PHYS 357 Pset 3. Due 11:59 PM Thursday Sep. 26

1. Starting with the angular momentum operators (Jx,Jy,Jz) you worked out in the last problem set (which should have been

$$J_z = \frac{\hbar}{2}\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \quad J_x = \frac{\hbar}{2}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \quad J_y = \frac{\hbar}{2}\begin{bmatrix}0 & -i\\ i & 0\end{bmatrix} \tag{1}$$

show that you get the cannonical commutation relations

$$[J_x, J_y] = i\hbar J_z, \quad [J_y, J_z] = i\hbar J_x, \quad [J_z, J_x] = i\hbar J_y \tag{2}$$

where $[A, B] = AB - BA$ for matrices A and B. *see python code*

2. Consider two coordinate systems (x, y, z) and (x', y', z) that have the same z-axis, but have their x- and y-axes rotated by some angle γ relative to each other (note the lack of a prime on z). Show that if I rotate a vector in the (x, y, z) coordinate system through some angle θ about the z-axis, I get the same physical vector if I transform the vector to the (x', y', z) coordinate system and carry out the rotation there.

If (x', y', z) is rotated by an angle ϕ relative to (x, y, z), then we can get a vector to the new coordinate system by multiplying by $R_z(-\phi)$ (where the notation means we're rotating by an angle $-\phi$ about the z-axis), and get it back to the original coordinate system by multiplying by $R_z(\phi)$. If the rotation of the vector v is by an angle θ, then in the original coordinate system, we have $v_{new} = R_z(\theta)v$ while rotating to the new coordinate system, rotating by θ and going back, we have $v_{new} = R_z(\phi)R_z(\theta)R_z(-\phi)v$. The short answer here is that rotations about the *same* axis *do* commute, so the $R_z(\pm\phi)$ rotations cancel, and we're left with just an $R_z(\theta)$. More formally, for any two angles,

$$R_z(\alpha)R_z(\beta) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$= \begin{bmatrix} \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta) & -\cos(\alpha)\sin(\beta) - \sin(\alpha)\cos(\beta) & 0 \\ \cos(\beta)\sin(\alpha) + \cos(\alpha)\sin(\beta) & -\sin(\beta\sin(\alpha) + \cos(\beta)\cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} \cos(\alpha+\beta) & -\sin(\alpha+\beta) & 0 \\ \sin(\alpha+\beta) & \cos(\alpha+\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

$$= R_z(\alpha+\beta) \tag{6}$$

Obviously, $R_z(\alpha+\beta) = R_z(\beta+\alpha)$ since scalar additions commute, so our same-axis rotations commute. That means that indeed $R_z(\phi)R_z(\theta)R_z(-\phi) = R_z(\theta)$ and so we get the same vector if we rotate about z no matter what we pick for the $x - y$ orientation.

3. If I have a set of arbitrary basis vectors (x', y', z'), where you have the (x, y, z) representation of each basis vector, write down the matrix that converts a vector in the (x, y, z) basis to one in the (x', y', z') basis, and vice-versa. How are these two matrices related? I suggest you write a function

   ```
   genbasis(xyz)
   ```

   that returns *a* valid matrix of basis vectors where xyz is one of the principal axes (a reminder that this isn't unique so you'll have to make a choice. The results of Problem 2 show that this choice won't matter so just do something easy.)

   for answers to questions 3,4,5,6, please see python code pset3_q3456_ans.py

4. We're now ready to work out an rotation matrix in 3D about an arbitrary axis. Start by picking a random rotation axis in the $\hat{n}$, direction. Start with random θ, ϕ and convert that to a vector in (x, y, z) coordinates. Then generate a pair of basis vectors to make an orthogonal space (hint - if you take the cross product of two vectors, you get something

that's perpendicular to both, so e.g. $\hat{z} \times \hat{n}$ will be perpenciular to both $\hat{z}$ and $\hat{n}$). Don't forget that basis vectors must have unit length. Now write out the matrices that convert from (x, y, z) to the $\hat{n}$ basis, rotate by an angle γ in that basis, then convert back to (x, y, z). I suggest you do this on a computer, and write a function

```
genrot(xyz,gamma)
```

that generates the individual matrices, multiplies them together, and returns the final rotation matrix in the (x, y, z) coordinate system. What is the rotation matrix for $\theta = \pi/4$, $\phi = \pi/6$, and gamma=0.01?

5. Pretend, in a shocking turn of events, the Earth's crust gets rotated so that the Royal Greenwich Observatory, England (current latitude 51.476852, longitude=0.000) moves to the north pole. What are Montreal's new latitude and longitude? (current: 45.50884, -73.58781). Verify that the current angle between Montreal and Greenwich corresponds to Montreal's new latitude (as it must, because the distance to the pole depends only on latitude, not longitude, and your rotation matrix had better not change relative angles between vectors). Hint - think what axis we have to be rotating about to move Greenwich to the North pole. (note - the usual definition of θ is the angle from the north pole, while latitude is relative to the equator, so the distance from the pole is 90-latitude, not latitude directly).

6. Use your the results (and code!) to show numerically that the rotation commutation relations hold in an arbitrary coordinate system. If you haven't done this sort of thing before, you can show that as you make gamma smaller, the largest term in the error shrinks faster than the largest term in the commutator. I found gammas in the range of 0.01-0.001 were sufficient to show this clearly.

7. Bonus: As I said in class, extra credit to anyone who figures out a physical reason why if we write a rotation matrix as

$$\exp(i\delta)\begin{bmatrix} 1 & 0 \\ 0 & exp(i\theta) \end{bmatrix}$$

that δ must equal $-\theta/2$.