DOING PHYSICS WITH PYTHON

VECTOR and MATRICES

Ian Cooper

Please email any corrections, comments, suggestions or additions:

matlabvisualphysics@gmail.com

DOWNLOAD DIRECTORIES FOR PYTHON CODE

op100.py

Google drive

GitHub

https://rezaborhani.github.io/mlr/blog_posts/Linear_Algebra/Vectors_and_matrices.html

SPECIFYING [3D] VECTORS

A **scalar** is completely characterised by its magnitude, and has no associated direction (mass, time, direction, work). A scalar is given by a simple number.

A **vector** has both a magnitude and direction (force, electric field, magnetic field). A vector can be specified in terms of its Cartesian or cylindrical (polar in [2D]) or spherical coordinates.

Cartesian coordinate system (XYZ right-handed rectangular: if we curl our fingers on the right hand so they rotate from the X axis to the Y axis then the Z axis is in the direction of the thumb).

A vector $\vec{V}$ in specified in terms of its X, Y and Z Cartesian components

$$\vec{V}\left(V_x, V_y, V_z\right) \qquad \vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$$

where $\left(\hat{i}, \hat{j}, \hat{k}\right)$ are unit vectors parallel to the X, Y and Z axes respectively.

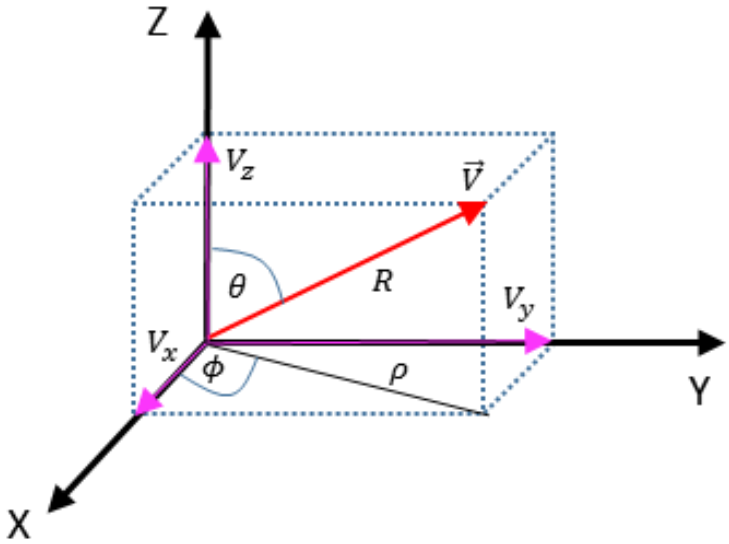


Fig. 1. Specifying a vector in an orthogonal coordinate system.

The **polar angle** θ is the angle down from the Z axis to the vector $\vec{V}$.

The **azimuthal angle** ϕ is the angle around from the X axis.

Polar angle $0 \le \theta \le \pi$

Azimuthal angle $0 \le \phi \le 2\pi$ or $-\pi \le \phi \le +\pi$

Angles can be measured in radians or in degrees where $2\pi \text{ rad} = 360^{\circ}$

You can use the Python functions **radians()** and **degrees()** for the conversions between radians and degrees

op100.py

```
#%% Cell 1: degrees <---> radians
# Input angle in degrees --> output in radians
thetaD = 180
thetaR = np.radians(thetaD)
print('Angle = %0.3f deg' % thetaD + '  = %0.3f rad' % thetaR)

# Input angle in radianss --> output in degrees
thetaR = pi/4
thetaD = np.degrees(thetaR)
print('Angle = %0.3f deg' % thetaD + '  = %0.3f rad' % thetaR)
```

→

```
Angle = 180.000 deg  = 3.142 rad
Angle = 45.000 deg  = 0.785 rad
```

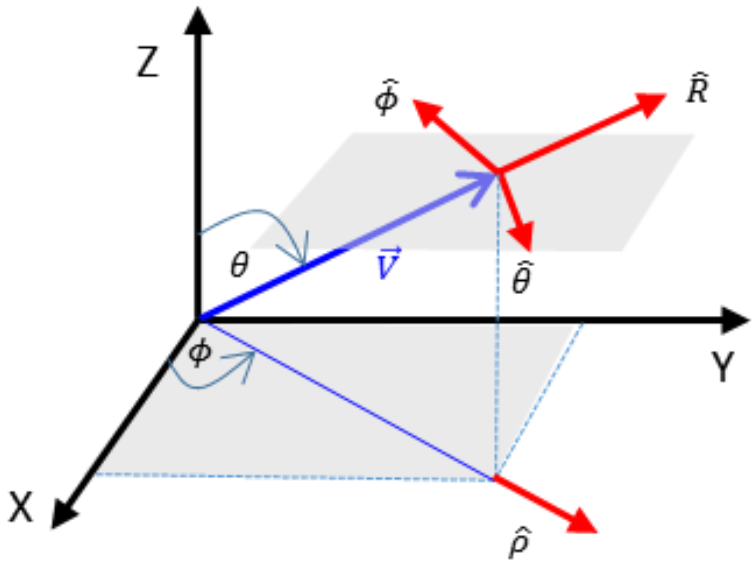


Fig. 2. The unit vectors $\hat{R}, \hat{\theta}, \hat{\phi}, \hat{\rho}$ pointing in the direction of an increase in the corresponding coordinate.

Cartesian components $\vec{V}\left(V_x, V_y, V_z\right)$ $\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$

Cylindrical components $\vec{V}\left(V_\rho, V_\phi, V_z\right)$ $\vec{V} = V_\rho \hat{\rho} + V_\phi \hat{\phi} + V_z \hat{k}$

Spherical components $\vec{V}\left(V_R, V_\theta, V_\phi\right)$ $\vec{V} = V_R \hat{R} + V_\theta \hat{\theta} + V_\phi \hat{\phi}$

Magnitudes

$$\left|\vec{V}\right| \equiv V \equiv R = \sqrt{V_x^{\,2} + V_y^{\,2} + V_z^{\,2}}$$

$$\rho = \sqrt{V_x^{\,2} + V_y^{\,2}}$$

Relationship between coordinates from figure 2

$V_x = R\sin\theta\cos\phi$ $V_y = R\sin\theta\sin\phi$ $V_z = R\cos\theta$

$V_x = \rho\cos\phi$ $V_y = \rho\sin\phi$ $V_z = V_z$

$$\tan\phi = \frac{V_y}{V_x} \qquad \tan\theta = \frac{\rho}{V_z} \qquad \cos\theta = \frac{V_z}{R}$$

Spherical coordinates

$$\hat{R} = \sin\theta\,\cos\phi\,\hat{i} + \sin\theta\,\sin\phi\,\hat{j} + \cos\theta\,\hat{k}$$
$$\hat{\theta} = \cos\theta\,\cos\phi\,\hat{i} + \cos\theta\,\sin\phi\,\hat{j} - \sin\theta\,\hat{k}$$
$$\hat{\phi} = -\sin\phi\,\hat{i} + \cos\theta\,\sin\phi\,\hat{j}$$

Cylindrical coordinates

$$\hat{\rho} = \cos\phi\,\hat{i} + \sin\phi\,\hat{j}$$
$$\hat{\phi} = -\sin\phi\,\hat{i} + \cos\theta\,\sin\phi\,\hat{j}$$
$$\hat{z} = \hat{z}$$

Python

In Python vectors and matrices are referred to as arrays. There are no row or column vectors as in Matlab and dealing with arrays in Python is more complex than in Matlab.

Consider the vectors using the Python code op100.py (Cell #2)

$$\vec{V}_1 = 3\hat{i} + 5\,\hat{j} - 6\hat{k} \qquad \vec{V}_2 = 5\hat{i} + 9\,\hat{j} + 2\hat{k} \qquad \vec{V}_3 = 5\hat{i} + 9\,\hat{j} + 2\hat{k}$$

By default, each vector is a row vector (1,3). Vector 3 can be reshaped to a column vector (3,1)

V1 = array([3.0,5,-6]) → array([3., 5., -6.])

V2 = array([5.0, 9, 2]) → array([5., 9., 2.])

V3 = array([5.0, 9, 2]) → array([5., 9., 2.])

V3.shape = (3,1) → array([[5.],

[9.],

[2.]])

Addition, subtraction, magnitudes

V12 = V1 + V2 → array([8., 14., -4.])

V21 = V1 - V2 → array([-2., -4., -8.])

V13 = V1 + V3 → array([[8., 10., -1.],

[12., 14., 3.],

[5., 7., -4.]])

V31 = V1 - V3 → array([[-2., 0., -11.],

[-6., -4., -15.],

[1., 3., -8.]])

V1mag = norm(V1) → 8.3666

V2mag = norm(V2) → 10.4881

V3mag = norm(V3) → 10.4881

V12mag = norm(V12) → 16.6132

V21mag = norm(V21) → 9.1652

V13mag = norm(V13) → 24.5764

V31mag = norm(V31) → 21.8174

The function norm gives the square root of the sum of the squares of each element of the array.

[3D] orientation of vector $\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$

Magnitude $R = \sqrt{V_x^2 + V_y^2 + V_z^2}$ $\rho = \sqrt{V_x^2 + V_y^2}$

Polar angle $\tan\theta = \frac{\rho}{V_z}$ $\cos\theta = \frac{V_z}{R}$

Azimuthal angle $\tan\phi = \frac{V_y}{V_x}$

$V = \vec{V}_2 = 5\hat{i} + 9\hat{j} + 2\hat{k}$

Magnitudes R, ρ

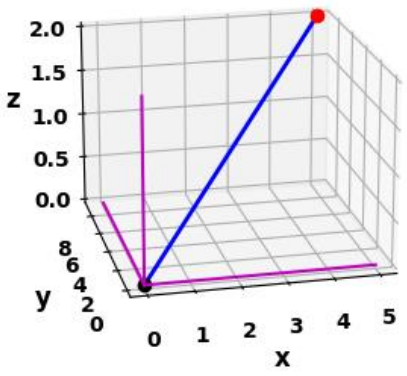
V2mag = norm(V2) → 10.4881

rho2 = sqrt(V2[0]**2 + V2[1]**2) → 10.2956

Polar and azimuthal angle

phi2R = atan2(V2[1],V2[0]) → 1.0637 rad

phi2Deg = np.degrees(phi2R) → 60.9454 deg



op100.py (Cell

VECTOR "MULTIPLICATION"

V1 → array([3., 5., -6.])
V2 → array([5., 9., 2.])
V3 → array([[5.],
[9.],
[2.]])
M1 = V1*V2 → array([15., 45., -12.])
M2 = V2*V1 → array([15., 45., -12.])
M3 = V1*V3 → array([[15., 25., -30.],
[27., 45., -54.],
[6., 10., -12.]])

M4 = V3*V1 → array([[15., 25., -30.],
[27., 45., -54.],
[6., 10., -12.]])

Dot product (scalar product) of two vectors

$$\vec{A}\bullet\vec{B} = AB\cos\theta = A_x B_x + A_y B_y + A_z B_z$$

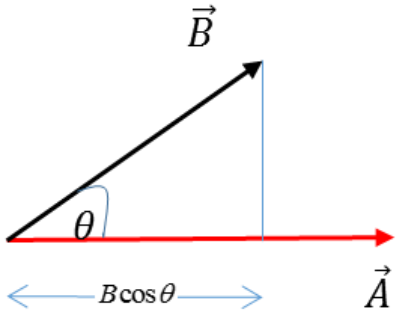
where θ is the angle between the two vectors when they are placed tail to tail

$\vec{A}\bullet\vec{B}$ scalar

$\vec{A}\bullet\vec{B} = \vec{B}\bullet\vec{A}$ commutative

$\vec{A}\bullet\left(\vec{B}+\vec{C}\right) = \vec{A}\bullet\vec{B}+\vec{A}\bullet\vec{C}$ distributive

Geometrically $\vec{A}\bullet\vec{B}$ is the product of $\vec{A}$ times the projection of $\vec{B}$ along $\vec{A}$ or the product of $\vec{B}$ times the projection of $\vec{A}$ along $\vec{B}$



Angle between the two vectors

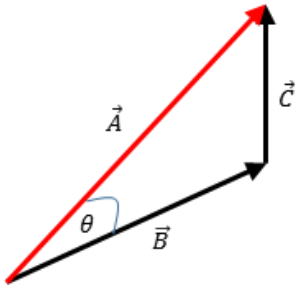
$$\cos\theta = \frac{\vec{A}\bullet\vec{B}}{AB} = \frac{A_x B_x + A_y B_y + A_z B_z}{AB}$$

Law of cosines

$$\vec{A} = \vec{B} + \vec{C} \qquad \vec{C} = \vec{A} - \vec{B}$$

$$\vec{C}\bullet\vec{C} = \left(\vec{A} - \vec{B}\right)\bullet\left(\vec{A} - \vec{B}\right) = \vec{A}\bullet\vec{A} + \vec{B}\bullet\vec{B} - 2\vec{A}\bullet\vec{B}$$

$$C^2 = A^2 + B^2 - 2AB\cos\theta$$



Python function **np.dot(A,B)**

mDot12 = np.dot(V1,V2) → 48

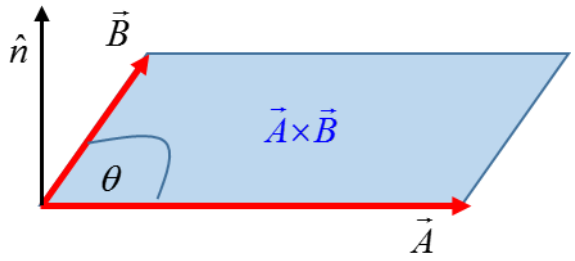
mDot21 = np.dot(V2,V1) → 48

mDot13 = np.dot(V1,V3) → 48

mDot31 = np.dot(V3,V1) → shapes (3,1) and (3,) not aligned: 1 (dim 1) != 3 (dim 0)

Cross product (vector product) of two vectors

$$\vec{A}\times\vec{B} = AB\sin\theta\,\hat{n}$$



where θ is the angle between the two vectors placed tail to tail and $\hat{n}$ is a unit vector that is normal to the plane defined by the two vectors and whose direction is determined by the right-hand rule (fingers curl from $\vec{A}$ to $\vec{B}$ then extended thumb points in direction of $\hat{n}$).

$\vec{A}\times\vec{B} = -\vec{B}\times\vec{A}$ non-commutative

$\vec{A}\times\left(\vec{B}+\vec{C}\right) = \vec{A}\times\vec{B} + \vec{A}\times\hat{C}$ distributive

$\vec{A}\|\vec{B} \quad\Rightarrow\quad \theta = 0 \quad \vec{A}\times\vec{B} = 0$

$\vec{A}\times\vec{A} = 0$

$\vec{A}\perp\vec{B} \quad\Rightarrow\quad \theta = \pi/2\ rad \quad \left|\vec{A}\times\vec{B}\right| = AB$

The cross product is the vector area of the parallelogram having $\vec{A}$ and $\vec{B}$ on adjacent sides.

Determinant form

$$\vec{A}\times\vec{B} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} = \left(A_y B_z - A_z B_y\right)\hat{i} + \left(A_z B_x - A_x B_z\right)\hat{j} + \left(A_x B_y - A_y B_x\right)\hat{k}$$

Python function **cross(A,B)** returns the cross product of the vectors A and B where A and B must be 3 element arrays.

Results of running the Python code op100.py (Cell #5)

Inputs vectors

A = [1 0 1] B = [0 1 1] C = [2 4 5]

A = array([1.0,0,1]); B = array([0.0,1,1]); C = array([2.0,4,5])

Dot product

AdotB = dot(A,B) → 1.0

BdotA = dot(B,A)) → 1.0

AdotC = dot(A,C) → 7.0

CdotA = dot(C,A) → 7.0

BdotC = dot(B,C) → 9.0

CdotB = dot(C,B) → 9.0

Magnitudes and angles between vectors

Amag = norm(A) → 1.4142

Bmag = norm(B) → 1.4142

Cmag = norm(C) → 6.70320

ABangle = acos(AdotB/(Amag*Bmag)) → 1.0472

ABdeg = degrees(ABangle) → 60

BAangle = acos(BdotA/(Amag*Bmag)) → 1.0472

BAdeg = degrees(BAangle) → 60

ACangle = acos(AdotC/(Amag*Cmag)) → 0.74089

ACdeg = degrees(ACangle) → 42.4502

BCangle = acos(BdotC/(Bmag*Cmag)) → 0.3218

BCdeg = degrees(BCangle) → 18.4349

Cross products and their magnitude

AB = cross(A,B) → array([-1., -1., 1.])

ABmag = norm(AB) → 1.7321

BA = cross(B,A) → array([1., 1., -1.])

BAmag = norm(BA) → 1.7321

AC = cross(A,C) → array([-4., -3., 4.])

ACmag = norm(AC) → 6.4031

CA = cross(C,A) → array([4., 3., -4.])

CAmag = norm(CA) → 6.4031

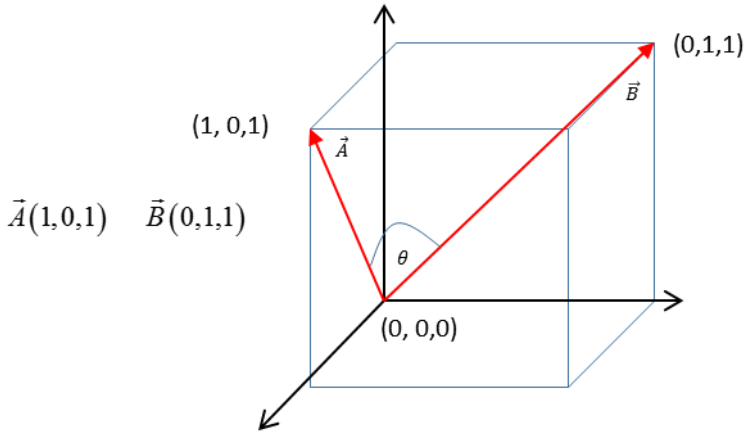
BC = cross(B,C) → array([1., 2., -2.])

BCmag = norm(BC) → 3.0

CB = cross(C,B) → array([-1., -2., 2.])

BCBmag = norm(CB) → 3.0

Example Find the angle between the face diagonals of a cube



⇒

The angle between the two vectors can be found from the cross product of the two vectors

$$\vec{A}\times\vec{B} = AB\sin\theta\ \hat{n}$$

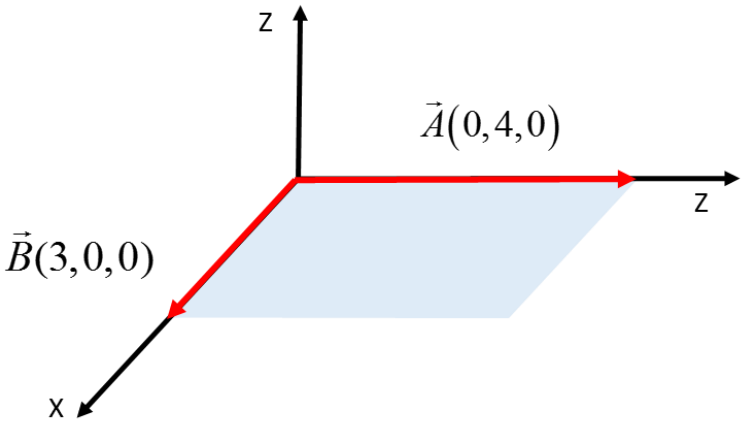
$$\sin\theta = \frac{\left|\vec{A}\times\vec{B}\right|}{\left|\vec{A}\right|\left|\vec{B}\right|}$$

A = [1 0 1] B = [0 1 1]

angle is θ = 1.0472 rad = 60.0000 deg

⇐

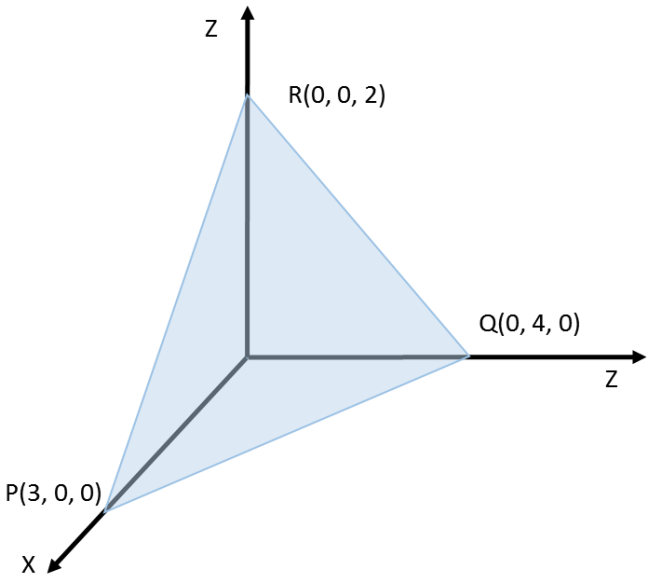
Example Find the components of the unit vector $\hat{n}$ perpendicular to the shaded regions formed by the vectors $\vec{A}$ and $\vec{B}$



$$\vec{C} = \vec{A} \times \vec{B} = -12\hat{k} = \left|\vec{C}\right|\hat{n} \quad n = \frac{\vec{C}}{\left|\vec{C}\right|} = \frac{-12\hat{k}}{12} = -\hat{k}$$

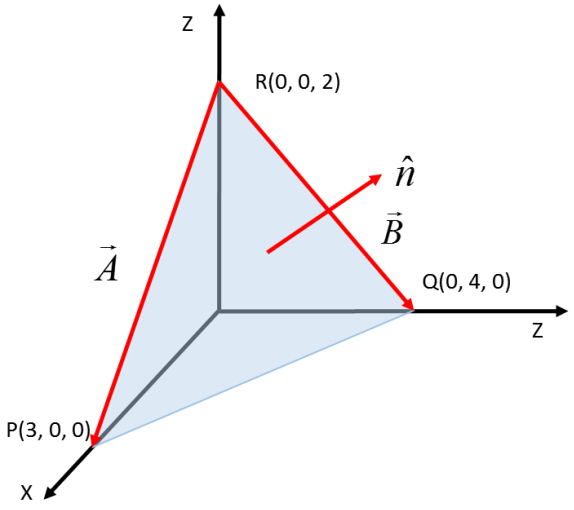
$$n_x = 0 \quad n_y = 0 \quad n_z = -1$$

Example Find the components of the unit vector $\hat{n}$ perpendicular to the shaded regions formed by the points R, P, Q.



Let $\vec{A}$ be the vector pointing from R to P and $\vec{B}$ be the vector pointing from R to Q. Then the vectors are $\vec{A}(3,0,-2)$ and $\vec{B}(0,4,-2)$.

$$\vec{A}\times\vec{B} = AB\sin\theta\ \hat{n}$$



A = [3 0 -2] B = [0 4 -2]

C = cross(A,B) C = [8 6 12]

Cmag = norm(C) Cmag = 15.6205

n = C./Cmag n = [0.5121 0.3841 0.7682

The Cartesian components of the vector $\hat{n}$ are (0.5121, 0.3841, 0.7682)

TRANSFORMATION OF COORDINATES DUE TO ROTATION

What is the change in the components of a vector due to a rotation of the coordinate system from X Y Z to X ’Y ’Z’?

The transformation matrix $\mathbf{R}$ due to a rotation uses the following notation

$X \& X' \rightarrow 1 \quad Y \& Y' \rightarrow 2 \quad Z \& Z' \rightarrow 3$

θ_{11} is the angle between axes X' and X

θ_{32} is the angle between axes Z' and Y

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \quad \text{where}$$

$$R_{nm} = \cos\left(\theta_{mn}\right) \qquad m = 1,2,3 \quad n = 1,2,3$$

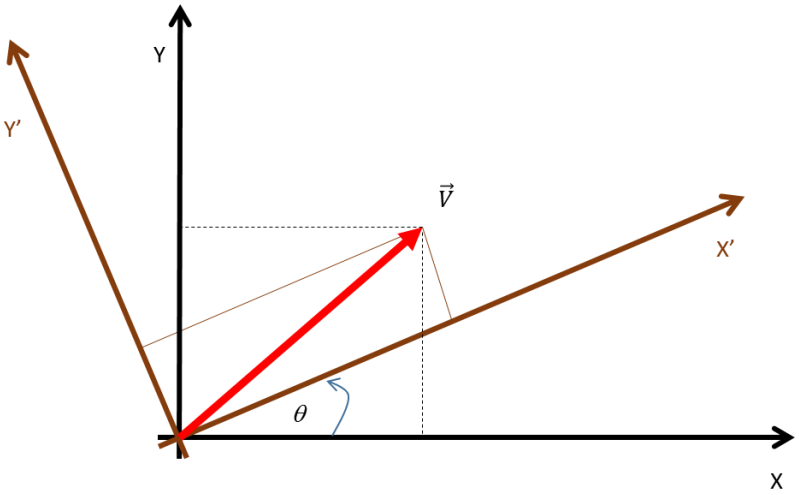
$\vec{V}\left(V_x, V_y, V_z\right)$ in the XYZ coordinate system

$\vec{V}'\left(V'_x, V'_y, V'_z\right)$ in the X’Y’Z’ coordinate system

$$\vec{V}' = \mathbf{R}\vec{V}^T \quad \text{where} \quad \vec{V}^T = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

Rotation of the XY axes around the Z axis

Consider the vector $\vec{V} = V_x \hat{i} + V_y \hat{j} + V_z \hat{k}$ in the unprimed frame of reference.



What will be its components in the primed frame of reference that is rotated by an angle θ in an anticlockwise direction in the XY plane?

Angles between the axes XYZ and X'Y'Z'

$$\theta_{11} = \theta \qquad \theta_{12} = 90^o - \theta \qquad \theta_{13} = 90^o$$
$$\theta_{21} = \theta + 90^o \qquad \theta_{22} = \theta \qquad \theta_{23} = 90^o$$
$$\theta_{31} = 90^o \qquad \theta_{32} = 90^o \qquad \theta_{33} = 0^o$$

Transformation of vector components

$$\left(V'_x, V'_y, V'_z\right) = \begin{pmatrix} \cos(\theta) & \cos(90^o - \theta) & \cos(90^o) \\ \cos(\theta + 90^o) & \cos(\theta) & \cos(90^o) \\ \cos(90^o) & \cos(90^o) & \cos(0^o) \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

$$\left(V'_x, V'_y, V'_z\right) = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix}$$

$$V'_x = \cos(\theta)V_x + \sin(\theta)V_y$$
$$V'_y = -\sin(\theta)V_x + \cos(\theta)V_y$$
$$V'_z = V_z$$

Example

V(2, 3, 0) in XYZ frame of reference

XYZ rotated by 30^o anticlockwise in XY plane to give the X'Y'Z' frame

What are the components of V in the X'Y'Z' frame?

$\Rightarrow$

Inputs: V = [2 3 0] theta = 30

Output: Vdash = [3.2321 1.5981 0]

$\Leftarrow$