
Subject: Re: 3D vector rotation to the Z axis
Posted by [James\[2\]](#) on Thu, 16 Dec 2010 19:48:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Do you have to use a rotation matrix? Quaternions (<http://en.wikipedia.org/wiki/Quaternion>) are more numerically stable, and there is an easy-to-use library by Craig Markwardt at <http://www.physics.wisc.edu/~craigm/idl/math.html>. I was just using them to draw some rotating polyhedra in IDL, and they work great! Your code would look like this:

```
;define the axis of rotation
rotaxis = crossp (input, [0,0,1])
```

```
;find the angle of rotation
rotangle = transpose(input) # [0,0,1]
```

```
;make the quaternion
q = qtcompose(rotaxis, rotangle)
```

```
;do it
rotated = qtvrot(input, q)
```

```
;there is even a routine to create a rotation matrix from the
quaternion:
rmatx = qtmat(q)
```

by the way, I like the # operator. It lets you treat 1D arrays as column vectors; then defining a matrix is just concatenating a group of column vectors across the second dimension. Matrix-by-vector multiplication works like you expect.

- James

On Dec 14, 12:28 pm, David Fanning <n...@dfanning.com> wrote:

```
> MartyL writes:
>> Any help appreciated.
>
> I don't have any suggestions. You obviously know more
> about this than I do. Just an observation. I know any
> time I try to translate matrix operations out of a book
> into IDL my head swells up to about three times its
> normal size trying to keep columns and rows straight.
>
> That said, using the # operator, rather than the ## operator
> always throws me back from discovering the solution at LEAST
> two days, sometimes more. :-(
```

>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
