

The 18.C06 Master Document

Jack D.V. Carson

jdcarson@mit.edu

Contents

| | |
|-------------------------------------|----------|
| 1 Linear Algebra | 1 |
| 1.1 The Basics | 1 |
| 1.2 Vector Spaces | 5 |
| 1.3 Factorization (pt. 1) | 9 |
| 1.4 Orthogonality | 10 |

1 Linear Algebra

This document assumes you already have a reasonable familiarity with the most basic ideas of linear algebra (i.e. What is a matrix? What is matrix multiplication? Why would one be compelled to multiply a matrix?). From hence, we shall begin with the beginning.

1.1 The Basics

- In linear algebra, we most often want to express some situation in terms of a formula $Ax = b$ for a matrix A , and vectors x, b . Then, once we have expressed them in this form, we wish to solve for the values of x that give b . The simplest way to do this is with **Gaussian Elimination** wherein rows are added to one another or multiplied by scalar constants to give a **diagonal matrix**. This algorithm always goes columnwise. I.e. start from the first column and try and clear everything such that everything below the pivot is 0. This may not always be the fastest approach but it is consistent. From this diagonal matrix, it is simple to **backsubstitute** to get the complete values of x .

$$\left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 11 & 6 & 35 \end{array} \right] \xrightarrow[r_3 - 3r_1]{r_2 - r_1} \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 2 & 3 & 8 \end{array} \right] \xrightarrow{r_3 + r_2} \left[\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & -2 & -2 & -8 \\ 0 & 0 & 1 & 0 \end{array} \right]$$

Here we have gone from the augmented matrix $[A|b]$ to a much nicer $[U|b]$. By recalling that matrices are just an abstraction of linear functions, we can look at the last row and say that $x_3 = 0$. We can *propagate* this upwards to say therefore that $-2x_2 + 0 = -8 \rightarrow x_2 = 4$

- If we get a 0 in a pivot position, we say that a matrix is **singular**. In this case $Ax = b$ may not have a solution, or it may have infinite solutions. But it certainly does not have a single solution. All linear equations of this type either have 0, 1, or ∞ solutions.
- It wouldn't hurt us to brush up on matrix multiplication either. It turns out there are many ways to think about this operation.

1. **Entry-wise:** For each $1 \leq i \leq m$ and $1 \leq jp$ we have

$$C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$$

2. **Inner product:** C_{ij} is the dot product (also known as “inner product”) of the i th row in A and the j th column in B . For instance,

$$C = \begin{bmatrix} \text{---} & x_1 & \text{---} \\ \text{---} & x_1 & \text{---} \\ & \vdots & \\ \text{---} & x_1 & \text{---} \end{bmatrix} \begin{bmatrix} \left| \right. & \left| \right. & \cdots & \left| \right. \\ b_1 & b_2 & \cdots & b_p \\ \left| \right. & \left| \right. & & \left| \right. \end{bmatrix} = \begin{bmatrix} x_1 \cdot b_1 & x_1 \cdot b_2 & \cdots & x_1 \cdot b_p \\ x_2 \cdot b_1 & x_2 \cdot b_2 & \cdots & b_2 \cdot b_p \\ \vdots & \vdots & \ddots & \vdots \\ x_m \cdot b_1 & x_m \cdot b_2 & \cdots & x_m \cdot b_p \end{bmatrix}$$

3. **Column-wise:** the j th column of a matrix C is the matrix-vector product of A and the j th column of B . For instance,

$$C = A \begin{bmatrix} \left| \right. & \left| \right. & \cdots & \left| \right. \\ b_1 & b_2 & \cdots & b_p \\ \left| \right. & \left| \right. & & \left| \right. \end{bmatrix} = \begin{bmatrix} \left| \right. & \left| \right. & \cdots & \left| \right. \\ Ab_1 & Ab_2 & \cdots & Ab_p \\ \left| \right. & \left| \right. & & \left| \right. \end{bmatrix}$$

4. **Outer product:** C is the sum of the product of i th column of A and the i th row of B such as

$$C = \begin{bmatrix} \left| \right. & \left| \right. & \cdots & \left| \right. \\ a_1 & a_2 & \cdots & a_n \\ \left| \right. & \left| \right. & & \left| \right. \end{bmatrix} \begin{bmatrix} \text{---} & y_1 & \text{---} \\ \text{---} & y_2 & \text{---} \\ & \vdots & \\ \text{---} & y_n & \text{---} \end{bmatrix} = \sum_{i=1}^n \begin{bmatrix} \left| \right. & \left| \right. \\ a_i & \left| \right. \end{bmatrix} \begin{bmatrix} \text{---} & y_i & \text{---} \end{bmatrix}$$

Properties of Matrix Multiplication

1. **Associative:** $A(BC) = (AB)C$
2. **Distributive:** $A(B + C) = AB + AC \iff (A + B)C = AC + BC$
3. **Non-commutative:** $AB \neq BA$
4. **Identity:** $IA = AI = A$

- Although it can often be lost in the abstraction of mathematics, matrices really, truly are **linear operators**. They transform spaces and vectors to other spaces and other vectors. As an example consider 2023 Recitation 1 Problem 1

Example: 1.1

1. Find a 2×2 matrix such that when you multiply a 2-D vector by it, the result is a reflection of the vector across the origin
2. Find a 3×3 matrix such that when you multiply a 3-D vector by it, it swaps the second and third coordinates.

3. If you have a 4×4 matrix A , find a 4-D vector x such that Ax is the second column of A .

Solution (1.1).

1. To reflect across the origin, $x \mapsto -x$, $y \mapsto -y$. Therefore, $A = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$.

$$2. \text{ Here } \begin{bmatrix} x \\ y \\ z \end{bmatrix} \mapsto \begin{bmatrix} x \\ z \\ y \end{bmatrix} = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + y \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + z \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_A \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Here A is a **permutation matrix**, as it permutes one or more of the variables. Permutation matrices have a variety of desirable properties such as $A^{-1} = A^T$

3. Since we are only interested in the second column, we want the products $\mathbf{c}_i x_i = 0$ for $i = 1, 3, 4$ to be 0. The only way to guarantee this is to specify $x_{1,3,4} = 0$ and $x_2 = 1$. Therefore $\mathbf{x} = [0 \ 1 \ 0 \ 0]^T$

Takeaways (1.1).

1. All of the matrix formulations here are useful to recognize, particularly the intuition of part (3). This will be seen many times.
2. One useful way to interpret the result in part (2) is the fact that left-multiplied matrices operate on the rows of that which they multiply. Right-multiplied matrices operate on columns. Because \mathbf{x} has only one column, right-multiplication is nonsensical.

Example: 1.2

1. Find a 3×3 matrix P such that in $B = PA$ is the result of subtracting the second row from the third row of A and then swapping the first and second rows.
2. Find a 4×4 matrix that right multiplies A such that result $C = AQ$ is A after dividing the first column by two, and then adding the first column to the second and third columns
3. Does the order of performing the operations in (1) and (2) matter?

Solution (1.2).

$$1. \ P = \underbrace{\begin{bmatrix} 1 & & \\ & 1 & \\ -1 & 1 & \end{bmatrix}}_{r_3=r_3-r_2} \underbrace{\begin{bmatrix} & 1 & \\ 1 & & \\ & & 1 \end{bmatrix}}_{\text{swap } \mathbf{c}_1, \mathbf{c}_2} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

$$2. Q = \begin{bmatrix} 1/2 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 1/2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \text{ This multiplication}$$

should be obvious by the column-wise definition above.

3. Although it might look like we are dealing with the commutative property here, we are really dealing with the associative property. In (1). $B = (P_1 P_2)A = P_1(P_2 A)$. So the order of operations does not matter. In (2) it is the same $C = A(Q_1 Q_2) = (A Q_1)Q_2$

Takeaways (1.2).

- Remember that left multiplication always affects rows only, and that right multiplication affects columns only. Complex operations can be formed by chaining linear operators together.
- Matrix multiplication is associative! When doing these chained operations, the order does not matter.
- Remember the forms that these kinds of matrices take. They are not always obvious.

- Before we close up, there are some other noteworthy operations we can perform with matrices that will follow us around. The **transpose** of a matrix A^T “flips” a matrix A such that

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

and with it we can define certain identities such as $(A^T)_{i,j} = A_{j,i}$ and $(AB)^T = B^T A^T$. Finally, we can define the dot product between two vectors $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b}$. All three of these facts will show up constantly.

- Last but not least, the **inverse** A^{-1} of a matrix is the matrix such that $AA^{-1} = A^{-1}A = I$. More interestingly we can say that in a situation $Ax = b$, $x = A^{-1}b$. This matrix inverse does not always exist, and is exceedingly impractical to calculate for large matrices, an issue we will deal with thoroughly in optimization. A matrix is said to be **invertible** if and only if it is square and full column rank. That is, every column has a pivot. The inverse, if it exists, is always unique. Like the transpose it is subject to the identity $(AB)^{-1} = B^{-1}A^{-1}$
- We can define more facts that will help us with these two operations. For instance $(A^T)^T$ and $(A+B)^T = A^T + B^T$, which follow directly from the definition of the transpose. More useful is $(A^{-1})^T = (A^T)^{-1}$. All of these identities will help us greatly.

1.2 Vector Spaces

Definition

A **vector space** V is a set of elements (e.g., vectors in \mathbb{R}^n , polynomials, diagonal 2×2 matrices) defined over a “field” F of scalars that are closed under

1. **Vector Addition:** For any vectors u and v in V , $u \pm v$ also belongs the vector space V .
2. **Scalar Multiplication:** For any vector v in V and scalar c in F , cv is an element of the vector space

- If $W \subseteq V$ is also a vector space with respect to the operations in V , W is a **subspace** of V . Specifically for any $v, w \in W$, $v + w \in W$ and $cv \in W$. For instance if we define a vector space \mathbb{R}^2 . W could be \mathbb{R}^2 or $W = \{0\}$. A more enlightening sample vector space is a line that passes through the origin (note that a line that does not pass through the origin would not be a vector space since $\alpha v \notin V$ for $\alpha = 0$). Every point on this line is closed under addition and scalar multiplication.
- There are an infinite number of obscure facts to say about spaces and subspaces. More interestingly, is how to show something is a space or a subspace. We can be convinced that for subspaces S_1 and S_2 $S = S_1 \cap S_2$ is also a subspace by showing that S is closed under addition and scalar multiplication.

Proof. Assuming that this is true, any $v, w \in S$ will also be $v, w \in S_1$ and $v, w \in S_2$ by the definition of an intersection. Then $v + w$ will be in S_1 and S_2 by the definition of subspace. Thus $v + w \in S_1 \cap S_2 = S$ which shows that it is closed under addition. Then we can define $v \in S_1 \cap S_2$ and $\alpha \in \mathbb{R}$. Since S_1 is a subspace, $\alpha v \in S_1$ and S_2 such that $\alpha v \in S_1 \cap S_2 = S$, showing that it is closed under scalar multiplication.

Common vector spaces

- The **column space** of an $m \times n$ matrix A , denoted $C(A)$ is the set of linear combinations of columns of A , also known as the **span** of A . Although the idea may seem a bit abstract, soon we will be very interested in what column space a vector is in. We can define useful facts such as
 1. Formally, $C(A) = \{Ax \mid x \in \mathbb{R}^n\}$
 2. $Ax = b$ has a solution if and only if $b \in C(A)$
 3. If $m = n$, then A is invertible if and only if $C(A) = \mathbb{R}^n$
- The **null space** of A , denoted $N(A)$ is the set of vectors x such that $Ax = \mathbf{0}$. We can define similar facts
 1. $N(A) = \{x \in \mathbb{R}^n \mid Ax = \mathbf{0}\}$
 2. If B is square and invertible, $N(A) = N(BA)$
 3. If $A \in \mathbb{R}^{n \times n}$, then $C(A) = \mathbb{R}^n$ is equivalent to $N(A) = \{\mathbf{0}\}$

Computing basis of a null space

- Before we start with computation, it helps to briefly examine our matrix. If we are looking at an invertible matrix, then its null space will clearly be $\{\mathbf{0}\}$, since there is no non-trivial x that will give $\mathbf{0}$. Even if A is not invertible, if every column has a pivot, then there are no variables that can move freely. This can happen easily for an “overdetermined” system where a matrix has more rows than columns. However, the most interesting case in which A is “underdetermined” such that it has many more columns than rows, and thus many free variables.
- Our goal for underdetermined systems will be to transform some matrix A

$$A \rightsquigarrow U = \begin{bmatrix} U_r & F \\ m - r \text{ rows of } 0's & \dots \end{bmatrix}$$

- Now how do we implement this scary looking transformation? In reality it is quite simple. First transform A into an upper-triangular matrix U with gaussian elimination.

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 4 & 5 & -3 \\ 1 & 6 & 7 & -7 \end{bmatrix} \rightsquigarrow \begin{bmatrix} \boxed{1} & 2 & 3 & 1 \\ & \boxed{2} & 2 & -4 \end{bmatrix} \quad (1)$$

Let pivot columns $U_r = \begin{bmatrix} 1 & 2 \\ 2 & -4 \end{bmatrix}$ and free columns $F = \begin{bmatrix} 3 & 1 \\ 2 & -4 \end{bmatrix}$ such that $U = \begin{bmatrix} U_r & F \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$.

- Then we can start with the actual linear algebra. To find $N(A)$ we want $Ax = \mathbf{0} \iff Ux = \mathbf{0}$. In order to compute this we define two vectors \mathbf{p} for the coefficients of x by which the pivot columns of U are multiplied, and a vector \mathbf{f} for the values of x that multiply the free columns. With this we can say,

$$U = \begin{pmatrix} U_r & F \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} U_r \mathbf{p} + F \mathbf{f} \\ \mathbf{0} \end{pmatrix} = \mathbf{0} \quad (2)$$

$$U_r \mathbf{p} + F \mathbf{f} = \mathbf{0} \quad (3)$$

$$\boxed{U_r \mathbf{p} = -F \mathbf{f}} \quad (4)$$

U_r is guaranteed to be invertible, so we can say $\mathbf{p} = U_r^{-1}(-F \mathbf{f})$ is guaranteed to be uniquely determined for any choice of \mathbf{f} . Then $\begin{pmatrix} \mathbf{p} \\ \mathbf{f} \end{pmatrix}$ is a basis vector of our nullspace. To show this, let's expand the boxed formula above for our example.

$$\begin{bmatrix} 1 & 2 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = - \begin{bmatrix} 3 & 1 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (5)$$

- For an n -dimensional null space, we can just make up whatever n linearly independent \mathbf{f} vectors we might fancy. For simplicity, say $\mathbf{f} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ such that

$$\begin{bmatrix} 1 & 2 \\ 2 & -4 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \underbrace{\begin{bmatrix} -3 \\ -2 \end{bmatrix}}_{-F \mathbf{f}} \longrightarrow \mathbf{p} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \text{ and } x = [-1 \quad -1 \quad 1 \quad 0]^T \quad (6)$$

Then we can do this again for $\mathbf{f} = [0 \quad 1]^T$ to complete the basis for $N(A)$.

- In the case that the pivot columns are not adjacent, you *can* interlace the p and f elements systematically.

Computing the basis of a columnspace

- Let's revisit our matrix A from the previous example.

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 4 & 5 & -3 \\ 1 & 6 & 7 & -7 \end{bmatrix} \rightsquigarrow U = \begin{bmatrix} 1 & 2 & 3 & 1 \\ & 2 & 2 & -4 \\ & & & 0 \end{bmatrix} \quad (7)$$

In U we can easily identify that c_1 and c_2 contain our pivots. Therefore, our basis for $C(A)$ is simply columns c_1 and c_2 of A itself. Don't forget, though, that $C(A) \neq C(U)$. Rather,

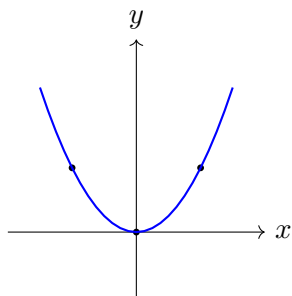
$$C(A) = \text{span} \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \right\} \quad (8)$$

Polynomial fitting

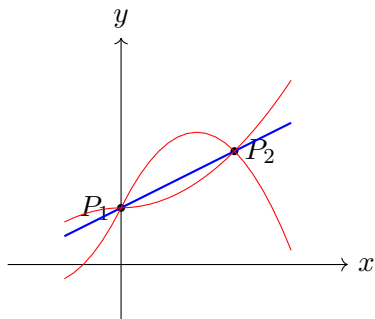
- The topic of optimization begins with polynomial fitting. This is a task we will revisit extensively in this course. A simple way to fit a polynomial is with a **Vandermonde Matrix**. A degree-2 polynomial follows a form with linear coefficients $p(x) = c_0 + c_1x + c_2x^2$, leading to the intuition

$$\underbrace{\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix}}_A \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

- This actually gives us a great way to visualize overdetermined and underdetermined systems.



$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix} \rightarrow \underbrace{A \in \mathbb{R}^{3 \times 3}}_{1\text{-sol}} \rightarrow N(A) = \{\mathbf{0}\}$$



$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \rightarrow \underbrace{A \in \mathbb{R}^{2 \times 3}}_{\text{underdetermined}} \rightarrow N(A) \neq \{\mathbf{0}\}$$

- As we can see with the figures, if we have two points, we can define exactly one line. If we have 3 points, we can define exactly 1 parabola. This is a formalization of the intuitive fact that we can draw an infinite number of parabolas to fit two points.

Example: 1.3

1. If the zero vector is in $C(A)$, then the columns of A are linearly dependent.
2. The columns of a matrix are a basis for the column space
3. Define the row space of A as the span of the row vectors. If A is square, then the row space equals the column space.
4. The row space of A is equal to the column space of A^T .
5. If the row space of A equals the column space, then $A^T = A$.
6. A 4×4 permutation matrix has $C(P) = \mathbb{R}^4$.
7. For $v \in N(A)$, if x is a solution to $Ax = b$, so is $x + v$

Solution (1.3).

1. **False.** $A = I$ is a counterexample. The zero vector is in the column span of every matrix!
2. **False.** This is true only if the columns are linearly independent. It is clearly not true for underdetermined systems where there are more columns than rows.
3. **False.** As a counterexample $A = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}$ where $C(A) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $R(A) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$
4. **True.** This is simply the definition of the row space, and the set of rows of A is identical to the set of columns of A^T .
5. **False:** Consider $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$. Both $C(A) = \mathbb{R}^2$ and $R(A) = \mathbb{R}^2$. However, $A \neq A^T$.
6. **True:** A permutation space simply permutes the order of the variables. If it were to not be in \mathbb{R}^4 , it would somehow lose one of the dimensions.
7. **True:** If $v \in N(A)$, $Av = \mathbf{0}$. Therefore $A(x + v) = Ax + av = b$ which is defined in $C(A)$

Takeaways (1.3).

1. We will see the row space $C(A^T)$ in more detail soon. This problem examines simple properties that it has. It is essential to remember that $C(A) \neq C(A^T)$, even if they have the same dimension.
2. Always look for simple matrices of size 2×2 or smaller to find counterexamples. Most false properties will collapse even by this size.

Example: 1.4

$AB = 0$ (the zero matrix) for matrices A and B . If the null space of $---$ is $\{=, \subseteq, \supseteq\}$ the column space of $---$?

Solution (1.4). Recalling our “Column-Wise” definition for matrix multiplication

$$AB = \begin{bmatrix} | & | & & | \\ Ab_1 & Ab_2 & \cdots & Ab_p \\ | & | & & | \end{bmatrix}$$

$AB = 0$ only if every column of B is in the null space of A . Therefore the $N(A)$ must contain any possible linear combination of B and possibly more. Therefore $N(B) \supseteq C(A)$

1.3 Factorization (pt. 1)

- Before we get to factorization, we should define some simple concepts that have already appeared. For instance, the **rank** of a matrix is simply the number of pivots. Simply, $\text{rank } A = \dim C(A) = \dim C(A^T)$, which should be intuitive. We can also define the **rank-nullity theorem** as $\dim N(A) = n - r$ for r pivots and n columns. Although we do not yet have the tools to introduce them all now, in this document we will examine **four critical factorizations**

$$A = LU \tag{9}$$

$$A = QR \tag{10}$$

$$S = Q\Lambda Q^T \tag{11}$$

$$A = U\Sigma V^T \tag{12}$$

LU Factorization

- It turns out that matrices can be described in many ways. Some of these ways can help you solve difficult problems. The first of these is the **LU-factorization**, which relates A to its upper-triangular form U . In the process, it tells us about the inverse of a matrix.
- The crux of the factorization is just Gaussian elimination of $A \rightsquigarrow U$, but storing the coefficients of each row operation in a left-multiplying matrix L

$$A = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 5 & 1 \\ -3 & 1 & -1 \end{bmatrix} \xrightarrow[r_3+3r_1]{r_2-2r_1} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 7 & -1 \end{bmatrix} \xrightarrow{r_3-7r_2} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & -8 \end{bmatrix} = U \tag{13}$$

$$\implies L = \begin{bmatrix} 1 & & \\ +2 & 1 & \\ -3 & +7 & 1 \end{bmatrix} \longrightarrow A = LU \tag{14}$$

- And as it turns out it is *much* easier to calculate the inverse of a triangular matrix than a dense matrix. We can rely on $A^{-1} = (LU)^{-1} = U^{-1}L^{-1}$, and then on the kind fact that we can find each of those inverses through back substitution of U and forward-substitution of L .

- In reality, it's worth noting that it's not *really* $A = LU$, but rather $PA = LU$. We must apply a permutation matrix to A in order to account for the fact that the pivots of A may not be perfectly ordered. We may at some point be forced to do a row swap, and hence apply a permutation at the end of the computation.

CR Factorization

- This is not one of our “critical factorizations”. However, it is useful for us. We learned a moment ago how to compute a basis for $C(A)$. We can use that to write $A = CR$, where $C \in \mathbb{R}^{m \times r}$ consists of *any* basis for C . The truth of this should be intuitive by how we have constructed the principle of a column space. We can take for example

$$A = \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 2 & 5 & -3 \\ 1 & 2 & 7 & -7 \end{bmatrix} \rightsquigarrow U = \begin{bmatrix} 1 & 2 & 0 & 7 \\ & & 1 & 2 \\ & & & \end{bmatrix}$$

where clearly \mathbf{c}_1 and \mathbf{c}_3 are the pivot rows such that $C(A) = \{\mathbf{c}_1, \mathbf{c}_3\}$. We can from this express a *specific case* of CR-Factorization as

$$A = \underbrace{\begin{bmatrix} 1 & 3 \\ 1 & 5 \\ 1 & 7 \end{bmatrix}}_C \underbrace{\begin{bmatrix} 1 & 2 & 0 & 7 \\ 0 & 0 & 1 & 2 \end{bmatrix}}_R$$

where, here, C is the *specific* column space basis corresponding to the pivot columns of the original matrix, and R is the reduced row echelon form of A , omitting the trivial last row. This does not generalize to all solutions for the factorization.

1.4 Orthogonality

- Let us recall that for $x, y \in \mathbb{R}^n$, i.e. two “n-component vectors”, we can define the dot product (a.k.a. the inner product) as $x^T y = [x_1 \ x_2 \ \dots] \begin{bmatrix} y_1 \\ y_2 \\ \vdots \end{bmatrix} = x_1 y_1 + x_2 y_2 + \dots$. From this we can define the L^2 **norm** or length of x as $\|x\| = \sqrt{x^T x} = \sqrt{x_1^2 + x_2^2 + \dots}$.
- Also we can recall the key property that a matrix transpose moves the matrix to the other side of a dot product. For example $u \cdot (Av) = u^T Av = (A^T u)^T v = (A^T u) \cdot v$
- In the two most common cases, orthogonal, and parallel, we will find nice properties. For $x \perp y$, $x^T y = 0$; for $x \parallel y$, $x^T y = \|x\| \|y\|$.
- In our original basis, we made no promises about what vectors defined the basis of the column space. This means that in order to solve $Ax = b$, we need to solve the whole system naively, which is computationally expensive. If we were to have a **nicer** basis, we could optimize some of that computation and inelegance. Imagine we have an **orthonormal basis** for a matrix. That is, the basis vectors of the matrix are orthogonal, and have magnitude 1. We will work on the details of how to compute one of these bases soon. For basis vectors $\mathbf{q}_1, \mathbf{q}_2$, we could define $\mathbf{b} = c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2$ and $\|\mathbf{b}\| = b^T b = (c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2)^T (c_1 \mathbf{q}_1 + c_2 \mathbf{q}_2) = c_1^2 + c_2^2$, or, the Pythagorean Theorem.

- In matrix form $Q = [\mathbf{q}_1 \quad \mathbf{q}_2]$, we have desirable properties such as $Q^T Q = I$, assuming that Q is square, meaning that the transpose of Q is its inverse! If Q is **not square**, we can say a lot less about it. It does not follow $Q^T = Q^{-1}$ and $Q^T Q \neq Q Q^T$.
- We can also define orthogonal subspaces S_1 and S_2 of V if every vector in S_1 is orthogonal to every vector in S_2 , or $x^T y = 0 \forall x \in S_1, y \in S_2$. Also, $\dim S_1 + \dim S_2 \leq \dim V$.
- Finally, we can define orthogonal complements of subspaces S^\perp for the subspace containing *all* vectors in V that are orthogonal to *every* vector in S . Therefore, S^\perp is the largest subspace that is orthogonal to S . It is formally defined as $S^\perp = \{w \in V : w^T x = 0 \text{ for any } x \in S\}$

Example: 1.5

Denote a subspace $V = \left\{ \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} : 2v_1 + 3v_2 + 5v_3 = 0 \right\}$. Find V^\perp (give a basis). Can you relate V and V^\perp to column and/or null spaces of some matrix.

Solution (1.5).

As we defined $V^\perp = \{w : w \cdot v = 0 \text{ for any } v \in V\}$. For $v = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$, $w = \begin{bmatrix} 2 & 3 & 5 \end{bmatrix}^T$ such that $v \cdot w = 0$ as given. Therefore, $V^\perp = \text{span} \left\{ \begin{bmatrix} 2 \\ 3 \\ 5 \end{bmatrix} \right\}$. Alternatively $A \equiv \begin{bmatrix} 2 & 3 & 5 \end{bmatrix}$ such that $Av = 0 \implies V = N(A)$ and $V^\perp = C(A^T)$.

Takeaways (1.5).

- Remember the formal definition of the orthogonal subspace: it is just the subspace for which the dot product with *anything* is 0.
- This orthogonal subspace is highly related to the nullspace. Always be looking for the nullspace and column space connections.

Example: 1.6

Suppose we have a subspace \mathcal{S} with orthogonal (but not necessarily orthonormal) basis $\{v_1, \dots, v_k\}$

$$v = \sum_{i=1}^k \alpha_i v_i$$

by the definition of a basis, for some constants $\alpha_1, \dots, \alpha_k$. Determine every α_j in terms of v_i, \dots, v_k . Will this work if the basis is not orthogonal? If we put the vectors v_i as columns of a matrix $V = \begin{bmatrix} v_1 & \dots & v_k \end{bmatrix}$, what special form does $V^T V$ have?

Solution (1.6).

In order to find α_j specifically, left multiply by v_j^T such that

$$v_j^T v = \sum_{i=1}^k v_j^T (\alpha_i v_i) = \sum_{i=1}^k \alpha_i v_j^T v_i = \alpha_j \|v_j\|^2$$

The genius of multiplying by v_j^T is that for all terms $i \neq j$, $v_j^T v_i = 0$ such that we are able to get rid of that nasty summation. This leads easily to $\alpha_j = \frac{v_j^T v}{\|v_j\|^2}$. Then if we take the matrix V , we will end with just

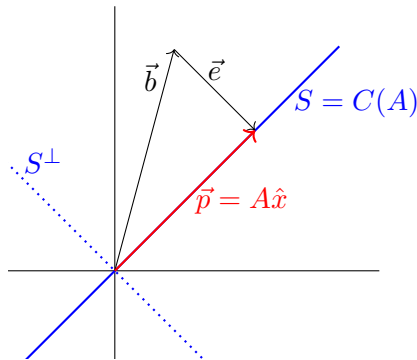
$$V^T V = \begin{bmatrix} \|v_1\|^2 & & & \\ & \|v_2\|^2 & & \\ & & \ddots & \\ & & & \|v_k\|^2 \end{bmatrix}$$

since V forms a linear multiple of the basis vectors.

Takeaways (1.6).

- The trick of left-multiplying by v_j^T is tricky as hell here yet. A good problem solving technique is to look for what you're trying to isolate and then the conditions that can help you simplify the opaque linear algebra expressions.
- For the basis vectors, it's important to realize this property of $v^T V$ that we will revisit on the section on symmetry later.

Projections



If we imagine some n -dimensional subspace $S \subseteq \mathbb{R}^n$, we can define a vector \mathbf{b} that exists outside the subspace. We can also define a vector \mathbf{p} , the orthogonal projection of \mathbf{b} onto S . Here we could say

$$\mathbf{b} = \underbrace{\mathbf{p}}_{\in S} + \underbrace{\mathbf{e}}_{\in S^\perp} = P\mathbf{b} + (I - P)\mathbf{b}$$

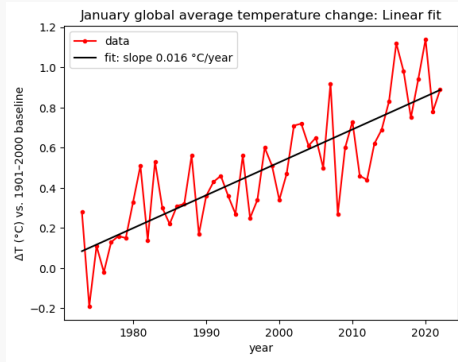
For our “projection matrix” P . Now, let's imagine we define $S = C(A)$ such that $S^\perp = N(A^T)$. Then, $\mathbf{p} = A\hat{x} \in C(A)$, and \hat{x} is the closest solution to \mathbf{b} . Therefore, $\mathbf{e} = \mathbf{b} - A\hat{x} \in N(A^T) \rightarrow A^T(\mathbf{b} - A\hat{x}) \rightarrow A^T\mathbf{b} = A^T A\hat{x}$. From this, we can define three **normal equations**.

$$\begin{aligned} \hat{x} &= (A^T A)^{-1} A^T \mathbf{b} \\ \mathbf{p} &= A\hat{x} = P\mathbf{b} \\ P &= A(A^T A)^{-1} A^T \end{aligned}$$

- We can also define a handful of neat properties about the projection matrix P .

1. $P^2 = P$ Proof. $A(A^T A)^{-1}(A^T A)(A^T A)^{-1}A^T = P$
2. $P^T = P$ Proof. $((A^T A)^{-1}A^T b)^T = (A^T)^T((A^T A)^{-1})^T A^T = P$
3. $C(P) = C(A)$
4. $N(P) = C(A)^\perp = N(A^T)$
5. $I - P$ is the projection matrix onto $N(A^T)$

- Our projection vector \mathbf{p} is going to minimize the norm $\|b - Ax\|$ as should be clear in the geometry above. Notably, it won't necessarily come up with an *exact solution*. Rather it is able to find the closest solution when there is no absolute solution. This is the beginning of our forays into optimization. As an example, let's examine the simplest optimization problem: least squares.



Given a time series of temperature data, we want to calculate the linear regression with our matrix methods. We are going to construct another Vandermonde matrix: this time with a highly overdetermined system.

$$\begin{pmatrix} 1 & y_1 - y_0 \\ 1 & y_2 - y_0 \\ \vdots & \vdots \\ 1 & y_m - y_0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \Delta T_1 \\ \Delta T_2 \\ \vdots \\ \Delta T_m \end{pmatrix}$$

To find our solution, we must simply minimize the orthogonal projection with the normal equations

$$\operatorname{argmin}_{x \in \mathbb{R}^2} \|b - Ax\| = (A^T A)^{-1} A^T b$$