```
(*Matrix and Vector Operations*)
```

In[•]:=
```
u = {a, b, c}
```

Out[•]=
```
{a, b, c}
```

In[•]:=
```
u1 = {2, 5, 7};
u2 = {2, 3, 6};
u1 + u2
```

Out[•]=
```
{4, 8, 13}
```

In[•]:=
```
u2 + u1
```

Out[•]=
```
{4, 8, 13}
```

In[•]:=
```
u1 + u2 == u2 + u1
```

Out[•]=
```
True
```

In[•]:=
```
u1 – u2
```

Out[•]=
```
{0, 2, 1}
```

In[•]:=
```
u2 – u1
```

Out[•]=
```
{0, –2, –1}
```

In[•]:=
```
u1 – u2 == u2 – u1
```

Out[•]=
```
False
```

In[•]:=
```
A = {7, 3, 9};
5 A
```

Out[•]=
```
{35, 15, 45}
```

In[•]:=
```
Norm[A]
```

Out[•]=
$$\sqrt{139}$$

In[•]:=
```
Norm[{1, 0, 0}]
```

Out[•]=
```
1
```

In[•]:=
```
Norm[{9, 2, –4}] // N
```

Out[•]=
```
10.0499
```

In[•]:=
```
Sqrt[A]
```

Out[•]=
$$\left\{\sqrt{7}, \sqrt{3}, 3\right\}$$

In[•]:=
```
? Dot
```

Out[•]=

Symbol

$a.b.c$ or Dot[a, b, c] gives products of vectors, matrices, and tensors.

In[•]:=
```
Dot[u1, u2]
```

Out[•]=
```
61
```

In[•]:=
```
u = {2, 4, 9}; v = {7, 5, 8};
Dot[u, v]
```

Out[•]=
```
106
```

In[•]:=
```
u.v
```

Out[•]=
```
106
```

In[•]:=
```
? Norm
```

Out[•]=

Symbol

Norm[*expr*] gives the norm of a number, vector, or matrix.

Norm[*expr*, *p*] gives the p-norm.

In[•]:=
```
m1 = {{2, 5}, {7, 8}}; m2 = {{1, 2}, {3, 9}}; v = {2, 5};
m1 + m2
```

Out[•]=
```
{{3, 7}, {10, 17}}
```

In[•]:=
```
m2 + m1
```

Out[•]=
```
{{3, 7}, {10, 17}}
```

In[•]:=
```
m1 + m2 == m2 + m1
```

Out[•]=
```
True
```

```
(*scalar multiplication of matrix*)
```

In[•]:=
```
c = 0;
c * m1
```

Out[•]=
```
{{0, 0}, {0, 0}}
```

In[•]:=
```
(*power of matrix*)
m1
m1^2
```

Out[•]=
```
{{2, 5}, {7, 8}}
```

Out[•]=
```
{{4, 25}, {49, 64}}
```

```
In[•]:= v
        m1
Out[•]= {2, 5}
Out[•]= {{2, 5}, {7, 8}}

In[•]:= (*colmn vector with matrix*)
        m1.v
Out[•]= {29, 54}

        (*row vector with matrix*)

In[•]:= v.m1
Out[•]= {39, 50}

In[•]:= x = {2, 4, 6}; y = {7, 5, 3};
        rv = x + y
Out[•]= {9, 9, 9}

In[•]:= mx = Norm[x]; my = Norm[y];
        dotxy = x.y;
        mxmy = mx * my;
        th = ArcCos[dotxy / mxmy]
        th = ArcCos[dotxy / mxmy] // N
        th = ArcCos[dotxy / mxmy] / Degree // N
Out[•]= ArcCos[13 Sqrt[2/581]]
Out[•]= 0.703274
Out[•]= 40.2946

In[•]:= (*Direction Vector and Angle*)
        l = {3, 5, 6};
        unitvector = uv = directionvector = dv = l / Norm[l]
        da = ArcCos[dv] / Degree // N
Out[•]= {3/Sqrt[70], Sqrt[5/14], 3 Sqrt[2/35]}
Out[•]= {68.9877, 53.3008, 44.1814}
```