Vectors (SKP)

In [1]:
```
import numpy as np
import sympy as smp
from sympy import *
from sympy.vector import *
```

In [2]:
```
x,y,z,t,u1,u2,u3,v1,v2,v3 = smp.symbols('x y z t u_1 u_2 u_3 v_1 v_2 v_3')
```

In [3]:
```
a = np.array([2,3,7])  # input vector a
b = np.array([2,4,1])  # input vector b
u = smp.Matrix([u1,u2,u3])
v = smp.Matrix([v1,v2,v3])
```

Vector products

In [4]:
```
# dot product
print('a.b =', np.dot(a,b))
display('u.v', u.dot(v))

# cross product
print('a X b =', np.cross(a,b))
display('u X v', u.cross(v))
```

```
a.b = 23
```

```
'u.v'
```

$$u_1v_1 + u_2v_2 + u_3v_3$$

```
a X b = [-25  12   2]
```

```
'u X v'
```

$$\begin{bmatrix} u_2v_3 - u_3v_2 \\ -u_1v_3 + u_3v_1 \\ u_1v_2 - u_2v_1 \end{bmatrix}$$

Length of vector

In [5]:
```
print('|a| =',np.linalg.norm(a))

unorm = u.norm()
display('|u|', unorm)
```

```
|a| = 7.874007874011811
```

```
'|u|'
```

$$\sqrt{|u_1|^2 + |u_2|^2 + |u_3|^2}$$

Vector projection

Projection of u on v,

$$\operatorname{proj}_v(u) = (u.\,\hat{v})\hat{v} = \frac{u \cdot v}{|v|^2} v$$

In [6]:
```
projab = np.dot(a,b)*b/np.linalg.norm(b)**2
print('projection of a on b =', projab)

projuv = u.dot(v)*v/v.norm()**2
display('projection of u on v', projuv)
```

```
projection of a on b = [2.19047619 4.38095238 1.0952381 ]
```

```
'projection of u on v'
```

$$\begin{bmatrix} \frac{v_1(u_1v_1+u_2v_2+u_3v_3)}{|v_1|^2+|v_2|^2+|v_3|^2} \\ \frac{v_2(u_1v_1+u_2v_2+u_3v_3)}{|v_1|^2+|v_2|^2+|v_3|^2} \\ \frac{v_3(u_1v_1+u_2v_2+u_3v_3)}{|v_1|^2+|v_2|^2+|v_3|^2} \end{bmatrix}$$

In [7]:
```
### Lines
```

$$\vec{r}(t) = \vec{r}_0 + t\vec{v}$$

In [8]:
```
r0 = smp.Matrix([1,1,1]) # input the vector
v = smp.Matrix([4,3,4])  # input the vector
r = r0 + t*v
r
```

Out[8]:

$$\begin{bmatrix} 4t+1 \\ 3t+1 \\ 4t+1 \end{bmatrix}$$

In [9]:
```
### Planes
```

$$\vec{n} \cdot (P_0 - \langle x, y, z \rangle) = 0$$

In [10]:
```
n = smp.Matrix([3,2,3])   # input the normal vector
P0 = smp.Matrix([2.2,3,2]) # input foot of the perpendicular
r = smp.Matrix([x,y,z])
eqnp = n.dot(P0 - r)
display('equation of the plane', eqnp)
```

```
'equation of the plane'
```

$$-3x - 2y - 3z + 18.6$$

Example: Find unit vector parallel to the line of intersection of the two planes $3x - 6y - 2z = 15$ and $2x + y - 2z = 5$. (Hint: It's going to be perpendicular to both normal vectors)

In [11]:
```
n1 = np.array([3,-6,-2])
n2 = np.array([2,1,-2])
vec = np.cross(n1,n2)
ans = vec/np.linalg.norm(vec)
ans
```

Out[11]:
```
array([0.67909975, 0.09701425, 0.72760688])
```

In [ ]:
```

```