EE951: Linear Algebra **eMasters**

Chapter 2: Vector Operations

Author: Ketan Rajawat

Contents

2.1 Vector Addition

Vector addition allows us to combine two vectors to obtain a new vector that represents their combined effect or displacement. Let's explore this concept using an example of adding two 3x1 vectors. Suppose we have two vectors:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$

To add these vectors, we simply add their corresponding components together. The resulting vector, denoted as $\mathbf{w}$, will have the same size and dimensions as the original vectors:

$$\mathbf{w} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \\ u_3 + v_3 \end{bmatrix}$$

For example, let's consider:

$$\mathbf{u} = \begin{bmatrix} 2 \\ 3 \\ -1 \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} 1 \\ -2 \\ 4 \end{bmatrix}$$

To find the sum $\mathbf{w} = \mathbf{u} + \mathbf{v}$, we add the corresponding components:

$$\mathbf{w} = \begin{bmatrix} 2+1 \\ 3+(-2) \\ (-1)+4 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}$$

Thus, the sum of vectors $\mathbf{u}$ and $\mathbf{v}$ is $\mathbf{w} = \begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}$.

Vector subtraction follows a similar principle. When subtracting one vector from another, we subtract the corresponding components to obtain the resulting vector. Using the same vectors $\mathbf{u}$ and $\mathbf{v}$, the difference $\mathbf{x} = \mathbf{u} - \mathbf{v}$ would be:

$$\mathbf{x} = \begin{bmatrix} 2-1 \\ 3-(-2) \\ (-1)-4 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ -5 \end{bmatrix}$$

Thus, the difference of vectors $\mathbf{u}$ and $\mathbf{v}$ is $\mathbf{x} = \begin{bmatrix} 1 \\ 5 \\ -5 \end{bmatrix}$.

2.1.1 Properties

Let us state some basic properties of vectors that you should already be familiar with.

1. Commutative Property: The commutative property of vector addition states that the order in which vectors are added does not affect the result. In other words, the sum of two vectors remains the same regardless of their order. For any vectors $\mathbf{u}$ and $\mathbf{v}$:
$$\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$$
2. Associative Property: The associative property of vector addition states that the grouping of vectors being added does not affect the result. In other words, the sum of three or more vectors remains the same regardless of how they are grouped. For any vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$:
$$(\mathbf{u} + \mathbf{v}) + \mathbf{w} = \mathbf{u} + (\mathbf{v} + \mathbf{w})$$
3. Additive Identity Property: The additive identity property states that there exists a special vector, called the additive identity or zero vector, denoted by $\mathbf{0}$, such that adding it to any vector leaves the vector unchanged. For any vector $\mathbf{u}$:
$$\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$$
4. Additive Inverse Property: The additive inverse property states that for every vector $\mathbf{u}$, there exists a vector called the additive inverse or negative of $\mathbf{u}$, denoted by $-\mathbf{u}$, such that adding $\mathbf{u}$ and its additive inverse results in the zero vector. For any vector $\mathbf{u}$:
$$\mathbf{u} + (-\mathbf{u}) = (-\mathbf{u}) + \mathbf{u} = \mathbf{0}$$

Remark 1. In some programming languages (notably MATLAB, python, and R), the addition of a scalar value to a vector is allowed and produces a new vector, where the scalar is added to each component of the vector. However, this notation should be used with caution, as it deviates from the standard mathematical convention for vector operations.

2.1.2 Displacements and Positions

When considering vectors $\mathbf{a}$ and $\mathbf{b}$ as displacements, the sum $\mathbf{a}+\mathbf{b}$ represents the net displacement obtained by sequentially displacing by $\mathbf{a}$ and then by $\mathbf{b}$. This can be visualized in Fig. 2.1. The result is the same if we first displace by $\mathbf{b}$ and then by $\mathbf{a}$.

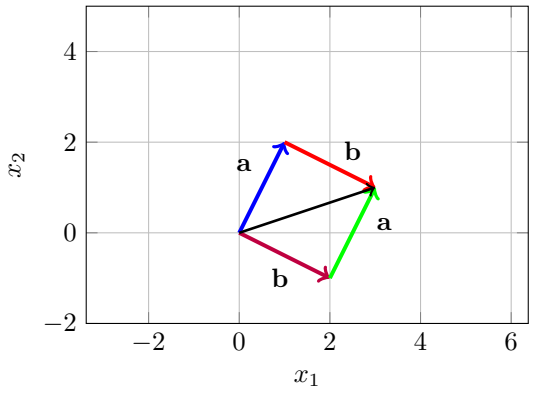

Figure 2.1: Illustration of vector addition: Displacing by **a** and then by **b** (blue and red arrows) yields the same vector as displacing by **b** and then by **a** (purple and green arrows).

If **p** is a position and **a** is a displacement, then $\mathbf{p} + \mathbf{a}$ is the position of a point that was originally at **p** but has now been displaced by **a**. Finally, if **p** and **q** are positions, then $\mathbf{p} - \mathbf{q}$ is the displacement vector from **q** to **p**, as illustrated in Fig. 2.2.

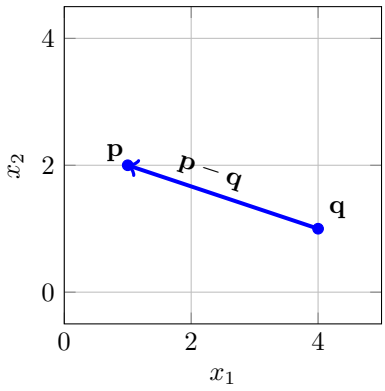

Figure 2.2: Illustration of vector subtraction: The displacement vector from **q** to **p**, denoted by $\mathbf{p} - \mathbf{q}$, is shown as a blue arrow. The positions **p** and **q** are marked with blue dots.

2.2 Scaling

Scalar-vector multiplication is an operation that allows us to scale a vector by a scalar quantity. In this operation, which is also called *scaling*, every element of the vector is multiplied by the scalar value. Mathematically, if we have a vector **v** and a scalar α, the scalar-vector multiplication is denoted as $\alpha\mathbf{v}$. Consider the example

$$\mathbf{v} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \quad \text{and} \quad \alpha = 3$$

To perform scalar-vector multiplication, we multiply each element of **v** by α:

$$\alpha\mathbf{v} = \alpha \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = 3 \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} = \begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix}$$

So, the result of scalar-vector multiplication in this case is the column vector $\begin{bmatrix} 6 \\ 12 \\ 18 \end{bmatrix}$, where each element of the original vector $\mathbf{v}$ has been multiplied by the scalar $\alpha = 3$. Other common notations are $\mathbf{v}/3 = \frac{1}{3}\mathbf{v}$ and $-\mathbf{v} = (-1)\mathbf{v}$. We list some basic properties of scalar-vector multiplication.

1. *Commutative property:* A scalar can either be multiplied from the right or the left, with the same result, i.e., $\alpha\mathbf{v} = \mathbf{v}\alpha$.
2. *Zero element:* When a vector is multiplied by the scalar zero, the result is the zero vector. For any vector $\mathbf{u}$, it holds that $0\mathbf{u} = \mathbf{0}$.
3. *Associative property:* Consider two scalars α, β, and a vector $\mathbf{v}$. The scalar product of $\alpha\beta$ with $\mathbf{v}$ is equal to the scalar product of α with the scalar product of β with $\mathbf{v}$, i.e., $(\alpha\beta)\mathbf{v} = \alpha(\beta\mathbf{v}) = (\alpha\mathbf{v})\beta$.
4. *Distributive property:* Scalar multiplication satisfies two versions of distributive properties, both from left and right each:

$$(\alpha + \beta)\mathbf{v} = \alpha\mathbf{v} + \beta\mathbf{v} \tag{2.1}$$
$$\alpha(\mathbf{v} + \mathbf{w}) = \alpha\mathbf{v} + \alpha\mathbf{w} \tag{2.2}$$

2.2.1 Scaling Displacements

For a vector $\mathbf{v}$ representing displacement, its scaled version $\alpha\mathbf{v}$ for $\alpha > 0$ represents a displacement which is in the same direction as $\mathbf{v}$ but with length scaled by α. On the other hand, its scaled version $\beta\mathbf{v}$ for $\beta < 0$ represents a displacement which is in the opposite direction as $\mathbf{v}$ but with length scaled by $|\beta|$. Fig. shows two such examples.

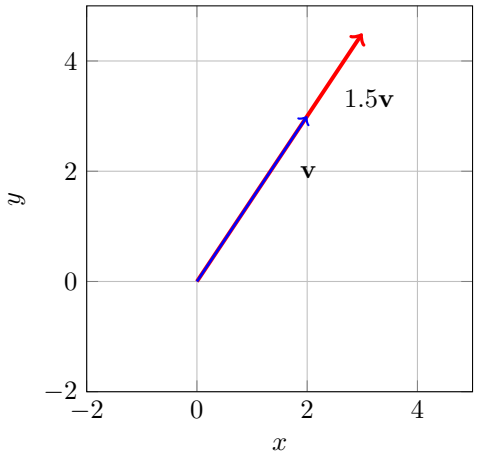

Figure 2.3: Scaling a vector $\mathbf{v}$ by a factor of 1.5.

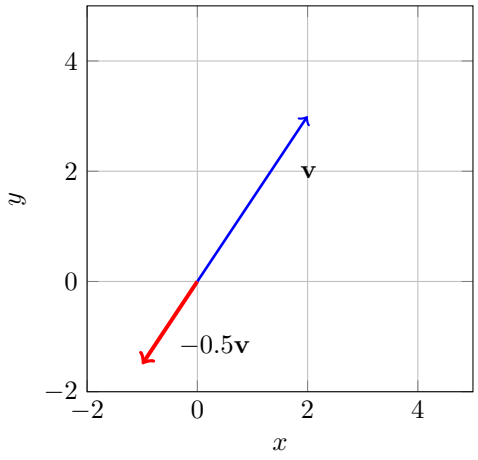

Figure 2.4: Scaling a vector $\mathbf{v}$ by a factor of -0.5.

2.3 Linear Combinations

A linear combination refers to the sum of scalar multiples of vectors. Given a set of vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ and corresponding scalars $\alpha_1, \alpha_2, \ldots, \alpha_m$, the linear combination of these vectors is given by:

$$\alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \ldots + \alpha_m\mathbf{v}_m$$

Here, $\alpha_1, \alpha_2, \ldots, \alpha_m$ are scalars, and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ are vectors. The scalars are multiplied with their corresponding vectors and then summed together. In this context, we often refer to the vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ as the "vector terms" or "component vectors" of the linear combination. The scalars $\alpha_1, \alpha_2, \ldots, \alpha_m$ are called the "coefficients" or "weights" associated with each vector, since they determine the relative importance or contribution of each vector to the linear combination.

Observation 1. Any given vector can be expressed as a linear combination of standard unit vectors. Given a vector $\mathbf{v}$ with entries v_1, ..., v_n, we have that

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 + \ldots + v_n\mathbf{e}_n. \tag{2.3}$$

For example, let's consider a vector $\mathbf{v}$ in three-dimensional space given by:

$$\mathbf{v} = 3\mathbf{e}_1 + (-2)\mathbf{e}_2 + 4\mathbf{e}_3 = \begin{bmatrix} 3 \\ -2 \\ 4 \end{bmatrix}$$

Here, the coefficients 3, -2, and 4 represent the weights or contributions of the standard unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, respectively.

2.4 Complexity of Vector Operations

In computers, real numbers are stored using a format called floating point. This format uses a block of 64 bits, which are essentially 0s and 1s, or 8 bytes (each byte consists of 8 bits grouped together). Real vectors are stored as arrays of floating point numbers. Generally, in order to operate on vectors, we need to load them on the random access memory (RAM) which is typically at least 4 GB or 32×10^9 bits. Since each vector takes 64 bits for storage, it is possible to load vectors of sizes at most 5×10^8 in 4GB RAM. Some operations, such as inner product, may require two vectors to be loaded into the RAM, which would further decrease the maximum allowable size of vectors that can be used in a given computer. We refer to the storage requirement of an algorithm (such as scaling or inner product calculation) as it storage complexity.

More importantly, we must also look at the computational complexity of various linear algebra algorithms. The concept of flop count provides a way to estimate the computational complexity of various vector and matrix operations. Flop count stands for "floating-point operations count" and refers to the number of arithmetic operations (additions, subtractions, multiplications, and divisions) performed on floating-point numbers during the execution of an algorithm or operation. By counting the number of floating-point operations required, we can estimate the computational cost or complexity of an algorithm. This measure is particularly useful in assessing the efficiency of algorithms for solving linear algebra problems since many of these problems involve large-scale computations with vectors and matrices.

As an example, we see that scaling an n-vector by a real α entails multiplying each entry with α, which amounts to n flops. Addition of two n-vectors requires one addition operation per-element, resulting in a total flop count of n.

It's important to note that these flop counts are based on the basic arithmetic operations involved in each operation. Actual implementation details, such as hardware optimization or specific algorithms, may influence the precise number of floating-point operations performed. Additionally, certain libraries or programming languages may provide optimized routines for these operations, potentially reducing the actual computational cost. It is customary to drop the constants, and report the flop counts as multiples of n, as in the table below.

Table 2.1: Flop counts for operations on n-vectors

Operation	**Flop count**
Scaling	n
Vector Addition	n