Dual Quaternion Demo

```
clc;clear;close all

a=DualQuaternion([1,2,3,4,4,3,2,1]);
b=DualQuaternion([2,1,3,3,4,1,3,1]);
```

+

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

```
ans = 1×8
     3     3     6     7     8     4     5     2
```

*****

```
c=10*a;c.v
```

```
ans = 1×8
    10    20    30    40    40    30    20    10
```

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

```
ans = 1×8
   -21     2     7    14   -15    13    25    41
```

conjugate

```
a.conj
```

```
ans =
  DualQuaternion with properties:

    v: [1 -2 -3 -4 4 -3 -2 -1]
```

swap the real quaternion and dual quaternion

```
a.swap
```

```
ans =
  DualQuaternion with properties:

    v: [4 3 2 1 1 2 3 4]
```

cross product

```
cross(a,b)
```

```
ans =
  DualQuaternion with properties:

    v: [0 2 7 14 0 13 25 41]
```

```
a.cross(b)
```

```
ans =
  DualQuaternion with properties:

    v: [0 2 7 14 0 13 25 41]
```

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

```
ans = 1×8
     0     2     7    14     0    13    25    41
```

get the vector or pure dual quaternion

```
a.vec
```

```
ans =
  DualQuaternion with properties:

    v: [0 2 3 4 0 3 2 1]
```

norm^2

```
a.norm
```

```
ans =
  DualQuaternion with properties:

    v: [60 0 0 0 0 0 0 0]
```

```
% verify
sum([1,2,3,4,4,3,2,1].*[1,2,3,4,4,3,2,1])
```

```
ans = 60
```

normd^2

```
a.normd
```

```
ans =
  DualQuaternion with properties:

    v: [30 0 0 0 40 0 0 0]
```

```
% note that a unit quaternion has 1 for normd but not norm
b=DQFromEulerVec(pi/2,[0,0,1],[0,0,3]);
b.norm
```

```
ans =
  DualQuaternion with properties:

    v: [3.2500 0 0 0 0 0 0 0]
```

```
b.normd
```

```
ans =
  DualQuaternion with properties:
```

```
    v: [1 0 0 0 0 0 0 0]
```

Real part & dual part

```
a.real
```

```
ans =
  Quaternion with properties:

    v: [1 2 3 4]
```

```
a.dual
```

```
ans =
  Quaternion with properties:

    v: [4 3 2 1]
```