

Quaternion Demo

```
clc;clear;close all
a=Quaternion([1,2,3,4]);
b=Quaternion([2,1,3,3]);
```

+

```
c=a+b;c.v
```

```
ans = 1x4
      3      3      6      7
```

*

```
c=a*b;c.v
```

```
ans = 1x4
     -21      2      7     14
```

```
% verify the * using built-in quaternion
aq=quaternion(a.v);
bq=quaternion(b.v);
c=aq*bq
```

```
c = quaternion
     -21 + 2i + 7j + 14k
```

conjugate

```
c=conj(a);c.v
```

```
ans = 1x4
      1     -2     -3     -4
```

cross product

```
c=cross(a,b);c.v
```

```
ans = 1x4
      0      2      7     14
```

```
% verify
c=(a*b-b.conj*a.conj)*(1/2);c.v
```

```
ans = 1x4
      0      2      7     14
```

norm^2

```
a.norm
```

```
ans =
Quaternion with properties:
```

```
v: [30 0 0 0]
```

```
% verify a.conj *a when unit  
an=a.normalize
```

```
an =  
  Quaternion with properties:  
  
v: [0.1826 0.3651 0.5477 0.7303]
```

```
an.conj*an
```

```
ans =  
  Quaternion with properties:  
  
v: [1.0000 0 0 0]
```

```
an*an.conj
```

```
ans =  
  Quaternion with properties:  
  
v: [1.0000 0 0 0]
```

normalize

```
a.normalize*a.normalize.conj
```

```
ans =  
  Quaternion with properties:  
  
v: [1.0000 0 0 0]
```

```
a.normalize.norm
```

```
ans =  
  Quaternion with properties:  
  
v: [1.0000 0 0 0]
```

get pure / vector quaternion

```
a.vec
```

```
ans =  
  Quaternion with properties:  
  
v: [0 2 3 4]
```

dot product .*

```
a.*b
```

```
ans =  
  Quaternion with properties:  
  
v: [25 0 0 0]
```

```
% verify  
[1,2,3,4]*[2,1,3,3]'
```

```
ans = 25
```