Basic Introduction to Vectors and Matrices

Mark A. Austin

University of Maryland

austin@umd.edu
ENCE 201, Fall Semester 2024

October 14, 2024

Overview

Definition of Vectors

Definition of Row and Column Vectors

Definition. A row vector is simply a $(1 \times n)$ matrix, i.e.,

$$V = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & \cdots & v_n \end{bmatrix} \quad (1)$$

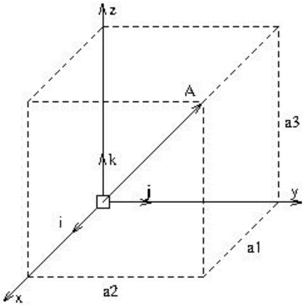
Definition. A column vector is a $(m \times 1)$ matrix, e.g.,

$$V = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \cdots \\ v_m \end{bmatrix} \quad (2)$$

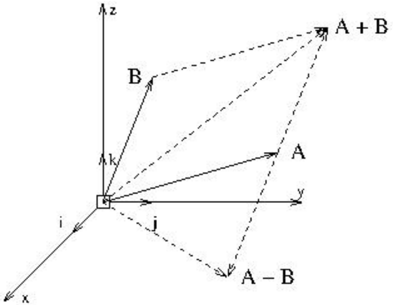
In both cases, the i-th element of the column vector is denoted v_i.

Vector Properties

Properties of Vector Arithmetic



Components of Three–Dimensional Vector



Vector Addition and Subtraction

- a + b = b + a
- a + 0 = a
- c(a + b) = ca + cb
- (a + b) + c = a + (b + c)
- a + (-a) = 0
- 1 a = a.

Dot Product

Definition. The dot product of two vectors $a = [a_1, a_2, a_3, \cdots, a_n]$ and $b = [b_1, b_2, b_3, \cdots, b_n]$ is:

$$a.b = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + a_3 b_4 + \cdots + a_n b_n. \tag{3}$$

Note: a.b = b.a. If a and b are perpendicular then a.b = 0.

Engineering Applications

- Mechanical work is the dot product of force and displacement vectors (Jou).
- Power is the dot product of force and velocity vectors (W).
- Fluid Mechanics.

Dot Product

Example 1. Let $a = [1, 2, 3]$ and $b = [0, -1, 2]$. The dot product:

$$a.b = \sum_{i=1}^{n} a_i b_i = 1 \times 0 + 2 \times -1 + 3 \times 2 = 4. \quad (4)$$

A dot product can also be written as a row vector multiplied by a column vector, e.g.,

$$\begin{bmatrix} 1, 2, 3 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -1 \\ 2 \end{bmatrix} = 4. \quad (5)$$

The vector dimensions are: $(1 \times 3)\ (3 \times 1) \rightarrow (1 \times 1)$.

Dot Product

Properties. Let $a = [a_1, a_2, a_3, a_4]$, $b = [b_1, b_2, b_3, b_4]$ and $c = [c_1, c_2, c_3, c_4]$. And let d be a non-zero constant.

The dot product:

$$a.b = a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4 \tag{6}$$

obeys the properties:

- a.a $= \|a\|^2$.
- a.(b + c) = a.b + a.c
- a.b = b.a
- $a.b = 0 \iff a = 0$ or $b = 0$ or $a \perp b$.
- 0.a = 0
- (da).b = d(a.b)
- a.b $= |a|.|b| \cos(\theta)$.

Dot Product

```
Fragment of Python Code:                Output:
------------------------------------    ------------------------

import numpy as np

# Define vectors a and b

a = np.array([1, 2, 3])                 [1 2 3]
b = np.array([4, 5, 6])                 [4 5 6]

# Compute dot product c = a.b ...

c = np.dot(a,b)                         32.000000

------------------------------------    ------------------------
```

Cross Product

Definition. Consider two vectors A and B in three dimensions:

$$A = a_1 \hat{i} + a_2 \hat{j} + a_3 \hat{k}$$
$$B = b_1 \hat{i} + b_2 \hat{j} + b_3 \hat{k}$$

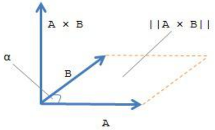
The cross product of A and B is:

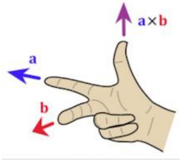
$$C = A \times B = \det \begin{pmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{pmatrix}$$
$$= (a_2 b_3 - a_3 b_2)\,\hat{i} + (a_3 b_1 - a_1 b_3)\,\hat{j} + (a_1 b_2 - a_2 b_1)\,\hat{k}.$$

Cross Product

Geometric Interpretation

$A \times B$ is a vector that is perpendicular to both A and B.





- The magnitude of $\|A \times B\|$ is equal to the area of the parallelogram formed using A and B as the sides.
- The angle between A and B is: $\|A \times B\| = \|A\| \, \|B\| \, sin(\alpha)$.
- The cross product is zero when the A and B are parallel.

Cross Product

```
Fragment of Python Code:                Output:
------------------------------------    ------------------------

# Define vectors a and b

a = np.array([1, 2, 3])                 [1 2 3]
b = np.array([4, 5, 6])                 [4 5 6]

# Compute cross product c = a x b ...

c = np.cross(a,b)                       [-3  6 -3]

# Check that cross product is perpendicular to a and b ...

d1 = np.dot(c,a)                        0.000000
d2 = np.dot(c,b)                        0.000000

------------------------------------    ------------------------
```

Linear Independence of Vectors

Linear Independence

A set of vectors $(v_1, v_2, v_3, \cdots, v_n)$ is said to be linearly independent if the equation

$$a_1 v_1 + a_2 v_2 + a_3 v_3 + \cdots + a_n v_n = 0. \tag{7}$$

can only be satisfied by $a_i = 0$ for i $=$ 1, ... n.

Put another way: no vector in the sequence can be written as a linear combination of the other vectors.

Linear Independence of Vectors

Example 1. Consider three vectors $v_1 = (1, 1)$, $v_2 = (\text{-}3, 2)$, and $v_3 = (2, 4)$ in two-dimensional space.

The vectors will be linearly independent if the only solutions to

$$a_1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + a_2 \begin{bmatrix} -3 \\ 2 \end{bmatrix} + a_3 \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{8}$$

are $a_1 = a_2 = a_3 = 0$. Writing these equations in matrix form:

$$\begin{bmatrix} 1 & -3 & 2 \\ 1 & 2 & 4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{9}$$

Linear Independence of Vectors

Apply row operations (details to follow):

$$\begin{bmatrix} 1 & 0 & 16/5 \\ 0 & 1 & 2/5 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{10}$$

which can be rearranged:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} + a_3 \begin{bmatrix} 16/5 \\ 2/5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{11}$$

We conclude that since a_1 and a_2 can be written in terms of a_3, the equations are linearly dependent.

Linear Independence of Vectors

A Few Observations

- Vectors v_1 through v_3 are two dimensional.
- Can show that three (or more) vectors in two-dimensional space will always be linearly dependent.
- Can show that four (or more) vectors in three-dimensional space will always be linearly dependent.
- This is why a stool with three legs (vectors) will always be steady (linearly independent), but one with four legs (vectors) will sometimes rock (linearly dependent).

Definition of Matrices

Definition of a Matrix

Definition. A matrix (or array) of order `m` by `n` is simply a set of numbers arranged in a rectangular block of `m` horizontal rows and `n` vertical columns. We say

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad (12)$$

is a matrix of size (or dimension) $(\mathtt{m} \times \mathtt{n})$.

In the double subscript notation a_{ij} for matrix element `a(i,j)`, the first subscript i denotes the row number, and the second subscript j denotes the column number.

Matrix Properties

Matrix Properties

Properties of Matrix A:

- A matrix having the same number of rows and columns is called square.
- A square matrix of order n is also called a $(n \times n)$ matrix.
- The elements $a_{11}, a_{22}, \cdots, a_{nn}$ are called the principal diagonal.
- A diagonal matrix with elements $a_{ii} = 1$, and all other matrix elements zero, is called the identity matrix I.

Matrix Transpose

Matrix Transpose. The transpose of a $(m \times n)$ matrix A is the $(n \times m)$ matrix obtained by interchanging the rows and columns of A. The tranpose is denoted A^T.

Example 1. The matrix transpose of

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \quad \text{is} \quad A^T = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix} \tag{13}$$

Properties

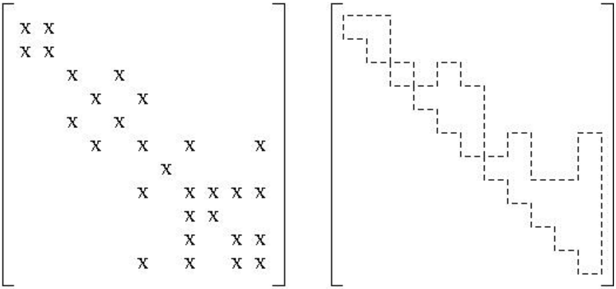
- $(A + B)^T = A^T + B^T$.
- $(ABC)^T = C^T B^T A^T$.

Symmetric and Skew-Symmetric Matrices

Matrix Symmetry:

- A square matrix A is symmetric if A $= A^T$.
- A square matrix A is skew-symmetric if A $=$ -A^T.

Large symmetric matrices play a central role in structural analysis.



Schematic of Non–Zero Matrix Elements   Skyline Storage Pattern

Matrix Inverse

Definition: When it exists, the inverse of matrix A is written A^{-1} and it has the property:

$$[A]\left[A^{-1}\right]=\left[A^{-1}\right][A]=I. \tag{14}$$

Nomenclature

- If matrix A has an inverse, then A is called non-singular.
- If matrix A has an inverse, then the inverse will be unique.
- If matrix A does not have an inverse, then A is called singular.

Theorem. For a $(n \times n)$ matrix A, the inverse A^{-1} exists $\iff$ rank(A) $=$ n.

- Conversely, matrix A is singular if rank(A) $<$ n (i.e., rank deficient).

Matrix Inverse

Computational Procedure. We want to carry out row operations such that:

$$[A|I] \xrightarrow{\text{row operations}} \left[I|A^{-1}\right]. \quad (15)$$

Example. Can apply row operations to get:

$$\left[\begin{array}{rrr|rrr} -1 & 1 & 2 & 1 & 0 & 0 \\ 3 & -1 & 1 & 0 & 1 & 0 \\ -1 & 3 & 4 & 0 & 0 & 1 \end{array}\right] \xrightarrow{\text{row ops}} \left[\begin{array}{rrr|rrr} 1 & 0 & 0 & -0.7 & 0.2 & 0.3 \\ 0 & 1 & 0 & -1.3 & -0.2 & 0.7 \\ 0 & 0 & 1 & 0.8 & 0.2 & -0.2 \end{array}\right]. \quad (16)$$

If A has rank(A) $<$ n, then the last row in echelon form will be the O (zero) vector, and the computation will fail.

Matrix Inverse

Properties:

$$\left[A^{-1}\right]^{-1} = A. \tag{17}$$

$$(AB)^{-1} = B^{-1}A^{-1}. \tag{18}$$

$$(ABC)^{-1} = C^{-1}B^{-1}A^{-1}. \tag{19}$$

$$\left[A^{T}\right]^{-1} = \left[A^{-1}\right]^{T}. \tag{20}$$

Lower and Upper Triangular Matrices

A lower triangular matrix `L` is one where $a_{ij} = 0$ for all entries above the diagonal.

An upper triangular matrix `U` is one where $a_{ij} = 0$ for all entries below the diagonal. That is,

$$L = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad U = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{mn} \end{bmatrix} \tag{21}$$

Matrix Arithmetic

Matrix Addition and Subtraction

Definition. If A is a $(m \times n)$ matrix and B is a $(r \times p)$ matrix, then the matrix sum $C = A + B$ is defined only when $m = r$ and $n = p$, and is a $(m \times n)$ matrix C whose elements are

$$c_{ij} = a_{ij} + b_{ij}, \text{ for } i = 1, 2, \cdots m \text{ and } j = 1, 2, \cdots n. \tag{22}$$

Properties

- (kA) B = k $(A.B)$
- $A(BC) = (AB)C.$
- $(A + B)C) = AB + AC.$
- $C(A + B) = CA + CB.$

Matrix Addition and Subtraction

Example 1. Let

$$A = \begin{bmatrix} 2 & 1 \\ 4 & 6 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix}. \tag{23}$$

The matrix sum is:

$$C = A + B = \begin{bmatrix} 2 & 1 \\ 4 & 6 \end{bmatrix} + \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 3 \\ 4 & 7 \end{bmatrix}. \tag{24}$$

Matrix Multiplication

Definition. Let A and B be $(m \times n)$ and $(r \times p)$ matrices, respectively.

The matrix product $A \cdot B$ is defined only when interior matrix dimensions are the same (i.e., $n \;=\; r$).

The matrix product $\mathrm{C} = \mathrm{A} \cdot \mathrm{B}$ is a $(m \times p)$ matrix whose elements are

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \; b_{kj} \tag{25}$$

for $i = 1, 2, \cdots m$ and $j = 1, 2, \cdots n$.

Matrix Multiplication

Example 1. Assuming that matrices A and B are as defined in the previous section:

$$C = A \cdot B = \begin{bmatrix} 2 & 1 \\ 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 4 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 \cdot 4 + 1 \cdot 0 & 2 \cdot 2 + 1 \cdot 1 \\ 4 \cdot 4 + 6 \cdot 0 & 4 \cdot 2 + 6 \cdot 1 \end{bmatrix} = \begin{bmatrix} 8 & 5 \\ 16 & 14 \end{bmatrix}. \quad (26)$$

Geometric Interpretation. Matrix element c_{ij} is the dot product of the i-th row of A with the j-th column of B.

Matrix Multiplication

Properties.

- A.B.C = (A.B).C = A.(B.C).
- A.(B + C) = A.B + A.C.
- (A + B).C = A.C + B.C.
- A.I = A.
- In general, A.B $\neq$ B.A.
- A.B = ϕ does not necessarily imply A = ϕ or B = ϕ. Counter example:

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix} \text{ and } B = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{27}$$

Matrix Operations with Python

```
Fragment of Python Code:                Output:
-------------------------------------   -------------------------

A = np.array([ [2, 1], [4, 6] ]);       Matrix: A
                                            2.00      1.00
                                            4.00      6.00

B = np.array([ [4, 2], [0, 1] ]);       Matrix: B
                                            4.00      2.00
                                            0.00      1.00
# Compute matrix transpose ...

C = A.T                                 Matrix: A^T
                                            2.00      4.00
                                            1.00      6.00

-------------------------------------   -------------------------
```

Matrix Operations with Python

```
Fragment of Python Code:                Output:
------------------------------------    ------------------------

# Use numpy add() for matrix addition ...

C = np.add(A,B)                         Matrix: np.add(A,B)
                                            6.00      3.00
                                            4.00      7.00

# Add matrices with + operator ...

C = A + B                               Matrix: A + B
                                            6.00      3.00
                                            4.00      7.00

------------------------------------    ------------------------
```

Matrix Operations with Python

```
Fragment of Python Code:        Output:
----------------------------    ------------------------

# Use numpy matmul() for matrix multiplication ...

C = np.matmul( A, B )           Matrix: np.matmul(A,B)
                                    8.00      5.00
                                   16.00     14.00

# Use * operator for matrix element-level multiplies ...

C = A * B                       Matrix: Element-level multiply A*B
                                    8.00      2.00
                                    0.00      6.00

----------------------------    ------------------------
```