

Vectors

Introduction to Linear Algebra

What is Linear Algebra?

What is a Matrix?

A matrix is a rectangular array of numbers, symbols, or expressions, arranged in rows and columns. Each element in a matrix is identified by its row and column index. Matrices are fundamental in various fields of mathematics and engineering, particularly in solving systems of linear equations, performing linear transformations, and representing data.

General form of a matrix

A general $n \times m$ matrix (n rows and m columns) looks like this:

$$\mathbf{A} = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Where a_{ij} represents the element in the i -th row and j -th column.

2x2 Matrix

A 2x2 matrix (2 rows and 2 columns) looks like this:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

Where a_{ij} represents the element in the i -th row and j -th column.

3x5 Matrix

A 3x5 matrix (3 rows and 5 columns) looks like this:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix}$$

Where b_{ij} represents the element in the i -th row and j -th column.

Matrix Operations

For matrices of the same size, we can perform the following operations addition, subtraction, and scalar multiplication. We can also multiply two matrices together, but the dimensions of the matrices must be compatible. Division of matrices is not defined!

Addition

The sum of two matrices of the same size is obtained by adding the corresponding elements of the matrices.

Let's consider two matrices **A** and **B** of the size 2×2 :

The sum of matrices **A** and **B** is given by:

$$\mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$$

Multiplication by a Scalar

Let's consider a matrix **A** of the size 2×2 and a scalar c :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The product of the scalar c and the matrix **A** is given by:

$$c \cdot \mathbf{A} = c \cdot \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} c \cdot a_{11} & c \cdot a_{12} \\ c \cdot a_{21} & c \cdot a_{22} \end{bmatrix}$$

Matrix Multiplication

General Form

Let's consider two matrices \mathbf{A} of size $n \times m$ and \mathbf{B} of size $m \times p$. The product of matrices \mathbf{A} and \mathbf{B} is a matrix \mathbf{C} of size $n \times p$.

Each element c_{ij} of the resulting matrix \mathbf{C} is calculated as:

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$$

for $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, p$.

In other words, the element in the i -th row and j -th column of \mathbf{C} is the sum of the products of the elements from the i -th row of \mathbf{A} and the j -th column of \mathbf{B} .

2x2 Matrices

Let's consider two matrices \mathbf{A} and \mathbf{B} of the size 2×2 :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

The product of matrices \mathbf{A} and \mathbf{B} is given by:

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Non-Commutativity of Matrix Multiplication

Consider a 1×2 row matrix \mathbf{A} and a 2×1 column matrix \mathbf{B} :

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}$$

The product $\mathbf{A} \cdot \mathbf{B}$ is:

$$\mathbf{A} \cdot \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \end{bmatrix} \cdot \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} = a_{11}b_{11} + a_{12}b_{21}$$

Now, let's consider the product $\mathbf{B} \cdot \mathbf{A}$:

$$\mathbf{B} \cdot \mathbf{A} = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} \end{bmatrix} = \begin{bmatrix} b_{11}a_{11} & b_{11}a_{12} \\ b_{21}a_{11} & b_{21}a_{12} \end{bmatrix}$$

As we can see, $\mathbf{A} \cdot \mathbf{B}$ is a scalar, while $\mathbf{B} \cdot \mathbf{A}$ is a (2×2) matrix. Therefore, $\mathbf{A} \cdot \mathbf{B} \neq \mathbf{B} \cdot \mathbf{A}$, demonstrating that matrix multiplication is not commutative.

Determinants

2x2 Matrices

3x3 Matrices

Laplace Expansion - Determinants of Larger Matrices

Inverse of a Matrix

Minors and Cofactors

Gauss-Jordan Elimination

Systems of Linear Equations

Gaussian Elimination

Matrix Form of a System of Equations

Cramer's Formula

Cartesian spaces

Coordinate Systems

```

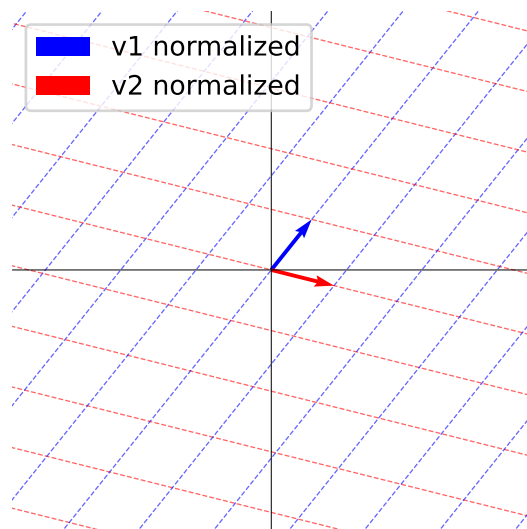
v1 = np.array([4, 5]) # some arbitrary vector
v2 = np.array([4, -1]) # another arbitrary vector

plot_coordinate_grid(v1, v2, grid_size=4)

```

Normalized v1: [0.62469505 0.78086881]
 Normalized v2: [0.9701425 -0.24253563]

Coordinate Grid Based on Normalized Unit Vectors



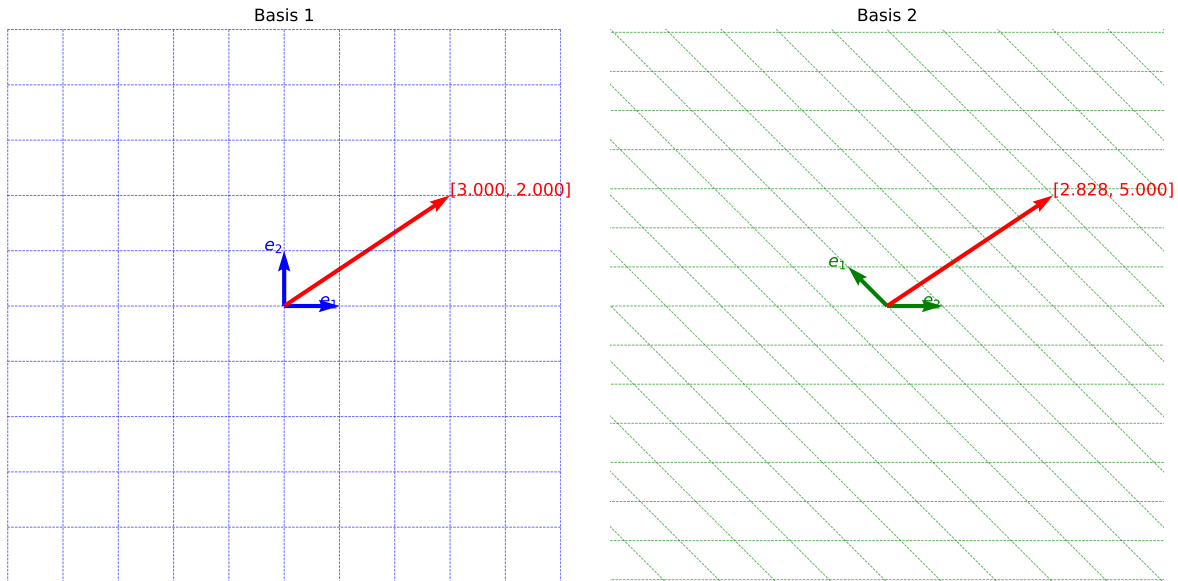
```

# Example: Using non-standard vectors and a vector in space
v1_basis1 = np.array([1,0]) # first versor of basis 1
v2_basis1 = np.array([0,1]) # second versor of basis 1
v1_basis2 = np.array([-1, 1]) # first versor of basis 2
v2_basis2 = np.array([1, 0]) # second versor of basis 2

vector = np.array([3, 2]) # Vector in cartesian coordinates

plot_side_by_side_systems(v1_basis1, v2_basis1, v1_basis2, v2_basis2, vector)

```



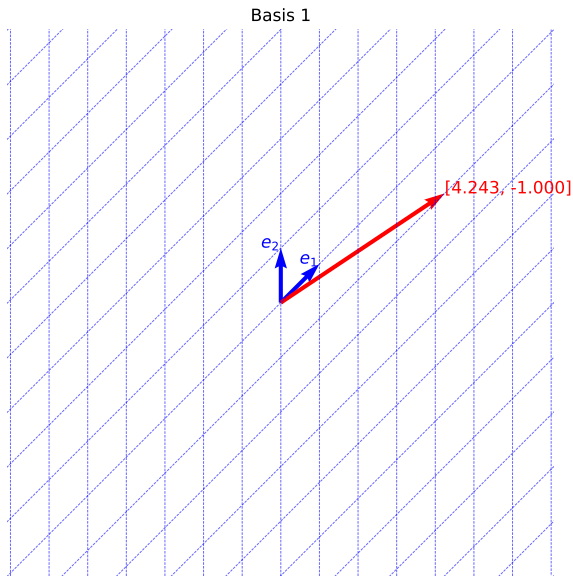
Vector components in basis 1: [3. 2.]

Vector components in basis 2: [2.82842712 5.]

```
# Example: Using non-standard vectors and a vector in space
v1_basis1 = np.array([1,1]) # first versor of basis 1
v2_basis1 = np.array([0,1]) # second versor of basis 1
v1_basis2 = np.array([-1, -1]) # first versor of basis 2
v2_basis2 = np.array([1, 0]) # second versor of basis 2

vector = np.array([3, 2]) # Vector in the cartesian coordinates

plot_side_by_side_systems(v1_basis1, v2_basis1, v1_basis2, v2_basis2, vector)
```



Vector components in basis 1: [4.24264069 -1.]
 Vector components in basis 2: [-2.82842712 1.]

Basis Vectors