qpyMatrix 参考文档

一、简要介绍 Brief introduction

这个库是一个简单的矩阵库，用于简单的量子计算。

This library is a simple matrix library for simple quantum computations.

在这个库中，我们使用二维数组作为矩阵的数据结构，这样在访问和打印矩阵时都比较方便。

In this library, we use a two-dimensional array as the data structure of the matrix, which makes it easier to access and print the matrix.

在这个库中，我们默认导入numpy库，并重命名为np。

In this library, we import the numpy library by default and rename it np

二、库主要内容 Main contents of the library

1. 类 Class

（1）矩阵类

在这个库中，包含一个简单的矩阵类，结构如下：

Included in this library is a simple matrix class with the following structure:

```
Matrix
    matrix:#一个二维数组，用于存放矩阵元素
    #A two-dimensional array of matrix elements.
    row:#矩阵的行数 Number of rows of the matrix
    col:#矩阵的列数 Number of columns of the matrix
```

类的初始化函数如下：

The initialization functions of the class are as follows:

```
def __init__(self,m,n,p=0)
```

m是行数，n是列数，p是初始值，不提供初始值则默认为0。

m is the number of rows, n is the number of columns, and p is the initial value, which defaults to 0 if not provided.

类中还有一个函数element，他的定义如下：

The class also has a function element who is defined as follows:

```
def element(self,i,j,p,rw=False)
```

i是横坐标，j是纵坐标，p是要修改的值，rw是控制变量（默认为False）。如果rw为True，则对相应元素进行修改，否则仅打印相应元素。

i is the horizontal coordinate, j is the vertical coordinate, p is the value to be modified, and rw is the control variable (default is False). If rw is True, the corresponding element is modified, otherwise only the corresponding element is printed.

运算符重载（2025-04-24更新）

在这个类中，我们重载了以下运算符：

```
+:  #将+重载为矩阵加法
*:  #将*重载为矩阵乘法
**: #将**重载为矩阵的张量积
```

运算符的定义如下：

```
def __add__(self,other):#重载加法+为矩阵加法
    return plus(self,other)
def __mul__(self,other):#重载乘法*为矩阵乘法
    return multiply(self,other)
def __pow__(self,other):#重载**为张量积
    return kron(self,other)
```

（2）量子程序类（2025-04-23更新）

这个库中包含一个量子程序类qProg，结构如下：

```
qProg:
    prog:用于存放量子线路的列表
```

类中定义了以下函数：

```
def __init__(self):#初始化量子线路
def add(self,qprog):#向线路中添加操作
def draw(self):#绘制线路图
def help(self):#关于操作帮助
def subMatrix(self):#计算线路的合并矩阵
```

①初始化函数 `__init__(self)`

为对象初始化一个量子线路prog。

②函数 `add(self,qprog)`

向量子线路中添加一个操作qprog，这个操作的格式为 `["op_1","op_2"]` 。暂时只支持两个比特的操作

③函数 `draw(self)`

将量子线路绘制成图像，在终端上打印。

④函数 `help(self)`

向终端打印量子线路门对应的符号。

⑤函数 `subMatrix(self)`

计算量子线路的等价矩阵并返回。

2. 普通预设函数 General Preset Functions

（1）矩阵加法 Matrix addition

定义如下：

Definitions are as follows:

```
def plus(A,B)
```

对两个矩阵进行相加，如果两个矩阵不满足矩阵加法法则，则打印“矩阵不符合矩阵加法法则”。

Adds two matrices and prints “Matrix does not satisfy the law of matrix addition” if the two matrices do not satisfy the law of matrix addition.

（2）矩阵乘法 Matrix multiplication

定义如下：

Definitions are as follows:

```
def multiply(A,B)
```

返回两个矩阵相乘的矩阵结果，如果两个矩阵不满足矩阵乘法法则，则打印“矩阵不符合矩阵乘法法则”。

Returns the matrix result of multiplying two matrices. If the two matrices do not satisfy the matrix multiplication rule, prints “Matrix does not satisfy the matrix multiplication rule”.

（3）计算矩阵行列式 Calculate the matrix determinant

定义如下：

Definitions are as follows:

```
def det(A)
```

返回矩阵的行列式结果，使用的是numpy中的numpy.linalg.det()函数。

Returns the determinant result of the matrix, using the numpy.linalg.det() function in numpy.

（4）求逆矩阵 Find the inverse matrix

定义如下：

Definitions are as follows:

```
def inverse(A)
```

返回一个矩阵的逆矩阵，如果矩阵不是方阵，则打印“矩阵不是方阵”。

Returns the inverse of a matrix, or prints “matrix not square” if the matrix is not square.

（5）计算转置矩阵 Calculate the transpose matrix

定义如下：

Definitions are as follows:

```
def T(A)
```

返回一个矩阵的转置矩阵。

Returns the transpose matrix of a matrix.

（6）构造对角矩阵 Construct diagonal matrices

定义如下：

Definitions are as follows:

```
def diag(n,p)
```

构造一个n*n的对角矩阵并返回，对角线元素值为p。

Constructs an n*n diagonal matrix and returns it, with diagonal element values p.

（7）计算矩阵的张量积 Calculate the tensor product of the matrix

定义如下：

Definitions are as follows:

```
def kron(A,B)
```

计算$A \otimes B$并返回。

Compute the tensor product of A and B and return.

（8）矩阵的数乘 （2025-04-24更新）

定义如下：

```
def cmul(c,A)
```

将矩阵A的每一个元素乘以c并返回

3. 预设量子门矩阵 Preset quantum gate matrix

在库里，我们定义了常用的量子门的矩阵形式，他们对应的符号如下：

In the library, we define the matrix form of commonly used quantum gates, and their corresponding notation is as follows:

```
I:单位矩阵(2025-04-23更新)
X:泡利X门(Pauli X-Gate)
Y:泡利Y门(Pauli Y-Gate)
Z:泡利Z门(Pauli Z-Gate)
H:阿达玛门(Hadamard Gate)
S:S门/相位门(S-Gate/Phase Gate)
qT:T门(T-Gate)
CNOT:CNOT门,低位比特为控制比特
(CNOT gate, the low bit is the control bit)
CNOT_H:CNOT门，高位比特为控制比特
(CNOT gate, the high bit is the control bit)
SWAP:SWAP门，交换两个比特的状态
(SWAP gate, which swaps the state of two bits)
iSWAP:SWAP门，增加一个相对相位，如果两个比特的状态不同，
则相位发生变化(SWAP gate, adding a relative phase,
if the state of the two bits is different, the phase changes)
CZ:控制Z门(Controlled Z-Gate)
CCNOT:Toffoli门当两个高位控制比特均为1时翻转低位比特
(Toffoli Gate, flips the low bit when both high control bits are 1)
```

4. 预设量子计算函数 Predefined quantum computing functions

（1）旋转X门 Revolving X-Gate

定义如下：

Definitions are as follows:

```
def RX(p)
```

返回旋转角度为p的旋转X门。

Returns a rotating X-gate with a rotation angle of p.

（2）旋转Y门 Revolving Y-Gate

定义如下：

Definitions are as follows:

```
def RY(p)
```

返回旋转角度为p的旋转Y门。

Returns a rotating Y-gate with a rotation angle of p.

（3）旋转Z门 Revolving Z-Gate

定义如下：

Definitions are as follows:

```
def RZ(p)
```

返回旋转角度为p的旋转Z门。

Returns a rotating Z-gate with a rotation angle of p.

（4）共轭转置 Conjugate transpose

定义如下：

Definitions are as follows:

```
def CT(A)
```

返回一个矩阵的共轭转置矩阵。

Returns the conjugate transpose matrix of a matrix.

（5）测量函数，m为测量基 Measurement function, m is the measurement base

定义如下：

Definitions are as follows:

```
def Measure(m)
```

返回一个测量基对应的测量矩阵。

Returns the measurement matrix corresponding to a measurement base.

（6）测量量子比特qubit在测量基m上的概率 Measure the probability of qubit qubit on the measurement basis m

定义如下：

Definitions are as follows:

```
def measure_p(m,qubit)
```

测量qubit对于测量基m的概率。

The probability of a measurement qubit for a measurement base m.

（7）计算单量子比特在布洛赫球上的z坐标(2025-04-24更新)

定义如下：

```
def bloch(v)
```

计算z坐标并返回

三、预设基向量及其矩阵

(2025-04-24更新)

我们定义了常用的基向量及其测量矩阵，定义如下：

```
v_zero=Matrix(2,1)#基态|0>
v_one=Matrix(2,1)#基态|1>
v_currect=Matrix(2,1)#基态|+>
v_minus=Matrix(2,1)#基态|->
v_i=Matrix(2,1)#基态|i>
v_minus_i=Matrix(2,1)#基态|-i>
M_0=v_zero*CT(v_zero)#|0>的测量基矩阵
M_1=v_one*CT(v_one)#|1>的测量基矩阵
M_current=v_currect*CT(v_currect)#|+>的测量基矩阵
M_minus=v_minus*CT(v_minus)#|->的测量基矩阵
M_i=v_i*CT(v_i)#|i>的测量基矩阵
M_minus_i=v_minus_i*CT(v_minus_i)#|-i>的测量基矩阵
```

四、结语 Concluding remarks

非常感谢你能够使用这个库!

Thank you so much for being able to use this library!

当然，你也可以下载库的源文件，并对他进行完善，让这个库不断壮大!

Of course, you can also download the source files of the library and refine him to keep the library growing!

五、注 Note

最后更新日期：2025-04-24

Last Updated: 2025-04-24