PHYS 357 Pset 2. Due 11:59 PM Thursday Sep. 19

1. Solve for the eigenvalues of a 2x2 matrix A in terms of the trace and determinant of A. Use this expression to show that the trace is the sum of the eigenvalues.

   If you need a hint to get started, we know that

   $$\mathrm{A}v = \lambda v$$

   so

   $$\mathrm{A} - \lambda \mathrm{I} v = 0$$

   For that to be true for non-zero v, $\mathrm{A} - \lambda\mathrm{I}$ must be singular, so its determinant must be zero.

2. Explicitly show using the definition of matrix multiplication that $(\mathrm{AB})^\dagger = \mathrm{B}^\dagger \mathrm{A}^\dagger$

3. Townsend 2.8

4. A) Working in the $z-$basis, express the projection operators $|{+y}\rangle\langle{+y}|$ and $|{-y}\rangle\langle{-y}|$ as 2x2 matrices.

   B) Show that the $+y$ projection matrix times an arbitrary vector (a, b) outputs a vector that is proportional to $|{+y}\rangle$ (*i.e.* it comes out as (c, ic) for some value c). Show the same for the $-y$.

5. A) For an arbitrary state $|{+n}\rangle, |{-n}\rangle$, write down the 2x2 projection operators in the $z-$basis. As a reminder, you can look at Townsend problem 1.3 for the state in an arbitrary direction.

   B) Show that the sum of these two matrices is the identity matrix. We expect this because the $|{+n}\rangle$ component of a state plus the $|{-n}\rangle$ component must give us the state we started with.

6. A) Work out the angular momentum operators J_x, J_y in the $z-$basis. Verify that they are Hermitian. If you want to do this on a computer, that's fine, but include the (very short!) code you used to generate them, and comment what you are doing.

   B) Work out the angular momentum operators J_x, J_y, J_z in the $|{\pm y}\rangle$ basis. Again, verify that they are Hermitian.

7. Work our the $\pi/2$ rotation matrix about the $y-$axis in the $|\pm z\rangle$-basis. Do this two ways - first by writing down what this matrix has to do to the $|+x\rangle$ and $|+z\rangle$ states. Then by combining the matrices that turn a state represented in the $\pm z$-basis into the $\pm y$-basis, the matrix that rotates about its own axis (the rotation about $|+n\rangle$ represented in the $|\pm n\rangle$ basis can't depend on $|n\rangle$), and the matrix that converts states in the $|\pm y\rangle$ back into the $|\pm z\rangle$ basis. Show that these matrices are the same, possibly up to an overall phase factor.

Audréanne Bernier (261100643)

Pset 2

1. Solve for the eigenvalues of a 2x2 matrix A in terms of the trace and determinant of A. Use this expression to show that the trace is the sum of the eigenvalues.

If you need a hint to get started, we know that

$$\mathrm{A}v = \lambda v$$

so

$$\mathrm{A} - \lambda \mathrm{I} v = 0$$

For that to be true for non-zero v, $\mathrm{A} - \lambda\mathrm{I}$ must be singular, so its determinant must be zero.

$Av = \lambda v \longrightarrow (A - \lambda I)v = 0$

Non-trivial solutions if $A - \lambda I$ is not invertible

$\Rightarrow \det(A - \lambda I) = 0$

$\det\left[\begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\right] = 0$

$\det\begin{pmatrix} a-\lambda & b \\ c & d-\lambda \end{pmatrix} = 0$

$(a-\lambda)(d-\lambda) - bc = 0$

$ad - \lambda a - \lambda d + \lambda^2 - bc = 0$

$\det A + \lambda^2 - \lambda\underbrace{(a+d)}_{trA} = 0$

$$\lambda = \frac{trA \pm \sqrt{(trA)^2 - 4\det A}}{2}$$

$\Sigma\lambda_i = \frac{trA}{2} + \cancel{\frac{\sqrt{(trA)^2 - 4\det A}}{2}} + \frac{trA}{2} - \cancel{\frac{\sqrt{(trA)^2 - 4\det A}}{2}}$

$\Sigma\lambda_i = trA$

2. Explicitly show using the definition of matrix multiplication that $(AB)^\dagger = B^\dagger A^\dagger$

let $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$, $B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$

$AB = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$ where $c_{ij} = \sum_k a_{ik} b_{kj}$

so $(AB)^\dagger = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}^\dagger = \overline{\begin{pmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \end{pmatrix}} = \begin{pmatrix} c_{11}^* & c_{21}^* \\ c_{12}^* & c_{22}^* \end{pmatrix}$ where $c_{ij}^* = \sum \underbrace{a_{ik}^* b_{kj}^*}_{\text{elements of } A^* B^*}$

$$= \begin{pmatrix} a_{11}^* b_{11}^* + a_{12}^* b_{21}^* & a_{21}^* b_{12}^* + a_{22}^* b_{22}^* \\ a_{11}^* b_{12}^* + a_{12}^* b_{21}^* & \cdots \end{pmatrix}$$

$$= \begin{pmatrix} b_{11}^* & b_{21}^* \\ b_{12}^* & b_{22}^* \end{pmatrix} \begin{pmatrix} a_{11}^* & a_{21}^* \\ a_{12}^* & a_{22}^* \end{pmatrix}$$

$$= \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}^{*T} \begin{pmatrix} a_{11}^* & a_{12}^* \\ a_{21}^* & a_{22}^* \end{pmatrix}^T$$

$$= B^\dagger A^\dagger$$

Q3

2.8. The column vector representing the state $|\psi\rangle$ is given by

$$|\psi\rangle \xrightarrow[S_z \text{ basis}]{} \frac{1}{\sqrt{5}}\begin{pmatrix} i \\ 2 \end{pmatrix}$$

Using matrix mechanics, show that $|\psi\rangle$ is properly normalized and calculate the probability that a measurement of S_x yields $\hbar/2$. Also determine the probability that a measurement of S_y yields $\hbar/2$.

① $\langle\psi|\psi\rangle = \frac{1}{\sqrt{5}}(-i \;\; 2)\frac{1}{\sqrt{5}}\begin{pmatrix} i \\ 2 \end{pmatrix} = \frac{1}{5}\cdot(1+4) = 1$ Normalized!

② $P(S_x = \hbar/2) = |\langle +x|\psi\rangle|^2 = \left|\frac{1}{\sqrt{2}}(1 \;\; 1)\frac{1}{\sqrt{5}}\begin{pmatrix} i \\ 2 \end{pmatrix}\right|^2 = \left|\frac{1}{\sqrt{10}}(i+2)\right|^2$

$= \frac{1}{10}(\sqrt{4+1})^2$

$= \frac{1}{10}\cdot 5$

$P(S_x = \hbar/2) = \frac{1}{2}$

③ $P(S_y = \hbar/2) = |\langle +y|\psi\rangle|^2 = \left|\frac{1}{\sqrt{2}}(1 \;\; -i)\frac{1}{\sqrt{5}}\begin{pmatrix} i \\ 2 \end{pmatrix}\right|^2 = \frac{1}{10}|(i-2i)|^2$

$P(S_y = \hbar/2) = \frac{1}{10}$

4. ~~A)~~ Working in the $z-$basis, express the projection operators $|+y\rangle\langle+y|$ and $|-y\rangle\langle-y|$ as 2x2 matrices.

~~B)~~ Show that the $+y$ projection matrix times an arbitrary vector (a, b) outputs a vector that is proportional to $|+y\rangle$ (*i.e.* it comes out as (c, ic) for some value c). Show the same for the $-y$.

a) ① $|+y\rangle\langle+y| = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ i\end{pmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}1 & -i\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & -i\\ i & 1\end{pmatrix}$

② $|-y\rangle\langle-y| = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\ -i\end{pmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}1 & i\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix}$

b) ① $+y$ projection: $\frac{1}{2}\begin{pmatrix}1 & -i\\ i & 1\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix} = \frac{1}{2}\begin{pmatrix}a-ib\\ ai+b\end{pmatrix} = \frac{1}{2}\begin{pmatrix}a-ib\\ i(a-ib)\end{pmatrix} = \frac{1}{2}\begin{pmatrix}c\\ ic\end{pmatrix} \propto \begin{pmatrix}1\\ i\end{pmatrix}$

② $-y$ projection: $\frac{1}{2}\begin{pmatrix}1 & i\\ -i & 1\end{pmatrix}\begin{pmatrix}a\\ b\end{pmatrix} = \frac{1}{2}\begin{pmatrix}a+ib\\ -ai+b\end{pmatrix} = \frac{1}{2}\begin{pmatrix}a+ib\\ -i(a+ib)\end{pmatrix} = \frac{1}{2}\begin{pmatrix}c\\ -ic\end{pmatrix} \propto \begin{pmatrix}1\\ -i\end{pmatrix}$

5. A) For an arbitrary state $|+n\rangle, |-n\rangle$, write down the 2x2 projection operators in the z-basis. As a reminder, you can look at Townsend problem 1.3 for the state in an arbitrary direction.

B) Show that the sum of these two matrices is the identity matrix. We expect this because the $|+n\rangle$ component of a state plus the $|-n\rangle$ component must give us the state we started with.

a) ① $|\psi\rangle_z = \begin{pmatrix} |+n\rangle\langle+n| \\ +|-n\rangle\langle-n| \end{pmatrix} |\psi\rangle_z$

↳ projection operators

② $|+n\rangle = \cos\frac{\theta}{2}|+z\rangle + e^{i\varphi}\sin\frac{\theta}{2}|-z\rangle$

$\langle+n| = \cos\frac{\theta}{2}\langle+z| + e^{-i\varphi}\sin\frac{\theta}{2}\langle-z|$

$\Rightarrow \quad |+n\rangle = \begin{pmatrix} \cos\theta/2 \\ e^{i\varphi}\sin\theta/2 \end{pmatrix}$

$\langle+n| = (\cos\theta/2 \quad e^{-i\varphi}\sin\theta/2)$

So $|+n\rangle\langle+n| = \begin{pmatrix} \cos\theta/2 \\ e^{i\varphi}\sin\theta/2 \end{pmatrix} (\cos\theta/2 \quad e^{-i\varphi}\sin\theta/2)$

$= \begin{pmatrix} \cos^2\theta/2 & e^{-i\varphi}\cos\theta/2\sin\theta/2 \\ e^{i\varphi}\cos\theta/2\sin\theta/2 & \sin^2\theta/2 \end{pmatrix}$

③ $|-n\rangle = \begin{pmatrix} \sin\theta/2 \\ -e^{i\varphi}\cos\theta/2 \end{pmatrix}$

$\langle-n| = (\sin\theta/2 \quad -e^{-i\varphi}\cos\theta/2)$

$|-n\rangle\langle-n| = \begin{pmatrix} \sin\theta/2 \\ -e^{i\varphi}\cos\theta/2 \end{pmatrix} (\sin\theta/2 \quad -e^{-i\varphi}\cos\theta/2)$

$= \begin{pmatrix} \sin^2\theta/2 & -e^{-i\varphi}\cos\theta/2\sin\theta/2 \\ -e^{i\varphi}\cos\theta/2\sin\theta/2 & \cos^2\theta/2 \end{pmatrix}$

④ $\begin{pmatrix} |+n\rangle\langle+n| \\ +|-n\rangle\langle-n| \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

6. ~~A~~) Work out the angular momentum operators J_x, J_y in the $z-$basis. Verify that they are Hermitian. If you want to do this on a computer, that's fine, but include the (very short!) code you used to generate them, and comment what you are doing.

~~B~~) Work out the angular momentum operators J_x, J_y, J_z in the $|\pm y\rangle$ basis. Again, verify that they are Hermitian.

a) ① $J_{xz} = R^{\dagger}_{z\to x} J_{xx} R_{z\to x}$

J_x in z basis

also done in code

② we know $J_{xx} = \begin{pmatrix} \hbar/2 & 0 \\ 0 & -\hbar/2 \end{pmatrix}$

$$R_{z\to x} = \begin{pmatrix} |+x\rangle_x \langle +x|_z \\ +|-x\rangle_x\langle -x|_z \end{pmatrix} = \begin{matrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} (1 \ \ 1) \\ + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \frac{1}{\sqrt{2}} (1 \ \ -1) \end{matrix} = \frac{1}{\sqrt{2}} \left[\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix} \right] = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$R^{\dagger}_{z\to x} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

③ so we have $J_{xz} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \hbar/2 & 0 \\ 0 & -\hbar/2 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

$$= \frac{1}{2} \begin{pmatrix} \hbar/2 & -\hbar/2 \\ \hbar/2 & \hbar/2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$J_{xz} = \begin{pmatrix} 0 & \hbar/2 \\ \hbar/2 & 0 \end{pmatrix}$$

④ $J_{yz} = R^{\dagger}_{z\to y} J_{yy} R_{z\to y}$

with $R_{z\to y} = \begin{pmatrix} |+y\rangle_y \langle +y|_z \\ +|-y\rangle_y\langle +y|_z \end{pmatrix}$ & $J_{yy} = \begin{pmatrix} \hbar/2 & 0 \\ 0 & -\hbar/2 \end{pmatrix}$

$\Rightarrow J_{yz} = \begin{pmatrix} 0 & -\frac{\hbar}{2} i \\ \frac{\hbar}{2} i & 0 \end{pmatrix}$ see code

⑤ Hermitian $\to J = J^{\dagger}$

⑥ code:

```python
import numpy as np
import scipy as sp

# constants
hbar = sp.constants.hbar

# state in its own basis
np_in_n = np.array([1,0])
nm_in_n = np.array([0,1])
Jnn = hbar/2*np.array([[1,0],[0,-1]])  # Jn in n basis (n=x,y,z)


# 6 a)
print('6 a)')
# x kets
xp_in_z = np.array([1,1])/np.sqrt(2)
xm_in_z = np.array([1,-1])/np.sqrt(2)
# y kets
yp_in_z = np.array([1,1j])/np.sqrt(2)
ym_in_z = np.array([1,-1j])/np.sqrt(2)

# Rotation matrices
R_ztox = np.outer(np_in_n,np.conj(xp_in_z.T)) + np.outer(nm_in_n,np.conj(xm_in_z.T))  # z to x basis
# print('Rotation z to x:\n', R_ztox)
# print(f'Checking Rz-x:\n {R_ztox@xp_in_z}, {xp_in_x}, {R_ztox@xm_in_z},{xm_in_x}')
R_ztoy = np.outer(np_in_n,np.conj(yp_in_z.T)) + np.outer(nm_in_n,np.conj(ym_in_z.T))  # z to y basis
# print('Rotation z to y:\n', R_ztoy)
# print(f'Checking Rz-y:\n {R_ztoy@yp_in_z},{yp_in_y}, {R_ztoy@ym_in_z}, {ym_in_y}')

# Momentum operators
Jxz = np.conj(R_ztox.T) @ Jnn @ R_ztox  # Jx in z basis
Jyz = np.conj(R_ztoy.T) @ Jnn @ R_ztoy  # Jy in z basis
print('Jxz:\n', Jxz)
print('Jyz:\n', Jyz)

# Hermitian check
print(f'Checking hermitian:\n(Jxz)\n {np.conj(Jxz.T)==Jxz}\n(Jyz)\n {np.conj(Jyz.T)==Jyz}')
```

```
6 a)
Jxz:
 [[-1.08453934e-51  5.27285909e-35]
 [ 5.27285909e-35 -1.08453934e-51]]
Jyz:
 [[0.+0.00000000e+00j 0.-5.27285909e-35j]
 [0.+5.27285909e-35j 0.+0.00000000e+00j]]
Checking hermitian:
(Jxz)
 [[ True  True]
 [ True  True]]
(Jyz)
 [[ True  True]
 [ True  True]]
```

b) ① $J_{yy} = \begin{pmatrix} \hbar/2 & 0 \\ 0 & -\hbar/2 \end{pmatrix}$

② $J_{xy} = R^{\dagger}_{y\to x} J_{xx} R_{y\to x} = \begin{pmatrix} 0 & -\frac{\hbar}{2}i \\ \frac{\hbar}{2}i & 0 \end{pmatrix}$

with $R_{y\to x} = \begin{pmatrix} |{+x}\rangle_x \langle{+x}|_y \\ + |{-x}\rangle_x \langle{-x}|_y \end{pmatrix}$

$J_{xx} = \begin{pmatrix} \hbar/2 & 0 \\ 0 & -\hbar/2 \end{pmatrix}$

③ $J_{zy} = R^{\dagger}_{y\to z} J_{zz} R_{y\to z} = \begin{pmatrix} 0 & \hbar/2 \\ \hbar/2 & 0 \end{pmatrix}$

with $R_{y\to z} = R^{\dagger}_{z\to y}$

code:

```python
# 6 b)
print('\n6 b)')
# converting the x and z to the y basis
# x kets
xp_in_y = R_ztoy@xp_in_z
xm_in_y = R_ztoy@xm_in_z
# z kets
zp_in_y = R_ztoy@np_in_n
zm_in_y = R_ztoy@nm_in_n

# Rotation matrices
R_ytox = np.outer(np_in_n,np.conj(xp_in_y.T)) + np.outer(nm_in_n,np.conj(xm_in_y.T))  # y to x basis
R_ytoz = np.conj(R_ztoy.T)  # y to z basis  (inverse of R_ztoy)

# Momentum operators
# Momentum operators
Jxy = np.conj(R_ytox.T) @ Jnn @ R_ytox  # Jx in y basis
Jzy = np.conj(R_ytoz.T) @ Jnn @ R_ytoz  # Jz in y basis
print('Jxy:\n', Jxy)
print('Jzy:\n', Jzy)

# Hermitian check
print(f'Checking hermitian:\n(Jxy)\n {np.conj(Jxy.T)}\n(Jzy)\n {np.conj(Jzy.T)==Jzy}')
```

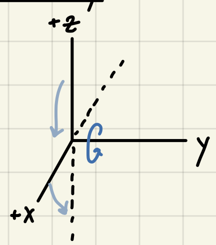
```
6 b)
Jxy:
 [[0.+5.08520145e-52j 0.-5.27285909e-35j]
 [0.+5.27285909e-35j 0.-5.08520145e-52j]]
Jzy:
 [[0.00000000e+00+0.j 5.27285909e-35+0.j]
 [5.27285909e-35+0.j 0.00000000e+00+0.j]]
Checking hermitian:
(Jxy)
 [[0.-5.08520145e-52j 0.-5.27285909e-35j]
 [0.+5.27285909e-35j 0.+5.08520145e-52j]]
(Jzy)
 [[ True  True]
 [ True  True]]
```

0

0

7. Work our the $\pi/2$ rotation matrix about the $y-$axis in the $|\pm z\rangle$-basis. Do this two ways - first by writing down what this matrix has to do to the $|+x\rangle$ and $|+z\rangle$ states. Then by combining the matrices that turn a state represented in the $\pm z$-basis into the $\pm y$-basis, the matrix that rotates about its own axis (the rotation about $|+n\rangle$ represented in the $|\pm n\rangle$ basis can't depend on $|n\rangle$), and the matrix that converts states in the $|\pm y\rangle$ back into the $|\pm z\rangle$ basis. Show that these matrices are the same, possibly up to an overall phase factor.

first way



In the $|\pm z\rangle$ basis, we have

$$|+x\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} \qquad |+y\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\i\end{pmatrix}$$

$$|-x\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} \qquad |-y\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-i\end{pmatrix}$$

① $R_z(\frac{\pi}{2}\hat{j})|+x\rangle = |-z\rangle$

$$R\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$$

② $R_z(\frac{\pi}{2}\hat{j})|+z\rangle = |+x\rangle$

$$R\begin{pmatrix}1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

combine...

$$\begin{pmatrix}0 & 1/\sqrt{2}\\ 1 & 1/\sqrt{2}\end{pmatrix} = R\begin{pmatrix}1/\sqrt{2} & 1\\ 1/\sqrt{2} & 0\end{pmatrix}$$

$$R = \begin{pmatrix}0 & 1/\sqrt{2}\\ 1 & 1/\sqrt{2}\end{pmatrix}\begin{pmatrix}1/\sqrt{2} & 1\\ 1/\sqrt{2} & 0\end{pmatrix}^{-1}$$

$$R = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}$$ computed w/ code

second way

Need: $R_{z\to y} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -i\\ 1 & i\end{pmatrix}$, $R(\theta) = \begin{pmatrix}e^{-i\theta/2} & 0\\ 0 & e^{i\theta/2}\end{pmatrix}$, $R_{y\to z} = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ i & -i\end{pmatrix}$

so we have $R_z = R_{y\to z}\, R_y(\pi/2)\, R_{z\to y}$

$$= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -i\\ 1 & i\end{pmatrix}\begin{pmatrix}e^{i\theta/2} & 0\\ 0 & e^{i\theta/2}\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ i & -i\end{pmatrix}$$

$$R_z = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & -1\\ 1 & 1\end{pmatrix}$$ computed with code

```python
import numpy as np

# state in its own basis
np_in_n = np.array([1,0])
nm_in_n = np.array([0,1])

# x kets
xp_in_z = np.array([1,1])/np.sqrt(2)
xm_in_z = np.array([1,-1])/np.sqrt(2)
# y kets
yp_in_z = np.array([1,1j])/np.sqrt(2)
ym_in_z = np.array([1,-1j])/np.sqrt(2)


# First way
print("First Way:")
left_matrix = np.zeros([2,2])
left_matrix[:,0]=nm_in_n
left_matrix[:,1]=xp_in_z
right_matrix = np.zeros([2,2])
right_matrix[:,0]=xp_in_z
right_matrix[:,1]=np_in_n

R_first_way = left_matrix @ np.linalg.inv(right_matrix)
print('R_first_way:\n', R_first_way)


# Second way
print("\nSecond Way:")
# Rotation matrices
R_ztoy = np.outer(np_in_n,np.conj(yp_in_z.T)) + np.outer(nm_in_n,np.conj(ym_in_z.T))  # z to y basis
R_ytoz = np.conj(R_ztoy.T)  # y to z basis  (inverse of R_ztoy)
def rotation_theta(theta):
    return np.array([[np.exp(-1j*theta/2), 0],[0, np.exp(1j*theta/2)]])  # rotation matrix for theta
R_theta = rotation_theta(np.pi/2)
# print(R_ztoy, '\n', R_ytoz)

R_second_way = R_ytoz @ R_theta @ R_ztoy
print('R_second_way:\n', R_second_way)
```

```
First Way:
R_first_way:
 [[ 0.70710678 -0.70710678]
 [ 0.70710678  0.70710678]]

Second Way:
R_second_way:
 [[ 0.70710678+0.j -0.70710678+0.j]
 [ 0.70710678+0.j  0.70710678+0.j]]
```