Initial coordinates are:

$$S\ (0.138, 0.0, -1.654)$$
$$C_1\ (0.474, 0.0, 1.059)$$
$$C_2\ (-0.621, 0.0, -0.007)$$
$$O\ (-0.125, 0.0, 2.357)$$
$$H\ (0.598, 0.0, 2.999)$$

The axis of rotation is then:

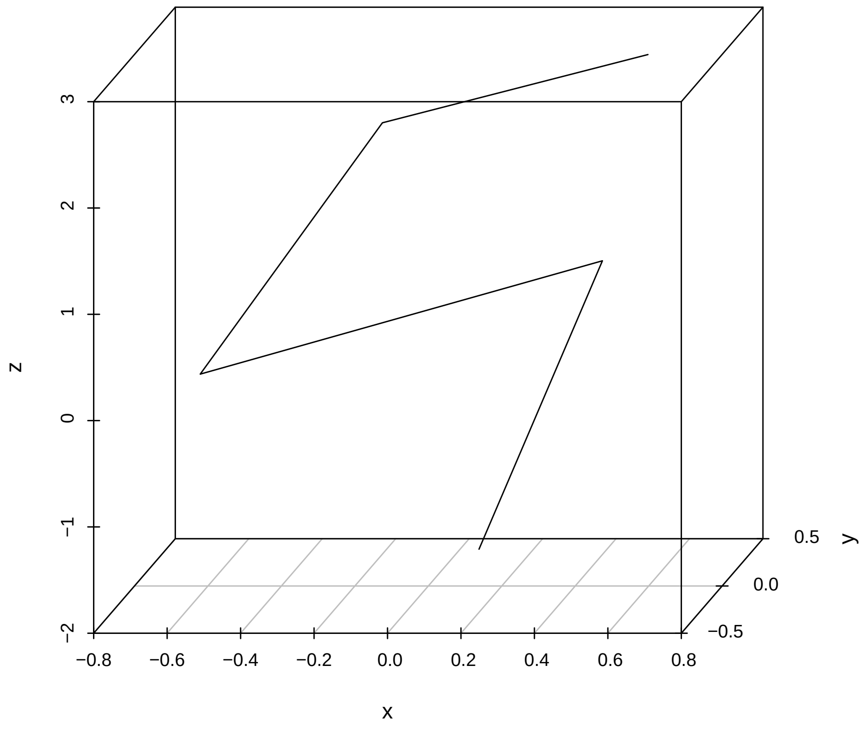
```r
library(scatterplot3d)
S = c(0.138,0.0,-1.654)
C1 = c(0.474, 0., 1.059)
C2 = c(-0.621,0.,-0.007)
O = c(-0.125, 0.0, 2.357)
H = c(0.598, 0.0, 2.999)
axis = C2-C1
message(sprintf("The axis of rotation is: %s",paste(axis,collapse=", ")))
```

```
## The axis of rotation is:  -1.095, 0, -1.066
```

```r
normaxis = axis/sqrt(sum(axis^2))
message(sprintf("The normed axis of rotation is: %s",paste(normaxis,collapse = ", ")))
```

```
## The normed axis of rotation is:  -0.716531434615578, 0, -0.697554803013887
```

```r
#Plot the molecule using scatterplot3d
x = c(S[1],C1[1],C2[1],O[1],H[1])
y = c(S[2],C1[2],C2[2],O[2],H[2])
z = c(S[3],C1[3],C2[3],O[3],H[3])
#data_frame = data.frame(x,y,z)
scatterplot3d(x,y,z,type = "l")
```



The quaternion that defines a $\frac{\pi}{2}$ radian rotation about this axis is:

$$\vec{q} = \cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2}) \cdot \frac{1}{\|a\|}\vec{a} = \cos(\frac{\pi}{4}) + \sin(\frac{\pi}{4}) \cdot \frac{1}{\|a\|}\vec{a}$$

This turns out to be:

```
library(onion)
q = cos(pi/4) + sin(pi/4) * normaxis[1] * Hi + sin(pi/4) * normaxis[2] * Hj +
    sin(pi/4) * normaxis[3] * Hk
message(sprintf("The quaternion of rotation is %s", paste(q, collapse = ", ")))

## The quaternion of rotation is 0.707106781186548, -0.50666423635,
0, -0.493245731460365
```

Finally we get the vectors we wish to rotate:

```
O = c(-0.125, 0, 2.357)
H = c(0.598, 0, 2.999)
v1 = O - C2
v2 = H - C2
message(sprintf("Vector 1 is %s", paste(v1, collapse = ", ")))

## Vector 1 is 0.496, 0, 2.364

message(sprintf("Vector 2 is %s", paste(v2, collapse = ", ")))

## Vector 2 is 1.219, 0, 3.006
```

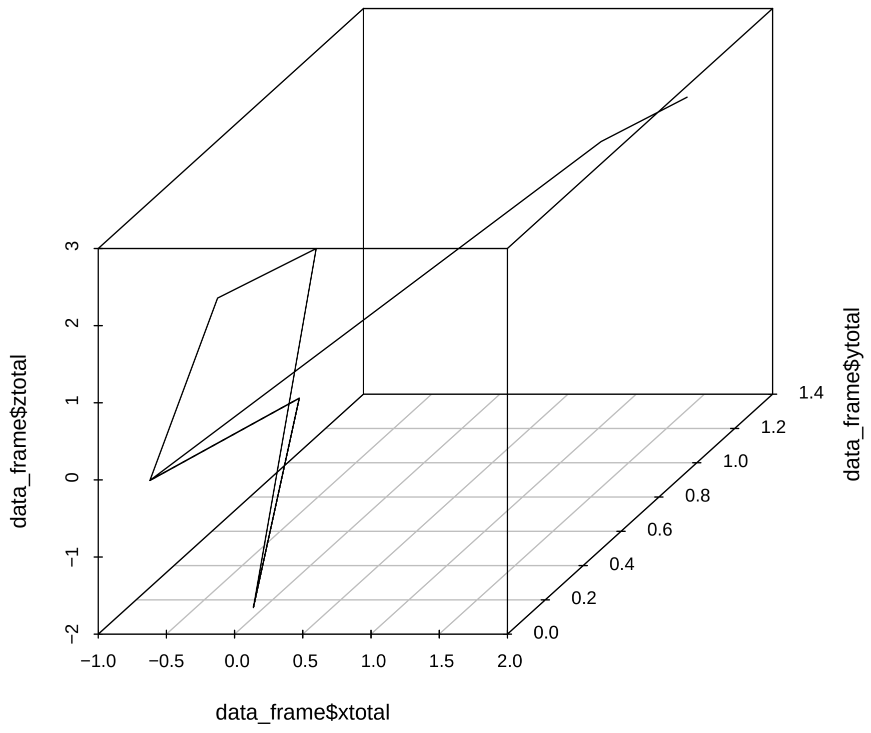
Applying the rotation quaternions we get:

```
v1_q = 0 + v1[1] * Hi + v1[2] * Hj + v1[3] * Hk
v2_q = 0 + v2[1] * Hi + v2[2] * Hj + v2[3] * Hk
v1_rot_q = (q * v1_q) * Conj(q)
v2_rot_q = (q * v2_q) * Conj(q)
v1_rot = c(i(v1_rot_q), j(v1_rot_q), k(v1_rot_q))
v2_rot = c(i(v2_rot_q), j(v2_rot_q), k(v2_rot_q))
message(sprintf("v1 rotated pi/2 radians is now %s", paste(v1_rot, collapse = ", ")))

## v1 rotated pi/2 radians is now 1.43622932617847, 1.34789312913634,
1.39819220247146

message(sprintf("v2 rotated pi/2 radians is now %s", paste(v2_rot, collapse = ",")))

## v2 rotated pi/2 radians is now 2.1283144356317,1.3035741875805,2.07194811724511
```

This translates to final coordinates of O and H of:

```
Onew = v1_rot + C2
Hnew = v2_rot + C2
message(sprintf("Onew is %s", paste(Onew, collapse = ", ")))

## Onew is 0.815229326178469, 1.34789312913634, 1.39119220247146

message(sprintf("Hnew is %s", paste(Hnew, collapse = ",")))

## Hnew is 1.5073144356317,1.3035741875805,2.06494811724511

# Plot the new dihedral
xnew = c(S[1], C1[1], C2[1], Onew[1], Hnew[1])
ynew = c(S[2], C1[2], C2[2], Onew[2], Hnew[2])
znew = c(S[3], C1[3], C2[3], Onew[3], Hnew[3])

xtotal = c(x, xnew)
ytotal = c(y, ynew)
```

```r
ztotal = c(z, znew)
category = c(rep("blue", 5), rep("red", 5))
data_frame = data.frame(xtotal, ytotal, ztotal)
data_frame$fac = factor(rep(LETTERS[1:2], each = 5))
scatterplot3d(data_frame$xtotal, data_frame$ytotal, data_frame$ztotal, type = "l",
    color = as.numeric(data_frame$fac))
```



The calculated dihedral angle from these new coordinates is:

```r
library(pracma)

##
## Attaching package: 'pracma'
## The following object is masked from 'package:onion':
##
##     Norm
```

```
b1 = C1 - S
b2 = C2 - C1
b3 = Onew - C2
b2norm = b2/sqrt(sum(b2^2))
n1 = cross(b1, b2)/sqrt(sum(cross(b1, b2)^2))
n2 = cross(b3, b2)/sqrt(sum(cross(b2, b3)^2))
m1 = cross(n1, n2)
angle = atan2((m1 %*% b2norm), (n1 %*% n2))
message(sprintf("The calculated dihedral angle of the resulting rotation is: %4.2f degrees",
    angle * 180/pi))
```

```
## The calculated dihedral angle of the resulting rotation is:  90.00
degrees
```