Density cube

Too Hon Lin

June 2018

1 Introduction

Density cube is a mathematical model that view quantum mechanics as a subset of a more general probabilistic theory.

2 Functions

2.1 gendcube2()

Input: -
Output: density cube(list)
Description: gendcube2 generates random density cube for 2-level system. The hermiticity constraint and normalisation condition for a two-level system is given below:

$$\rho^*_{112} = \rho_{121} = \rho_{121} = \rho_{211}$$
$$\rho^*_{122} = \rho_{122} = \rho_{212} = \rho_{221}$$
$$\rho_{111} + \rho_{222} = 1$$
$$\rho_{iii} \geq 0, i = 1, 2$$

There are 3 independent real parameters: ρ_{111},$\rho_{112}$and ρ_{122}. Thus, any two-level density cube must be in the following form:

$$\rho = \left\{ \begin{bmatrix} \rho_{111} & \rho_{112} \\ \rho_{112} & \rho_{122} \end{bmatrix}, \begin{bmatrix} \rho_{112} & \rho_{122} \\ \rho_{122} & 1-\rho_{111} \end{bmatrix} \right\}$$

Thus, this function first creates 2 2x2 matrices, where $mat1 = \begin{bmatrix} \rho_{111} & \rho_{112} \\ \rho_{112} & \rho_{122} \end{bmatrix}$ and $mat2 = \begin{bmatrix} \rho_{112} & \rho_{122} \\ \rho_{122} & 1-\rho_{111} \end{bmatrix}$. Then, ρ_{111}, ρ_{112} and ρ_{122} are generated using random.uniform()function to generate pseudo-random float between 0 and 1.The two matrices are then concatenated into a multidimensional list dcube.

2.2 qm2dc(mat)

Input: mat(matrix)
Output: dcube(array)
Description: qm2dc(mat) takes a matrix and its level as input, maps it to density cube, and return its respective density cube as output. The current function can only take 2×2 or 3×3 matrices. The function will first calculate if the input matrix is a 2-level of 3-level system matrix using the len() function. Then, it generates an empty 2×2 or 3×3 density cube, depending on the level system. The function will map each elements in the matrix to their respective places in the density cube using the following transformation:

$$\rho_{iii} = \rho_{ii}$$

$$\rho_{iij} = \sqrt{\frac{2}{3}} Re\rho_{ij}, i < j$$

$$\rho_{ijj} = \sqrt{\frac{2}{3}} Im\rho_{ij}, i < j$$

2.3 pauli()

Input: -
Output: 4 pauli cubes(array)
Description: *pauli* generates 4 Pauli cubes.

2.4 getH(dcube,n)

Input: dcube(density cube), n(integer)
Output: density cube(array)
Description: getH takes dcube and n, level system of the density cube as input and return the conjugate transpose of the density cube.The conjugate transpose function must satisfy the hermiticity condition, where $\rho^* = \rho$. Thus, the following transformation is performed to obtain the "conjugate transpose" of a density cube.

1. If i=j=k, then $\rho^*_{iii} = \rho_{iii} \subseteq \mathbb{R}$,
2. If i=j or j=k or j=k, then $\rho^*_{iij} = \rho_{iji} = \rho_{jii} \subseteq \mathbb{R}$
3. If $i \neq j \neq k$, then $\rho^*_{ijk} = \rho_{jik} = \rho_{ikj} = \rho_{kji}$

2.5 ishermit(dcube)

Input: density cube(array)
Output: True/False (boolean)
Description: ishermit(dcube) takes density cube as input and return true if the density cube is hermitian, false if vice versa.

The hermiticity condition states that $\rho = \rho^*$. For a density cube to satisfy hermiticity, the following conditions follow:

$$\rho_{ijk} = \rho^*_{jik}$$

$$\rho_{ijk} = \rho^*_{kij}$$

$$\rho_{ijk} = \rho^*_{ikj}$$

From here, a few lemmas are derived:

1. If i=j=k, then $\rho^*_{iii} = \rho_{iii} \subseteq \mathbb{R}$,
2. If i=j or j=k or j=k, then $\rho^*_{iij} = \rho_{iji} = \rho_{jii} \subseteq \mathbb{R}$
3. If $i \neq j \neq k$, then $\rho^*_{ijk} = \rho_{jik} = \rho_{ikj} = \rho_{kji}$

2.6 dot(dcube1,dcube2)

Input:density cube(array)
Output: inprod(float)
Description:dot() takes 2 density cubes,$\rho 1$ and $\rho 2$ of same level system and perform inner product on the cubes. The inner product is defined as:

$$inprod = \Sigma_{ijk} \rho 1^*_{ijk} \rho 2_{ijk}$$

The probability of obtaining a certain state after a measurement can also be obtained using the same function. In density cube model, the probability,p is given by

$$p = \Sigma_{ijk} \rho^*_{ijk} \sigma_{ijk}$$

whereρ is the state density cube's element, and σ is the projective density cube's measurement.

2.7 ispure(dcube)

Input:density cube(array)
Output: True/False(boolean)
Description: If the state density cube is a pure state density cube, the function will return true and vice versa. Using the same definition as a density matrix, the dot product of a pure state density cube is 1.

2.8 isorth(dcube1,dcube2)

Input:density cube(array)
Output: True/False(boolean)
Description: If the state density cube is a pure state density cube, the function will return true and vice versa. Using the same definition as a density matrix, the dot product of a pure state density cube is 1.