1. The algebra isomorphisms

```
> restart: with(linalg): print('whattype(I)'=whattype(I),'its_square'=I^2);
Warning, the protected names norm and trace have been redefined and unprotected
```

$$\mathrm{whattype}(I) = complex,\ its\_square = -1$$

We shall first construct two isomorphisms *f* and *F* onto the matrix representation, choosing a real respectively complex setting.

```
> f:=q->matrix(2,2,[[q[1]+q[2]*I,-q[3]-q[4]*I],[q[3]-q[4]*I,q[1]-q[2]*I]]):
  'f([a,b,c,d])'=f([a,b,c,d]);
```

$$\mathrm{f}([a, b, c, d]) = \begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}$$

```
> F:=Q->map(z->evalc(z),matrix(2,2,[[Q[1],-Q[2]],[conjugate(Q[2]),conjugate(Q[1])]])):

  'F([a+b*I,c+d*I])'=F([a+b*I,c+d*I]);
```

$$\mathrm{F}([a+b\,I, c+d\,I]) = \begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}$$

We now separately construct the inverse isomorphisms followed by a verification that they are inverse indeed.

```
> g:=A->map(z->evalc(z),[Re(A[1,1]),Im(A[1,1]),-Re(A[1,2]),-Im(A[1,2])]):
  'g(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]]))'=g(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]])
  );
```

$$\mathrm{g}\left(\begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}\right) = [a, b, c, d]$$

```
> G:=A->[A[1,1],-A[1,2]]:
  'G(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]]))'=G(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]])
  );
```

$$\mathrm{G}\left(\begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}\right) = [a+b\,I, c+d\,I]$$

```
> '(g@f)([a,b,c,d])'=(g@f)([a,b,c,d]);
```

$$(g@f)([a, b, c, d]) = [a, b, c, d]$$

```
> '(f@g)(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]]))'=(f@g)(matrix([[a+b*I,-c-I*d],[c-I*d,
  a-I*b]]));
```

$$(f@g)\left(\begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}\right) = \begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}$$

```
> '(G@F)([a+b*I,c+d*I])'=(G@F)([a+b*I,c+d*I]);
```

$$(G@F)([a+b\,I, c+d\,I]) = [a+b\,I, c+d\,I]$$

```
> '(F@G)(matrix([[a+b*I,-c-I*d],[c-I*d,a-I*b]]))'=(F@G)(matrix([[a+b*I,-c-I*d],[c-I*d,
  a-I*b]]));
```

$$(F@G)\left(\begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}\right) = \begin{bmatrix} a+b\,I & -c-d\,I \\ c-d\,I & a-b\,I \end{bmatrix}$$

As a bonus we get two, mutually inverse, identifying maps between the 4-d real and 2-d complex linear representations of ***H***.

```
> '(G@f)([a,b,c,d])'=(G@f)([a,b,c,d]);
```

$$(G@f)([a, b, c, d]) = [a+b\,I, c+d\,I]$$

```
> '(g@F)([a+b*I,c+d*I])'=(g@F)([a+b*I,c+d*I]);
```

$$(g@F)([a+b\,I, c+d\,I]) = [a, b, c, d]$$

2. Quaternion product, conjugate and norm

We shall now express the quaternion product by means of the corresponding matrix product and verify that *V*, the subspace of quaternions with zero scalar part, is closed under multiplication. The relation to inner and outer product on *V* becomes apparent.

```
> p:=(x,y)->g(evalm(f(x)&*f(y))):
  p([a[1],b[1],c[1],d[1]],[a[2],b[2],c[2],d[2]]);
```

$$[a_1\,a_2 - b_1\,b_2 - c_1\,c_2 - d_1\,d_2,\ a_1\,b_2 + b_1\,a_2 + c_1\,d_2 - d_1\,c_2,\ a_1\,c_2 - b_1\,d_2 + c_1\,a_2 + d_1\,b_2,\ a_1\,d_2 + b_1\,c_2 - c_1\,b_2 + d_1\,a_2]$$

```
> p([0,b[1],c[1],d[1]],[0,b[2],c[2],d[2]]);
```

$$[-b_1\,b_2 - c_1\,c_2 - d_1\,d_2,\ c_1\,d_2 - d_1\,c_2,\ -b_1\,d_2 + d_1\,b_2,\ b_1\,c_2 - c_1\,b_2]$$

```
> innerprod([b[1],c[1],d[1]],[b[2],c[2],d[2]]);
```

$$b_1\, b_2 + c_1\, c_2 + d_1\, d_2$$

```
> crossprod([b[1],c[1],d[1]],[b[2],c[2],d[2]]);
```

$$[c_1\, d_2 - d_1\, c_2, -b_1\, d_2 + d_1\, b_2, b_1\, c_2 - c_1\, b_2]$$

Let us check that the set of solutions of the quadratic equation $X^2 + 1 = 0$ is the unit sphere S in V given by $b^2 + c^2 + d^2 = 1$ (and a=0).

```
> p([a,b,c,d],[a,b,c,d]);
```

$$[a^2 - b^2 - c^2 - d^2, 2\,a\,b, 2\,a\,c, 2\,a\,d]$$

For notational convenience the quaternion *conjugate* (an algebra anti-isomorphism of order 2) and *Norm* (a multiplicative homomorphism) are defined.

```
> conj:=q->[q[1],-q[2],-q[3],-q[4]]: 'conj([a,b,c,d])'=conj([a,b,c,d]);
  N:=q->(q[1]^2+q[2]^2+q[3]^2+q[4]^2): 'N([a,b,c,d])'=N([a,b,c,d]);
```

$$\mathrm{conj}([a, b, c, d]) = [a, -b, -c, -d]$$

$$\mathrm{N}([a, b, c, d]) = a^2 + b^2 + c^2 + d^2$$

3. The action of the automorphism group of *H* as the orthogonal group *O*(3)

We shall now construct the internal automorphism $f_q = v \to \sigma(q, v)$ of ***H***, see below, for quaternions q with $\mathrm{N}(q) = 1$ only. This is simpler and sufficient for our purpose of showing that f_q acts on S as the full rotation group $O(3)$. By the way, ***H*** has no outer algebra automorphisms.

```
> sigma:=(q,v)->p(p(q,[v[1],v[2],v[3],v[4]]),conj(q)):
  map(w->collect(w,[x,y,z],distributed),sigma([a,b,c,d],[0,x,y,z]));
```

$$[0, (a^2 - d^2 - c^2 + b^2)\,x + (2\,c\,b - 2\,d\,a)\,y + (2\,c\,a + 2\,d\,b)\,z, (2\,d\,a + 2\,c\,b)\,x + (c^2 - b^2 - d^2 + a^2)\,y + (-2\,b\,a + 2\,c\,d)\,z, (-2\,c\,a + 2\,d\,b)\,x + (2\,b\,a + 2\,c\,d)\,y + (a^2 - b^2 + d^2 - c^2)\,z]$$

Since the centralizer of a nonscalar q in H is the 2-d subspace generated by 1 and q , the vectorial part of q (q minus its scalar part) lies on the rotation axis and vectors in the 2-d orthogonal complement in V of this axis, the rotation plane, stay orthogonal when f_q is applied.

```
> subs(a^2+b^2+c^2+d^2=1,map(z->factor(z),sigma([a,b,c,d],[0,lambda*b,lambda*c,lambda*
  d])));
```

$$[0, \lambda\, b, \lambda\, c, \lambda\, d]$$

```
> map(v->innerprod([a,b,c,d],v),{sigma([a,b,c,d],[0,0,-d,c]),sigma([a,b,c,d],[0,-d,0,b
  ]),sigma([a,b,c,d],[0,-c,b,0])});
```

$$\{0\}$$

More generally f_q is an orthogonal mapping in V, that is, it preserves inner products:

```
> subs(a^2+b^2+c^2+d^2=1,factor(innerprod(sigma([a,b,c,d],[0,x[1],y[1],z[1]]),sigma([a
  ,b,c,d],[0,x[2],y[2],z[2]]))));
```

$$y_1\, y_2 + z_2\, z_1 + x_2\, x_1$$

So by continuity f_q neccesarily is a rotation (with determinant +1, instead of the other a priori possibility -1). We finally determine the *common* angle of rotation ϕ , up to a possible minus sign, for two linearly independent vectors in the rotation plane within V (with equation $b\,x + c\,y + d\,z = 0$). Note that a^2 may vary between 0 and 1. In particular, if q itself is in V, then $\phi = \pi$.

```
> innerprod([0,-c,b,0],sigma([a,b,c,d],[0,-c,b,0])/(b^2+c^2)):
  innerprod([0,-d,0,b],sigma([a,b,c,d],[0,-d,0,b])/(b^2+d^2)):
  print(cos(phi[1])=subs(b^2=1-a^2-c^2-d^2,%),
  cos(phi[2])=subs(b^2=1-a^2-c^2-d^2,%%));
```

$$\cos(\phi_1) = -1 + 2\,a^2, \cos(\phi_2) = -1 + 2\,a^2$$

The outer product may also be used. Since $\mathrm{N}(q) = 1$, the outcome may be written as $2\,a\,\sqrt{1 - a^2}\,[b, c, d]/\sqrt{b^2 + c^2 + d^2}$, where, $2\,a\,\sqrt{1 - a^2} = \sin(\phi)$, (or $-\sin(\phi)$).

```
> simplify(crossprod([-c,b,0],sigma([a,b,c,d],[0,-c,b,0])[2..4]/(b^2+c^2))):
  simplify(crossprod([-d,0,b],sigma([a,b,c,d],[0,-d,0,b])[2..4]/(b^2+d^2))):
  print(%, %%);
```

$$[2\,b\,a, 2\,c\,a, 2\,d\,a], [2\,b\,a, 2\,c\,a, 2\,d\,a]$$