

20191119—

## 3.2\_Basic\_Operations\_on\_Quantum\_STATES\_OPERATORs

February 3, 2020

### 1 20191119—QuTip\_States\_Operators

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

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

#### 1.0.1 States - Operators Examples

- Manually specifying the data for each quantum object is inefficient. Even more so when most objects correspond to commonly used types such as the ‘ladder operators’ of a harmonic oscillator, the ‘Pauli spin operators’ for a twolevel system, or ‘state vectors’ such as Fock states.
- Therefore, QuTiP includes predefined objects for a variety of states: please go to the Qutip documentary page no 14 ##### NOTE
- basis command is belongs to - Fock state vector -

#### command

- basis(N,#m)

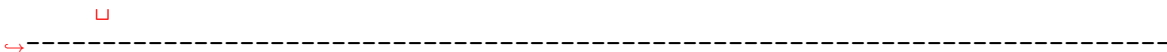
#### Input -

- N = the number of levels in Hilbert space
- =# means optional
- m = the level containing excitation (0, if m is not given)

#### NOTE:

- basis(N,#m)
  - Here ‘N’ should not be less than ‘m’
  - If ‘m’ is more than ‘N’ then - ValueError: basis vector index need to be in n <= N-1

```
[3]: basis()
```



```
TypeError                                Traceback (most recent call↳
↳last)
```

```
<ipython-input-3-e8e348c201bd> in <module>
----> 1 basis()
```

```
TypeError: basis() missing 1 required positional argument: 'N'
```

```
[4]: basis(0)
```

```
↳
↳-----
```

```
ValueError                                Traceback (most recent call↳
↳last)
```

```
<ipython-input-4-1f6e4b276558> in <module>
----> 1 basis(0)
```

```
~\Anaconda3\envs\qutip-env\lib\site-packages\qutip\states.py in basis(N,↳
↳n, offset)
    103
    104     if n - offset > (N - 1): # check if n is within bounds
--> 105         raise ValueError("basis vector index need to be in n <= N-1")
    106
    107     data = np.array([1], dtype=complex)
```

```
ValueError: basis vector index need to be in n <= N-1
```

```
[5]: basis(1)
```

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

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

```
[6]: basis(2)
```

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

$$\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

```
[7]: basis(3)
```

```
[7]: Quantum object: dims = [[3], [1]], shape = (3, 1), type = ket
```

$$\begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

```
[8]: basis(1,0)
```

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

$$(1.0)$$

```
[9]: basis(1,1)
```

```

↳ -----
ValueError                                Traceback (most recent call↳
↳ last)

<ipython-input-9-d19035ac5346> in <module>
----> 1 basis(1,1)

~\Anaconda3\envs\qutip-env\lib\site-packages\qutip\states.py in basis(N,↳
↳ n, offset)
    103
    104     if n - offset > (N - 1): # check if n is within bounds
--> 105         raise ValueError("basis vector index need to be in n <= N-1")
    106
    107     data = np.array([1], dtype=complex)

ValueError: basis vector index need to be in n <= N-1
```

```
[12]: basis(2,0)
```

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

$$\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

```
[13]: basis(2,1)
```

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

$$\begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}$$

```
[14]: basis(2,2)
```

```

      □
↳ -----

      ValueError                                Traceback (most recent call↳
↳ last)

      <ipython-input-14-871c56c10f7b> in <module>
      ----> 1 basis(2,2)

      ~\Anaconda3\envs\qutip-env\lib\site-packages\qutip\states.py in basis(N,↳
↳ n, offset)
      103
      104     if n - offset > (N - 1): # check if n is within bounds
      --> 105         raise ValueError("basis vector index need to be in n <= N-1")
      106
      107     data = np.array([1], dtype=complex)

      ValueError: basis vector index need to be in n <= N-1
```

```
[15]: basis(3,0)
```

```
[15]: Quantum object: dims = [[3], [1]], shape = (3, 1), type = ket
```

$$\begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

```
[16]: basis(3,1)
```

```
[16]: Quantum object: dims = [[3], [1]], shape = (3, 1), type = ket
```

$$\begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

```
[17]: basis(3,2)
```

```
[17]: Quantum object: dims = [[3], [1]], shape = (3, 1), type = ket
```

$$\begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}$$

```
[18]: basis(3,3)
```

```

      □
↳ -----

      ValueError                                Traceback (most recent call↳
↳ last)

      <ipython-input-18-852c2207af92> in <module>
      ----> 1 basis(3,3)

      ~\Anaconda3\envs\qutip-env\lib\site-packages\qutip\states.py in basis(N,↳
↳ n, offset)
      103
      104     if n - offset > (N - 1): # check if n is within bounds
--> 105         raise ValueError("basis vector index need to be in n <= N-1")
      106
      107     data = np.array([1], dtype=complex)

      ValueError: basis vector index need to be in n <= N-1
```

```
[19]: basis(5,2)
```

```
[19]: Quantum object: dims = [[5], [1]], shape = (5, 1), type = ket
```

$$\begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

```
[20]: basis(7,4)
```

```
[20]:
```

Quantum object: `dims = [[7], [1]], shape = (7, 1), type = ket`

$$\begin{pmatrix} 0.0 \\ 0.0 \\ 0.0 \\ 0.0 \\ 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

## 1.1 NOTE - something different but interesting with more confusion

- Let me try to understand first and later i will try to do on this

### 1.1.1 ~~—————END—————~~

[ ]:

[ ]: