20191119—3.2_Basic_Operations—QuTip—Qobj()

February 3, 2020

0.1 20191119—3.2_Basic_Operations—QuTip—Qobj()

[1]:
```
from qutip import *
```

[2]:
```
import numpy as np
```

0.1.1 Quantum Object Class

Qobj()

[3]:
```
Qobj(4)
# Here the output is 0.0 without any input.....how.?
# based on output, here defined its dimension, or based on input..?
```

[3]:

Quantum object: dims = [[1], [1]], shape = (1, 1), type = bra

$$\begin{pmatrix} 4.0 \end{pmatrix}$$

[4]:
```
np.array(5).ndim
```

[4]: 0

0.1.2 1D array

[5]:
```
shape(np.array([]))
```

[5]: (0,)

[6]:
```
np.array([]).ndim
```

[6]: 1

[7]:
```
Qobj([[]])
# why its not executed but 2D quantum executed....what my mistake was.?
# Here my doubt is - In numpy we have the output even for a single braces but␣
↪not in Quantum ...why.?
# look above numpy example with dimension as well
```

[7]:
Quantum object: dims = [[1], [0]], shape = (1, 0), type = bra

$$\begin{pmatrix} \end{pmatrix}$$

0.1.3 2D array

[8]:
```
shape(np.array([[]]))
```

[8]: (1, 0)

[9]:
```
np.array([[]]).ndim
```

[9]: 2

[10]:
```
Qobj([[],[],[]])
```

[10]:
Quantum object: dims = [[3], [0]], shape = (3, 0), type = other

$$\begin{pmatrix} \\ \\ \end{pmatrix}$$

0.1.4 3D array

[11]:
```
shape(np.array([[[]]]))
```

[11]: (1, 1, 0)

[12]:
```
np.array([[[]]]).ndim
```

[12]: 3

[14]:
```
#Qobj([[[]]])
```

0.2 As per the output, EXECUTED in the above, what i understood is, QUANTUM consider UPTO 2D only. WHY.?

one more example using numpy

[15]:
```
a11=[[1,2,3],[4,5,6]]
#a11=np.asarray(a11)
a11=np.array(a11)
a11
```

[15]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

[16]:
```
print("Shape: ",shape(a11))
print("Dim: ",a11.ndim)
```

```
Shape:  (2, 3)
Dim:  2
```

Let's try the same input in Quantum

[17]:
```python
Qobj([[1,2,3],[4,5,6]])
```

[17]:

```
Quantum object: dims = [[2], [3]], shape = (2, 3), type = other
```

$$\begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \end{pmatrix}$$

[18]:
```python
Qobj([[1,],[2]])
```

[18]:

```
Quantum object: dims = [[2], [1]], shape = (2, 1), type = ket
```

$$\begin{pmatrix} 1.0 \\ 2.0 \end{pmatrix}$$

[19]:
```python
Qobj([[1,2,3],[4,5,6],[7,8,9]])
```

[19]:

```
Quantum object: dims = [[3], [3]], shape = (3, 3), type = oper, isherm = False
```

$$\begin{pmatrix} 1.0 & 2.0 & 3.0 \\ 4.0 & 5.0 & 6.0 \\ 7.0 & 8.0 & 9.0 \end{pmatrix}$$

Note - Both matrices are not square and my doubt is, for 2*1 matrix the type shows as KET, for 2*3 matrix the type is OTHER, but in the below 3*3 matrix type is OPER. Why.? #### if m=n for m*n matrix then type is "OPER"

i have to study more on these theoretically from quantum book.

0.2.1 I understood both Dimension & Shape but not type which is executed in the output and also type is changing in all 4 examples…what is that means - other, ket, oper, bra -.?

- `bra-ket are belongs to Dirac Notations which belongs to Hilbert space`………`i have to study on those.`

[20]:
```python
np.array([[7],[8],[9]]).ndim
```

[20]: 2

[21]:
```python
np.array([[7,8,9]]).ndim
```

[21]: 2

[22]:
```
Qobj(np.random.rand(4,4))
```

[22]:
```
Quantum object: dims = [[4], [4]], shape = (4, 4), type = oper, isherm = False
```

$$\begin{pmatrix} 0.891 & 0.867 & 0.932 & 0.229 \\ 0.746 & 0.194 & 0.379 & 0.786 \\ 0.750 & 0.542 & 0.018 & 0.375 \\ 0.006 & 0.499 & 0.835 & 0.599 \end{pmatrix}$$

———–End of this Qobj()———-

1 Next topic — Quantum STATES_Operators

[ ]:
```
```

[ ]:
```
```