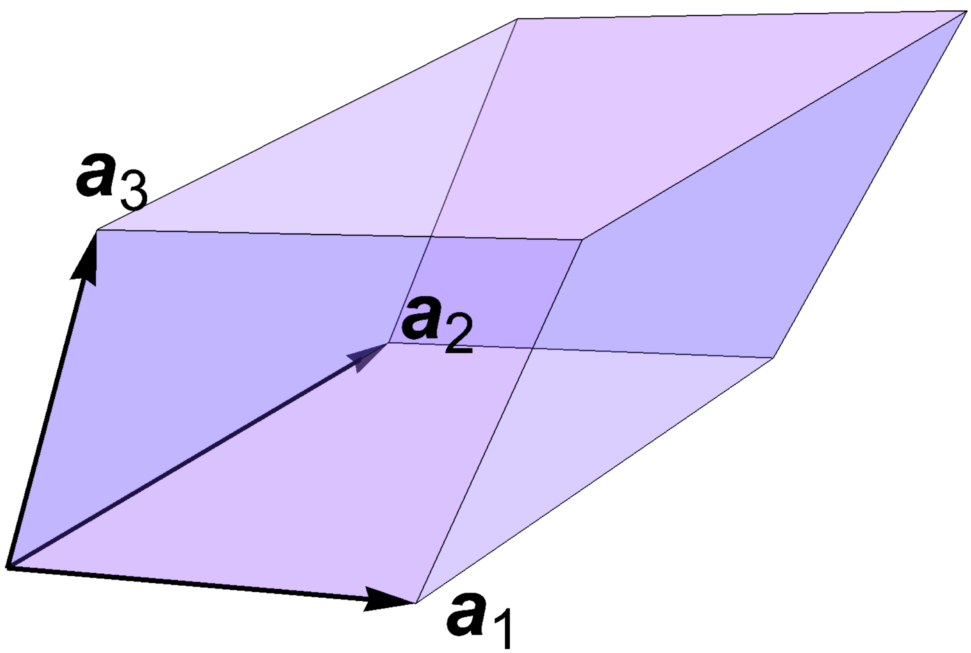
A Crash Course in Linear Algebra for Differential Equations
Part I: Matrices, Systems, and Determinants

Katharine Long
Department of Mathematics and Statistics
Texas Tech University

January 9, 2022



Volume of V = $\det([\boldsymbol{a}_1|\boldsymbol{a}_2|\boldsymbol{a}_3])$

Contents

1 What you're assumed to know **4**

2 Notation **5**
- 2.1 Sets . . . 5
- 2.2 Vectors . . . 5
  - 2.2.1 Vector arithmetic . . . 6
  - 2.2.2 Dot product . . . 6
  - 2.2.3 Magnitude . . . 6
- 2.3 Matrices . . . 7

3 Linear combinations and matrix-vector multiplication **7**
- 3.1 Core definitions . . . 7
  - 3.1.1 Linear combination . . . 7
  - 3.1.2 Vector spaces and subspaces . . . 8
  - 3.1.3 Matrix-vector multiplication . . . 8
  - 3.1.4 Linear dependence and independence . . . 9
  - 3.1.5 Null space and range space . . . 9
  - 3.1.6 Basis . . . 10
- 3.2 Examples . . . 10
  - 3.2.1 Span . . . 10
  - 3.2.2 Matrix-vector multiplication . . . 11
  - 3.2.3 Linear dependence\independence . . . 11
  - 3.2.4 Null space . . . 12
  - 3.2.5 Range space . . . 12
  - 3.2.6 Basis . . . 13

4 Systems of equations **13**
- 4.1 Homogeneous equations . . . 13
- 4.2 Nonhomogeneous equations . . . 16
  - 4.2.1 Nonsingular A: unique solution . . . 16
  - 4.2.2 Singular A; no solution . . . 17
  - 4.2.3 Singular A; infinitely many solutions . . . 18
- 4.3 Superposition . . . 19

5 Matrix-matrix multiplication and the inverse **20**
- 5.1 The matrix-matrix multiplication formula . . . 21
  - 5.1.1 An alternate derivation . . . 21
  - 5.1.2 Properties of matrix multiplication . . . 22

5.1.3 The identity matrix and the Kronecker delta . . . 22
5.2 The matrix inverse . . . 23
5.2.1 Constructing the inverse . . . 24
5.2.2 Don't compute an inverse . . . 25
5.2.3 Some properties of the inverse . . . 26

6 Some special matrices **27**
6.1 Orthogonal and unitary matrices . . . 27
6.2 Permutation matrices . . . 28
6.3 Triangular matrices . . . 29
6.3.1 Diagonal matrices . . . 29

7 The determinant **29**
7.1 The determinant as an oriented hypervolume . . . 29
7.2 Properties of the determinant . . . 30
7.2.1 The determinant of the identity is one . . . 30
7.2.2 The determinant of a product is the product of the determinants . . . 30
7.2.3 If A contains a zero column (or row) then $\det(A) = 0$ . . . 31
7.2.4 The determinant of a permutation matrix is ± 1 . . . 31
7.2.5 $\det(A) = 0$ iff A is singular . . . 31
7.2.6 The determinant of an inverse . . . 32
7.2.7 Summary of determinants . . . 33
7.3 Why you should *never* use a floating point determinant calculation to check for singularity of a matrix . . . 33
7.4 Cramer's Rule . . . 34

8 Putting it all together: the Big Matrix Theorem **34**

A Important notation **35**
A.1 Abbreviations . . . 35
A.2 Mathematical notation . . . 35

B Formal definition of a vector space **36**

C Three ways of looking at matrix-vector multiplication **37**

D Refresher: The area of a parallelogram **37**

E Refresher: Gaussian elimination **38**
E.1 Simplest case: no row exchanges . . . 39
E.2 Elimination with row exchanges . . . 39
E.3 When Gaussian elimination fails . . . 40
E.3.1 Consistent versus inconsistent singular systems . . . 41

F Practical algorithms for solving systems of equations **42**
F.1 Row pivoting . . . 43
F.2 The LU factorization . . . 43
F.3 Why you shouldn't compute a matrix inverse . . . 44

G Computing determinants numerically **45**
G.1 How to wait a lifetime and get the wrong answer: expansion in minors . . . 46

H Further reading **47**

References **47**

These notes are not a substitute for a linear algebra course. They're intended to be a bare minimum to get you ready to use matrix methods for differential equations and related application in science and engineering. There's no attempt to cover every topic in full generality or depth.

1 What you're assumed to know

1. Vector algebra: addition, subtraction, multiplication by scalars, and the dot product. Refer to your calculus books and notes now if you need a refresher.

2. Complex numbers, arithmetic, and conjugation. You should have seen these in either calculus or in your first course in differential equations. Look back at your books and notes for a refresher if needed.

3. Solving systems of linear equations by row reduction (Gaussian elimination). You might not have seen elimination presented as a systematic algorithm; if that's the case, then I refer you to Appendix E for a brief introduction and for references to books and online resources. You should recall the following:

   - If row reduction (Gaussian elimination) on a square matrix A succeeds without producing a zero on the diagonal, then A is called ***nonsingular***, and the solution to $A\mathbf{x} = \mathbf{b}$ can be found by ***backsubstitution***. Otherwise A is called ***singular***.

4. Computation of determinants by expansion in minors (also known as a Laplace expansion). You'll have seen Jacobians in multivariable calculus and Wronskians in your first differential equations course; whether you realized it or not, you were computing these using expansion of a determinant in minors. Review these as needed; we'll use determinants in the hand calculation of eigenvalues for small matrices.

2 Notation

2.1 Sets

Members of a ***set*** will be written in curly braces. We can describe sets in several ways, such as

- Listing the members explicitly: $S = \{s_1, s_2, \cdots, s_n\}$
- Listing members implicitly with indexing: $S = \{s_j\}_{j=1}^{n}$
- Giving conditions for membership in the set: $S = \{s : \text{ condition for } s \text{ to be a member}\}$. For example:
$$S = \{s : s = 2k,\ k = 1, 2, 3, \cdots\}$$
is the set of even numbers.

2.2 Vectors

This should be familiar from working with 2D and 3D vectors; I'm writing everything out to establish notation and to generalize to complex vectors. The space$\mathbb{R}^m$ is the m-dimensional real space, the m-dimensional generalization of the familiar 3D position space of classical physics and everyday experience. A point $\mathbf{x} \in \mathbb{R}^m$ is described with an ordered m-tuple that we organize as a column vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}. \tag{1}$$

The space $\mathbb{C}^m$ is an m-dimensional space of complex numbers.

Transposing a vector – denoted with a superscipt T – turns a column into a row or vice-versa; the vector in equation 1 can be written

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T.$$

To save space I'll often write vectors in this way, as transposed rows. When working with complex vectors, we'll find need for the complex conjugate of the transpose, the ***conjugate transpose***. This will be important enough to get its own symbol, a superscript $*$:

$$\mathbf{x}^* = \begin{bmatrix} \overline{x}_1 & \overline{x}_2 & \cdots & \overline{x}_m \end{bmatrix}^T.$$

Recall that if $z = a + ib$, its conjugate is $\overline{z} = a - ib$. A superscript "dagger" † is sometimes used to denote the conjugate transpose. We'll use the asterisk ("star") instead of the dagger.

The ***standard unit vectors*** in $\mathbb{R}^m$ or $\mathbb{C}^m$ are

$$\begin{aligned}
\hat{\boldsymbol{e}}_1 &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \\
\hat{\boldsymbol{e}}_2 &= \begin{bmatrix} 0 & 1 & \cdots & 0 \end{bmatrix}^T \\
&\vdots \\
\hat{\boldsymbol{e}}_m &= \begin{bmatrix} 0 & 0 & \cdots & 1 \end{bmatrix}^T .
\end{aligned}$$

2.2.1 Vector arithmetic

Vectors add componentwise, *e.g.*

$$[x_1, x_2]^T + [y_1, y_2]^T = [x_1 + y_1, x_2 + y_2]^T ,$$

and scalars multiply into all components of a vector, *e.g.*,

$$\alpha [x_1, x_2]^T = [\alpha x_1, \alpha x_2]^T .$$

Vectors of different sizes can't be added, *e.g.*,

$$[x_1, x_2]^T + [y_1, y_2, y_3]^T$$

is nonsense.

2.2.2 Dot product

The dot product between two vectors in $\mathbb{R}^m$ is

$$\mathbf{x} \cdot \mathbf{y} = \sum_{j=1}^{m} x_j y_j .$$

We won't use dot product notation much; we'll usually write the same expression as $\mathbf{x}^T \mathbf{y}$. With complex vectors, the *conjugate* transpose should be normally be used:

$$\mathbf{x}^* \mathbf{y} = \sum_{j=1}^{m} \overline{x}_j y_j .$$

2.2.3 Magnitude

The ***magnitude*** of a vector $\mathbf{x} \in \mathbb{R}^m$ is denoted $\|\mathbf{x}\|$, and is computed using the Pythagorean formula

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_m^2} = \sqrt{\mathbf{x}^T \mathbf{x}} .$$

If the vector is complex, $\mathbf{x} \in \mathbb{C}^m$, then this is slightly modified, using the conjugate transpose in place of the transpose,

$$\|\mathbf{x}\| = \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_m|^2} = \sqrt{\mathbf{x}^* \mathbf{x}} .$$

Recall that the magnitude of a complex number $x = a + ib$ is $|x| = \sqrt{a^2 + b^2} = \sqrt{z\overline{z}}$. If $x_j = a_j + ib_j$, then

$$\|\mathbf{x}\| = \sqrt{a_1^2 + b_1^2 + a_2^2 + b_2^2 + \cdots + a_m^2 + b_m^2} .$$

2.3 Matrices

Given m vectors $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n\}$ each in $\mathbb{C}^m$, we can form an $m \times n$ ***matrix*** A by putting the vector $\mathbf{a}_j$ into the j-th column:

$$A = \left[\ \mathbf{a}_1 \,\middle|\, \mathbf{a}_2 \,\middle|\, \cdots \,\middle|\, \mathbf{a}_n\ \right].$$

If we denote the i-th entry in $\mathbf{a}_j$ by A_{ij}, then the matrix A is written

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & A_{22} & \\ \vdots & & \ddots \end{bmatrix}.$$

Like vectors, matrices add componentwise and a scalar multiplying into a matrix multiplies all components:

$$\begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & A_{22} & \\ \vdots & & \ddots \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} & \cdots \\ B_{21} & B_{22} & \\ \vdots & & \ddots \end{bmatrix} = \begin{bmatrix} A_{11}+B_{11} & A_{12}+B_{12} & \cdots \\ A_{21}+B_{21} & A_{22}+B_{22} & \\ \vdots & & \ddots \end{bmatrix}$$

and

$$\alpha \begin{bmatrix} A_{11} & A_{12} & \cdots \\ A_{21} & A_{22} & \\ \vdots & & \ddots \end{bmatrix} = \begin{bmatrix} \alpha A_{11} & \alpha A_{12} & \cdots \\ \alpha A_{21} & \alpha A_{22} & \\ \vdots & & \ddots \end{bmatrix}.$$

The set of all $m \times n$ real matrices is called $\mathbb{R}^{m\times n}$. The analogue for complex matrices is $\mathbb{C}^{m\times n}$.

3 Linear combinations and matrix-vector multiplication

3.1 Core definitions

Now that notation has been described, we can establish some definitions that will be fundamental to our study of linear algebra and differential equations. I'll write out all the definitions with some commentary, and only afterwards give some examples.

In the definitions that follow, S is the set of vectors $S = \{\mathbf{a}_j\}_{j=1}^n$ with each $\mathbf{a}_j \in \mathbb{C}^m$. All the definitions will be stated in terms of complex spaces; to restrict the definitions to real spaces, replace $\mathbb{C}$ by $\mathbb{R}$.

3.1.1 Linear combination

This is a term we'll use often, so we might as well define it formally.

Definition 1 A ***linear combination*** (LC) of the members of S is a scalar-weighted sum such as

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

where the n coefficients $\{x_j\}_{j=1}^{n}$ are scalars. The ***span*** of S, denoted $\operatorname{span}(S)$, is the set of all linear combinations that can be formed from the members of S.

3.1.2 Vector spaces and subspaces

Definition 2 A set V of vectors is called a ***vector space*** if it obeys the axioms listed in Appendix B. We needn't go into all the axioms for this course; as long as one defines addition and scalar multiplication "reasonably," then it is sufficient to consider the ***closure conditions***

- $\mathbf{x} + \mathbf{y} \in U$ for all $\mathbf{x}, \mathbf{y} \in U$
- $\alpha \mathbf{x} \in U$ for all $\mathbf{x} \in U$ and for all scalars α

and the ***zero condition***,

- U contains the zero vector $\mathbf{0}$.

A sub*set* of V that is itself a vector space is called a ***subspace***.

The closure conditions mean that if you form a LC of members of a subspace U, the result must be in U. Like they say about Las Vegas, whatever happens in a vector space stays in that vector space. You'd have to use some imagination to come up with a set that obeys the closure and zero conditions but isn't a vector space, and such cases aren't anything we'd encounter in practice.

I will assert without proof that the sets $\mathbb{R}^n$ and $\mathbb{C}^n$ are vector spaces. A useful result, also stated without proof, is the ***subspace theorem***.

Theorem 1 (Subspace theorem) Let V be a vector space, and let U be a subset of V. If U satisfies the closure conditions from Definition 2, then U is a vector space (and therefore a subspace of V).

For subspaces, the zero condition is implied by the closure conditions (to prove this, consider the operations $\mathbf{x} - \mathbf{x}$ and $0\mathbf{x}$). Often the easiest way to show something is *not* a subspace is to check the zero condition.

3.1.3 Matrix-vector multiplication

There are several equivalent[1] ways to define matrix vector multiplication (see Appendix C). For our purposes, it will be most useful to think of multiplication of a matrix A into a vector $\mathbf{x}$ as forming a linear combination of the columns of A, weighted by the entries in $\mathbf{x}$.

[1] The equivalent definitions of matrix-vector multiplication will all produce the same results. We choose the one that is most useful for *understanding* how to use matrices for solving differential equations.

Definition 3 Form the matrix $A \in \mathbb{C}^{m \times n}$ by packing the members of S into its columns. Given a vector $\mathbf{x} \in \mathbb{C}^n$, the ***matrix-vector product*** $A\mathbf{x}$is the linear combination of the $\mathbf{a}_j$'s weighted by the entries in $\mathbf{x}$:

$$A\mathbf{x} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n.$$

3.1.4 Linear dependence and independence

Definition 4 The set S is called ***linearly dependent*** (LD) if there is a *nonzero* vector $\mathbf{x} \in \mathbb{C}^n$ such that

$$\mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n = \mathbf{0}.$$

If no such set of nonzero scalars exists, then we call S ***linearly independent*** (LI).

If you've taken DE1, you're familiar with linear independence of sets of functions; this definition in terms of vectors is identical, but with vectors in place of functions.

3.1.5 Null space and range space

Definition 5 The ***null space*** (sometimes called the ***kernel***) of a matrix $A \in \mathbb{C}^{m \times n}$, denoted $\operatorname{null}(A)$, is the set of vectors $\mathbf{x} \in \mathbb{C}^n$ such that $A\mathbf{x} = \mathbf{0}$. That is, $\operatorname{null}(A)$ is the set of all solutions to the homogeneous equation $A\mathbf{x} = \mathbf{0}$. Notice that $\operatorname{null}(A)$ always contains the zero vector. If a null space contains *only* the zero vector, that null space is called ***trivial***; otherwise, it is called ***nontrivial***.

The use of the word "space" in this definition isn't casual: By working with the subspace theorem and the definition of $\operatorname{null}(A)$, it is not hard to show that $\operatorname{null}(A)$ is a subspace of $\mathbb{C}^n$.

Linear independence/dependence and the null space are intimately related: from the definitions of LI and the null space, we see that $\operatorname{null}(A)$ is trivial iff the columns of A are LI.

Preview: the *eigenvalue problem* will be to identify when a certain matrix has a nontrivial null space, and then finding the members of that null space. The eigenvalue problem will be essential in solving systems of ODEs and also PDEs, so perk up your ears whenever you hear "null space."

Definition 6 The ***range space*** or ***column space*** of a matrix A, denoted $\operatorname{range}(A)$ or $\operatorname{col}(A)$, is the set of all linear combinations of the columns of A:

$$\operatorname{range}(A) = \{\mathbf{y} : \exists \mathbf{x} \text{ such that } \mathbf{y} = A\mathbf{x}\}.$$

Notice that we can also view this as $\operatorname{range}(A) = \operatorname{span}(\text{columns of } A)$.

The range space of A is the set of vectors that can be "reached" by matrix-vector multiplication by A. The theorem follows immediately from the definition.

Theorem 2 The problem $A\mathbf{x} = \mathbf{b}$ has a solution iff $\mathbf{b} \in \operatorname{range}(A)$

As with the null space, the range space is a vector space. In particular, $\operatorname{range}(A)$ is a subspace of $\mathbb{C}^m$. Notice that $\operatorname{null}(A)$ is a subspace of $\mathbb{C}^n$ while $\operatorname{range}(A)$ is a subspace of $\mathbb{C}^m$; when A isn't square, $\mathbb{C}^m$ and $\mathbb{C}^n$ aren't the same spaces.

Notice that $\mathbf{0}$ is in the range of every matrix, with preimage $\mathbf{x} = \mathbf{0}$.

3.1.6 Basis

Definition 7 A set B is a ***basis*** for a vector space V if B is LI and $\text{span}\,(B) = V$.

3.2 Examples

3.2.1 Span

Example 1

Here are some examples set in $\mathbb{R}^2$

- The span of $\{\hat{e}_1, \hat{e}_2\}$ is all of $\mathbb{R}^2$, since every 2D vector can be written as

$$\mathbf{x} = x_1\hat{e}_1 + x_2\hat{e}_2.$$

- The span of the one-member set $\{\hat{e}_1 + 2\hat{e}_2\}$ is the line

$$\mathbf{x} = (\hat{e}_1 + 2\hat{e}_2)\, s = \begin{bmatrix} 1 \\ 2 \end{bmatrix} s.$$

- The span of $\{\hat{e}_1, \hat{e}_1 + 2\hat{e}_2\}$ is $\mathbb{R}^2$, since an arbitrary LC of the members is

$$\mathbf{x} = \hat{e}_1 s + (\hat{e}_1 + 2\hat{e}_2)\, t = \begin{bmatrix} s+t \\ s \end{bmatrix}.$$

  Any vector in $\mathbb{R}^2$ can be represented in this form: given x_1 and x_2, choose $s = x_2$ and $t = x_1 - s$.

- The span of $\{\hat{e}_1, \hat{e}_2, \hat{e}_1 + \hat{e}_2\}$ is also $\mathbb{R}^2$. An arbitrary LC of the members is

$$\mathbf{x} = \hat{e}_1 s + \hat{e}_2 t + (\hat{e}_1 + \hat{e}_2)\, u = \begin{bmatrix} s+u \\ t+u \end{bmatrix}.$$

  Given any x_1,x_2, choose $s + u = x_1$, $t + u = x_2$. There are infinitely many choices of s, t, and u that all represent the same vector.

Example 2

In $\mathbb{R}^3$, the span of $\{\hat{e}_1 + \hat{e}_2, \hat{e}_3\}$ is the plane containing the points

$$\mathbf{x} = \begin{bmatrix} s \\ s \\ t \end{bmatrix}.$$

3.2.2 Matrix-vector multiplication

Example 3

Let $\mathbf{a}_1 = \begin{bmatrix} -1 & 5 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 2 & -3 \end{bmatrix}^T$. Form

$$A = \begin{bmatrix} \mathbf{a}_1 \mid \mathbf{a}_2 \end{bmatrix}^T = \begin{bmatrix} -1 & 2 \\ 5 & -3 \end{bmatrix},$$

and multiply by $\mathbf{x} = \begin{bmatrix} 4 & 7 \end{bmatrix}^T$. Compute the matrix-vector product

$$A\mathbf{x} = \begin{bmatrix} -1 & 2 \\ 5 & -3 \end{bmatrix} \begin{bmatrix} 4 \\ 7 \end{bmatrix} = 4 \begin{bmatrix} -1 \\ 5 \end{bmatrix} + 7 \begin{bmatrix} 2 \\ -3 \end{bmatrix} = \begin{bmatrix} 10 \\ -1 \end{bmatrix}.$$

Example 4

Let $\mathbf{a}_1 = \begin{bmatrix} -1 & 5 & 2 \end{bmatrix}^T$, $\mathbf{a}_2 = \begin{bmatrix} 1 & 0 & 3 \end{bmatrix}^T$, and $\mathbf{x} = \begin{bmatrix} 4 & 7 \end{bmatrix}^T$. Then $A\mathbf{x}$ is

$$A\mathbf{x} = \begin{bmatrix} -1 & 1 \\ 5 & 0 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 4 \\ 7 \end{bmatrix} = 4 \begin{bmatrix} -1 \\ 5 \\ 2 \end{bmatrix} + 7 \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} = \begin{bmatrix} 3 \\ 20 \\ 29 \end{bmatrix}.$$

3.2.3 Linear dependence\independence

Example 5

The vectors $\mathbf{a}_1 = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 5 & 10 \end{bmatrix}^T$ are LD, because the linear combination with the scalars $x_1 = -5$, $x_2 = 1$, is zero: $-5\mathbf{a}_1 + \mathbf{a}_2 = \mathbf{0}$. Notice that there are, in fact, infinitely many such sets of scalars: any pair of scalars such that $x_1 = -5x_2$ will work.

Example 6

The vectors $\mathbf{a}_1 = \begin{pmatrix} 1 & 2 \end{pmatrix}^T$ and $\mathbf{a}_2 = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ are LI. To show this, suppose there were a pair of *nonzero* constants x_1, x_2 such that $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 = 0$. To find them, we would write out the equation

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 0 \end{bmatrix} x_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

which is equivalent to the system of equations

$$x_1 + x_2 = 0$$

$$2x_1 = 0.$$

The only solution to this system is $x_1 = 0$, $x_2 = 0$. Therefore the vectors are LI.

3.2.4 Null space

We'll revisit examples 3.2.3 and 3.2.3 from the point of view of the null space.

Example 7

Using the vectors from example 3.2.3, form the matrix

$$A = \begin{bmatrix} 1 & 5 \\ 2 & 10 \end{bmatrix}.$$

Then $\text{null}\,(A)$ consists of all vectors $\mathbf{x} = s \begin{bmatrix} -5 & 1 \end{bmatrix}^T$ where s is any real number.

Example 8

Using the vectors from example 3.2.3, form the matrix

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

In example 3.2.3 we saw that $x_1\mathbf{a}_1 + x_2\mathbf{a}_2$ has only the trivial solution $\mathbf{x} = \mathbf{0}$. Therefore $\text{null}\,(A) = \{\mathbf{0}\}$: the null space is trivial.

3.2.5 Range space

We next revisit examples 3.2.3 and 3.2.3 from the point of view of the range space.

Example 9

Using the vectors from example 3.2.3, form the matrix

$$A = \begin{bmatrix} 1 & 5 \\ 2 & 10 \end{bmatrix}.$$

Then $\text{range}\,(A)$ consists of all vectors $\mathbf{y}$ that can be formed by matrix-vector multiplication with A; equivalently, it is $\text{span}\,(\{\mathbf{a}_1, \mathbf{a}_2\})$. Choose an arbitrary vector $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ and multiply A into it,

$$A\mathbf{x} = x_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + x_2 \begin{bmatrix} 5 \\ 10 \end{bmatrix} = \begin{bmatrix} x_1 + 5x_2 \\ 2x_1 + 10x_2 \end{bmatrix} = (x_1 + 5x_2) \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Conclude that $\text{range}\,(A)$ is the set of all scalar multiples of $\begin{bmatrix} 1 & 2 \end{bmatrix}^T$.

Example 10

Using the vectors from example 3.2.3, form the matrix

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

Let $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ and multiply A into it,

$$A\mathbf{x} = x_1 \begin{bmatrix} 1 \\ 2 \end{bmatrix} + x_2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ 2x_1 \end{bmatrix}.$$

Now pick any $\mathbf{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$, and set $x_1 = \frac{1}{2}y_2$, $x_2 = y_1 - \frac{1}{2}y_2$; we have $A\mathbf{x} = \mathbf{y}$. Since any $\mathbf{y} \in \mathbb{R}^2$ has a preimage $\mathbf{x} \in \mathbb{R}^2$, we have that range $(A) = \mathbb{R}^2$.

3.2.6 Basis

Example 11

We recall the vectors from example 3.2.3, $\mathbf{a}_1 = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$. From example 3.2.3 we know $\mathbf{a}_1$ and $\mathbf{a}_2$ are LI. From example 3.2.5 we know that span $(\{\mathbf{a}_1, \mathbf{a}_2\}) = \mathbb{R}^2$. Therefore, these two vectors are a basis for $\mathbb{R}^2$.

Example 12

The standard unit vectors $\{\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_m\}$ are a basis for $\mathbb{R}^m$. You can check that they meet both conditions: they are LI and span $\mathbb{R}^m$.

4 Systems of equations

Central to our study of differential equations will be this: represent a function (say, a solution to a DE) as a linear combination of convenient functions (say, exponentials, polynomials, or trig functions). We'll start with a simpler problem: representing a specified vector $\mathbf{b} \in \mathbb{R}^m$ as a LC of m vectors $\{\mathbf{a}_j\}$. Not only will this be illustrative of more general problems, it will be useful in its own right. This problem takes the form

$$\sum_{j=1}^{m} \mathbf{a}_j x_j = \mathbf{b}$$

which we can write compactly as

$$A\mathbf{x} = \mathbf{b}.$$

Notice that the problem description implies that A is square. This problem with non-square matrices is also interesting and leads to least-squares problems; however, we will not discuss such problems here.

4.1 Homogeneous equations

Start very simple, with the homogeneous problem $A\mathbf{x} = \mathbf{0}$. This looks like a stupidly useless problem. It isn't, so bear with me and make sure you understand this problem. You'll see it again.

There's an obvious solution to the homogeneous problem, $\mathbf{x} = \mathbf{0}$. We'll call this the ***trivial solution***. The more interesting question is whether there any *nontrivial* solutions $\mathbf{x} \neq \mathbf{0}$.

Example 13

Solve $A\mathbf{x} = \mathbf{0}$ with

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

this is the problem of finding a linear combination of $\mathbf{a}_1 = \begin{bmatrix} 2 & 4 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 3 & 5 \end{bmatrix}^T$ that comes out zero. It's also the system of simultaneous equations

$$2x_1 + 3x_2 = 0$$

$$4x_1 + 5x = 0.$$

Remember how to solve a system of linear equations: use row reduction on an augmented matrix to eliminate everything below the diagonal.

$$\left[\begin{array}{cc|c} 2 & 3 & 0 \\ 4 & 5 & 0 \end{array}\right] \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 0 & -1 & 0 \end{array}\right].$$

The bottom row encodes the equation $-x_2 = 0$. The first row encodes the equation $2x_1 + 3 \cdot 0 = 0$, so $x_1 = 0$ as well. The only solution to this problem is $\mathbf{x} = \mathbf{0}$, which is trivial.

Before going on to another problem, a few observations.

- With the RHS $\mathbf{b} = \mathbf{0}$, the column of zeros in the augmented matrix will remain zero regardless of what row operations are performed. It can be dropped from the calculation.
- If, after row reduction of a problem $A\mathbf{x} = \mathbf{0}$ all diagonal entries are nonzero, then the only solution is the trivial solution $\mathbf{x} = \mathbf{0}$. No need to proceed with backsubstitutions; you know it will produce nothing but zeros.

Example 14

Solve $A\mathbf{x} = \mathbf{0}$ with

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}.$$

Do row reduction,

$$\left[\begin{array}{cc|c} 2 & 3 & 0 \\ 4 & 6 & 0 \end{array}\right] \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{cc|c} 2 & 3 & 0 \\ 0 & 0 & 0 \end{array}\right].$$

The bottom row encodes the equation

$$0x_1 + 0x_2 = 0;$$

any vector $\mathbf{x}$ satisfies that. It contains no useful information. Move on to the next row, which encodes the equation

$$2x_1 + 3x_2 = 0.$$

This is one equation in two unknowns, and has infinitely many solution: any vector $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ with $x_1 = -\frac{3}{2}x_2$. Some possible solutions are

$$\mathbf{x} = \begin{bmatrix} -\frac{3}{2} \\ 1 \end{bmatrix}; \quad \mathbf{x} = \begin{bmatrix} 9 \\ -6 \end{bmatrix}; \quad \mathbf{x} = \begin{bmatrix} -360 \\ 240 \end{bmatrix}$$

and so on. Of course $\mathbf{x} = \mathbf{0}$ is a solution as well. All of these solutions can be represented as follows: choose *any* solution, say $x_1 = 3$, $x_2 = -2$, introduce a scalar parameter s, and write the solution to the problem as

$$\mathbf{x} = s \begin{bmatrix} 3 \\ -2 \end{bmatrix}.$$

This can represent any solution simply by changing the value of s. It doesn't matter which solution you choose as the "basis" vector; any solution will do. We could have just as well chosen $\mathbf{x} = s \begin{bmatrix} -360 & 240 \end{bmatrix}^T$. The same set of possible solutions is represented.

The word "basis" is in quotes because we've not shown that this vector is a basis for some vector space. Spoiler alert: it is.

We'll go a long way with 2×2 problems in this class, but particularly for this type of problem it's important to understand what happens for systems of any size. If after row reduction of an $m \times m$ matrix we find k zero rows, then there will be k free variables.

Example 15

Solve $A\mathbf{x} = \mathbf{0}$ with

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

No row reduction is needed. Only the first row is informative; we have one equation

$$x_1 + 2x_2 + 3x_3 = 0$$

in three unknowns. We have two free variables; set $x_2 = s$ and $x_3 = t$. Then $x_1 = -2s - 3t$, so any solution can be written as

$$\mathbf{x} = \begin{bmatrix} -2s - 3t \\ s \\ t \end{bmatrix}.$$

It is useful to write this as the sum of two vectors, each multiplied by only one of the two parameters s and t,

$$\mathbf{x} = s \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix} + t \begin{bmatrix} -3 \\ 0 \\ 1 \end{bmatrix}.$$

Any solution can be written as a linear combination of the two vectors $\mathbf{v}_1 = \begin{bmatrix} -2 & 1 & 0 \end{bmatrix}^T$ and $\mathbf{v}_2 = \begin{bmatrix} -3 & 0 & 1 \end{bmatrix}^T$. Again with the spoiler alert: these two vectors will be a basis for something.

With some examples in hand, let's analyze the nonhomogeneous problem in general.

By looking at the definitions of $\text{null}\,(A)$ and the homogeneous problem $A\mathbf{x} = \mathbf{0}$, notice that the set of solutions to the homogeneous problem $A\mathbf{x} = \mathbf{0}$ is *exactly* the null space of A. If the only solution is $\mathbf{x} = \mathbf{0}$ (and remember, $\mathbf{0}$ will *always* solve the homogeneous problem) then the null space is trivial.

Let's now consider the more complicated case of nontrivial solutions. This will occur iff row reduction produces one or more rows of zeros. With k nonzero rows there will be k free variables $\{s_j\}_{j=1}^k$, and we will find k nontrivial, LI vectors $\{\mathbf{v}_j\}_{j=1}^k$ such that any solution $\mathbf{x}$ can be expressed as

$$\mathbf{x} = s_1\mathbf{v}_1 + s_2\mathbf{v}_2 + \cdots + s_k\mathbf{v}_k.$$

The $\mathbf{v}_j$'s form a basis for $\text{null}\,(A)$.

Notice that the homogeneous equation $A\mathbf{x} = \mathbf{0}$ has either exactly one solution – the trivial solution $\mathbf{x} = \mathbf{0}$ – or an entire space of solutions parametrized by the free variables.

4.2 Nonhomogeneous equations

Let's now study the nonhomogeneous equation $A\mathbf{x} = \mathbf{b}$. The solution procedure is row reduction by Gaussian elimination; if row reduction produces no zero rows, then the solution $\mathbf{x}$ is constructed by backsubstitution. If no zeros appear on the diagonal, we call the matrix ***nonsingular***. Otherwise, the matrix is called ***singular***.

4.2.1 Nonsingular A: unique solution

Example 16

Solve $A\mathbf{x} = \mathbf{b}$with

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}.$$

Do row reduction,

$$\left[\begin{array}{cc|c} 2 & 3 & 4 \\ 4 & 5 & 6 \end{array}\right] \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{cc|c} 2 & 3 & 4 \\ 0 & -1 & -2 \end{array}\right],$$

there are no zeros on the diagonal so the matrix is nonsingular. The first row is unchanged; the second encodes the one-variable equation $x_2 = 2$. Substitute x_2 back into the first equation,

$$2x_1 + 3 \cdot 2 = 4$$
$$2x_1 = -2,$$

so $x_1 = -1$ and the solution vector is $\mathbf{x} = \begin{bmatrix} -1 & 2 \end{bmatrix}^T$.

If you think about it, if there are no zeros on the diagonal then backsubstitution will always produce exactly one solution. If a matrix is nonsingular, then the equation $A\mathbf{x} = \mathbf{b}$ has a unique solution.

The case where row reduction produces zeros on the diagonal takes a little more work and thought.

4.2.2 Singular A; no solution

Example 17

Solve $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}.$$

Do row reduction,

$$\left[\begin{array}{cc|c} 2 & 3 & 4 \\ 4 & 6 & 6 \end{array}\right] \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{cc|c} 2 & 3 & 4 \\ 0 & 0 & -2 \end{array}\right].$$

The second row encodes the equation $0x_1 + 0x_2 = -2$. There is no solution.

The nonhomogeneous problem $A\mathbf{x} = \mathbf{b}$ must fail to have a solution any time row reduction on the *augmented* matrix produces a row of zeros to the left of the bar (*i.e.*, in the part corresponding to the original matrix) with a nonzero in that row to the right of the bar. To produce that row of zeros, A must be singular.

It's important to understand why the problem $A\mathbf{x} = \mathbf{b}$ in example 4.2.2 has no solution. We can look at it in several equivalent ways.

- Geometry of the rows, figure 1 left. The rows in example 4.2.2 correspond to the equations

  $$2x_1 + 3x_2 = 4$$
  $$4x_1 + 6x_2 = 6.$$

  Those equations correspond to finding the intersection between two parallel lines (they have the same slope) that don't intersect (they have different x_2-intercepts). These lines can't intersect, so there's no solution.

- Geometry of the columns, figure 1 right. Seen columnwise, we're looking for a LC of the vectors $\mathbf{a}_1 = \begin{bmatrix} 2 & 4 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 3 & 6 \end{bmatrix}^T$ that will produce the vector

$\mathbf{b} = \begin{bmatrix} 4 & 6 \end{bmatrix}^T$. But $\mathbf{a}_1$ and $\mathbf{a}_2$ are parallel, and $\mathbf{b}$ doesn't lie along them. No LC of $\mathbf{a}_1$ and $\mathbf{a}_2$, can possibly produce $\mathbf{b}$, therefore, the problem has no solution.

- Abstract: range (A) is the space of all vectors of the form $s \begin{bmatrix} 1 & 2 \end{bmatrix}^T$ for $s \in \mathbb{R}$, but $\mathbf{b} = \begin{bmatrix} 4 & 6 \end{bmatrix}^T$ is not in that space. From theorem 2 there can be no solution.

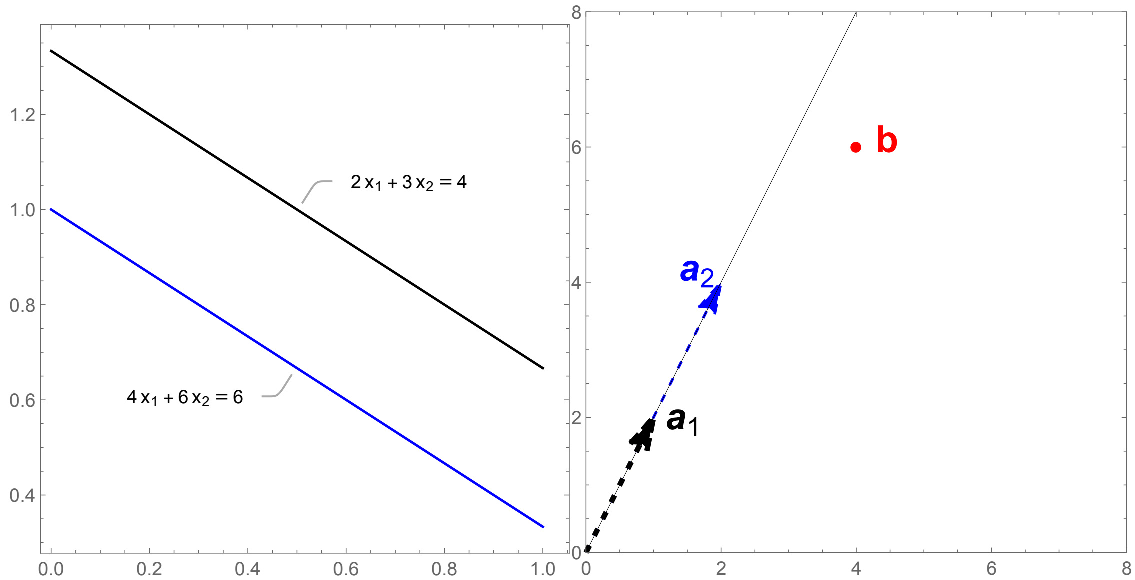


Figure 1: Geometry of a failure to solve $A\mathbf{x} = \mathbf{b}$. Left: distinct parallel lines don't intersect. Right: vectors can't combine to match a vector not in their span.

4.2.3 Singular A; infinitely many solutions

What happens if A is singular and $\mathbf{b} \in \text{range}(A)$. Since $\mathbf{b} \in \text{range}(A)$, there must be at least one solution.

Example 18

Solve $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix}; \quad \mathbf{b} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}.$$

Do row reduction,

$$\left[\begin{array}{cc|c} 2 & 3 & 4 \\ 4 & 6 & 8 \end{array}\right] \xrightarrow{R_2 \leftarrow R_2 - 2R_1} \left[\begin{array}{cc|c} 2 & 3 & 4 \\ 0 & 0 & 0 \end{array}\right].$$

In this case, the second row is all zero and therefore uninformative. The first row encodes the equation

$$2x_1 + 3x_2 = 4,$$

which is the equation of a line. Choose one of the variables to be free, for example, set $x_2 = t$. Then $x_1 = 2 - \frac{3}{2}t$, and the solutions are

$$\mathbf{x} = \begin{bmatrix} 2 \\ 0 \end{bmatrix} + t \begin{bmatrix} -\frac{3}{2} \\ 1 \end{bmatrix}.$$

There are infinitely many solutions, one for each value of t.

What's going on here?

- Geometry of the rows: the equations

$$2x_1 + 3x_2 = 4$$

$$4x_1 + 6x_2 = 8$$

  are two equations for the same line (divide the second equation by two). All points $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ on that line are solutions.

- Geometry of the columns: The vectors $\mathbf{a}_1 = \begin{bmatrix} 2 & 4 \end{bmatrix}^T$ and $\mathbf{a}_2 = \begin{bmatrix} 3 & 6 \end{bmatrix}^T$ can be combined infinitely many ways to produce the point $\mathbf{b} = \begin{bmatrix} 4 & 8 \end{bmatrix}^T$.

- Abstract: Since $\mathbf{b} \in \text{range}\,(A)$ we know there's at least one solution; call it $\mathbf{x}_p$. Add in any member of $\text{null}\,(A)$, say $\mathbf{x}_h \in \text{null}\,(A)$. Then $A\left(\mathbf{x}_p + \mathbf{x}_h\right) = A\mathbf{x}_p = \mathbf{b}$, so we find that $\mathbf{x}_p + \mathbf{x}_h$ is also a solution.

If A is singular, then $A\mathbf{x} = \mathbf{b}$ will have either no solution or infinitelymany solutions depending on whether $\mathbf{b} \in \text{range}\,(A)$. If $\mathbf{b} \notin \text{range}\,(A)$, then there is no solution. If A is singular and $\mathbf{b} \in \text{range}\,(A)$ then there are infinitely many solutions; this case is called ***singular but consistent***.

The observation that we can add any $\mathbf{x}_h \in \text{null}\,(A)$ to a solution and still have a solution, is an example of ***superposition***. Superposition is one of the most widely useful ideas in the analysis of linear equations (whether linear algebraic equations or linear differential equations). The next section is devoted to looking at several manifestations of superposition.

4.3 Superposition

Theorem 3 (Superposition I) If $\mathbf{u}$ and $\mathbf{v}$ are in $\text{null}\,(A)$, then every linear combination $\alpha\mathbf{u} + \beta\mathbf{v}$ is also in $\text{null}\,(A)$.

Proof. This is an immediate consequence of distributivity of matrix-vector multiplication over addition: If $A\mathbf{u} = \mathbf{0}$ and $A\mathbf{v} = \mathbf{0}$, then $A\left(\mathbf{u} + \mathbf{v}\right) = A\mathbf{u} + A\mathbf{v} = \mathbf{0}$. □

Notice that this theorem is just a statement that linear combinations of members of the null space obey the closure conditions. If you've taken differential equations, you'll recognize this idea in another form. If $y_1(x)$ and $y_2(x)$ are solutions to a homogeneous linear differential equation, then so is $\alpha y_1(x) + \beta y_2(x)$.

Theorem 4 (Superposition II) Suppose $\mathbf{x}_p$ is a solution to the homogeneous problem $A\mathbf{x} = \mathbf{0}$, and $\mathbf{x}_h$ is a solution to the nonhomogeneous problem $A\mathbf{x} = \mathbf{b}$. Then $\mathbf{x}_g = \mathbf{x}_p + \mathbf{x}_h$ is also a solution to the nonhomogeneous problem.

We've seen this form of superposition in action in 4.2.3. Once again, if you've taken differential equations you'll recognize this idea in another form. If $y_p(x)$ is a particular solution to a nonhomogeneous linear differential equation, and if $y_h(x)$ is a solution to the homogeneous version of the problem, then $y_g(x) = y_h(x) + y_p(x)$ is also a solution. There is one more form of superposition to introduce.

Theorem 5 (Superposition III) Suppose we have solutions $\{\mathbf{x}_k\}_{k=1}^{N}$ to a collection of N nonhomogeneous problems involving the same $m \times n$ matrix A,

$$A\mathbf{x}_k = \mathbf{b}_k, \;\; k = 1, 2, \cdots, N.$$

Then the problem

$$A\mathbf{x} = \sum_{k=1}^{N} \beta_k \mathbf{b}_k$$

has solution

$$\mathbf{x}_g = \mathbf{x}_h + \sum_{k=1}^{N} \beta_k \mathbf{x}_k$$

where $\mathbf{x}_h$ is any solution to the homogeneous problem.

You might have used this idea when solving differential equations such as

$$y' + y = 5x + 3e^x.$$

Solve the subproblems

$$y_1' + y_1 = x; \quad y_2' + y_2 = e^x; \quad y_h' + y_h = 0,$$

and form the general solution

$$y_g(x) = y_h(x) + 5y_1(x) + 3y_2(x).$$

All three superposition theorems 3, 4, and 5 are of the form "vector such-and-such is a solution to this-or-that equation." To prove any theorem of that form, just plug vector such-and-such into equation this-or-that, and verify the equation is satisfied.

5 Matrix-matrix multiplication and the inverse

You've probably learned how to do matrix-matrix multiplication, but might not have been told *why* it's defined in such a strange way. Here's the fundamental idea:

- Matrix-matrix multiplication is defined so that the operations $A(B\mathbf{x})$ and $(AB)\mathbf{x}$ give the same result.

First of all, notice that the output of $B\mathbf{x}$ must have the same dimension as the input to $A(\cdot)$, that is, if B is $k \times n$, then $B\mathbf{x}$ has dimension k, so A must have k columns. The matrix-matrix product AB is defined only if A is $m \times k$ and B is $k \times n$; the "middle" dimension must be the same. If the matrices have the right combination of shapes for AB to make sense, we call them ***compatible***. Note that two square matrices of the same size are always compatible.

5.1 The matrix-matrix multiplication formula

Suppose A and B are compatible. We're going to *derive* the formula for the matrix-matrix product AB by working out $A(B\mathbf{x})$ and insisting that $(AB)\mathbf{x}$ must give the same result. First, use the column-wise form of matrix-vector multiplication to write

$$B\mathbf{x} = \sum_{j=1}^{n} \mathbf{b}_j x_j$$

and then multiply through (from the left) by A,

$$\begin{aligned} A(B\mathbf{x}) &= A\sum_{j=1}^{n} \mathbf{b}_j x_j \\ &= \sum_{j=1}^{n} A\mathbf{b}_j x_j. \end{aligned}$$

That's $A(B\mathbf{x})$. Now let's construct $(AB)\mathbf{x}$ to get the same answer. We don't (yet) know what the columns of AB actually look like, but we do know that $(AB)\mathbf{x}$ is the LC of the columns of AB weighted by the entries in $\mathbf{x}$:

$$(AB)\mathbf{x} = \sum_{j=1}^{n} [j - \text{th column of } AB]\, x_j.$$

We can now determine the unknown expression $[j - \text{th column of } AB]$ by setting $(AB)\mathbf{x}$ equal to $A(B\mathbf{x})$:

$$\underbrace{\sum_{j=1}^{n} A\mathbf{b}_j x_j}_{A(B\mathbf{x})} = \underbrace{\sum_{j=1}^{n} [j - \text{th column of } AB]\, x_j}_{(AB)\mathbf{x}}$$

Comparing the vectors in each sum, we see that the j-th column of AB is $A\mathbf{b}_j$. Therefore, the matrix-matrix product AB is the matrix written out columnwise as

$$AB = \left[\; A\mathbf{b}_1 \;\middle|\; A\mathbf{b}_2 \;\middle|\; \cdots \;\middle|\; A\mathbf{b}_n \;\right].$$

This is a simple formula to follow for hand calculation, and it will also be important theoretically in eigenvalue analysis.

> **Pro tip**: when computing $AB\mathbf{x}$, it's usually more efficient to compute $B\mathbf{x}$ first, then multiply A into the result, than it is to compute AB then multiply into $\mathbf{x}$. This is true for hand calculations and for computer calculations.

5.1.1 An alternate derivation

Here's an alternate derivation of matrix-matrix multiplication using the summation form of the matrix-vector product. Start by computing the ℓ-th entry in $B\mathbf{x}$, which we write out in summation form as

$$(B\mathbf{x})_\ell = \sum_{j=1}^{n} B_{\ell j} x_j.$$

Then multiply from the left by A, writing the i-th entry in the result as

$$[A(B\mathbf{x})]_i = \sum_{\ell=1}^{k} A_{i\ell}\left(\sum_{j=1}^{n} B_{\ell j}x_j\right)$$

$$= \sum_{j=1}^{n}\left(\sum_{\ell=1}^{k} A_{i\ell}B_{\ell j}\right)x_j \quad \text{(exchange order of summation)}.$$

Now recognize that this is just matrix-vector multiplication of $\mathbf{x}$ by the matrix whose i, j element is

$$\sum_{\ell=1}^{k} A_{i\ell}B_{\ell j}.$$

You should convince yourself that the $j-$th column of this matrix is $A\mathbf{b}_j$, as found above.

5.1.2 Properties of matrix multiplication

Here are a few observations about matrix-matrix multiplication:

- Matrix-matrix multiplication is not, in general, commutative: AB does not always equal BA.
- Matrix-matrix multiplication does distribute over addition: $A(B+C) = AB + AC$. This follows from the fact that matrix-*vector* multiplication distributes over addition, and that matrix-matrix multiplication is built up from matrix-vector multiplication:

$$A(B+C) = \left[\ A(\mathbf{b}_1+\mathbf{c}_1) \mid A(\mathbf{b}_2+\mathbf{c}_2) \mid \cdots \mid A(\mathbf{b}_n+\mathbf{c}_n)\ \right]$$
$$= \left[\ A\mathbf{b}_1 \mid A\mathbf{b}_2 \mid \cdots \mid A\mathbf{b}_n\ \right] + \left[\ A\mathbf{c}_1 \mid A\mathbf{c}_2 \mid \cdots \mid A\mathbf{c}_n\ \right].$$
$$= AB + AC.$$

- Notice that matrix-vector multiplication $A\mathbf{x}$ can be regarded as a special case of matrix-matrix multiplication, with the column vector $\mathbf{x}$ regarded as an $n \times 1$ matrix.
- Notice that if we transpose a column vector $\mathbf{x}$ to a row vector

$$\mathbf{x}^T = \begin{pmatrix} x_1 & x_2 & \cdots & x_m \end{pmatrix}$$

  then the row-vector times matrix product $\mathbf{x}^T A$ makes sense as a product between $1 \times m$ and $m \times n$ matrices.

5.1.3 The identity matrix and the Kronecker delta

The $m \times m$ matrix

$$I = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

whose entries are

$$I_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

is called the $m \times m$ ***identity matrix***. It's easily checked that multiplication of $\mathbf{x}$ by I simply returns $\mathbf{x}$ unchanged.

Related notation is the ***Kronecker delta***, defined by

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

5.2 The matrix inverse

Now that we know about matrix-matrix products, we ask whether we can find a matrix A^{-1} with which we can solve the equation

$$A\mathbf{x} = \mathbf{b}$$

by multiplication, similarly to how we use the reciprocal of a scalar a to solve $ax = b$. Let I be the identity matrix, and suppose that for a matrix A there exists an ***inverse matrix*** A^{-1} such that

$$A^{-1}A = I.$$

Then, solution of $A\mathbf{x} = \mathbf{b}$ follows immediately: multiply through by A^{-1} from the left,

$$A^{-1}A\mathbf{x} = A^{-1}\mathbf{b}$$

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

Hooray, the equation has been solved!

Application of a matrix inverse is ***not*** division by a matrix, so ***never*** write

$$\mathbf{x} = \frac{\mathbf{b}}{A}.$$

Division by matrices and vectors is never allowed!

There are a few pending questions:

1. Does A have an inverse? (and is it unique, etc)
2. If A does have an inverse, how do we find it?
3. If A has an inverse and we know how to find it, is using the inverse to compute $\mathbf{x} = A^{-1}\mathbf{b}$ an efficient way to solve $A\mathbf{x} = \mathbf{b}$?

Our answer to the first question will be *constructive*, meaning that we will show the existence of an inverse by giving an algorithm to *construct* one; thus, we will answer the second question ("how do we find it?") along the way to answering the question of existence.

Before embarking on construction of an inverse, let's think about when we'd expect *not* to have one. An inverse applied by matrix-vector multiplication $A^{-1}\mathbf{b}$ should produce exactly one solution $\mathbf{x} = A^{-1}\mathbf{b}$ for each $\mathbf{b}$, and it should break down when the problem has no solution. By the nature of matrix-vector multiplication the operation $A^{-1}\mathbf{b}$ can't produce more than one result. Furthermore, matrix-vector multiplication can't ever break down. To avoid nonsense results we expect A^{-1} to fail to exists whenever A is singular. We will see that this is the case.

5.2.1 Constructing the inverse

We are going to solve $A\mathbf{x} = \mathbf{b}$ by *superposition*. The basic idea is this: *build up a solution to an equation by summing simpler solutions to simpler problems.* The "simpler" problems will be $A\mathbf{x}_j = \hat{\boldsymbol{e}}_j$. These aren't really any simpler, but bear with me.

Suppose that A is nonsingular. Let's write $\mathbf{b}$ as a LC of standard Cartesian unit vectors,

$$\mathbf{b} = b_1\hat{\boldsymbol{e}}_1 + b_2\hat{\boldsymbol{e}}_2 + \cdots + b_n\hat{\boldsymbol{e}}_n.$$

Suppose we've solved each of the "simple" problems $A\mathbf{x}_j = \hat{\boldsymbol{e}}_j$ for the vectors $\mathbf{x}_j$. Then the solution $\mathbf{x}$ to the original problem is, by superposition,

$$\mathbf{x} = b_1\mathbf{x}_1 + b_2\mathbf{x}_2 + \cdots + b_n\mathbf{x}_n.$$

Glue the vectors $\mathbf{x}_j$ into the columns of a matrix X, and observe that $\mathbf{x} = X\mathbf{b}$. Thus, when A is nonsingular there is a matrix X such that the solution $\mathbf{x}$ to $A\mathbf{x} = \mathbf{b}$ can be written as $\mathbf{x} = X\mathbf{b}$. By the definition of the inverse, we have $A^{-1} = X$. Every nonsingular matrix has an inverse.

Moreover, if you're paying attention, you'll see that we've actually shown to to construct the inverse. The j-th column of A^{-1} is the vector $\mathbf{z}_j$, formed by solving $A\mathbf{z}_j = \hat{\boldsymbol{e}}_j$. Because A is nonsingular, each of those "simple" problems has a unique solution; therefore, the inverse X formed from those solutions is unique.

Will this procedure work if A is singular? It can't.

Theorem 6 *A has an inverse iff A is nonsingular. That inverse is unique.*

Case 1. We've already shown that if A is nonsingular, then it has a unique inverse. For the "only if" direction: if A is singular, then row reduction on A (possibly involving row exchanges) will produce at least one row of nonzeros. For simplicity, focus on the case where there are no row exchanges, and suppose that (at least) the final row m is entirely zero. We'll need to solve $A\mathbf{x}_m = \hat{\boldsymbol{e}}_m$. When we form the augmented matrix for that problem, it will look like this:

$$\left[\begin{array}{cccc|c} u_{11} & u_{12} & \cdots & u_{1m} & 0 \\ 0 & u_{22} & & & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \end{array}\right].$$

This is because the m-th (bottom) row is formed as a linear combination of the original m-th row of A with nonzero multiples of the rows above it. But since all entries in $\hat{\boldsymbol{e}}_m$ save the last are zero, that process leaves $\hat{\boldsymbol{e}}_m$ unchanged, and as a result the equation encoded in the final row has no solution. Therefore, the construction of the inverse must fail for at least one of the unit vectors, and the inverse cannot exist.

In a linear algebra class, you might have learned to compute the inverse of A by setting up the augmented system

$$\left[\, A \,\middle|\, I \,\right]$$

and doing Gauss-Jordan elimination; after that process the inverse will be in the second block column of the augmented system. Now you can see why that works: the procedure solves the equations $A\mathbf{z}_j = \hat{\boldsymbol{e}}_j$ simultaneously in the same table, with the solution $\mathbf{z}_j$ being written into the j-th column of the second block column. In practice, you'd not do Gauss-Jordan elimination; you'd do Gaussian elimination followed by backsubstitution to find each column of A^{-1}. If you ever need to compute an inverse – and you probably don't – do it with Gaussian elimination and backsubstitution.

5.2.2 Don't compute an inverse

Now that I've shown you how to compute an inverse, I'm going to tell you never to do it. Well, almost never. It's a matter of efficiency. Notice that the "simple" problems $A\mathbf{z}_j = \hat{\boldsymbol{e}}_j$ that arose in the computation of the inverse were, in fact, no smaller or easier than the original problem $A\mathbf{x} = \mathbf{b}$. You might conclude from that observation that computing the inverse isn't such a useful idea after all, and you'd be right. As a general rule, *don't compute a matrix inverse*. Just solve the system $A\mathbf{x} = \mathbf{b}$.

You might think that if you're solving *many* systems $A\mathbf{x} = \mathbf{b}$ with the same A but different RHS vectors[2] $\mathbf{b}_1, \mathbf{b}_2, \cdots$, then the up-front cost invested in computing A^{-1} might save time in the long run when solving the sequence of systems. My undergraduate linear algebra textbook, and my teacher, claimed this; they were wrong. See F.3 for an explanation of why that is, and for the right way to solve multiple systems of equations efficiently.

So if we shouldn't compute it, what good is the inverse?

- It is most useful as *notation*. When I write $\mathbf{x} = A^{-1}\mathbf{b}$ I mean "solve the equation $A\mathbf{x} = \mathbf{b}$ for $\mathbf{x}$ using any efficient method you like". Keeping in mind the Pro Tip on how to compute $ABC\mathbf{x}$ (***don't*** do the matrix-matrix multiplications to form ABC), this lets you make sense out of what to do with expressions like $ABA^{-1}\mathbf{y}$. You can evaluate that with (a) no inverse computations and (b) no matrix-matrix multiplications, as follows:
  - First compute $\mathbf{p} = A^{-1}\mathbf{y}$ by solving $A\mathbf{p} = \mathbf{y}$ for $\mathbf{p}$ using, *e.g.*, Gaussian elimination.
  - Compute $\mathbf{q} = B\left(A^{-1}\mathbf{y}\right)$ by multiplying B into $\mathbf{p}$ computed in the first step
  - Compute $\mathbf{r} = A\left(B\left(A^{-1}\mathbf{y}\right)\right)$ by multiplying A into $\mathbf{q}$ computed in the second step.

[2] This is very common in practice.

The procedure takes one run of Gaussian elimination plus two matrix-vector multiplies, compared to the *much* more expensive proposition of one inverse calculation and two matrix-matrix multiplies.

- Sometimes the inverse can be found easily and efficiently.
  - Nonsingular diagonal matrices are trivially inverted: just take the reciprocals of the diagonal entries. If one of the diagonal entries is zero then the matrix is singular.
  - There is a simple formula for the inverse of a two-by-two matrix. The formula is:
  $$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$
  For a general two-by two system, using this formula is about as efficient as any other method.
  - The inverses of *orthogonal* matrices and *unitary* matrices (see 6.1) can be found with no calculations at all.

5.2.3 Some properties of the inverse

- The inverse of A^{-1}, $\left(A^{-1}\right)^{-1}$, is A.
- When A is an invertible matrix and α is a scalar, then $(\alpha A)^{-1} = \frac{1}{\alpha}A^{-1}$.
- The inverse of a product AB is the product of the inverses, in reversed order: $(AB)^{-1} = B^{-1}A^{-1}$
  - Here's why: to solve the system
  $$AB\mathbf{x} = \mathbf{b},$$
  you would first solve $A\,(B\mathbf{x}) = \mathbf{b}$ for the unknown $B\mathbf{x}$:
  $$B\mathbf{x} = A^{-1}\mathbf{b},$$
  and then solve that system for $\mathbf{x}$,
  $$\mathbf{x} = B^{-1}A^{-1}\mathbf{b}.$$
  Since this is the solution to $(AB)\,\mathbf{x} = \mathbf{b}$, we have
  $$(AB)^{-1} = B^{-1}A^{-1}.$$

Those are some things you can do with inverses. Here's something that *doesn't* work, however desperate one might be to solve a problem.

The inverse of a sum is ***not*** , in general, the sum of the inverses:

$$(A+B)^{-1} \neq A^{-1}+B^{-1}.$$

To prove the formula false, a single counterexample is sufficient: try $B=-A$.

There is a formula for $(A+B)^{-1}$, but it is an infinite series (similar to the familiar geometric series for $(a+b)^{-1}$ where a and b are real). I will not state it here, because the convergence conditions involve concepts (matrix norms) beyond the level of these notes. We will not use this formula in this course.

Example 19

The inverse of $AB^{-1}C$ is

$$\begin{aligned}\left(AB^{-1}C\right) &= C^{-1}\left(B^{-1}\right)^{-1}A^{-1}\\ &= C^{-1}BA^{-1}.\end{aligned}$$

6 Some special matrices

6.1 Orthogonal and unitary matrices

Suppose the columns of a real matrix Q are orthonormal: $\mathbf{q}_i^T\mathbf{q}_j=\delta_{ij}$. Then $Q^TQ=I$, and the matrix Q is called ***column orthogonal***. If Q is square, then $Q^TQ=QQ^T=I$, and the matrix is called ***orthogonal***. Since $Q^TQ=I$, conclude that $Q^{-1}=Q^T$; every orthogonal matrix is invertible, and it's easy to find the inverse of an orthogonal matrix: no computations needed, just transpose! Therefore, you can solve a system $Q\mathbf{x}=\mathbf{b}$ right away: $\mathbf{x}=Q^T\mathbf{b}$.

If you multiply two orthogonal matrices, the result is orthogonal. The proof will be left to you.

Orthogonal matrices have one further important property: multiplication of $\mathbf{x}$ by Q *preserves the magnitude of* $\mathbf{x}$. Here's why: compute the magnitude of $Q\mathbf{x}$,

$$\|Q\mathbf{x}\| = \sqrt{(Q\mathbf{x})^T(Q\mathbf{x})} = \sqrt{\mathbf{x}^TQ^TQ\mathbf{x}} = \sqrt{\mathbf{x}^TI\mathbf{x}} = \|\mathbf{x}\|.$$

This will be important for certain differential equations.

If Q is a *complex* matrix, then the condition for orthonormal columns is $\mathbf{q}_i^*\mathbf{q}_j=\delta_{ij}$. If Q is also square, then it is called ***unitary***. All properties of orthogonal matrices carry over to unitary matrices by simply replacing transposition by transposition plus conjugation. Orthogonal matrices are a special case of unitary matrices.

Unitary matrix properties:

- The columns are orthonormal.

- The inverse of a unitary matrix is its conjugate transpose: $Q^{-1} = Q^*$.
- If Q_1 and Q_2 are unitary, then $Q_1 Q_2$ is unitary.
- Unitary matrices preserve magnitude: $\|Q\mathbf{x}\| = \|\mathbf{x}\|$.

We'll often use the symbol Q for a generic orthogonal matrix and either Q or U for a generic unitary matrix.

6.2 Permutation matrices

Start with the identity matrix and swap two rows or columns, for instance, swap columns 2 and 3 in the 3×3 identity,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Now multiply into a vector:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \\ x_2 \end{bmatrix}.$$

Multiplying from the left permutes rows 2 and 3 in the vector. Any such permutated identity is called a ***permutation matrix***; multiplication from the left does the same permutation on the *rows* of the input. Multiplication from the right permutes the *columns* of the input.

Example 20

Let P be the permutation matrix

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

Then

$$P\mathbf{x} = \begin{bmatrix} x_3 \\ x_4 \\ x_2 \\ x_1 \end{bmatrix}$$

and

$$\mathbf{x}^T P = \begin{bmatrix} x_3 & x_4 & x_2 & x_1 \end{bmatrix}.$$

Notice that the columns of any permutation matrix are orthonormal. Therefore, every permutation matrix is orthogonal.

To undo the action of a permutation matrix, apply the inverse matrix. This is easy: since the permutation matrix is unitary, its inverse is its transpose (no conjugation needed, since permutation matrices are real.)

Properties of permutation matrices

- A permutation matrix is formed by permuting the identity.
- The product of two permutation matrices is a permutation matrix.
- Every permutation matrix is unitary.

6.3 Triangular matrices

An upper(lower) triangular matrix has all zeros below(above) the main diagonal:

$$U = \begin{bmatrix} U_{11} & U_{12} & U_{13} & \cdots \\ 0 & U_{22} & U_{23} & \\ 0 & 0 & U_{33} & \\ \vdots & & & \ddots \end{bmatrix}; \quad L = \begin{bmatrix} L_{11} & 0 & 0 & \cdots \\ L_{21} & L_{22} & 0 & \\ L_{31} & L_{32} & L_{33} & \\ \vdots & & & \ddots \end{bmatrix}.$$

Tradition symbols are U for upper triangular matrices and L for lower triangular matrices; it will usually be clear from context whether U denotes upper triangular or unitary.

Solving a system $U\mathbf{x} = \mathbf{b}$ is easy: solve the bottom equation for x_n, then the next up for x_{n-1}, and so on. This will fail iff one of the diagonal entries is zero. For a system $L\mathbf{x} = \mathbf{b}$,work downward from the first row.

A triangular matrix is singular iff the main diagonal contains a zero entry.

6.3.1 Diagonal matrices

A diagonal matrix has zeros everywhere but the diagonal (though some diagonal entries may be zero). As a special case of a triangular matrix, a diagonal matrix is singular iff the main diagonal contains a zero entry. The inverse of a diagonal matrix is easy.

If D is a nonsingular diagonal matrix, then D^{-1} is the diagonal matrix whose i, i entry is $1/D_{ii}$.

7 The determinant

7.1 The determinant as an oriented hypervolume

You have almost certainly seen how to compute the determinant of a matrix by an ugly, messy technique called expansion in minors. I'm not going to write that out here; a quick search online will show you the procedure. Instead, I'll tell you what the determinant *means* geometrically. We'll be able to deduce a surprising amount about determinants from this geometric point of view.

We start in 2 dimensions. You should know from calculus that the signed area of the parallelogram defined by vectors **a** and **b** is given by a determinant:

$$\det\begin{pmatrix} a_x & b_x \\ a_y & b_y \end{pmatrix} = a_x b_y - a_y b_x.$$

If you don't understand why this is true, please refer to Appendix D for a proof using trigonometry and vector algebra. The relationship between the determinant of a matrix and signed area (or more generally, signed hypervolume (SHV)) of the parallelogram defined by the columns of that matrix extends to any dimension. In fact, we'll *define* the determinant so that this is true.

Definition 8 Let P be the m-dimensional parallelepiped formed by the vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_m$. Pack these vectors columnwise into a matrix

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

The determinant of A, denoted $\det(A)$, is the signed hypervolume of P.

The ugly formula for expansion by minors follows from this definition. The proof is by induction on dimension. For our purposes the conclusion is much more important than the proof; if you're interested see, for example [2].

7.2 Properties of the determinant

7.2.1 The determinant of the identity is one

The columns of the $m \times m$ identity matrix form the unit hypercube. The signed hypervolume of the unit hypercube is one, therefore, $\det(I) = 1$.

7.2.2 The determinant of a product is the product of the determinants

Theorem 7 For any $m \times m$ A and B, we have $\det(AB) = \det(A)\det(B)$.

When I was a beginning student, I found this theorem to be mysterious; even though the result was simple it made no intuitive sense at all to me, and the proof was hideous. This was because I was only taught the definition of the determinant as an expansion in minors without any geometric interpretation. Trying to prove $\det(AB) = \det(A)\det(B)$ by algebraic calculation is horrendously messy. Thinking geometrically, however, the argument is very simple.

Start with the trivial observation that $B = BI$. The parallelepiped constructed from the columns of B is also the result of applying B to each of the n standard Cartesian unit vectors $\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \cdots, \hat{\boldsymbol{e}}_n$ that define the unit hypercube. Geometrically, B maps the unit hypercube into a parallelepiped. Since transformation by B is linear, and since the unit hypercube has signed hypervolume one, we can also think of $\det(B)$ as the factor by which any signed hypervolume is scaled during transformation by B.

Now apply A to the columns of B, producing a new parallelepiped. We started with the unit hypercube with hypervolume 1, multiplied the unit vectors by B to form a new parallelepiped BI with hypervolume $\det(B)$, and then multiplied those new vectors by A. That second transformation scales the signed hypervolume by $\det(A)$. Put it together and we have

$$\underbrace{\det(AB)}_{\text{SHV of the result of } AB} = \underbrace{\det(A)}_{\text{(scaling due to transf. by } A)} \underbrace{\det(B)}_{\text{(scaling due to transf. by } B)} \underbrace{\det(I)}_{\text{(SHV of unit hypercube)}}$$

or

$$\det(AB) = \det(A)\det(B).$$

7.2.3 If A contains a zero column (or row) then $\det(A) = 0$

If a column is of A zero, then one of the "legs" forming the parallelepiped has zero length; the hypervolume must be zero.

If some row, say row i, of A is zero, then none of the "legs" extends at all into the i-th dimension. The hypervolume must be zero.

7.2.4 The determinant of a permutation matrix is ± 1

A permutation matrix applied to the identity exchanges two or more of the legs in the parallelepiped. One exchange reverses the orientation and thus has determinant -1. Two exchanges reverses the orientation twice, and so on.

7.2.5 $\det(A) = 0$ iff A is singular

Start with a simple case: we'll first show that if T is triangular, then $\det(T) = 0$ iff T is singular. Start in 2D, and WLOG suppose T is upper triangular,

$$T = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}.$$

The parallelogram formed by T has one leg of length T_{11} along the x axis, and another along the vector $\begin{bmatrix} T_{12} & T_{22} \end{bmatrix}^T$. The parallelogram collapses to a line iff the second leg has no component along the y axis, that is, iff $T_{22} = 0$. But from 6.3 we know that T is singular iff if it contains zero on the diagonal. Therefore, a 2×2 triangular matrix has zero determinant iff it is singular.

Now consider an $m \times m$ matrix

$$T = \begin{bmatrix} T_{11} & T_{12} & T_{13} & \\ 0 & T_{22} & T_{23} & \\ 0 & 0 & T_{33} & \\ & & & \ddots \end{bmatrix}$$

and think about the volume formed by the three columns. If T_{11} is zero, then we have a zero leg and the volume collapses to zero. If T_{22} is zero, then the first two legs are parallel and the volume collapses to zero. If $T_{33} = 0$, then leg 3 is coplanar with legs 1 and 2, and the volume collapses to zero. And so on. If any diagonal entry is zero, then the volume is zero. If no diagonal entries are zero, then each successive column extends into a dimension not yet "explored" by the others, so the volume is nonzero. If T is lower triangular, make the same argument starting with the last column. Therefore: if T is triangular, then $\det(T) = 0$ iff T is singular.

Why spend so much time on triangular matrices? Because the operations of row reduction (other than row exchanges) can be represented as multiplication by lower triangular matrices.

Ignoring row exchanges, row reduction of an $m \times m$ matrix A to an upper triangular matrix U can be written as

$$L_1 L_2 \cdots L_{m-1} A = U,$$

where each L_i is lower triangular with ones on the diagonal. Given that restriction on L_i, we have $\det(L_i) \neq 0$. Using the determinant product formula, we have

$$\det(L_1)\det(L_2)\cdots\det(L_{m-1})\det(A) = \det(U).$$

Since none of the L's have zero determinants, we find that $\det(A) = 0$ iff $\det(U) = 0$. But U will have a zero on the diagonal, and therefore a zero determinant, iff A is singular.

We're almost there. What about row exchanges? Row exchanges are represented by permutation matrices. Row reduction becomes

$$P_1 L_1 P_2 L_2 \cdots P_{m-1} L_{m-1} A = U.$$

But from 7.2.4 we know $\det(P_i) \neq 0$, so we reach the same conclusion.

$\det(A) = 0$ iff A is singular.

In this course, this will be the most important property of the determinant.

7.2.6 The determinant of an inverse

If A is invertible, then $A^{-1}A = I$. From the product formula, find

$$\det\left(A^{-1}\right)\det(A) = \det(I),$$

and then conclude

$$\det\left(A^{-1}\right) = \frac{1}{\det(A)}.$$

Notice that $\det\left(A^{-1}\right)$ is undefined iff $\det(A) = 0$, which is to be expected since $\det(A) = 0$ iff A is singular: a singular matrix has no inverse!

7.2.7 Summary of determinants

To summarize:

- $\det(A)$ is the signed hypervolume of the parallelepiped formed by the columns of A
- $\det(I) = 1$
- $\det(AB) = \det(A)\det(B)$
- $\det(A) = 0$ iff A is singular
- $\det\left(A^{-1}\right) = \frac{1}{\det(A)}$

We figured all of that out without touching expansions in minors.

Unfortunately, determinants are a subject (eigenvalue problems will be another) where the simplest methods for hand calculation turn out to be *terrible* methods for real-world numerical calculations. In hand calculations, it's easy to do expansion in minors to compute determinants of 2×2 or 3×3 matrices; we'll do many such calculations when solving eigenvalue problems. However, when doing numerical calculations, never use an expansion in minors for any matrix bigger than 2×2. If you're interested in computing, see G.1 for why.

7.3 Why you should *never* use a floating point determinant calculation to check for singularity of a matrix

Since I'm a computational mathematician, I'm morally obligated to warn you about ways to screw up when using a computer.

Probably the most important property of the determinant is that $\det(A) = 0$ iff A is singular. This is useful and we'll use this often when working with eigenvalues. But beware: ***never*** use this as a test for singularity when working in floating point arithmetic.

Suppose you have two matrices, A and B , and find after floating point calculation that

$$\det(A) \approx 10^{-18}$$

and

$$\det(B) \approx 10.$$

Since 10^{-18} is smaller than machine precision and 10 is bigger, it's tempting to conclude that A is singular and B is not. But you would be wrong.

It has to do with scaling. Remember: $\det(A)$ is a hypervolume. Scaling an $m \times m$ matrix A by α scales $\det(A)$ by α^m. The hypervolume of a six-dimensional hypercube, one meter on a side, is $1\,\mathrm{m}^6$. Now find the volume of that same box, but working in kilometers. Each side is 10^{-3} km in length, for a hypervolume of $10^{-18}\,\mathrm{km}^6$. The value of the determinant depends on the units you choose, and the problem gets worse the higher the dimension of the problem.

In floating point calculations, a very small determinant does ***not*** imply a singular matrix.

So, we can't trust very small determinants to indicate singular matrices. But can we at least trust a moderate size determinant to indicate a nonsingular matrix? Alas, no. The problem is accumulation of roundoff error.

7.4 Cramer's Rule

Don't use it. Use Gaussian elimination instead. If your problem too large for Gaussian elimination (in which case it will be *much* too large to use Cramer's Rule) use an iterative method such as conjugate gradients, GMRES, or multilevel.

Why avoid Cramer's Rule?

If you compute all the determinants the smart, efficient, and robust way, doing LU factorizations instead of expansions in minors, it will take you $m+1$ LU factorizations. Gaussian elimination requires one LU factorization. Cramer's Rule takes m times the work of Gaussian elimination, and that's if you do it the smart way. See G.1 for what you can expect if you do it the dumb way.

8 Putting it all together: the Big Matrix Theorem

You've now learned a bunch of facts regarding solution of $A\mathbf{x} = \mathbf{b}$, involving inverses, determinants, null spaces, range spaces, and more. All of this can be put together into a single theorem. A version of this theorem appears in every linear algebra textbook, but is rarely given a name. So you can remember it when I refer to it, let's call it ***The Big Matrix Theorem***.

Theorem 8 (The Big Matrix Theorem)

Suppose A is an $m \times m$ complex matrix. The following statements are equivalent (meaning that if any one is true, then all are true; if any one is false, then all are false.) A matrix for which these are true is called ***nonsingular*** (or ***invertible***). *A matrix for which they are false is called* ***singular*** (or ***non-invertible***).

1. The null space, $\text{null}\,(A)$, is trivial (*i.e.,* it contains only the zero vector)
2. The equation $A\mathbf{x} = \mathbf{0}$ has only the trivial solution $\mathbf{x} = \mathbf{0}$
3. The equation $A\mathbf{x} = \mathbf{b}$ has a unique solution
4. A has a unique inverse, A^{-1}, such that $A^{-1}A = I$ and $AA^{-1} = I$.
5. The columns of A are linearly independent and are a basis for $\mathbb{C}^n$
6. The rows of A are linearly independent and are a basis for $\mathbb{C}^n$
7. The determinant of A is nonzero

8. Zero is not an eigenvalue of A
9. The Gaussian elimination algorithm (perhaps with row exchanges) can reduce A to upper triangular form (zeros below the diagonal) with no zeros on the diagonal.

While I've shown how some of these statements are connected, I've not carefully proved the equivalence of all these statements and we don't have the time (or the need) to do so in this course. To see a complete proof of equivalence, consult a good linear algebra textbook.

A Important notation

A.1 Abbreviations

- ew – eigenvalue (from German *Eigenwert*)
- ev – eigenvector (from German *Eigenvektor*)
- iff – "if and only if"
- LC – linear combination
- LD – linearly dependent
- LI – linearly independent

A.2 Mathematical notation

- $\in$ – Membership in a set. For example, $x \in \mathbb{R}$ means x is a member of the set $\mathbb{R}$
- $\forall$ – For all
- $\exists$ – There exists
- $\mathbf{x}^T$ – The transpose of a vector $\mathbf{x}$
- $\mathbf{x}^*$ – The conjugate transpose of a vector $\mathbf{x}$
- $\|\mathbf{x}\|$ – the magnitude of a vector $\mathbf{x}$
- A^{-1} – The inverse of the matrix A
- A^T – The transpose of the matrix A
- A^{-T} – The inverse transpose of the matrix A
- A^* – The adjoint (conjugate transpose) of the matrix A
- A^{-*} – The inverse adjoint of the matrix A

- $\mathbb{C}$ – The set of complex numbers
- $\mathbb{C}^n$ – The set of complex vectors of length n
- $\mathbb{C}^{m\times n}$ – The set of complex $m \times n$ matrices
- $\mathbb{N}_0$ – The set of natural ("counting") numbers, starting with zero
- $\mathbb{N}_1$ – The set of natural ("counting") numbers, starting with one
- $\mathbb{R}$ – The set of real numbers
- $\mathbb{R}^n$ – The set of real vectors of length n
- $\mathbb{R}^{m\times n}$ – The set of real $m \times n$ matrices
- $\mathbb{Z}$ – The set of integers (from the German *Zahlen*)
- $\bar{z}$ – complex conjugate of z. For a and b real, the conjugate of $z = a + ib$ is $a - ib$.

B Formal definition of a vector space

Definition 9 A set V is called a ***vector space*** if the following properties hold:

1. There is a binary operation between vectors, called *addition*, denoted "$+$", such that $u + v \in V$ for all $u, v \in V$.
2. Addition is *commutative*: $u + v = v + u$ for every $u, v \in V$
3. Addition is *associative*: $u + (v + w) = (u + v) + w$ for every $u, v, w \in V$
4. V contains an *additive identity*, called $\mathbf{0}$, such that $v + \mathbf{0} = v$ for all $v \in V$.
5. For every $v \in V$, there is an *additive inverse*, called $-v$, such that $v + (-v) = \mathbf{0}$.
6. There is a binary operation between a scalar and a vector, called *scalar multiplication*, such that $\alpha v \in V$ for all $\alpha \in \mathbb{C}$, $v \in V$.
7. Scalar multiplication is associative: $\alpha(\beta v) = (\alpha\beta) v$ for all $\alpha, \beta \in \mathbb{C}$ and for all $v \in V$.
8. Scalar multiplication distributes over addition: $\alpha(u + v) = \alpha u + \alpha v$ for all $\alpha \in \mathbb{C}$, $u, v \in V$.
9. The scalar zero times any vector v produces the zero vector: $0v = \mathbf{0}$ for all $v \in V$.
10. The scalar one times any vector v produces v: $1v = v$ for all $v \in V$.

The property that $u + v \in V$ is called ***closure under addition***. The property that $\alpha v \in V$ is called ***closure under scalar multiplication***.

Pick any two linear algebra textbooks and the precise statements of these properties will differ, but all will have logically equivalent sets of properties.

C Three ways of looking at matrix-vector multiplication

Let A be an $m \times n$ matrix and $\mathbf{x}$ a vector of dimension n . Then the matrix vector product $A\mathbf{x}$ can be defined in the three equivalent ways:

1. $A\mathbf{x}$ is the vector formed by the LC of cols (A) weighted by the entries of $\mathbf{x}$:
$$A\mathbf{x} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \cdots + \mathbf{a}_n x_n.$$

2. $A\mathbf{x}$ is the vector whose i-th entry is
$$(A\mathbf{x})_i = \sum_{j=1}^{n} A_{ij} x_j.$$

3. $A\mathbf{x}$ is the vector whose i-th entry is the dot product between $\mathbf{x}$ and the i-th row of A.

You should convince yourself that the three ways of doing matrix-vector multiplication are equivalent. The order matters: $A\mathbf{x}$ is defined as above; $\mathbf{x}A$ is nonsense.

When calculating $A\mathbf{x}$ by hand, most of us use the "dot product" method (#3 in the list); it's more convenient. However, for understanding the applications of matrices, the summation (#2) or the LC method (#1) will be more useful.

You should also convince yourself of the following:

- Matrix-vector multiplication distributes over linear combinations: $A(\alpha\mathbf{x} + \beta\mathbf{y}) = \alpha A\mathbf{x} + \beta A\mathbf{y}$.
- Matrix-vector multiplication and matrix-matrix addition are defined consistently, meaning that
$$(A + B)\mathbf{x} = A\mathbf{x} + B\mathbf{x}.$$

D Refresher: The area of a parallelogram

Here I will show that the oriented area of a parallelogram P defined by vectors $\mathbf{a}$ and $\mathbf{b}$. From the construction in figure 2, the area of P is identical to the area of the rectangle outlined in blue dashed lines. The rectangle has sides of length $\|\mathbf{a}\|$ and $\|\mathbf{b}\| \sin\theta_{ba}$, so the area of P is
$$\text{Area}(P) = \|\mathbf{a}\| \, \|\mathbf{b}\| \sin\theta_{ba}. \tag{2}$$

We compute $\sin\theta_{ba}$ with angle addition:

$$\begin{aligned}\sin\theta_{ba} &= \sin(\theta_b - \theta_a) \\ &= \cos\theta_a \sin\theta_b - \cos\theta_b \sin\theta_a\end{aligned}$$

which we can express in terms of the Cartesian components of **a** and **b**,

$$\sin\theta_{ba} = \frac{a_x}{\|\mathbf{a}\|}\frac{b_y}{\|\mathbf{b}\|} - \frac{a_y}{\|\mathbf{a}\|}\frac{b_x}{\|\mathbf{b}\|}.$$

Then compute the area using equation 2 and the expansion of $\sin\theta_{ba}$,

$$\begin{aligned} \text{Area}\,(P) &= \|\mathbf{a}\|\,\|\mathbf{b}\|\sin\theta_{ba} \\ &= a_x b_y - a_y b_x \\ &= \det\begin{pmatrix} a_x & b_x \\ a_y & b_y \end{pmatrix}. \end{aligned}$$

Most multivariate calculus books (*e.g.* [4]) will show how to extend this construction to 3D. Inductive extension to arbitrary dimension n is shown in [2].

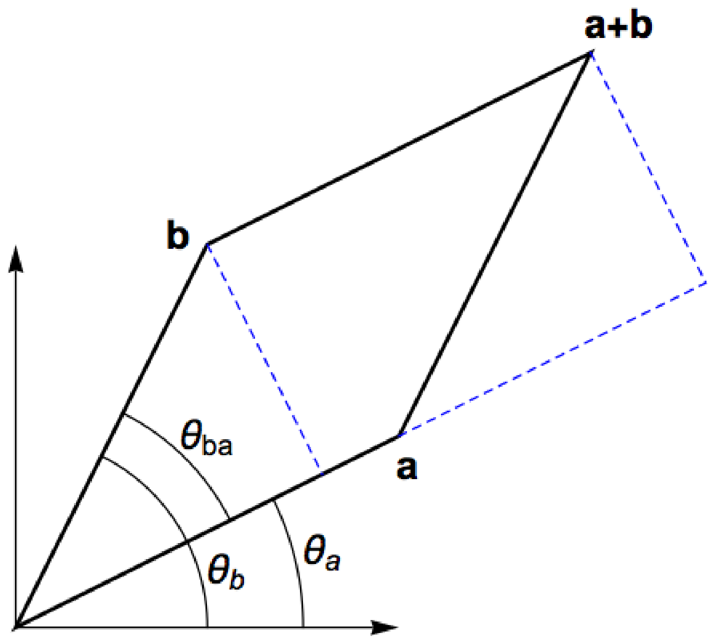


Figure 2: The area of the parallelogram defined by **a** and **b** is equal to the area of the rectangle outlined in blue. That rectangle has sides of length $\|\mathbf{a}\|$ and $\|\mathbf{b}\|\sin\theta_{ba}$.

E Refresher: Gaussian elimination

You've very likely seen row reduction and Gaussian elimination before. Recall that we solve $Ax = b$ by constructing an augmented matrix $\left[\begin{array}{c|c} A & b \end{array}\right]$, then doing row operations to put the augmented matrix in row echelon form. The solution can then by computed by backsubstitution.

In addition to my presentation here, you might look at the very good MIT OpenCourseWare video at `https://www.youtube.com/watch?v=xCIXkm3-ocQ` in which an example is worked out in detail.

E.1 Simplest case: no row exchanges

Let's solve

$$\begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & 3 \\ 1 & 1 & 5 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}$$

in this way. Form the augmented matrix. The element to be eliminated is circled; the row operation that eliminates it is indicated above the arrow. Start by eliminating the 2,1 element:

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 2 & 1 & 3 & 0 \\ 1 & 1 & 5 & 2 \end{bmatrix} \overset{R_2 \leftarrow R_2 - 2R_1}{\Longrightarrow} \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & -1 & -2 \\ 1 & 1 & 5 & 2 \end{bmatrix}.$$

Then the 3,1 element

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & -1 & -2 \\ 1 & 1 & 5 & 2 \end{bmatrix} \overset{R_3 \leftarrow R_3 - R_1}{\Longrightarrow} \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & -1 & -2 \\ 0 & 1 & 3 & 1 \end{bmatrix}$$

and the 3,2 element

$$\begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & -1 & -2 \\ 0 & 1 & 3 & 1 \end{bmatrix} \overset{R_3 \leftarrow R_3 - R_2}{\Longrightarrow} \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 4 & 3 \end{bmatrix}.$$

The system is now in row echelon form so we're ready to do backsubsitution. Work up from the bottom: the third row encodes an equation for c_3

$$\begin{bmatrix} 0 & 0 & 4 & 3 \end{bmatrix} \quad \text{means} \quad 4c_3 = 3 \;\; \text{so} \;\; c_3 = \frac{3}{4}.$$

Then use the second row to solve for c_2

$$\begin{bmatrix} 0 & 1 & -1 & -2 \end{bmatrix} \quad \text{means} \quad c_2 - c_3 = -2 \;\; \text{so} \;\; c_2 = -2 + \frac{3}{4} = -\frac{5}{4}$$

and the first row to solve for c_1

$$\begin{bmatrix} 1 & 0 & 2 & 1 \end{bmatrix} \quad \text{means} \quad c_1 + 0c_2 + 2c_3 = 1 \;\; \text{so} \;\; c_1 = 1 - \frac{3}{2} = -\frac{1}{2}.$$

E.2 Elimination with row exchanges

Sometimes you'll encounter a zero on the diagonal. That's a problem, because you can't use it to eliminate entries below it and ultimately because you'll need to divide by the diagonal entries.. When you hit a zero on the diagonal, swap rows until you find a nonzero. Here we solve the system

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 2 & 3 \\ 1 & 2 & 5 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}.$$

The first step introduces a zero on the diagonal:

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 2 & 2 & 3 & 0 \\ 1 & 2 & 5 & 2 \end{bmatrix} \overset{R_2 \leftarrow R_2 - 2R_1}{\Longrightarrow} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & -1 & -2 \\ 1 & 2 & 5 & 2 \end{bmatrix}$$

which causes no problems with the next step

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & -1 & -2 \\ 1 & 2 & 5 & 2 \end{bmatrix} \overset{R_3 \leftarrow R_3 - R_1}{\Longrightarrow} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & -1 & -2 \\ 0 & 1 & 3 & 1 \end{bmatrix}$$

but at this point, we can't use rows 2 and 3 to eliminate the 3,2 entry because the 2,2 entry is zero. (Why can't we just use row 1 to eliminate 3,2? Think about it: what would go wrong if we did that?) So swap rows 2 and 3, moving the zero away from the diagonal.

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & -1 & -2 \\ 0 & 1 & 3 & 1 \end{bmatrix} \overset{R_2 \leftrightarrows R_3}{\Longrightarrow} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 3 & 1 \\ 0 & 0 & -1 & -2 \end{bmatrix}.$$

Now the matrix is in row echelon form (in a large system we might have additional rows to reduce, but with only 3 rows we're done). Do backsubstitution:

$$c_3 = 2$$

$$c_2 = 1 - 3c_3 = -5$$

$$c_1 = 1 - c_2 - 2c_3 = 2.$$

E.3 When Gaussian elimination fails

Unfortunately, in some cases we may be unable to clear all zeros from the diagonal. Here's such a case: try to solve

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix}.$$

Eliminate the 2,1 entry

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 1 & 2 \end{bmatrix} \overset{R_2 \leftarrow R_2 - R_1}{\Longrightarrow} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

The 3,1 entry happened to be zero so we can skip that step. Move on to eliminate the 3,2 entry:

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \overset{R_3 \leftarrow R_3 - R_2}{\Longrightarrow} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 3 \end{bmatrix}.$$

Oops. We could eliminate 3,2 but in doing so we also killed 3,3. There are no more rows to swap, so we're stuck with a zero diagonal. Now the row

$$\begin{bmatrix} 0 & 0 & 0 & 3 \end{bmatrix}$$

is shorthand for the equation

$$0c_1 + 0c_2 + 0c_3 = 3$$

which has no solution for any c_1, c_2, c_3.

A system where after row reduction we have one or more zeros on the diagonal is called *singular.* A case such as this one, where no solution was found, is called *singular and inconsistent*.

E.3.1 Consistent versus inconsistent singular systems

In the previous example we had no solution. That's not always the case. Consider the slightly different problem

$$\begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}.$$

Notice that the matrix is the same, but the right hand side differs in the 3rd entry.

Proceed with Gaussian elimination: eliminate the 2,1 entry

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \underset{\Longrightarrow}{R_2 \leftarrow R_2 - R_1} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 \end{bmatrix},$$

skip the 3,3 entry (which is already zero) and eliminate the 3,2 entry:

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \underset{\Longrightarrow}{R_3 \leftarrow R_3 - R_2} \begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Once again, we killed the diagonal 3,3, but the final row is now

$$\begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}$$

is shorthand for the equation

$$0c_1 + 0c_2 + 0c_3 = 0.$$

This equation can be solved by *any* c_1, c_2, c_3. Ordinarily, we'd use the final row to solve for c_3, but in this case, we can't determine c_3 since any c_3 will do. So let's pick c_3 to be an arbitrary parameter, say, α. Then proceed back to the second row,

$$0c_1 + c_2 + c_3 = -1$$

or

$$c_2 = -1 - \alpha.$$

Use the first row to solve for c_1,

$$\begin{aligned} c_1 + c_2 + 2c_3 &= 1 \\ c_1 &= 1 - c_2 - 2c_3 \\ &= 1 + (1+\alpha) - 2\alpha \\ &= 2 - \alpha. \end{aligned}$$

The solution vector is

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 2-\alpha \\ -1-\alpha \\ \alpha \end{bmatrix}$$

or

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} \alpha.$$

This is the parametric equation for a line in 3D, with parameter α. There are infinitely many points on the line, one for each α, so we have infinitely many solutions.

This system is still singular; we encountered a zero diagonal we couldn't remove. But unlike the previous example having no solutions, here we had infinitely many solutions. We call such a system ***singular and consistent***.

Notice that these last two examples both had the same matrix

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

but different vectors on the right hand sides. Whether a system is singular depends on the matrix, not the right hand side, so it's appropriate to describe a *matrix* as singular or not. Whether a system is consistent depends on both the matrix and the right hand side.

F Practical algorithms for solving systems of equations

In real world problems with large matrices and floating-point arithmetic, solving $A\mathbf{x} = \mathbf{b}$ is a challenge. Simple variants of Gaussian elimination can be used for problems up to a few tens of thousands of unknowns, and if the matrix is ***sparse*** (meaning most entries are zero; this happens quite often in engineering problems) more complicated variants of Gaussian elimination that arrange the calculation to ignore any zero entries can solve problems up to a few hundred thousand unknowns. For large problems, ***iterative*** methods must be used; these construct an approximate solution through a sequence of matrix-vector multiplications.

F.1 Row pivoting

In the examples above we exchanged rows only when forced to do so by the presence of a zero on a diagonal. In exact calculations that is sufficient. However, when rounded-off arithmetic is used, straightforward Gaussian elimination is highly susceptible to roundoff error. The solution is simple: when starting down a new column, examine the values in that column on the diagonal or below, and then exchange rows so that the largest (in absolute value) entry goes on the diagonal. This is called ***Gaussian elimination with row pivoting*** (GERP) and it is seen in practice to be very robust. The worst-case behavior of GERP is very poor (meaning that in the worst case, roundoff errors can be amplified exponentially) but, interestingly and fortunately, that worst-case behavior almost never happens except in contrived examples. For reasons that are still not fully understood, in real-world problems GERP is observed to be much more stable than its theoretical worst-case roundoff error growth rate would suggest. For all practical purposes you can consider it to be robust and reliable on any problem you're likely to encounter in the real world. It's theoretically *possible* for it to go horribly wrong, but it almost certainly won't. Or, as Jim Wilkinson – the man who first derived the very pessimistic error bounds back in 1961 – put it,

> *"Anyone that unlucky will have already been run over by a bus."*

A more stable variant of GE is to exchange *both* rows and columns to put the largest element on the diagonal at each step; this is called ***Gaussian elimination with full pivoting***. However, this is more expensive and is rarely used. GERP is reliable enough in practice that full pivoting is not needed.

F.2 The LU factorization

Suppose you need to solve two systems having the same matrix but different right-hand sides: $A\mathbf{x}_1 = \mathbf{b}_1$ and $A\mathbf{x}_2 = \mathbf{b}_2$. You could go through Gaussian elimination twice, but if you think about it when you do row reduction on the two different augmented matrices you're doing exactly the same operations on all but the final column of the augmented matrices. It would probably occur to you to record both the sequence of operations and the final upper triangular form. Then, when solving a new system you could do the recorded operations to the right-hand side vector and then do backsubstitution against the recorded upper triangular form to find the solution.

Sure enough, somebody[3] has already thought to do just that. Remarkably, all the row operations can be recorded in a convenient form as a product of a permutation matrix P

[3] Like most good ideas, more than one person thought of it; Doolittle, Banacheiewicz, Crout, Dwyer, and Turing all contributed. The great British mathematician Alan Turing was perhaps the first to appreciate fully its implications for automated computing. He also formulated the problem in modern notation, and made the idea widely known. Turing also pioneered computer programming and theoretical computer science, proposed the "Turing test" for artficial intelligence, and explained biological pattern formation in terms of reaction-diffusion equations. Perhaps his most important contribution to civilization was his work on breaking Nazi codes during the Second World War; this work was essential to winning the Battle of the Atlantic.

times a lower triangular matrix L^{-1} having ones on the diagonal.

$$L^{-1}PA = U$$

or

$$PA = LU.$$

A permutation matrix is an identity matrix with certain rows exchanged; multiplication from the left by a permutation matrix, PA, produces a copy of A with rows exchanged as specified by P. Permutation matrices have the property that $P^{-1} = P^T$, so that it is easy to solve systems involving P (this shouldn't be surprising; to apply P^{-1} simply undo the row exchange done by P). What is surprising is that the matrix L^{-1} which records the row operations can be inverted very simply with no calculations.

Once you've factored $A = P^T LU$, you can solve a system $Ax = b$ with two triangular solves: first solve $Ly = Pb$ for y, then $Ux = y$ for x. Triangular solves require backsubstitution, which is easy.

F.3 Why you shouldn't compute a matrix inverse

The cost (in floating point operations) of doing row reduction on an $m \times m$ matrix to upper triangular form by Gaussian elimination with row pivoting works out to be $\frac{2}{3}m^3$. The LU factorization is simply a convenient encoding of that algorithm, with no extra cost. The cost formula can be derived by induction, but it's more useful to understand it as follows: to do row reduction, you need to eliminate the $\approx \frac{1}{2}m^2$ entries below the diagonal. To kill each subdiagonal entry you need to do one row operation. Each *full length* row operation takes $\approx 2m$ elementary floating point arithmetic operations (flops), so a first estimate for the cost would be $\approx m^3$. But notice that as you proceed through the algorithm, the columns already zeroed out can be ignored so the row operations become cheaper as you go, making the m^3 estimate an overestimate. The exact factor can be worked out (pretty easily), and it turns out to be $2/3$.

I'll ignore pivots (row exchanges) because they don't change the operation counts; there's a slight cost to pivoting because indirect addressing is necessary, but for moderate size dense matrices (up to a few thousand) the whole thing fits in cache memory so addressing is essentially free. To include pivots, just write A as $P^T LU$ where P is a permutation matrix.

Once you've factored $A = LU$, you can solve a system $Ax = b$ with two triangular solves: first solve $Ly = b$ for y, then $Ux = y$ for x. The cost of one triangular solve is m^2 flops (you do two operations with half the elements, so twice $m^2/2$). So solving $Ax = b$ with one RHS costs $\frac{2}{3}m^3 + 2m^2$. Solving r systems with different right-hand sides costs

$$\underbrace{\frac{2}{3}m^3}_{LU\text{ factorization}} + \underbrace{2rm^2}_{2r\text{ triangular solves}}.$$

Now compare to the cost of solving these systems by computing A^{-1} once, then multiplying A^{-1} by each of r right-hand sides. To find A^{-1}, solve the system $AX = I$ for X. The best way to do this is to factor A into LU, then use it to solve $AX_j = e_j$ (where e_j is the j-th column of the identity) using the method above. Since there are m such systems to solve, the cost of forming the inverse is given by the formula above with $r = m$, or

$$\frac{8}{3}m^3.$$

Now that you have the inverse, solving $Ax = b$ requires a matrix-vector multiplication $x = A^{-1}b$. It's easy to see that each matrix-vector multiplication costs $2m^2$ operations. Doing r matrix-vector multiplications requires $2rm^2$ operations, exactly the same as the cost of doing the triangular solves in the LU method; that is, once you've formed A^{-1} the marginal cost of solving one more system is exactly the same as if you'd simply stored LU. Unfortunately, since you already had to find LU to compute $A^{=1}$, finding and using the inverse is never cost effective.

To summarize:

- Cost of factoring $A = LU$, then doing 2 triangular solves for each of r right-hand sides:
$$\underbrace{\frac{2}{3}m^3}_{LU\text{ factorization}} + \underbrace{2rm^2}_{2r\text{ triangular solves}}.$$
- Cost of computing A^{-1}, then doing matrix-vector multiplication for each of r right-hand sides:
$$\underbrace{\frac{2}{3}m^3}_{LU\text{ factorization}} + \underbrace{2m^3r}_{2m\text{ triangular solves to find }A^{-1}} + \underbrace{2rm^2}_{r\text{ matrix-vector multiplications }A^{-1}b}$$
$$= 2m^3 \text{ overhead beyond solving the same problem without the inverse.}$$

Elementary textbooks often say that you only compute an inverse when you're going to solve multiple systems, but you can now see that even then it's simply not efficient.

You might use an inverse in cases where forming the inverse is trivial: for example, to invert a diagonal matrix just replace the diagonal elements by their reciprocals. Another important example is an orthogonal matrix, where the inverse turns out to be the transpose, no need for any calculations. Finally, there's a simple formula for the inverse of a two-by-two matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

that's handy for quick calculations.

In any real production calculation, ***you should never compute a non-trivial inverse***. And ***never, never use Cramer's Rule with Laplace expansion of the determinant for numerical calculation***.

G Computing determinants numerically

The first thing to say is this: If you think you need to compute a determinant numerically, you probably don't. There's really only one reason to compute one, and that's if you need to compute a volume or hypervolume. If you're doing a numerical integral, yep, you'll need a determinant. Otherwise, probably not. If you're thinking of computing a determinant numerically to solve a system or to compute eigenvalues, stop now and read about the right ways to do those calculations. The right ways don't involve determinants.

Okay, suppose you really do need to compute $\det(A)$ for some numerical matrix A. Here's how.

1. Use your favorite LU factorization code to factor A into PLU, where P is a permutation matrix, L is lower triangular with ones on the diagonal, and U is upper triangular.
2. $\det(P)$ will be -1 for an odd permutation and 1 for an even permutation.
3. $\det(L)$ will be one.
4. $\det(U)$ will be the product of the diagonals entries in U.
5. $\det(A) = \det(P)\det(L)\det(U)$, or

$$\det(A) = \det(P)\prod_{i=1}^{m} U_{ii},$$

   where, again, $\det(P)$ is ± 1 depending on the parity of the permutations.

G.1 How to wait a lifetime and get the wrong answer: expansion in minors

If A is $m \times m$, then computing $\det(A)$ by expansion in minors takes $m!$ operations. (Can you see why? Hint: induction.) The factorial grows *very* fast.

A typical GPU can do a teraflop: 10^{12} floating point operations per second. There are 3.15×10^7 seconds in a year. So your GPU will take $\approx 3 \times 10^{-20}$ years (a thousandth of a nanosecond) to do one flop. The fact that I'm using *years*, not seconds, as my time units should alarm you. The problem is that $m!$ grows so darn fast with m.

m	$m!$
10	3.63×10^8
20	2.43×10^{18}
22	1.12×10^{21}
25	1.55×10^{25}
30	2.65×10^{32}

- For $m = 10$, your determinant will be done in about a microsecond. So far, so good.
- For $m = 20$, your determinant will be done in 0.07 years, a bit under a month. Be sure you have a good battery backup, and you'll get there.
- For $m = 22$, you'll wait 33.7 years. Start now, and you'll be done when your grandkids are in elementary school.

- For $m = 30$... you'll wait 8×10^{12} years. For comparison, the Big Bang was 1.3×10^{10} years ago.

To add insult to injury, after patiently waiting a few weeks, or 8×10^{12} years, for the determinant, it will be wildly inaccurate. Remember all those alternating signs in the expansion by minors? That means you're doing lots of subtractions of quantities of similar magnitude, which is ripe for roundoff error.

Don't do it. Use the LU factorization method instead. An LU factorization takes $\frac{2}{3}m^3$ flops; you'll be able to do large systems in the blink of an eye rather than a lifetime, and you'll have an accurate determinant when done.

Even for a 3 by 3 matrix, use the LU method.[4]

H Further reading

- There are many introductory linear algebra textbooks. One of the better choices is [5]. For a concise and inexpensive introduction to linear algebra, it's hard to go wrong with the Schaum's Outline, [3].
- A linear algebra textbook aimed specifically at applications to differential equations is [6].
- For more information on numerical computation with matrices, see [8], or the more advanced [7].
- Introductory books for the more mathematically mature[5] student are [2] and [1]. Lax starts with determinants (from the geometric viewpoint), while Axler holds them off to the very end; there are many ways to approach the same subject.

References

[1] Sheldon Axler. *Linear algebra done right*, volume 2. Springer, 1997.

[2] Peter D Lax. *Linear Algebra and its Applications*. 2007.

[3] Seymour Lipschutz and Marc Lipson. *Schaum's Outline of Linear Algebra*. McGraw Hill Professional, 2017.

[4] Karl J. Smith, Monty J. Strauss, and D. Toda, Magdalena. *Calculus*. Kendall Hunt, 2013.

[4] I say this from experience. Once when writing a finite element code – which requires a *lot* of 3D numerical integrals – I got lazy and hand-coded the easy formula for the determinant of a 3×3 instead of doing the right thing and hooking in an LU function. I knew the right way to do it, but was in a rush and figured a 3×3 would be small enough to avoid catastrophic roundoff. Sure enough, upon testing, the code was coming in below accuracy requirements and I tracked it down to roundoff error in the stupid 3×3 determinants.

[5] "Mathematically mature" means "totally comfortable working with proofs"

[5] Gilbert Strang. *Linear algebra and its applications*. 4th edition, 2006.

[6] Gilbert Strang. *Differential equations and linear algebra*. 2014.

[7] Lloyd Nicholas Trefethen and David Bau. *Numerical linear algebra*. SIAM, 1997.

[8] David Watkins. *Fundamentals of matrix computations*. Wiley, 3rd edition, 2010.