

---

# **Curso de Introducción a la Computación Cuántica**

**Javier Mas Solé**

**Feb 13, 2024**



# CONTENTS

<b>1</b>	<b>Formalismo Matemático</b>	<b>3</b>
1.1	Números Complejos . . . . .	3
1.2	Vectores . . . . .	20
1.3	Operadores . . . . .	30
1.4	Tensores . . . . .	66
1.5	Probabilidades . . . . .	79
<b>2</b>	<b>Fundamentos de Mecánica Cuántica</b>	<b>87</b>
2.1	Axiomas . . . . .	87
2.2	Medidas y valores esperados . . . . .	89
2.3	Evolución temporal . . . . .	92



Aquí vamos a recopilar una serie de commando útiles para JupyterBooks.

- Las reglas del juego
  - *Formalismo Matemático*
    - \* *Números Complejos*
    - \* *Vectores*
    - \* *Operadores*
    - \* *Tensores*
    - \* *Probabilidades*
  - *Fundamentos de Mecánica Cuántica*

Feb 13, 2024 | 17 words | 0 min read



## FORMALISMO MATEMÁTICO

Aquí vamos a explicar las herramientas matemáticas necesarias para la computación cuántica.

- *Números Complejos*
- *Vectores*
- *Operadores*
- *Tensores*
- *Probabilidades*

Feb 13, 2024 | 4 min read

### 1.1 Números Complejos

```
%run ../../macro_tQ.py
import sys
sys.path.append('../../')
import macro_tQ as tQ

import numpy as np
import scipy.linalg as la
from IPython.display import display,Markdown,Latex
import matplotlib.pyplot as plt
from qiskit.tools.visualization import array_to_latex
import copy
```

#### 1.1.1 Introducción

La mecánica cuántica guarda una relación muy estrecha con los números complejos. Para introducirlos, empezaremos recordando que un número real, al cuadrado, *siempre* da un resultado positivo

$$a^2 > 0$$

por ejemplo  $2^2 = 4$  pero también  $(-2)^2 = 4$ . Por eso, la raíz cuadrada de un número real *sólo* existe si dicho número es positivo.

Por ejemplo  $\sqrt{4} = \pm 2$ , mientras que

$$\sqrt{-4} = ?$$

**Pregunta:** ¿cómo podríamos definir la raíz cuadrada de un número real negativo?

**Respuesta:** para hacerlo, es necesario ampliar el conjunto de los números reales.

Definición: número  $i$

se postula la existencia de un nuevo número,  $i$ , que es la solución única de la ecuación

$$i^2 = -1$$

Equivalentemente podríamos haber requerido que  $i = \sqrt{-1}$ .

Con esto podemos ahora encontrar la raíz de cualquier número negativo. Por ejemplo,  $-4$

$$(2i)(2i) = 4i^2 = -4$$

y  $2i$  será la raíz buscada.

Con el número  $i$  se opera igual que con los números reales

$$i + i = 2i$$

$$i - i = 0$$

$$i + 2i = 3i$$

$$i^3 = i * i * i = i^2 * i = -i$$

$$\frac{i}{i} = 1$$

Observar que el *inverso multiplicativo*  $1/i$  también es  $-i$

$$\begin{aligned} \frac{i}{i} &= 1 \\ i(-i) &= -i^2 = 1 \end{aligned} \tag{1.1}$$

por tanto hay una identificación importante

$$-i = \frac{1}{i}$$

---

### Resumen

La solución al problema planteado consiste en extender el cuerpo de los números reales  $\mathbb{R}$  al de los complejos  $\mathbb{C}$  que, ahora, incluyen el número  $i$

---

## 1.1.2 Formas cartesiana y polar

### Forma Cartesiana

Un *número complejo*,  $z \in \mathbb{C}$ , se representa en *forma cartesiana* mediante dos números reales  $x, y \in \mathbb{R}$

$$z = x + iy \quad \text{donde} \quad \begin{cases} x & \text{es la parte real} \\ y & \text{es la parte imaginaria} \end{cases}$$

Un número complejo se representa en el **plano complejo**: la parte real en el *eje horizontal*, y la parte imaginaria en el *eje vertical*



## Números complejos en python:

En *python*, el número imaginario  $i$  se representa con la letra  $j$ .

Añadiendo  $+0j$  convertimos un float en un complex

```
print(isinstance(1+0j,complex))
```

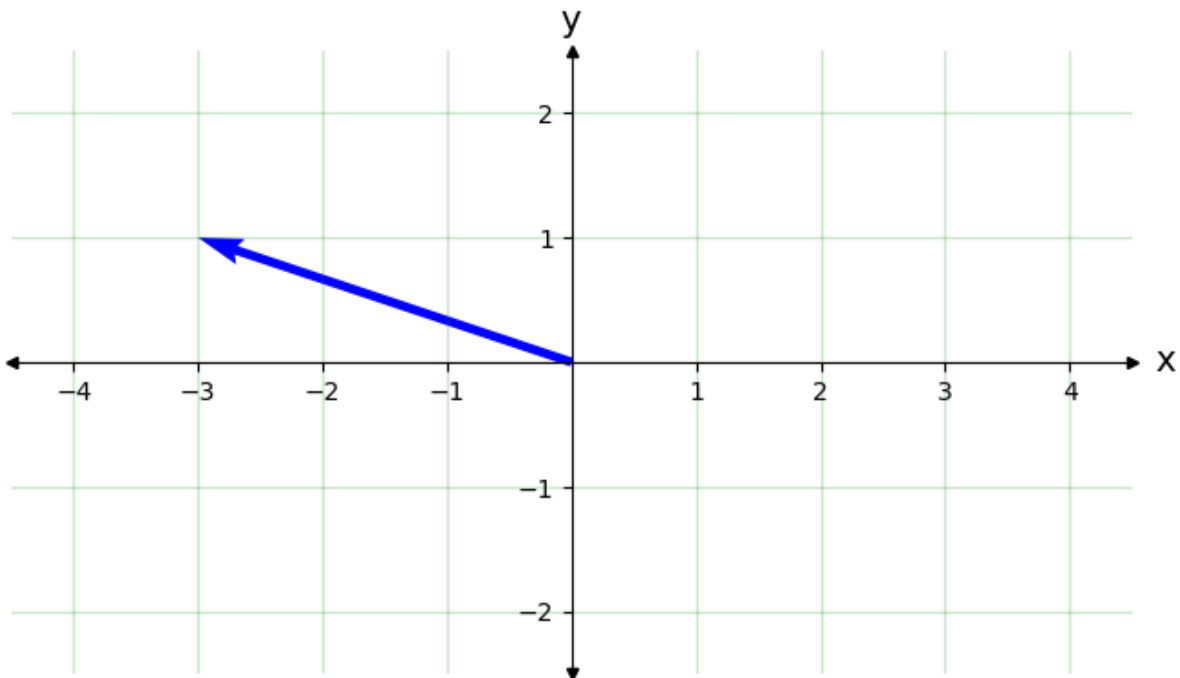
```
True
```

```
'''Forma cartesiana'''
z = -3 + 1j

'''Extraemos las partes real e imaginaria'''
x=z.real
y=z.imag
print('z=x+iy=',x + 1j*y)

''' Representación en el plano complejo '''
'''esta función está en archivo: macro_CURSO.py '''
''' Representación en el plano complejo '''
tQ.plot_2D_plane(left=-
    int(abs(x))-1,right=int(abs(x))+1,up=int(abs(y))+1,down=-int(abs(y))-1)
tQ.draw_vector(x,y,'b')
```

```
z=x+iy= (-3+1j)
```



## Forma Polar

Teorema: Fórmula de Euler

Dado un ángulo  $\theta \in (0, 2\pi)$  las dos expresiones siguientes son equivalentes

$$\cos \theta + i \sin \theta = e^{i\theta}$$

## Demostración

La demostración de la Fórmula de Euler viene de expandir ambos miembros en serie de Taylor en torno a  $\theta = 0$  y comprobar que ambas series son iguales

$$\begin{aligned} e^{i\theta} &= 1 + i\theta + \frac{1}{2}(i\theta)^2 + \frac{1}{3!}(i\theta)^3 + \dots \\ &= 1 - \frac{1}{2}\theta^2 + \dots + i\left(\theta - \frac{1}{3!}\theta^3 + \dots\right) \\ &= \cos \theta + i \sin \theta \end{aligned}$$

El número  $z = x + iy$  se puede representar en *forma polar*

$$z = \rho e^{i\theta} = \rho(\cos \theta + i \sin \theta)$$

de donde obtenemos las componentes cartesianas

$$x = \rho \cos \theta, \quad y = \rho \sin \theta$$

y por tanto  $\rho^2 = x^2 + y^2$

Los números reales  $\rho$  y  $\theta$  se denominan *módulo* y *fase*.

Las fases  $\theta$  y  $\theta + 2\pi$  representan el *mismo* número complejo

$$z = \rho e^{i\theta} = \rho e^{i(\theta+2\pi)}$$

Ello se debe a que las funciones  $\cos \theta$  y  $\sin \theta$  son periódicas

$$\sin \theta = \sin(\theta + 2\pi) \quad \cos \theta = \cos(\theta + 2\pi)$$

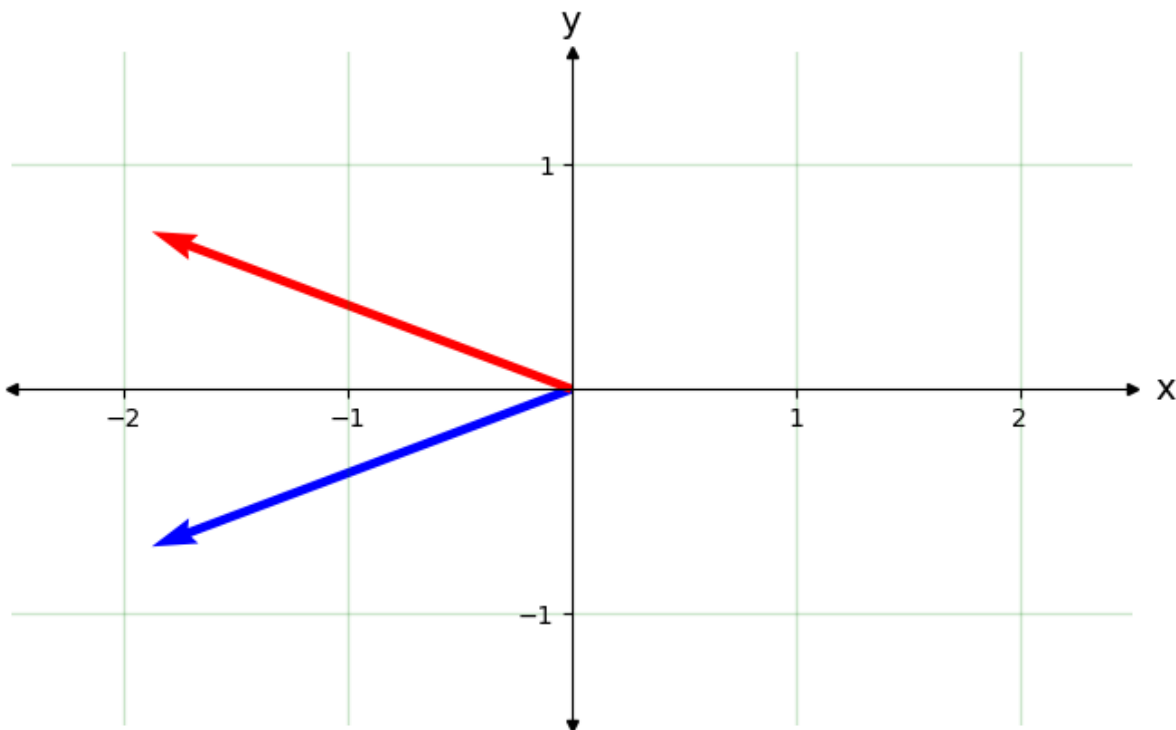
En forma polar la *conjugación compleja* se obtiene *cambiando el signo* de la fase

$$z = \rho e^{i\theta} \rightarrow z^* = \rho e^{-i\theta} = \rho \cos \theta - i \rho \sin \theta$$

```
'''Forma polar'''
r=2
th=3.5
z = r*np.exp(th*1j)
print(z)
x=z.real
y=z.imag
print('z=r exp(i th)=' ,np.round(x,3) + np.round(y,3)*1j)

''' Dibujamos en el plano complejo '''
tQ.plot_2D_plane(left=-
    int(abs(x))-1,right=int(abs(x))+1,up=int(abs(y))+1,down=-int(abs(y))-1)
tQ.draw_vector(z.real,z.imag)
tQ.draw_vector(z.conjugate().real,z.conjugate().imag,vcolor='r')
```

```
(-1.8729133745815927-0.7015664553792397j)
z=r exp(i th)= (-1.873-0.702j)
```



Usar la forma polar es útil en situaciones en las que aparecen productos y potencias del número  $i$

$$i = e^{i\pi/2} \quad (1.2)$$

$$-1 = e^{i\pi} \quad (1.3)$$

$$i = e^{3i\pi/2} = e^{-i\pi/2} \quad (1.4)$$

$$i^i = (e^{i\pi/2})^i = e^{i^2\pi/2} = e^{-\pi/2} \quad (1.5)$$

$$i^{2+i} = (e^{i\pi/2})^{(2+i)} = e^{i\pi/2(2+i)} = e^{i\pi} e^{-\pi/2} = -e^{-\pi/2} \quad (1.6)$$

### Ejercicio

Haz una lista de 10 números (fases)  $\theta_i$ ,  $i = 0, \dots, 9$ , equi-espaciadas entre 0 y  $2\pi$  y pinta los números complejos  $z_i = r \exp(i\theta_i)$  y sus complejos conjugados,  $z_i^*$ .

### Conversión entre formas cartesiana y polar

La conversión de la representación *polar a cartesiana* es muy sencilla gracias a las fórmula de Euler

$$z = re^{i\theta} = x + iy \quad \text{con} \quad \begin{cases} x = r \cos \theta \\ y = r \sin \theta \end{cases}$$

La conversión inversa, *de cartesiana a polar* es un poco más delicada. Formalmente sería

$$z = x + iy = re^{i\theta} \quad \text{con} \quad \begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \arctan(y/x) \end{cases}$$

A la hora de la verdad hay que fijar el signo de la función  $\arctan(y/x)$ . La siguiente función examina esto mirando a los signos de  $x$  y de  $y$  para saber en qué cuadrante estamos.

```
'Conversión de Cartesianas a Polares'
def cartes2polar(z):

    r = np.abs(z)
    y = z.imag
    x = z.real

    if r==0:
        print('el número 0+i0 no admite representación polar')
        th='indefinido'
    elif x==0 and y>0:
        th=np.pi/2
    elif x==0 and y<0:
        th=3*np.pi/2
    elif x>0 and y>=0:
        th=np.arctan(y/x)
    elif x<0 and y>=0:
        th=np.arctan(-y/x)+np.pi/2
    elif x<0 and y<0:
        th=np.arctan(y/x)+np.pi
    elif x>0 and y<0:
        th=np.arctan(-y/x)+3*np.pi/2.

    return r,th

#el signo correcto también se puede conseguir usando la funcion np.arctan2(x,y)
```

---

### Ejercicio

Calcula la forma polar del número complejo  $z = 4 + 3i$  a mano y verificalo con la función que acabamos de definir

---

```
'A la inversa no es necesario definir ninguna funcion,
↪ ya que numpy directamente escribe un numero complejo en forma cartesiana'
z = 3*np.exp(1j*0.5)
print(np.round(z,2))
```

```
(2.63+1.44j)
```

## Conjugacion compleja

Todo número complejo,  $z$ , lleva *asociado* otro,  $z^*$ , denominado el *complejo conjugado* que se obtiene cambiando  $i \rightarrow -i$

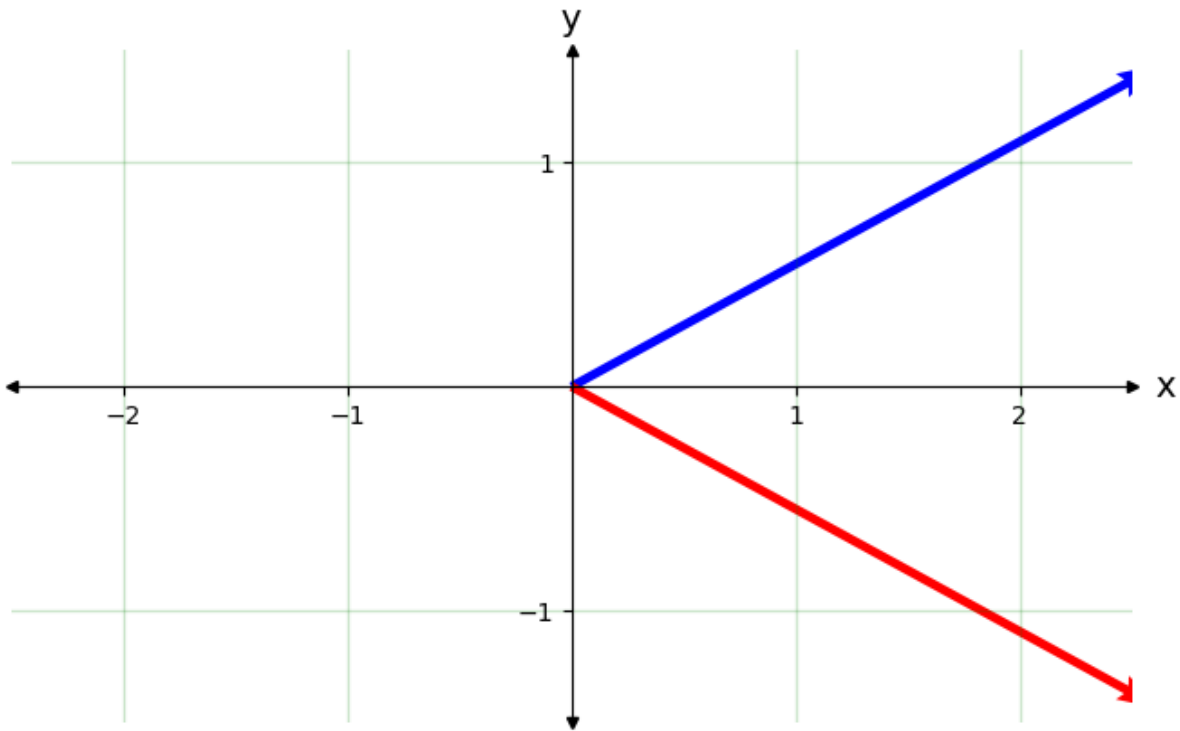
$$z = x + iy \quad \leftrightarrow \quad z^* = x - iy$$

Es evidente que  $(z^*)^* = z$ , es una *involución*.

```
'''Conjugacion compleja'''
zc = z.conjugate()
print('z*=(x+iy)*=',z.real +1j*zc.imag)

''' Representación en el plano complejo '''
tQ.plot_2D_plane(left=-
<int(abs(x))-1,right=int(abs(x))+1,up=int(abs(y))+1,down=-int(abs(y))-1)
tQ.draw_vector(z.real,z.imag,'b')
tQ.draw_vector(zc.real,zc.imag,vcolor='r')
```

```
z*=(x+iy)*= (2.6327476856711183-1.438276615812609j)
```



### 1.1.3 Operaciones básicas

Los números complejos  $\mathbb{C}$  forman una estructura matemática denominada *cuerpo*. Esto quiere decir que admiten dos operaciones *internas*: la **suma** y la **multiplicación**. Vamos a estudiarlas por separado

#### Suma

En representación *cartesiana* se suman las partes real e imaginaria por separado

$$(a + \imath b) + (c + \imath d) = (a + c) + \imath(b + d)$$

La resta es obvia, ya que  $a, b, c, d$  pueden ser números negativos.

```
'''Suma en cartesianas'''

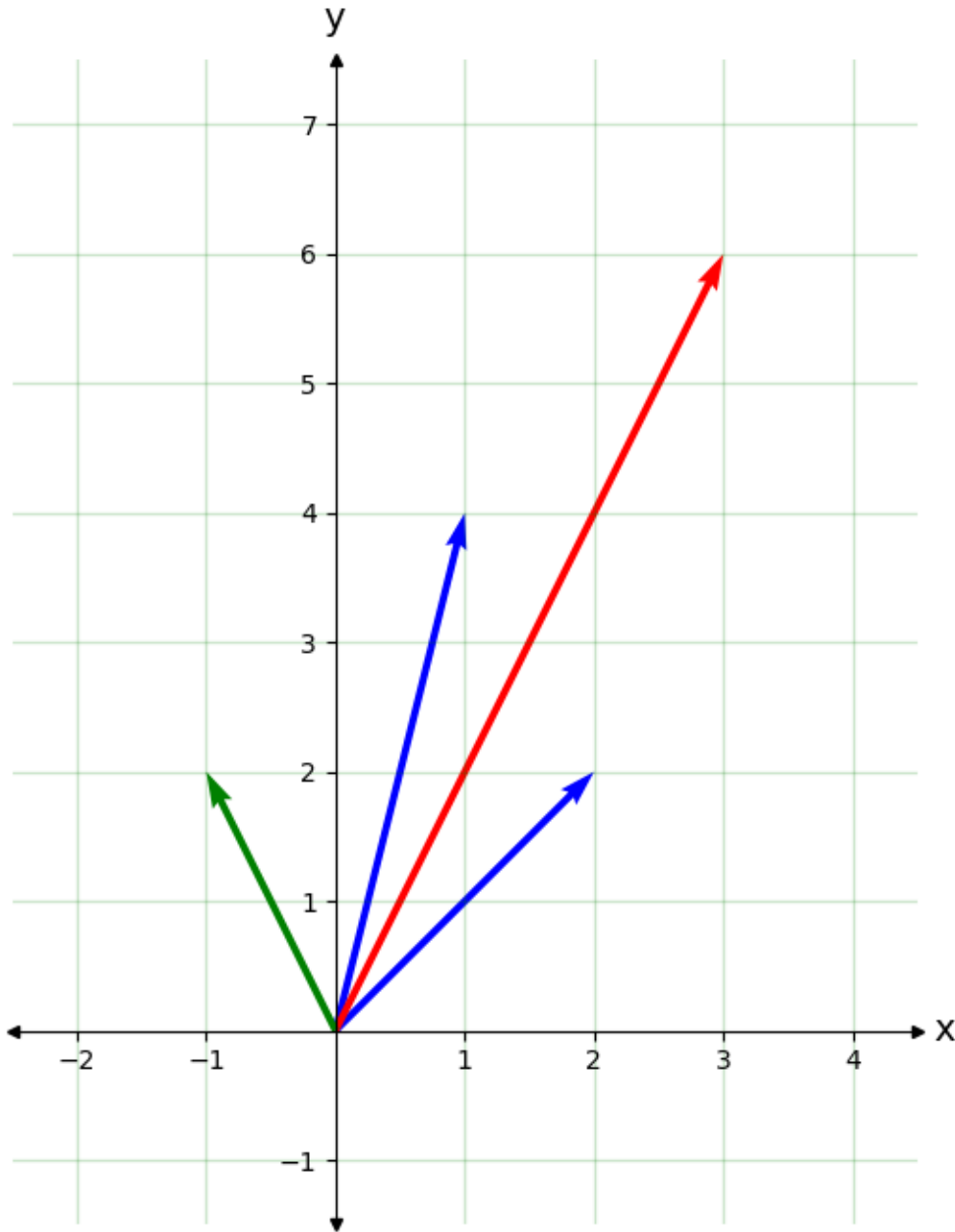
z1 = 1+4j
z2 = 2+2j

'''Suma y resta'''
zs = z1+z2
zd = z1-z2

print('z1+z2=',zs)
print('z1-z2=',zd)
print('*****')

tQ.plot_2D_plane(left=-
    2,right=4,up=7,down=-1) #cambiar las dimensiones para que encuadrar la figura
tQ.draw_vector(z1.real,z1.imag,'b')
tQ.draw_vector(z2.real,z2.imag,'b')
tQ.draw_vector(zs.real,zs.imag,vcolor='r')
tQ.draw_vector(zd.real,zd.imag,vcolor='g')
```

```
z1+z2= (3+6j)
z1-z2= (-1+2j)
*****
```



En *forma polar*, la suma de dos números complejos no admite ninguna simplificación, y deben transformarse primeramente a forma cartesiana, para sumarse.

$$z + w = \rho e^{i\theta} + \sigma e^{i\phi} = (\rho \cos \theta + \sigma \cos \phi) + i(\rho \sin \theta + \sigma \sin \phi)$$

```
'python directamente escribe un numero complejo en forma cartesiana'
z1 = 3*np.exp(1j*0.5)
z2 = 1*np.exp(-1j*0.7)

'''Suma y resta'''
zs = z1+z2
zd = z1-z2
```

(continues on next page)

(continued from previous page)

```

print('z1+z2=',np.round(zs,4))
print('z1-z2=',np.round(zd,4))
print('*****')

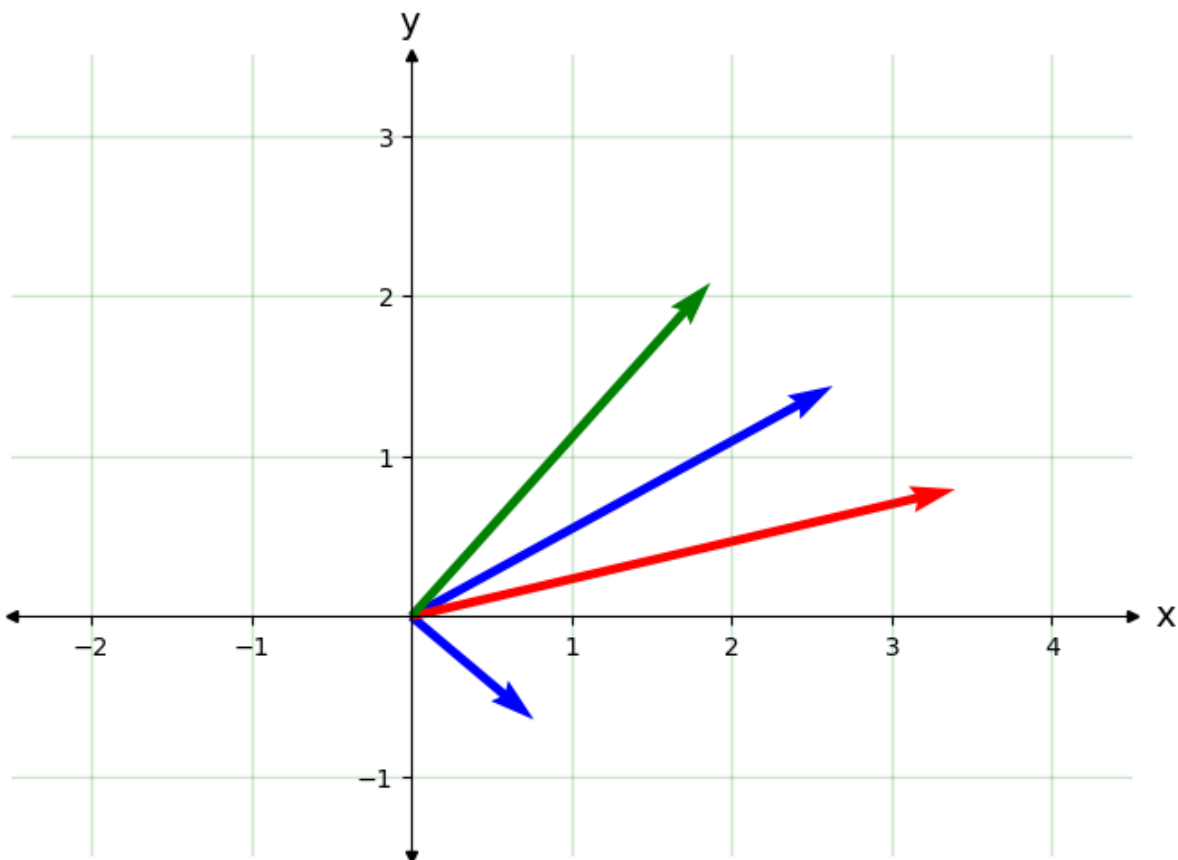
tQ.plot_2D_plane(left=-
    ↵2,right=4,up=3,down=-1) #cambiar las dimensiones para que encuadrar la figura
tQ.draw_vector(z1.real,z1.imag,'b')
tQ.draw_vector(z2.real,z2.imag,'b')
tQ.draw_vector(zs.real,zs.imag,vcolor='r')
tQ.draw_vector(zd.real,zd.imag,vcolor='g')

```

```

z1+z2= (3.3976+0.7941j)
z1-z2= (1.8679+2.0825j)
*****

```





## Multiplicación

En *forma cartesiana* la multiplicación es complicada, debiendo multiplicarse todos los factores entre sí, y teniendo en cuenta que  $i^2 = -1$

$$(a + ib)(c + id) = ac + aid + ibc + i^2bd = (ac - bd) + i(ad + bc)$$

Para hallar el producto de dos números complejos  $z = re^{i\theta}$  y  $w = se^{i\phi}$  escritos en forma polar, se multiplican los módulos y se suman las fases

$$zw = re^{i\theta}se^{i\phi} = rs e^{i(\theta+\phi)}$$

```
z1 = 3*np.exp(1j*0.5)
z2 = 1*np.exp(-1j*0.7)
'''Producto'''
print('z1*z2 = ', np.round(z1*z2,4))
print('z1**3 = ', np.round(pow(z1,6)))
print('*****')
```

```
z1*z2 = (2.9402-0.596j)
z1**3 = (-722+103j)
*****
```

## Valor absoluto

El cuadrado de un número real  $a \in \mathbb{R}$  es otro número real positivo  $a^2 > 0$ . Ello nos permite definir el valor absoluto  $|a| = \sqrt{a^2}$  que es el mismo para  $a$  y para  $-a$ .

Esto no sucede con un número complejo  $z$ . En efecto,

$$z^2 = x^2 - y^2 + 2ixy$$

es complejo. Sin embargo, el producto de un número por su conjugado es un número *real y positivo*

$$zz^* = (x + iy)(x - iy) = x^2 + y^2 > 0$$

Ello nos permite definir el *valor absoluto* de un número complejo

$$|z| = \sqrt{zz^*} = \sqrt{x^2 + y^2}$$

El *valor absoluto* de una fase es 1

$$|e^{i\theta}| = \sqrt{e^{i\theta}e^{-i\theta}} = \sqrt{e^{i(\theta-\theta)}} = \sqrt{e^0} = 1$$

El *valor absoluto* de un número complejo con el *módulo* escrito en forma polar

$$|z| = \sqrt{zz^*} = \sqrt{\rho e^{i\theta} \rho e^{-i\theta}} = \sqrt{\rho^2} = \rho$$

## Ejercicio

Verifica el valor absoluto de un producto de números complejos es el producto de sus valores absolutos

## División

Al igual que la multiplicación, en forma cartesiana, la división **no es simple**. Sea  $z = a + ib$  y  $w = c + id$

$$\frac{z}{w} = \frac{z w^*}{w w^*} = \frac{(a + ib)(c - id)}{|w|^2} = \frac{ac + bd + i(bc - ad)}{c^2 + d^2} = \frac{ac + bd}{c^2 + d^2} + i \frac{bc - ad}{c^2 + d^2}$$

En forma polar la división es tan sencilla como la multiplicación. Se toma el cociente de los módulos y la resta de las fases

$$\frac{z}{w} = \frac{\rho e^{i\theta}}{\sigma e^{i\phi}} = \frac{\rho}{\sigma} e^{i(\theta-\phi)}$$

```
'''Valor absoluto'''
print('|z1|=',abs(z1))
print('comprobación |z1|=',np.sqrt(z1*z1.conjugate()).real)
print('*****')

'''Division'''
print('z1/z2=',np.round(z1/z2,5))
print('comprobación z1/z2=', np.round(z1*z2.conjugate()/(z2*z2.conjugate()),5))
```

```
|z1|= 3.0
comprobación |z1|= 3.0
*****
z1/z2= (1.08707+2.79612j)
comprobación z1/z2= (1.08707+2.79612j)
```

### 1.1.4 Casos particulares

#### Sumas nulas

En muchas ocasiones nos encontraremos la siguiente representación del numero cero (complejo)  
 $0 = 0 + i0$

$$\sum_{k=0}^{N-1} e^{2\pi i k/N} = e^{2\pi i 0/N} + e^{2\pi i 1/N} + \dots + e^{2\pi i (N-2)/N} + e^{2\pi i (N-1)/N} = 0$$

Para convencerse de que esta identidad es cierta vamos a representar los números complejos y su suma. Puedes cambiar  $N$  y también multiplicar por un módulo constante

```
# cambiar el número N
N=9
rho=1

''' Creamos las fases'''
lista_de_fases=np.exp(2*np.pi*1j*np.array(range(N))/N)
#print('lista de fases =', np.round(lista_de_fases,2))
```

(continues on next page)

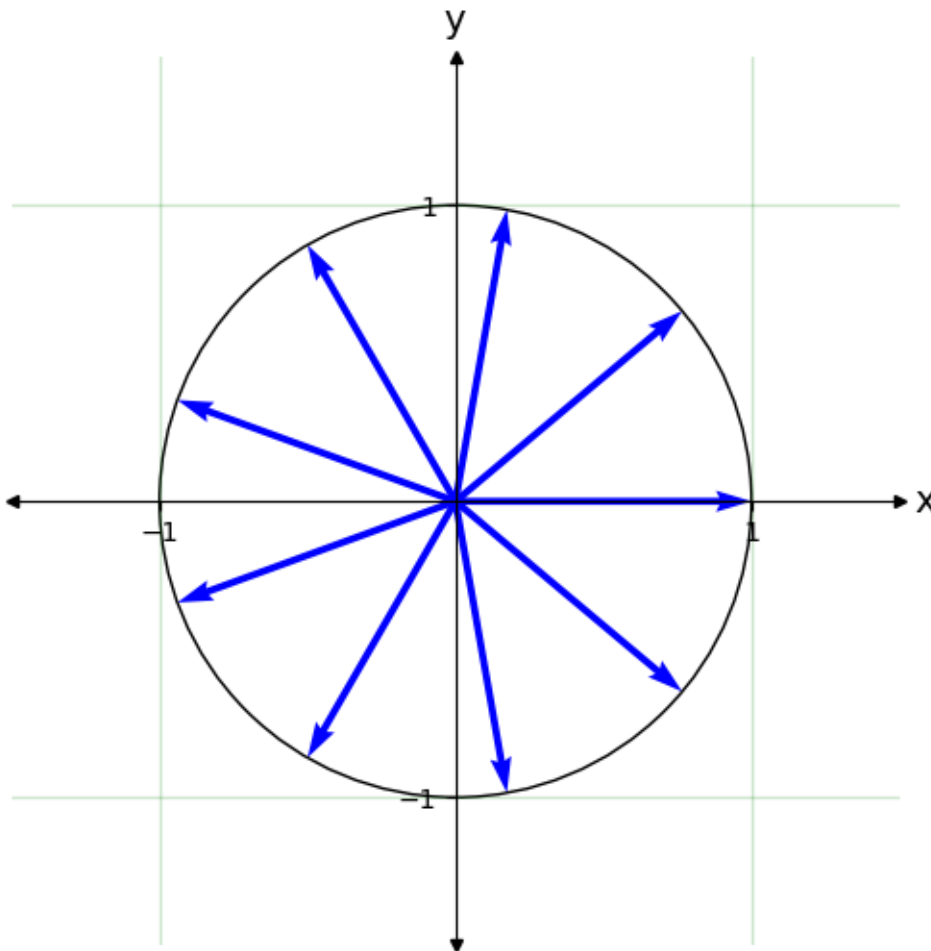
(continued from previous page)

```
''' Dibujamos los números complejos '''
tQ.plot_2D_plane(fsize=(6,6))
for vec in rho*lista_de_fases:
    tQ.draw_vector(x=vec.real,y=vec.imag)

#draw_unit_circle()
plt.gca().add_patch(plt.Circle((0.,0.),1.,color='black',fill=False))

''' Calculamos la suma. '''
#print(lista_de_fases)
print(np.round(sum(rho*lista_de_fases),10))
```

```
(-0+0j)
```



Sea  $1 \leq j \leq N - 1$  un entero por el que multiplicamos todas las fases. El resultado es el mismo

$$\sum_{k=0}^{N-1} e^{2\pi i j k / N} = e^{2\pi i j 0 / N} + e^{2\pi i j / N} + \dots + e^{2\pi i j (N-2) / N} + e^{2\pi i j (N-1) / N} = 0$$

```

# cambiar el número N
N=9
rho=1
j=3
''' Creamos las fases'''
lista_de_fases=np.exp(2*j*np.pi*1j*np.array(range(N))/N)
#print('lista de fases =', np.round(lista_de_fases,2))

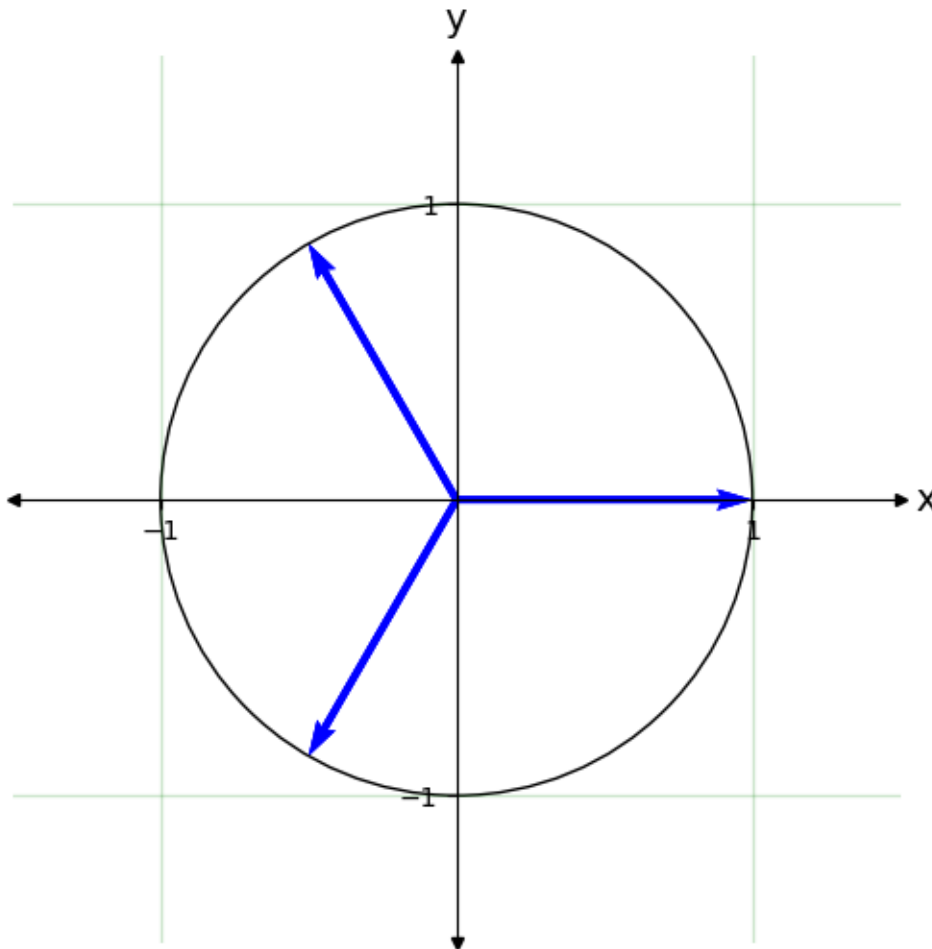
''' Dibujamos los números complejos '''
tQ.plot_2D_plane(fsize=(6,6))
for vec in rho*lista_de_fases:
    tQ.draw_vector(x=vec.real,y=vec.imag)

#draw_unit_circle()
plt.gca().add_patch(plt.Circle((0.,0.),1.,color='black',fill=False))

''' Calculamos la suma. '''
#print(lista_de_fases)
print(np.round(sum(rho*lista_de_fases),10))

```

```
(-0+0j)
```



Sin embargo si  $j = 0, N, 2N, \dots = 0 \bmod N$ , entonces la suma no se anula y su valor es igual a  $N$ .

Tomemos por ejemplo  $j = N$

$$\sum_{k=0}^{N-1} e^{2\pi i(3N)k/N} = \sum_{k=0}^{N-1} e^{2\pi i k} = \sum_{k=0}^{N-1} 1 = N$$

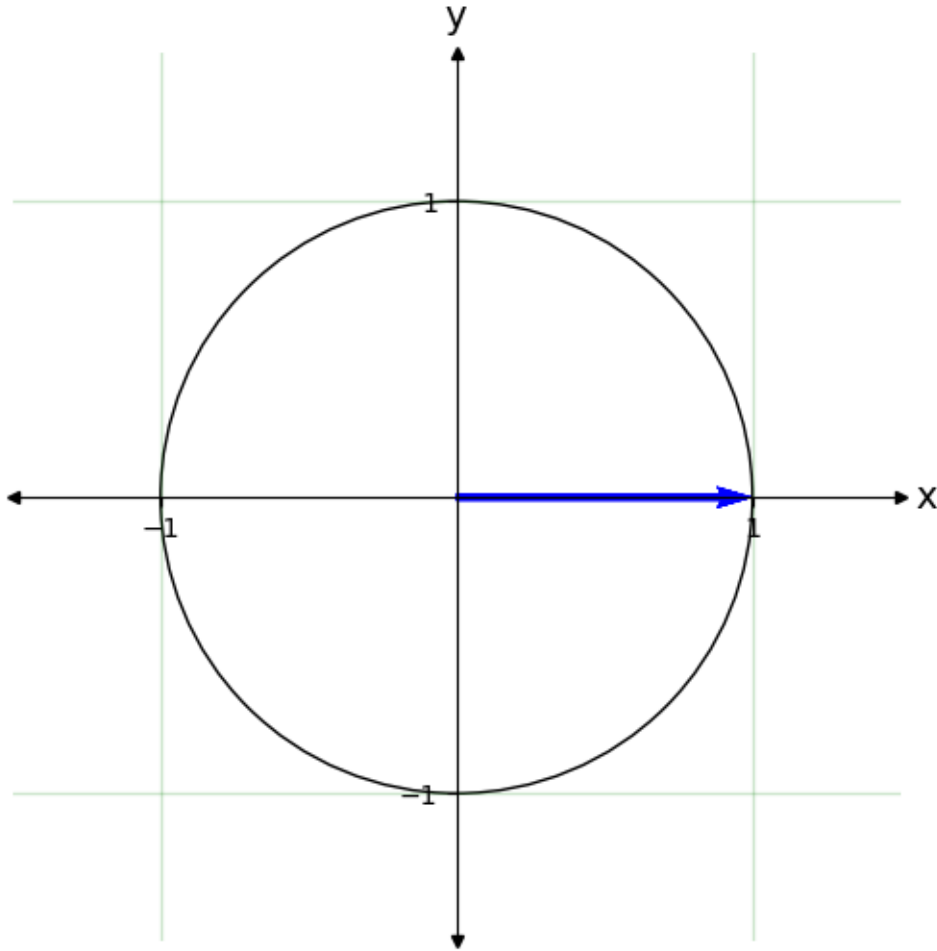
```
# cambiar el número N
N=9
rho=1
j=N
''' Creamos las fases'''
lista_de_fases=np.exp(2*j*np.pi*1j*np.array(range(N))/N)
#print('lista de fases =', np.round(lista_de_fases,2))

''' Dibujamos los números complejos '''
tQ.plot_2D_plane(fsize=(6,6))
for vec in rho*lista_de_fases:
    tQ.draw_vector(x=vec.real,y=vec.imag)

#draw_unit_circle()
plt.gca().add_patch(plt.Circle((0.,0.),1.,color='black',fill=False))

''' Calculamos la suma. '''
#print(lista_de_fases)
print(np.round(sum(rho*lista_de_fases),10))
```

(9-0j)




---

### Ejercicio

Modifica la lista de fases para convencerte de que todos los resultados anteriores son correctos

---

Una manera de resumir todos los casos anteriores en una sola expresión involucra la función  $\delta$  de Kronecker

$$\delta_{ij} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$$

Con ella podemos enunciar el siguiente resultado

$$\frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi i j k / N} = \delta_{j \bmod N}$$

que usaremos con profusión al estudiar la transformada de Fourier cuántica.

## Desigualdad triangular

El módulo de la suma de dos números complejos verifica que

$$|z + w| \leq |z| + |w|$$

Donde la igualdad sólo se verifica cuando ambos números complejos son paralelos en el plano complejo.

```
'''Comprueba
que sólo cuando z1 y z2 son paralelos, se satura la desigualdad triangular'''

'''Suma en cartesianas'''
z1 = 1+2j

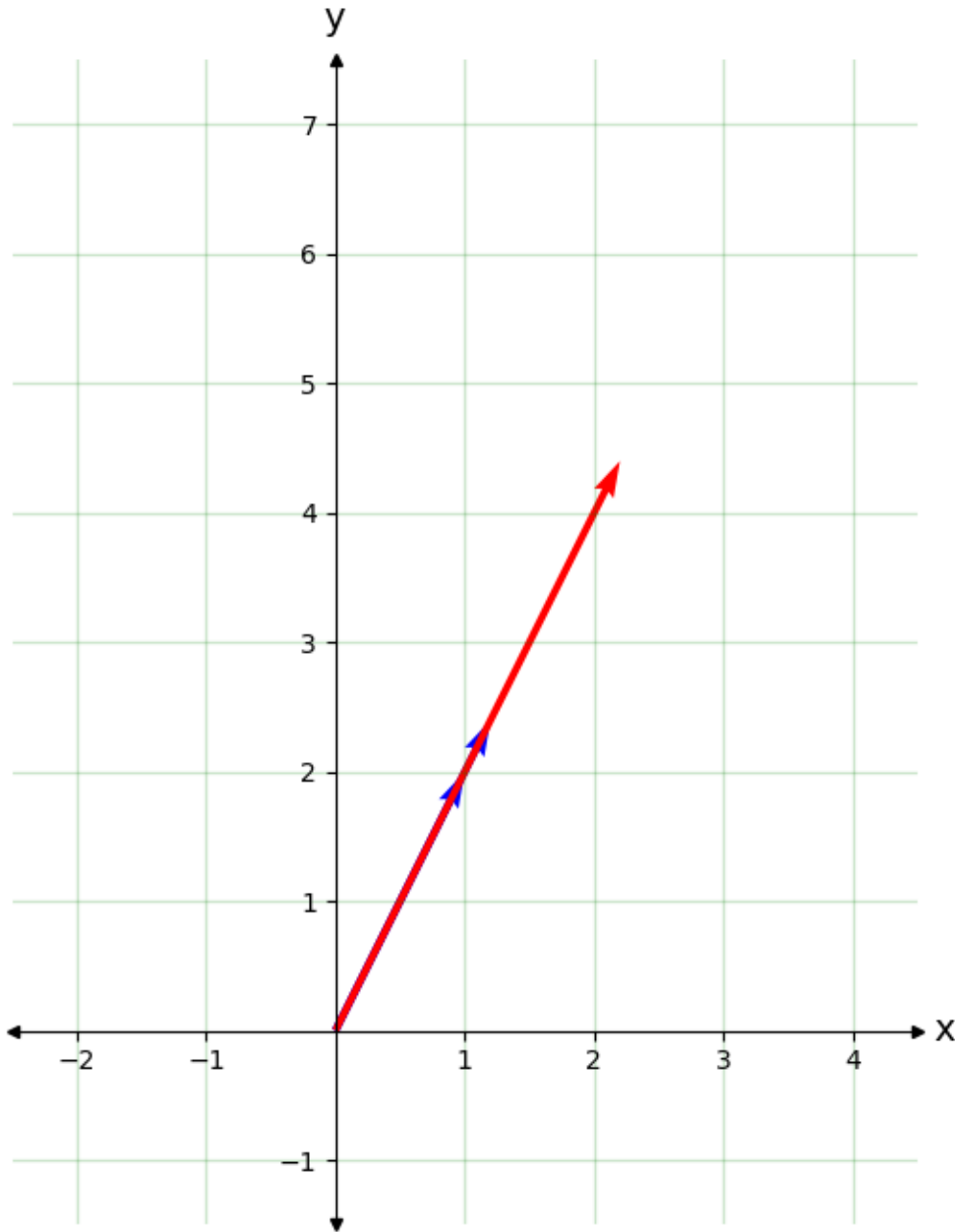
ang = 0. #el ángulo entre z1 y z2
z2 = z1*(1.2*np.exp(1j*ang))

'''Suma '''
zs = z1+z2

print('|z1|+|z2|=',abs(z1)+abs(z2))
print('|z1+z2|=',abs(z1+z2))

tQ.plot_2D_plane(left=-
    2,right=4,up=7,down=-1) #cambiar las dimensiones para que encuadrar la figura
tQ.draw_vector(z1.real,z1.imag,'b')
tQ.draw_vector(z2.real,z2.imag,'b')
tQ.draw_vector(zs.real,zs.imag,vcolor='r')
```

```
|z1|+|z2|= 4.919349550499538
|z1+z2|= 4.919349550499538
```



Feb 13, 2024 | 6 min read

## 1.2 Vectores

```
%run ../../macro_tQ.py
import sys
sys.path.append('../../')
import macro_tQ as tQ
```

(continues on next page)



(continued from previous page)

```
import numpy as np
from IPython.display import display, Markdown, Latex
from qiskit.tools.visualization import array_to_latex
from qiskit.quantum_info import Statevector
```

## 1.2.1 Espacio Vectorial Complejo

### Definición

De forma poco rigurosa, definiremos un *vector de dimensión  $N$*  como una columna de  $N$  números complejos

$$|u\rangle = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}$$

- El símbolo  $|u\rangle$  representa al vector y se denomina *ket* en la *notación de Dirac*
- Los números complejos  $u_i \in \mathbb{C}$  con  $i = 1, \dots, N$  se denominan *componentes* del vector  $|u\rangle$  (en una base dada).

Definición: Espacio Vectorial

La colección de todos los posibles vectores de  $N$  componentes, con las propiedades de suma y multiplicación forman un espacio vectorial,  $V$  de dimension compleja  $N$

Es decir, en un espacio vectorial tenemos dos operaciones posibles:

- Sumar dos vectores

$$|u\rangle + |v\rangle = \begin{pmatrix} u_1 + v_1 \\ u_2 + v_2 \\ \vdots \\ u_N + v_N \end{pmatrix} = |w\rangle$$

- Multiplicar un vector por número complejo  $\lambda \in \mathbb{C}$

$$\lambda|u\rangle = \begin{pmatrix} \lambda u_1 \\ \lambda u_2 \\ \vdots \\ \lambda u_N \end{pmatrix} \equiv |\lambda u\rangle$$

Todo vector de  $V$  se denota mediante el símbolo  $|v\rangle$  menos uno, el *elemento neutro* que se escribe como 0.

La existencia de un elemento opuesto y de un elemento neutro es una de las propiedades que definen un espacio vectorial

$$\begin{aligned} |v\rangle + 0 &= |v\rangle \\ |v\rangle + |-v\rangle &= |v\rangle - |v\rangle = 0 \end{aligned}$$

---

### Notar

La dimensión es igual al número de cantidades (grados de libertad) que debemos fijar para especificar un vector.

Pero  $N$  números complejos equivalen a  $2N$  números reales.

Entonces, podemos decir que:

la dimensión compleja de un espacio vectorial complejo  $V$  es  $N$ , o que su dimensión real es  $2N$

$$\dim_{\mathbb{C}} V = N \iff \dim_{\mathbb{R}} V = 2N$$

Formalmente, este espacio vectorial es equivalente (*isomorfo*) al producto cartesiano

$$V_N \sim \mathbb{C}^N = \overbrace{\mathbb{C} \times \mathbb{C} \times \dots \mathbb{C}}^N$$

formado por ristas de  $N$  números complejos  $N$ -tuplas, con la propiedad de suma y multiplicación

Adelantándonos un poco, el caso  $N = 2$  es el que juega un papel fundamental en computación cuántica: un *cúbit* es un vector de  $\mathbb{C}^2$ . El caso  $N = d$  hablamos de *cúdit*.

```
' qiskit tiene un visualizador '
from qiskit.tools.visualization import array_to_latex

uket=np.array([[1 + 1.j],[2-3*1.j],[2],[+1.j]])

display(array_to_latex(uket))
Statevector(uket).draw('latex')
```

$$\begin{bmatrix} 1+i \\ 2-3i \\ 2 \\ i \end{bmatrix}$$

$$(1+i)|00\rangle + (2-3i)|01\rangle + 2|10\rangle + i|11\rangle$$

## Conjugación adjunta

La operación *conjugación adjunta*,  $\dagger$ , es una *extensión* de la *conjugación compleja* de  $\mathbb{C}$  a todos los elementos de  $H$

Asociado a cada *ket*  $|u\rangle$ , definimos un vector **adjunto**, o *bra*  $\langle u| \equiv (|u\rangle)^\dagger$ , que representamos mediante un vector fila con las componentes conjugadas complejas.

$$\dagger : |u\rangle = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} \rightarrow (|u\rangle)^\dagger \equiv \langle u| = (u_1^* \quad u_2^* \quad \dots \quad u_N^*)$$

Consistentemente encontramos para el producto de un vector por un número complejo  $\lambda$

$$\dagger : \lambda|u\rangle = |\lambda u\rangle \rightarrow (\lambda|u\rangle)^\dagger = \lambda^* \langle u| = \langle u| \lambda^* = \langle \lambda u|$$

ya que el producto de un vector por un número es conmutativo.

Al igual que la conjugación compleja, la conjugación adjunta es una *involución*: su aplicación sucesiva devuelve el vector original

$$(|u\rangle^\dagger)^\dagger = \langle u|^\dagger = |u\rangle$$

es decir,  $\dagger^2 = I$ , el operador identidad.

```
'definamos un ket'
uket=np.array([[1+1j],[2-3*1.j]])
display(array_to_latex(uket))
#print(uket.shape)

'el bra asociado será una fila formada por las componentes conjugadas complejas'
ubra=uket.conj().T

display(array_to_latex(ubra))
#print(ubra.shape)
```

$$\begin{bmatrix} 1+i \\ 2-3i \end{bmatrix}$$

$$[1-i \quad 2+3i]$$

## 1.2.2 Bases

Definición:

En un espacio vectorial  $V$  de dimensión  $N$  una base es una colección de  $N$  vectores  $\{|e_1\rangle, \dots, |e_N\rangle\}$  tales que, cualquier vector  $|v\rangle \in V$  se puede expresar como una combinación lineal de ellos

$$|v\rangle = \sum_{i=1}^N v_i |e_i\rangle$$

Los coeficientes  $v_i$  son las componentes de  $|v\rangle$  en la base dada.

Existen *infinitas* bases en un espacio vectorial. Podemos escoger una de ellas y asociarle el siguiente conjunto de columnas

$$|e_1\rangle \sim \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad |e_2\rangle \sim \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad \dots \quad |e_{N-1}\rangle \sim \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \quad |e_N\rangle \sim \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Cualquier vector, escrito como una combinación lineal de sus elementos adquiere la representación usual

$$\begin{aligned} |u\rangle &= u_1 |e_1\rangle + u_2 |e_2\rangle + \dots + u_N |e_N\rangle = \sum_{i=1}^N u_i |e_i\rangle \sim \\ &\sim u_1 \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} + u_2 \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} + \dots + u_{N-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ 0 \end{pmatrix} + u_N \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \\ u_N \end{pmatrix} \end{aligned}$$

Podemos usar la clase **Statevector** de *qiskit* para tener una expansión en la base  $\{|e_i\rangle = |b(i)\rangle\}$ , donde  $b(i)$  es la representación binaria del índice  $i$ . Por ejemplo  $b(3) = 11$ .

```
from qiskit.quantum_info import Statevector
uket=np.array([[ 1.+1.j], [ 2.-3.j],[ 2.+2.j],[-1.-1.j]])
display(array_to_latex(uket))

Statevector(uket).draw('latex')
```

$$\begin{bmatrix} 1+i \\ 2-3i \\ 2+2i \\ -1-i \end{bmatrix}$$

$$(1+i)|00\rangle + (2-3i)|01\rangle + (2+2i)|10\rangle + (-1-i)|11\rangle$$

## Cambio de base

Existen *infinitas* bases en un espacio vectorial de dimensión finita. Todas ellas sirven para representar un vector arbitrario.

Consideremos dos bases  $\{|e_i\rangle\}$  y  $\{|\tilde{e}_j\rangle\}$  donde  $i, j = 1, \dots, N$ :

cualquier *elemento* (vector) de una base se puede expresar como una *combinación lineal de elementos* de la otra

$$|\tilde{e}_j\rangle = \sum_{i=1}^N C_{ij}|e_i\rangle \quad j = 1, \dots, N$$

Los coeficientes  $C_{ij} \in \mathbb{C}$  constituyen la *matriz*  $C_{ij}$  de *cambio de base*

---

### Notar

La forma en que están sumados los índices

$$|\tilde{e}_j\rangle = \sum_{i=1}^N C_{ij}|e_i\rangle$$

Esto hace que, si queremos escribir esta ecuación en notación matricial, debamos poner los vectores en una fila

$$(|\tilde{e}_1\rangle \quad \dots \quad |\tilde{e}_N\rangle) = (|e_1\rangle \quad \dots \quad |e_N\rangle) \begin{pmatrix} C_{11} & \dots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{N1} & \dots & C_{NN} \end{pmatrix}$$

---

### Ejemplo

Sea una nueva base  $\{|f_1\rangle, |f_2\rangle\}$  definida en términos de la antigua mediante

$$|f_1\rangle = \frac{1}{\sqrt{2}}(|e_1\rangle + i|e_2\rangle), \quad |f_2\rangle = \frac{1}{\sqrt{2}}(|e_1\rangle - i|e_2\rangle).$$

La matriz de coeficientes es

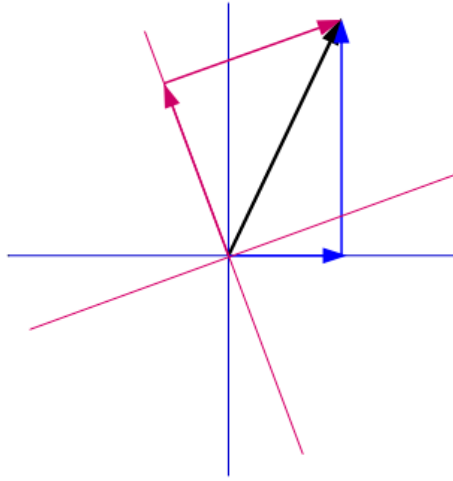
$$C_{ij} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix},$$

El cambio de base, usando esta matriz se escribe de forma matricial

$$(|f_1\rangle, |f_2\rangle) = (|e_1\rangle, |e_2\rangle) \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}$$

Un vector puede ser expandido en ambas bases

$$|v\rangle = \sum_i v_i |e_i\rangle = \sum_j \tilde{v}_j |\tilde{e}_j\rangle$$



La relación entre los coeficientes se sigue de la misma matriz de cambio de base

$$\begin{aligned} |v\rangle = \sum_i v_i |e_i\rangle &= \sum_j \tilde{v}_j |f_j\rangle \\ &= \sum_j \tilde{v}_j \left( \sum_i C_{ij} |e_i\rangle \right) \\ &= \sum_j \sum_i \tilde{v}_j C_{ij} |e_i\rangle \\ &= \sum_i \left( \sum_j C_{ij} \tilde{v}_j \right) |e_i\rangle \end{aligned}$$

Comparando obtenemos la relación de coeficientes

$$v_i = \sum_j C_{ij} \tilde{v}_j$$

que podemos invertir

$$\tilde{v}_i = \sum_j C_{ij}^{-1} v_j$$

Notar

1. La forma en que están sumados los índices hace que, en notación matricial esta operación se represente como sigue

$$\begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \begin{pmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{pmatrix} \begin{pmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_n \end{pmatrix}$$

2. La misma matriz  $C$ , que lleva  $|e_i\rangle \rightarrow |\tilde{e}_i\rangle$ , lleva  $\tilde{v}_j \rightarrow v_j$
- 

En particular esto dice que la matriz  $C_{ij}$  se lee inmediatamente de expresar la nueva base en la antigua

---

### Ejemplo (continuación)

$$|f_1\rangle \sim \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}, \quad |f_2\rangle \sim \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix}.$$

Poniendo las dos columnas en una sola matriz, obtenemos la matriz  $C_{ij}$  que efectúa el cambio de los vectores de la base

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

---

### Ejercicio

Considera las bases ortonormales  $\{|0\rangle, |1\rangle\}$  y  $\{|+\rangle, |-\rangle\}$  donde  $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ .

Las componentes del vector  $|u\rangle \sim \begin{pmatrix} 3 \\ 1 \end{pmatrix}$  están escritas en la primera base.

Halla las componentes de  $|u\rangle$  en la segunda.

---

## 1.2.3 Espacio de Hilbert

Definición:

Un espacio de Hilbert,  $H$ , es un espacio vectorial dotado de una operación interna denominada producto escalar.

### Producto Escalar

Definición:

El producto escalar de dos vectores  $|u\rangle$  y  $|v\rangle$  es un número complejo  $a \in \mathbb{C}$  que denotamos  $\langle u|v \rangle$

$$a \equiv \langle u|v \rangle$$

si verifica las dos propiedades siguiente

- linealidad:  $\rightarrow \langle u|(|v\rangle + |w\rangle) = \langle u|v\rangle + \langle u|w\rangle$
  - hermiticidad:  $\rightarrow \langle v|u\rangle = \langle u|v\rangle^*$
  - positividad:  $\langle u|u\rangle > 0$  para todo  $\text{ket } |u\rangle \neq 0$
  - no-degeneración: si  $\langle u|v\rangle = 0$  para todo  $\langle u|$ , entonces necesariamente  $|v\rangle = 0$
-

Combinando ambas propiedades, el producto escalar también es lineal en el primer argumento

$$(\langle u| + \langle w|)|v\rangle = \langle u|v\rangle + \langle w|v\rangle$$

Notar que la propiedad de hermiticidad es precisamente la extensión de la conjugación compleja de la que hablábamos al definir la aplicación *adjunta*

$$\langle u|v\rangle^* = \langle u|v\rangle^\dagger = \langle v|u\rangle$$

Hemos usado que  $\langle u|^\dagger = |u\rangle$  y  $|v\rangle^\dagger = \langle v|$  pero le hemos añadido una regla más: *al tomar el mapa adjunto es necesario invertir el orden de los elementos*

De no haber seguido esta regla, habríamos obtenido un resultado **erróneo**

$$\langle u|v\rangle^\dagger \rightarrow |u\rangle\langle v|$$

que no es ni siquiera un número complejo.

## Norma

Una **norma** es una función real  $\|\cdot\| : \text{Lin}(H) \rightarrow \mathbb{R}$  con las siguientes propiedades

- ser definida positiva  $\|A\| \geq 0$  con  $\|A\| = 0 \Leftrightarrow A = 0$
- homogeneidad  $\|\lambda A\| = |\lambda| \|A\|$
- triangle inequality.  $\|A + B\| \leq \|A\| + \|B\|$

Un espacio de Hilbert es, automáticamente un espacio normado. La positividad del producto escalar de un vector por sí mismo permite definir su *norma*.

$$\| |v\rangle \| = \sqrt{\langle v|v\rangle}$$

## Notar

- En contraste con la definición de producto escalar en espacios vectoriales reales, en el caso complejo se hace necesario conjugar el bra, para que la norma de un vector sea siempre real y positiva. Esta es la idea detrás de la definición de la conjugación adjunta.
- El único vector que tiene norma nula en un espacio de Hilbert es el elemento neutro

$$\langle v|v\rangle = 0 \Leftrightarrow |v\rangle = 0$$

## Distancia

Dados dos elementos  $|u\rangle$  y  $|v\rangle$  de  $H$ , podemos definir la *distancia* entre ellos como la *norma de su diferencia*

$$d(|v\rangle, |w\rangle) = \| |v\rangle - |w\rangle \|$$

En particular

- $d(|v\rangle, |w\rangle) = d(|w\rangle, |v\rangle)$
- $d(|v\rangle, |v\rangle) = 0$

### 1.2.4 Base ortonormales

Hasta ahora, a los vectores de una base  $\{|e_i\rangle\}$  sólo se les ha pedido que sean  $N$  vectores *linealmente independientes*, donde  $N$  es la dimensión del espacio vectorial  $V$ :

En un espacio de Hilbert  $H$  tiene sentido calcular el producto escalar de dos elementos de una base.

Definición:

Una base ortonormal se caracteriza por la siguiente lista de productos escalares

$$\langle e_i | e_j \rangle = \delta_{ij}$$

- Por un lado, dos elementos distintos de la base son ortogonales  $\langle e_1 | e_2 \rangle = 0$ .
- Por otro, todos están normalizados  $\|e_i\| = \sqrt{\langle e_i | e_i \rangle} = \sqrt{1} = 1$ .

En este curso siempre supondremos que las bases con las que trabajamos son ortonormales. Ello se justifica en base al siguiente teorema:

Teorema: Teorema de Gram-Schmidt

Dada una base general  $\{\langle f_i | f_j \rangle \neq \delta_{ij}\}$  de vectores no ortonormales, existe un procedimiento iterativo (de Gram-Schmidt) para construir, a partir de ella, una nueva base ortonormal  $\{\langle e_i | e_j \rangle = \delta_{ij}\}$ .

Dado un vector  $|v\rangle = \sum_{i=1}^N v_i |e_i\rangle$  escrito en una base ortonormal, la *componente*  $v_i$  se extrae mediante la *proyección ortogonal*

$$v_i = \langle e_i | v \rangle$$

---

#### Ejercicio

Verifica esta expresión

---

#### Solución

$$\begin{aligned} \langle e_k | v \rangle &= \langle e_k | \left( \sum_{j=1}^N v_j |e_j\rangle \right) \\ &= \sum_{j=1}^N v_j \langle e_k | e_j \rangle \\ &= \sum_{j=1}^N v_j \delta_{kj} = v_k \end{aligned} \tag{1.7}$$

En una base ortonormal, calcular el valor de un *producto escalar*  $a = \langle u | v \rangle$  es muy simple

$$\begin{aligned} a = \langle u | v \rangle &= \left( \sum_i u_i^* \langle e_i | \right) \left( \sum_j v_j |e_j\rangle \right) = \sum_{ij} u_i^* v_j \langle e_i | e_j \rangle = \sum_{ij} u_i^* v_j \delta_{ij} \\ &= \sum_i u_i^* v_i = (u_1^* \quad u_2^* \quad \cdots \quad u_N^*) \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} \end{aligned}$$



## Notar

La expresión de la izquierda  $a = \langle u|v \rangle$  no hace referencia a ninguna base. Por tanto, el resultado  $\sum_{i=1}^n u_i^* v_i$  debe ser independiente de la base que utilizamos para representar estos vectores mediante sus componentes  $u_i$  y  $v_i$ .

Subrayamos la importancia de esto:  $\langle u|v \rangle$  puede ser calculado en la base más conveniente.

## Ejercicio (Explícaselo a tu ordenador)

- Escribe una función  $\text{braket}(u, v)$  que calcule y devuelva el producto escalar  $\langle u|v \rangle$ , y, con ella, una función  $\text{norm}(u)$  que calcule la norma  $\|u\|$ . Verifica que  $\|u\| = \sqrt{\langle u|u \rangle}$  coincide con el resultado que da la función  $\text{np.linalg.norm}$ .
- Escribe una función  $\text{random\_ket}$  que genere un vector normalizado  $|v\rangle \in H$  de dimensión  $d$ .

Añade ambas funciones al archivo `macrotQ.py`

```
uket = tq.random_ket(5)
vket = tq.random_ket(5)

tq.braket(uket, vket)
```

```
(0.7254644130171193+0.24753624258665705j)
```

## Descomposición ortogonal

Definición:

Decimos que  $H$  admite una suma directa ortogonal  $H = H_1 \oplus H_2$  cuando todo vector  $|u\rangle \in H$  se puede escribir en la forma

$$|u\rangle = |u_1\rangle + |u_2\rangle$$

donde  $|u_1\rangle \in H_1$ ,  $|u_2\rangle \in H_2$  y  $\langle u_1|u_2\rangle = 0$ .

Dada una elección de  $|u_1\rangle$  el vector  $|u_2\rangle$  es único.

Este resultado no se refiere a un espacio vectorial  $H$  de dimensión 2. Cada línea en la figura anterior representa un subespacio vectorial  $|u_1\rangle \in H_1$  y  $|u_2\rangle \in H_2$ .

De modo que, lo que el teorema afirma es que hay infinitas descomposiciones de la forma  $H = H_1 \oplus H_2$  donde las dimensiones se suman.

$$\dim H = \dim H_1 + \dim H_2$$

La anterior expresión define lo que se conoce como *suma directa* de espacios vectoriales. En particular, de una suma directa *ortogonal*

Feb 13, 2024 | 19 min read

## 1.3 Operadores

```
%run ../../macro_tQ.py
import sys
sys.path.append('../../')
import macro_tQ as tQ

import numpy as np
import scipy.linalg as la
from IPython.display import display, Markdown, Latex
import matplotlib.pyplot as plt
from qiskit.tools.visualization import array_to_latex
from IPython.display import display, Latex
```

### 1.3.1 Operadores y Matrices

En un espacio vectorial, además de los vectores, será esencial entender la manera en que estos se pueden transformar entre sí,

Definición: Operador lineal

Un operador lineal es una aplicación  $A : H \rightarrow H$  que transforma un vector en otro

$$A : |u\rangle \rightarrow |v\rangle$$

de forma lineal.

La linealidad se refiere a la siguiente propiedad:

$$A : (\alpha|u\rangle + \beta|w\rangle) \rightarrow |v\rangle = \alpha A|u\rangle + \beta A|w\rangle$$

---

#### Nota (Sobre la notación)

Escribimos también  $|v\rangle = A|u\rangle \equiv |Au\rangle$  (donde  $Au$  debe entenderse como una etiqueta)

---

#### Ejemplo

Un operador fácil de visualizar es el operador de rotación en un plano. Dado un ángulo  $\theta \in (0, 2\pi)$  el operador  $A = R(\theta)$  gira cualquier vector un ángulo  $\theta$  en el sentido antihorario

Un vector en el plano  $\mathbf{u} = (u_1, u_2)$  es equivalente al número complejo  $u = u_1 + iu_2$  en el plano complejo  $V = \mathbb{C}$ .

Escrito en polares,  $u = |u|e^{i\phi}$ , y sabemos que una rotación de ángulo  $\theta$  es equivalente a añadirle dicho ángulo a la fase

$$v = R(\theta)u = |u|e^{i(\phi+\theta)} = |u|e^{i\phi}e^{i\theta} = u \cdot e^{i\theta}$$

Por tanto, para rotar un número complejo un ángulo  $\theta$  basta con multiplicarlo por la fase  $e^{i\theta}$ , que se corresponde con el operador  $R(\theta)$  en el espacio vectorial  $V = \mathbb{C}$ .

La propiedad fundamental de una rotación es la de mantener invariante el módulo  $|v| = |u|$ .

---

### Ejercicio

Usando el ejemplo anterior, define una función  $R$ , que recibe un vector en el plano  $(u_1, u_2)$  y devuelve el vector  $(v_1, v_2)$  de componentes rotadas un ángulo  $\theta$ .

### Solución

```
def R(u1,u2,theta):
    u = u1 + u2*1j
    v = u*np.exp(1j*theta) # u rotado un angulo theta
    return v.real,v.imag
```

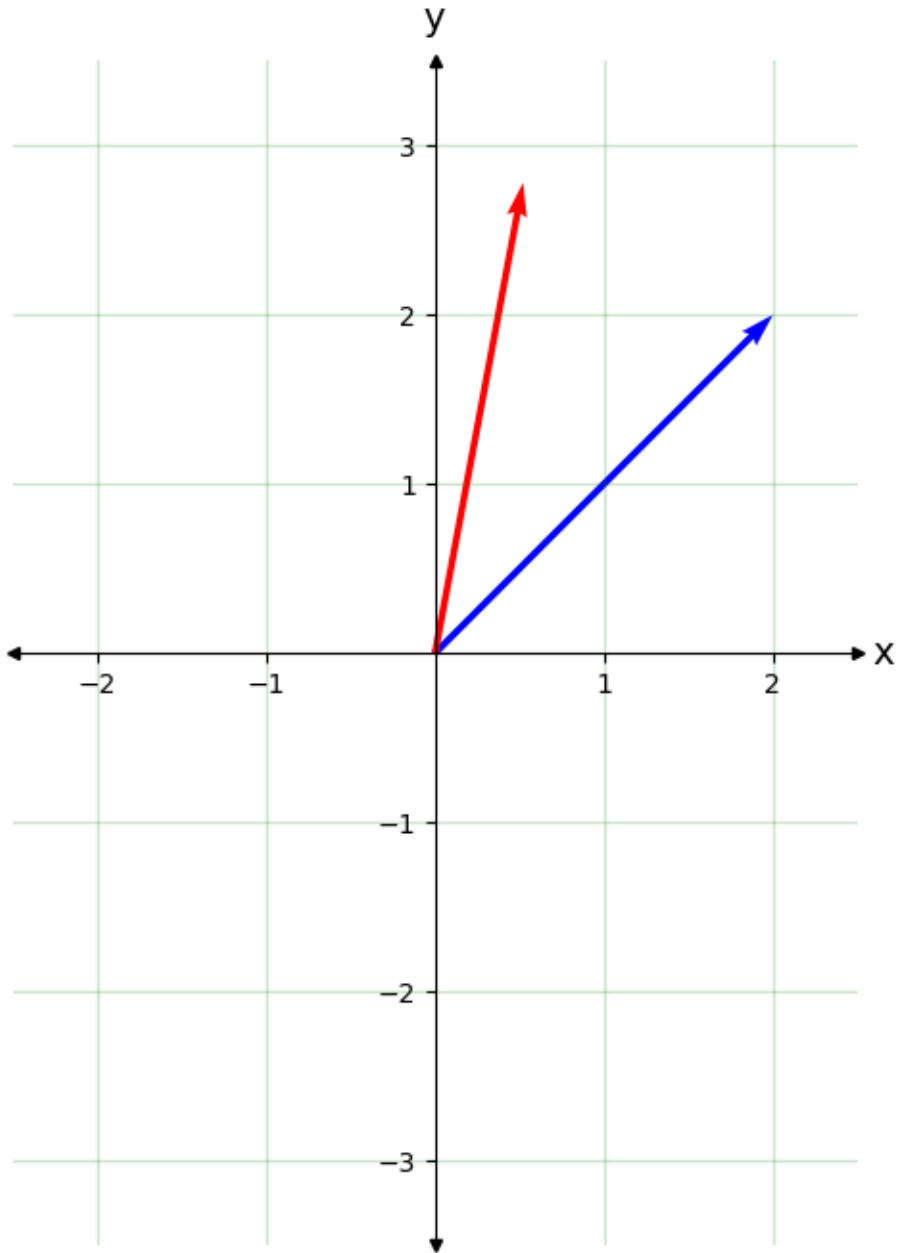
```
'''ángulo que queremos rotar'''
theta=0.6

'''vector a rotar'''
u1=2.
u2=2.

'''v1 y v2 a partir de u1, u2 y theta'''
def R(u1,u2,theta):
    u = u1 + u2*1j
    v = u*np.exp(1j*theta) # u rotado un angulo theta
    return v.real,v.imag

v1,v2 = R(u1,u2,theta)

''' Representación en el plano complejo '''
v = v1**2+v2**2
tQ.plot_2D_plane(left=-int(abs(v1))-2,right=int(abs(v1))+2,up=int(abs(v2))+1,
    down=-int(abs(v2))-1)
tQ.draw_vector(u1,u1,vcolor='b')
tQ.draw_vector(v1,v2,vcolor='r')
```



### Matriz de un operador

Dada una base  $|i\rangle$

$\Rightarrow$  un vector queda especificado por una *columna de componentes*

$$|v\rangle \sim \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}$$

⇒ un operador queda definido por una *matriz de componentes*.

$$A \sim \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix}$$

Efectivamente, en una base, la relación  $|v\rangle = A|u\rangle$  equivale a una ecuación que relacione las componentes de ambos vectores

$$v_i = \sum_{j=1}^N A_{ij} u_j.$$

Esta operación se corresponde con la siguiente composición de matrices

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}$$

### Ejemplo

Continuando con el ejemplo del operador de rotación en un plano, hemos visto que las componentes de  $u = u_1 + iu_2$  y las de  $R(\theta)u = v = v_1 + iv_2$  se obtienen mediante la multiplicación por una fase pura

$$v = ue^{i\theta}$$

Vamos a desarrollar cada miembro en cartesianas, separando las partes real e imaginaria

$$\begin{aligned} v_1 + iv_2 &= (u_1 + iu_2)(\cos \theta + i \sin \theta) \\ &= (\cos \theta u_1 - \sin \theta u_2) + i(\sin \theta u_1 + \cos \theta u_2) \end{aligned}$$

es decir las coordenadas del vector origen y el vector rotado imagen se relacionan en la forma

$$v_1 = \cos \theta u_1 - \sin \theta u_2, \quad v_2 = \sin \theta u_1 + \cos \theta u_2$$

que podemos expresar en forma matricial

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

## 1.3.2 Base de Operadores

### Producto externo

Dependiendo del orden en que los componemos,  $\langle u|v\rangle$  ó  $|v\rangle\langle u|$ , el resultado es muy distinto

- El **producto interno**, o *producto escalar* es un *número complejo*

$$a = \langle u|v\rangle = \langle v|u\rangle^*$$

- El **producto externo** es un *operador*

$$A = |v\rangle\langle u|$$

Para comprender por qué es un operador, observamos que dicha expresión aplicada a un vector  $|w\rangle$  da otro,

$$A : |w\rangle \rightarrow A|w\rangle = |v\rangle\langle u|w\rangle = |v\rangle b = b|v\rangle$$

---

### Notar

1. El orden en que escribimos las cosas es muy relevante.
    - $\langle u|v\rangle$  y  $|v\rangle\langle u|$  son objetos radicalmente distintos: el primero es un número y el segundo es un operador.
    - En cambio  $|v\rangle b = b|v\rangle$ , así como  $\langle u|b = b\langle u|$ , es decir, los números complejos y los *kets* o *bras* pueden escribirse en cualquier orden (decimos que conmutan).
  2. La acción del operador  $A = |v\rangle\langle u|$  es muy fácil de expresar con palabras:
    - el operador  $A$  toma cualquier vector  $|w\rangle$  y lo convierte en un vector paralelo a  $|v\rangle$  proporcionalmente a su proyección  $b = \langle u|w\rangle$ .
    - si la proyección es nula  $b = 0$ , el operador aniquila, es decir, da el elemento neutro.
- 

### Producto externo en componentes

La diferencia entre el *producto interno*  $a = \langle u|v\rangle$  y el *externo*  $A = |u\rangle\langle v|$  tiene su reflejo en una base expresando ambos vectores,  $|u\rangle = \sum_i u_i |i\rangle$  y  $|v\rangle = \sum_j v_j |j\rangle$ , en componentes en una base ortonormal

- el *número complejo*  $a$  es el *producto escalar*

$$a = \langle u|v\rangle = (u_1^*, \dots, u_N^*) \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} = \sum_i u_i^* v_i$$

- la matriz  $A_{ij}$  *representa* el operador  $A$  en la base  $\{|i\rangle = |e_i\rangle\}$

$$A = |v\rangle\langle u| \sim \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} (u_1^*, \dots, u_N^*) = \begin{pmatrix} v_1 u_1^* & v_1 u_2^* & \dots & v_1 u_N^* \\ v_2 u_1^* & v_2 u_2^* & \dots & v_2 u_N^* \\ \vdots & \vdots & \ddots & \vdots \\ v_N u_1^* & \dots & \dots & v_N u_N^* \end{pmatrix} = A_{ij}$$

### Base canónica de operadores

Consideremos el *producto externo* de dos elementos de la base ortonormal  $|i\rangle\langle j|$

- La acción de  $|i\rangle\langle j|$  sobre otro vector,  $|k\rangle$ , de la base es sencilla

$$|i\rangle\langle j|k\rangle = |i\rangle\delta_{jk} = \begin{cases} 0 & \text{si } k \neq j \\ |i\rangle & \text{si } k = j \end{cases}$$

- La matriz asociada al operador tiene sólo un 1 en el elemento  $(ij)$  y cero en todos los demás. Por ejemplo, supongamos que  $N = 4$

$$|2\rangle\langle 3| \rightarrow \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow A_{ij} = \delta_{i2}\delta_{j3}$$

Los elementos de matriz  $A_{ij}$  expresan las componentes de un operador en la base de operadores  $|i\rangle\langle j|$

$$A = \sum_{i,j=1}^N A_{ij} |i\rangle\langle j|$$

## Consistencia

Verifiquemos que actúa de la forma correcta

$$\begin{aligned} A|u\rangle &= \sum_{i,j} A_{ij} |i\rangle\langle j| \left( \sum_k u_k |k\rangle \right) \\ &\stackrel{\text{linealidad}}{=} \sum_{i,j} \sum_k A_{ij} |i\rangle u_k \langle j|k\rangle \\ &\stackrel{\text{ortonormalidad}}{=} \sum_{i,j,k} A_{ij} |i\rangle u_k \delta_{jk} \\ &= \sum_{ij} A_{ij} |i\rangle u_j = \sum_i \left( \sum_j A_{ij} u_j \right) |i\rangle \\ &= \sum_i v_i |i\rangle \\ &= |v\rangle \end{aligned}$$

## Elementos de matriz

De la misma manera que obteníamos las componentes de un vector proyectando sobre un elemento de la base

$$v_i = \langle i|v\rangle$$

ahora podemos obtener los *elementos de matriz* de un operador  $A$  en la forma

$$A_{ij} = \langle i|A|j\rangle$$

---

## Ejercicio

Comprueba la consistencia de las expresiones  $A = \sum_{i,j=1}^N A_{ij} |i\rangle\langle j|$  y  $A_{ij} = \langle i|A|j\rangle$

---

## Resumen

Dada una base  $\{|i\rangle\}$  podemos expresar un operador mediante una matriz  $A_{ij}$ . La relación concreta es

- como operador  $\rightarrow A = \sum_{ij} A_{ij} |i\rangle\langle j|$
  - como elemento de matriz  $\rightarrow A_{ij} = \langle i|A|j\rangle$
-

## Cambio de base

Dos bases ortonormales  $|e_i\rangle$  y  $|\tilde{e}_i\rangle$  se relacionadas linealmente mediante una matriz

$$|e_j\rangle \rightarrow |\tilde{e}_j\rangle = \sum_{ij} U_{ij} |e_i\rangle$$

La relación *adjunta* es muy sencilla de obtener

$$\langle e_j| \rightarrow \langle \tilde{e}_j| = \sum_{ij} U_{ij}^* \langle e_i|$$

En cada base, un operador  $A$  se representa mediante elementos de matriz distintos

$$A_{ij} = \langle e_i|A|e_j\rangle \quad , \quad \tilde{A}_{ij} = \langle \tilde{e}_i|A|\tilde{e}_j\rangle .$$

Podemo encontrar la relación sustituyendo el cambio de base

$$\begin{aligned} \tilde{A}_{ij} &= \langle \tilde{e}_i|A|\tilde{e}_j\rangle \\ &= \sum_k U_{ki}^* \langle e_k|A \sum_l U_{lj} |e_l\rangle \\ &= \sum_{k,l} U_{ik}^\dagger \langle e_k|A|e_l\rangle U_{lj} = \sum_{k,l} U_{ik}^\dagger A_{kl} U_{lj} . \end{aligned}$$

Lema:

Bajo un cambio de bases ortonormales  $|e_j\rangle \rightarrow |\tilde{e}_j\rangle = \sum_i U_{ij} |e_i\rangle$  las componentes de un vector  $|v\rangle$  y de un operador  $A$  cambian siguiendo la regla:

$$\tilde{v}_i = (U^\dagger \cdot v)_i$$

$$\tilde{A}_{ij} = (U^\dagger \cdot A \cdot U)_{ij}$$

---

## Notar

La regla mnemotécnica es que las columnas se multiplican por  $U^\dagger$  y las filas por  $\cdot U$

$$\begin{pmatrix} \tilde{v}_1 \\ \vdots \\ \tilde{v}_N \end{pmatrix} = U^\dagger \cdot \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} \quad ; \quad \begin{pmatrix} \tilde{A}_{11} & \cdots & \tilde{A}_{1N} \\ \vdots & \ddots & \vdots \\ \tilde{A}_{N1} & \cdots & \tilde{A}_{NN} \end{pmatrix} = U^\dagger \cdot \overbrace{\begin{pmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{pmatrix}}^{\cdot U}$$


---

## Ejercicio

Escribe una función *basis\_change* que reciba una matriz  $U_{ij}$  de cambio de base  $|\tilde{e}_j\rangle = \sum_i U_{ij} |e_i\rangle$ , la componentes  $v_i$  de un vector, ó  $A_{ij}$  de un operador, y devuelva las componentes  $\tilde{v}_i$  ó  $\tilde{A}_{ij}$  en la nueva base.

La matriz  $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$  representa un operador  $\sigma_y$  en la base  $\{|0\rangle, |1\rangle\}$ . Escribe  $\sigma_y$  en la base  $\{|+i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)\}$

---



## Relación de completitud

La acción del operador identidad es

$$I|v\rangle = |v\rangle$$

En particular sobre todo elemento de la base  $I|i\rangle = |i\rangle$ . En otras palabras, el operador identidad  $I$  tiene por matriz  $I_{ij} = \delta_{ij}$  = diagonal  $(1, 1, \dots, 1)$  con lo que

$$I = \sum_i |i\rangle\langle i| = \sum_{ij} \delta_{ij} |i\rangle\langle j|$$

Esta expresión se conoce también como relación de completitud o, también, relación de cierre y se utiliza muy frecuentemente.

---

### Notar

La relación de completitud es, en realidad, una propiedad de cualquier base. Dicho de otro modo, si  $\{|e_i\rangle\}$  y  $\{|\tilde{e}_j\rangle\}$  son, ambas, bases entonces  $I|e_i\rangle = |e_i\rangle$  y  $I|\tilde{e}_j\rangle = |\tilde{e}_j\rangle$ , entonces  $+1$  es el único autovalor de  $I$  en cualquier base, y la descomposición espectral da

$$I = \sum_i |e_i\rangle\langle e_i| = \sum_j |\tilde{e}_j\rangle\langle \tilde{e}_j|.$$

---

La relación de cierre, o completitud, siempre **se puede insertar** en cualquier momento del cálculo. Se utiliza con frecuencia para efectuar cambios de base.  $|u\rangle = \sum_i u_i I|e_i\rangle = \sum_i u_i \left( \sum_j |\tilde{e}_j\rangle\langle \tilde{e}_j| \right) |e_i\rangle = \sum_j \left( \sum_i \langle \tilde{e}_j|e_i\rangle u_i \right) |\tilde{e}_j\rangle = \sum_j \tilde{u}_j |\tilde{e}_j\rangle$

De aquí se sigue que

$$\tilde{u}_j = \sum_i C_{ji} u_i$$

donde la matriz de cambio es

$$C_{ji} = \langle \tilde{e}_j|e_i\rangle$$

### 1.3.3 El Espacio Vectorial $\text{Lin}(H)$

El conjunto de **todos** los operadores lineales sobre un espacio vectorial  $H$  tiene, de forma natural, una estructura de espacio vectorial que denominamos  $\text{Lin}(H)$

En efecto, dados dos operadores,  $A$  y  $B$  tanto la suma  $C = A + B$  como la multiplicación por un número complejo  $D = \lambda A$  son *nuevos operadores* definidos por su acción sobre un vector cualquiera  $|v\rangle \in H$

$$C|v\rangle = (A + B)|v\rangle = A|v\rangle + B|v\rangle$$

$$D|v\rangle = (\lambda A)|v\rangle = \lambda(A|v\rangle)$$

## Operador Adjunto

La conjugación *adjunta* se puede extender a  $\text{Lin}(H)$

$$\dagger \rightarrow \begin{cases} z & \leftrightarrow & z^* \\ |u\rangle & \leftrightarrow & \langle u| \\ A & \leftrightarrow & A^\dagger \end{cases}$$

y hay dos reglas más que permiten aplicar  $\dagger$  a sumas y productos de objetos  $a \in \{z, |u\rangle, A\}$

- *linealidad*  $(a + b)^\dagger = a^\dagger + b^\dagger$
- *trasposición*  $(ab)^\dagger = b^\dagger a^\dagger$  (sólo relevante cuando  $a$  y  $b$  no conmuten)

---

### Ejemplos

1.  $|v\rangle = A|u\rangle \Leftrightarrow \langle v| = \langle u|A^\dagger$  donde el operador en la derecha actúa sobre el *bra* a su izquierda. Notar que, como  $|v\rangle^\dagger = |Au\rangle^\dagger = \langle Au|$  la ecuación anterior implica

$$\langle Au| = \langle u|A^\dagger$$

2.

$$\langle w|A|u\rangle^* = (\langle w|A|u\rangle)^\dagger = \langle u|A^\dagger|w\rangle$$


---

## Matriz adjunta

Estas reglas nos permiten obtener el adjunto de un operador

$$A^\dagger = \sum_{ij} (A_{ij}|i\rangle\langle j|)^\dagger = \sum_{ij} |j\rangle\langle i|A_{ij}^* = \sum_{ji} A_{ji}^*|i\rangle\langle j|$$

donde en la última ecuación hemos intercambiado los nombres  $i \leftrightarrow j$

Vemos que la matriz que representa  $A^\dagger$  es la *matriz adjunta* de  $A_{ij}$ , es decir, la traspuesta y conjugada

$$(A^\dagger)_{ij} = A_{ji}^* = (A_{ij}^*)^t \equiv (A_{ij})^\dagger$$

donde  $^\dagger$  significa el adjunto de un operador a la izquierda, y de una matriz a la derecha.

Si  $H$  tiene dimensión  $N$ , un *operador general*  $A \in \text{Lin}(H)$  se especifica mediante una matriz de  $N^2$  números complejos  $\Rightarrow A = A_{ij}|e_i\rangle\langle e_j|$ .

$N^2$  números complejos equivalen a  $2N^2$  números reales.

En otras palabras:  $A$  tiene  $N^2$  grados de libertad complejos y, por tanto, ésta es la dimension del espacio  $\text{L}(H)$  es

$$\dim_{\mathbb{C}}(\text{Lin}(H)) = N^2 \iff \dim_{\mathbb{R}}(\text{Lin}(H)) = 2N^2$$

### 1.3.4 Clases de operadores

Dentro de  $\text{Lin}(H)$  podemos definir subconjuntos de operadores que mantienen ciertas propiedades importantes. Algunos formarán subespacios vectoriales, y otros no.

Podemos definir dichas **clases de operadores** pidiendo que satisfagan alguna *condición* o *restricción*.

#### Operador Unitario

Definición:

Un operador unitario  $U$  es tal que su adjunto es igual a su inverso

$$U^\dagger = U^{-1}$$

Naturalmente, esta ecuación se traduce en la misma ecuación para las matrices asociadas en *cualquier base*

$$(U_{ij})^\dagger = U_{ji}^* = U_{ij}^{-1}$$

Veamos ahora por qué hemos definido esta clase de operadores.

Teorema:

La acción de un operador unitario conserva intacto el producto escalar de dos vectores cualesquiera.

#### Demostración

Sea  $U$  un operador unitario, y  $|\varphi'\rangle = U|\varphi\rangle$  y  $|\psi'\rangle = U|\psi\rangle$  dos vectores transformados por  $U$ , entonces

$$\langle\varphi'|\psi'\rangle = (\langle\varphi|U^\dagger)U|\psi\rangle = \langle\varphi|U^\dagger U|\psi\rangle = \langle\varphi|\psi\rangle$$

particularizando para  $|\varphi\rangle = |\psi\rangle$  tenemos que un operador unitario *conserva la norma*.

$$\|U|\varphi\rangle\| = \||\varphi\rangle\|$$

En particular, conserva la norma de cualquier vector.

Por tanto, conserva la *distancia* entre dos vectores  $d(|v\rangle, |w\rangle) = \||v\rangle - |w\rangle\|$ .

En un lenguaje más formal, se dice que un operador unitario es una *isometría*

- La *composición* de operadores unitarios **es** unitaria

$$(UV)^\dagger = V^\dagger U^\dagger = V^{-1}U^{-1} = (UV)^{-1}$$

Matemáticamente esto quiere decir que los operadores unitarios forman un *grupo*

- La combinación lineal de operadores unitarios **no es** unitaria

$$(aU + bV)^\dagger = a^*V^\dagger + b^*U^\dagger = a^*V^{-1} + b^*U^{-1} \neq (aU + bV)^{-1}$$

Matemáticamente esto quiere decir que los operadores unitarios **no forman** un *subespacio vectorial* de  $\text{Lin}(H)$

- Por tanto, los operadores unitarios no forman un subespacio vectorial dentro de  $\text{Lin}(H)$ .

La estructura matemática que forman se denomina grupo: el grupo unitario  $U(d)$  actúa sobre  $H$  de dimensión  $d$ .

- Aun así, forman una *variedad*: un conjunto continuo que se puede parametrizar mediante una colección de parámetros, la *dimensión de la variedad*.
- Como hay una relación 1 a 1 entre un operador una matriz (en una base), esa dimensión será igual a la *dimensión del conjunto de matrices unitarias*

### Ejercicio

Resta de  $\dim_{\mathbf{R}}(\mathcal{L}(H)) = 2N^2$  el número de ecuaciones que restringen la matriz de un operador unitario y halla así la dimensión (real) del grupo  $U(N)$  de operadores unitarios de dimensión  $N$ .

### Bases ortonormales

- Como caso particular, aplicando un operador unitario  $U$  a una base ortonormal  $\{|e_i\rangle\}$  obtenemos otra base ortonormal  $\{|\tilde{e}_i\rangle\}$

$$\left. \begin{array}{l} |\tilde{e}_i\rangle = U|e_i\rangle \\ U^{-1} = U^\dagger \end{array} \right\} \iff \langle \tilde{e}_i | \tilde{e}_j \rangle = \langle \tilde{e}_i | U^\dagger U | \tilde{e}_j \rangle = \langle e_i | e_j \rangle = \delta_{ij}$$

- Inversamente, dadas dos bases ortonormales,  $\{|e_i\rangle\}$  y  $\{|\tilde{e}_i\rangle\}$ , el operador que las relaciona es un operador unitario

$$U = \sum_i |\tilde{e}_i\rangle \langle e_i| \Rightarrow U|e_j\rangle = |\tilde{e}_j\rangle$$

$$U^\dagger = \sum_i |e_i\rangle \langle \tilde{e}_i| \Rightarrow U^\dagger |\tilde{e}_j\rangle = |e_j\rangle \Rightarrow U^\dagger = U^{-1}$$

- Un operador ortogonal es un caso particular de operador unitario con *elementos de matriz reales*. El operador de rotación  $R(\theta)$  que hemos estudiado al comienzo de este tema es un operador ortogonal. Es inmediato comprobar que

$$R(\theta)^\dagger = R(\theta)^t = R(-\theta) = R(\theta)^{-1}$$

```
U=np.matrix([[1,1j],[1j, + 1]])/np.sqrt(2)
array_to_latex(U)
```

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}i}{2} \\ \frac{\sqrt{2}i}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

```
Uadj=U.getH() # getH es un método de la clase matrix que devuelve la matriz
↪ conjugada hermitica
array_to_latex(Uadj)
```

$$\begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}i}{2} \\ -\frac{\sqrt{2}i}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

```
print('comprobamos que U es unitaria')
array_to_latex(np.dot(Uadj,U))
```

comprobamos que  $U$  es unitaria

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

## Operador Normal

Definición:

un operador  $N$  es normal si conmuta con su adjunto

$$NN^\dagger = N^\dagger N$$

La matrix  $N_{ij}$  posee una propiedad importante

Teorema:

$N_{ij}$  es la matriz de un operador normal,  $[N, N^\dagger] = 0$ , si y sólo si es unitariamente equivalente a una matriz diagonal. Es decir, si existe  $U$  con  $U^{-1} = U^\dagger$  tal que

$$N'_{ij} = (U^\dagger \cdot N \cdot U)_{ij} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_D \end{bmatrix}$$

## Operador Hermítico

Definición:

un operador  $H$  es Hermítico (o autoadjunto) si verifica la siguiente ecuación

$$H = H^\dagger$$

Evidentemente, un *operador hermítico*  $\Rightarrow$  es un *operador normal*, pero a la inversa no tiene por qué ser verdad.

La *combinación lineal* de operadores *hermíticos* con coeficientes *reales* **es** *hermítica*

$$C^\dagger = (aA + bB)^\dagger = a^*A^\dagger + b^*B^\dagger = aA + bB = C$$

Matemáticamente: los operadores autoadjuntos forman un subespacio vectorial real  $\text{Her}(H) \subset \text{Lin}(H)$ .

La composición de operadores hermíticos, en general **no es** hermítica

$$(AB)^\dagger = B^\dagger A^\dagger = BA \neq AB$$

Matemáticamente, no forman grupo salvo que  $A$  y  $B$  conmuten entre sí, en cuyo caso forman un *grupo Abelian*

La matriz asociada a un operador hermítico también se llama hermítica, y coincide con su traspuesta y conjugada

$$A_{ij} = A_{ij}^\dagger \equiv A_{ij}^{*t} = A_{ji}^*$$

---

### Notar

A partir de cualquier operador  $C \neq C^\dagger$  siempre podemos construir un operador hermítico  $H = H^\dagger$  mediante la combinación lineal

$$H = C + C^\dagger$$

donde  $a$  es un número real. Esto se extiende trivialmente a las matrices que los representan en cualquier base

$$H_{ij} = C_{ij} + C_{ji}^*$$

---

### Ejercicio

Resta de  $\dim_{\mathbf{R}}(\mathcal{L}(H)) = 2N^2$  el número de ecuaciones que restringen la matriz de un operador hermítico y halla así la dimensión (real) de la subespacio vectorial de operadores hermíticos.

---

Si has hecho los dos últimos ejercicios habrás encontrado la misma respuesta en ambos. Eso quiere decir que podría haber una relación entre matrices hermíticas y unitarias.

### Operador semi-definido positivo

Definición:

Decimos que un operador  $A$  es semi-definido positivo (o, no-negativo) si verifica que

$$\langle u|A|u\rangle \geq 0$$

para todo  $|u\rangle \in H$ .

En este caso se dice que  $A \geq 0$ . Si la desigualdad es estricta  $\langle u|A|u\rangle > 0 \forall |u\rangle$ , decimos que  $A$  es un *operador positivo*,  $A > 0$ .

El siguiente teorema es el equivalente al hecho de que un número real  $a \in \mathbb{R}$  admite una raíz si y solo si es no-negativo.

Teorema:

Un  $A$  hermítico, es semi-definido positivo existe otro operador  $B$  tal que

$$A = B^\dagger B$$

para todo  $|u\rangle \in H$ .

### Proyectores

El operador  $P = |u\rangle\langle u|$  proyecta cualquier vector en la dirección de  $|u\rangle$

$$P|w\rangle = |u\rangle\langle u|w\rangle = a|u\rangle$$

donde  $a = \langle u|w\rangle$  es la *proyección*

De su forma se siguen dos propiedades que caracterizan un operador de proyección

- es hermítico

$$P^\dagger = |u\rangle\langle u| = P$$

- es idempotente

$$P^2 = |u\rangle\langle u|u\rangle\langle u| = |u\rangle\langle u| = P$$

Definición:

Un proyector es un operador hermítico que verifica la ecuación

$$P^2 = P$$

---

### Notar

- $P^2 = P = P^\dagger = P^{-1}$
- El proyector es un operador no-unitario: en general, la proyección reduce la norma

$$\|P|w\rangle\|^2 = \langle w|P^\dagger P|w\rangle = \langle w|P|w\rangle = \langle w|u\rangle\langle u|w\rangle = |\langle u|w\rangle|^2 < \|u\|\|w\| = 1$$

donde hemos aplicado la desigualdad de Cauchy Schwarz estricta si suponemos que  $|u\rangle \neq |w\rangle$ .

---

### Matriz asociada a un proyector

Si  $|u\rangle = |e_1\rangle$  el operador  $P_1 = |e_1\rangle\langle e_1|$  proyecta cualquier vector sobre su componente a lo largo de  $|e_1\rangle$ .

En forma matricial

$$|e_1\rangle\langle e_1| = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}$$

de modo que

$$|e_1\rangle\langle e_1|u\rangle = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} = \begin{pmatrix} u_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = u_1|e_1\rangle$$

Si  $|u\rangle = \sum_i u_i|e_i\rangle$  es un vector unitario  $\|u\rangle\| = 1$ , entonces el proyector a lo largo de  $|u\rangle$  viene dado por

$$P(u) = |u\rangle\langle u| = \sum_{i,j} u_i u_j^* |e_i\rangle\langle e_j|$$

Es decir, le está asociada una matriz dada por  $P_{ij} = u_i u_j^*$ . Es trivial verificar que

$$P_{ik}^2 = \sum_j P_{ij} P_{jk} = \sum_j u_i u_j^* u_j u_k^* = u_i \left( \sum_j u_j^* u_j \right) u_k^* = u_i u_k^* = P_{ik}$$

como corresponde a un proyector.

## Proyector ortogonal

Sea  $P = |u\rangle\langle u|$  un proyector a lo largo de un vector  $|u\rangle$ . Entonces el operador  $\Rightarrow P_{\perp} = I - P$  verifica que

- es proyector

$$P_{\perp}^2 = P_{\perp}$$

- es perpendicular a  $P$

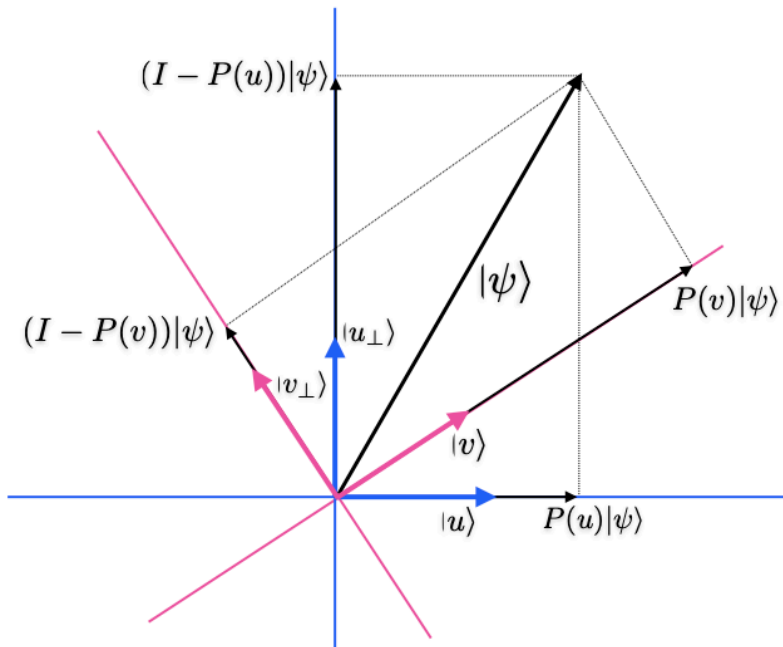
$$P_{\perp}P = (I - P)P = P - P^2 = P - P = 0 \quad (1.8)$$

Dado un vector  $|\psi\rangle$ , podemos descomponer cualquier otro vector  $|\psi\rangle$  en sus proyecciones paralela y perpendicular

$$|\psi\rangle = (P + P_{\perp})|\psi\rangle = a|u\rangle + b|u_{\perp}\rangle$$

donde  $a = \langle u|\psi\rangle$  y  $b = \langle u_{\perp}|\psi\rangle$

La ecuación  $|\psi\rangle = a|u\rangle + b|u_{\perp}\rangle$  quiere decir que los tres vectores están en un mismo hiperplano



```
d = 3

' generate a random vector'
u = tq.random_ket(d)
display(array_to_latex(u))

' build the parallel and perpendicular projectors'
P_par = tq.ket_bra(u,u);
P_perp = np.identity(d) - P_par

display(array_to_latex(P_par))
display(array_to_latex(P_perp))
```



$$\begin{bmatrix} 0.471010764 - 0.3033902688i \\ 0.239801871 + 0.4636412817i \\ -0.429936505 + 0.478319591i \end{bmatrix}$$

$$\begin{bmatrix} 0.313896795 & -0.0277149906 - 0.2911335884i & -0.347622231 - 0.0948551242i \\ -0.0277149906 + 0.2911335884i & 0.2724681755 & 0.1186691299 - 0.3140382451i \\ -0.347622231 + 0.0948551242i & 0.1186691299 + 0.3140382451i & 0.4136350295 \end{bmatrix}$$

$$\begin{bmatrix} 0.686103205 & 0.0277149906 + 0.2911335884i & 0.347622231 + 0.0948551242i \\ 0.0277149906 - 0.2911335884i & 0.7275318245 & -0.1186691299 + 0.3140382451i \\ 0.347622231 - 0.0948551242i & -0.1186691299 - 0.3140382451i & 0.5863649705 \end{bmatrix}$$

```
' check properties P^2 = P, and orthogonality '
A = P_par@P_par - P_par
B = P_perp@P_perp - P_perp
C = P_par@P_perp

display(array_to_latex(C))

' obtain parallel and perpendicular components of another vector '
v = tq.random_ket(d)

v_par = np.dot(P_par,v)
v_perp = np.dot(P_perp,v)

' check perpendicularity '
print(np.round(tq.braaket(v_par,v_perp),4))
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

-0j

### Ejercicio (reflector)

Dado un vector unitario  $|u\rangle$ , escribe

- el operador  $R_u^\perp$  que refleja la componente perpendicular a  $|u\rangle$  de cualquier vector  $|\psi\rangle$
- el operador  $R_u^\parallel$  que refleja la componente paralela a  $|u\rangle$  de cualquier vector  $|\psi\rangle$

## Proyectores sobre un subespacio

Consideremos una base ortonormal  $\{|e_i\rangle\}$ ,  $i = 1, \dots, N$  de  $H$  y dividámosla en dos subconjuntos

$$\{|e_i\rangle\}, i = 1, \dots, N_1 \quad , \quad \{|e_{j+N_1}\rangle, j = 1, \dots, N_2$$

Cualquier vector admite una descomposición ortogonal

$$|\psi\rangle = \sum_{i=1}^N a_i |e_i\rangle = \sum_{i=1}^N a_i |e_i\rangle + \sum_{i=1}^{N_2} a_{i+N_1} |e_{i+N_1}\rangle \equiv |\psi_1\rangle + |\psi_2\rangle \quad (1.9)$$

con  $\langle\psi_1|\psi_2\rangle = 0$ .

Decimos que el espacio  $H$  se descompone en la *suma directa de subespacios ortogonales*

$$H = H_1 \oplus H_2$$

de dimensiones  $N_1 + N_2 = N$ ,

Los operadores

$$P_1 = \sum_{i=1}^{N_1} |e_i\rangle\langle e_i| \quad , \quad P_2 = \sum_{i=1}^{N_2} |e_{i+N_1}\rangle\langle e_{i+N_1}| = I - P_1$$

Verifican la ecuación que define un proyector

$$P_1^2 = P_1 \quad , \quad P_2^2 = P_2$$

Su acción extrae de un vector su componente en el subespacio asociado

$$\begin{aligned} P_1 |\psi\rangle &= \sum_{i=1}^{N_1} |e_i\rangle\langle e_i| \left( \sum_{k=1}^N a_k |u_k\rangle \right) = \sum_{i=1}^{N_1} a_i |e_i\rangle = |\psi_1\rangle \\ P_2 |\psi\rangle &= \sum_{i=1}^{N_2} |e_{i+N_1}\rangle\langle e_{i+N_1}| \left( \sum_{k=1}^N a_k |u_k\rangle \right) = \sum_{i=1}^{N_2} a_{i+N_1} |e_{i+N_1}\rangle = |\psi_2\rangle \end{aligned}$$

Claramente verifican

$$(P_1 + P_2)|\psi\rangle = |\psi\rangle$$

En resumen, vemos que satisfacen

$$P_1 P_2 = P_2 P_1 = 0 \quad , \quad P_1 + P_2 = I$$

Notar que  $P_1 \neq P_v = |v\rangle\langle v|$  donde  $|v\rangle = \sum_i |i\rangle$ . Este operador proyectaría cualquier vector en la dirección de  $|v\rangle$ .

## 1.3.5 Autovalores y autovectores

Definición: Autovalores y autovectores

Existen vectores,  $|\lambda\rangle$  para los cuales la acción de un operador  $A$  devuelve un vector paralelo

$$A|\lambda\rangle = \lambda|\lambda\rangle$$

Decimos que  $|\lambda\rangle$  es un autovector (o vector propio) de  $A$  con autovalor (o valor propio) asociado  $\lambda \in \mathbb{C}$

Supongamos que  $A$  tiene  $d$  vectores propios  $|\lambda_j\rangle = \sum_i v_{ij}|e_i\rangle$ ,  $j = 1, \dots, d$

Sea  $U_{ij} = v_{ij}$  la matriz formada por las componentes de los vectores propios (apilados por columnas)

Entonces

$$A_{diag} = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{pmatrix} = U^{-1}AU$$

Si  $A$  es normal, la matriz  $U$ , que diagonaliza, es unitaria  $U^{-1} = U^\dagger$

```
d = 2
' en general una matriz no será normal '
A = np.matrix(np.random.rand(d,d)+ np.random.rand(d,d) * 1j)
display(array_to_latex(np.dot(A,A.getH()) - np.dot(A.getH(),A)))

eigvals, eigvecs = np.linalg.eig(A)
print('valprop =',eigvals)

'verificamos que los autovectores son las columnas de v'
m=1 #cambiar a otro valor
display(array_to_latex(np.dot(A, eigvecs[:, m]) - eigvals[m] * eigvecs[:, m],
    prefix=r'A|\lambda_m \rangle - \lambda_m |\lambda_m \rangle = '))

' diagonalizamos A '
U = np.matrix(eigvecs);

array_to_latex(np.dot(U.getI(),np.dot(A,U)),prefix='A_{diag} = U^{-1} A U = ')

'U no es unitaria'
array_to_latex(np.dot(U,U.getH()))
```

$$\begin{bmatrix} -0.7295770872 & -0.2507132779 + 0.2382776812i \\ -0.2507132779 - 0.2382776812i & 0.7295770872 \end{bmatrix}$$

```
valprop = [0.04510097+0.3274616j  0.81932658+0.99592662j]
```

$$A|\lambda_m\rangle - \lambda_m|\lambda_m\rangle = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0.5087716298 & -0.1688066649 + 0.1604337075i \\ -0.1688066649 - 0.1604337075i & 1.4912283702 \end{bmatrix}$$

```
d = 2
'veamos ahora una matriz normal'
A = np.matrix([[1, -1], [1, 1]])
display(array_to_latex(np.dot(A,A.getH()) - np.dot(A.getH(),A)))

eigvals, eigvecs = np.linalg.eig(A)
print('valprop =',eigvals)

'verificamos que los autovectores son las columnas de v'
m=1 #cambiar a otro valor
display(array_to_latex(np.dot(A, eigvecs[:, m]) - eigvals[m] * eigvecs[:, m],
    prefix=r'A|\lambda_m \rangle - \lambda_m |\lambda_m \rangle = '))
```

(continues on next page)

(continued from previous page)

```
' diagonalizamos A '
U = np.matrix(eigvecs);

array_to_latex(np.dot(U.getH(),np.dot(A,U)),prefix='A_{diag} = U^{\dagger} A U =
↪')

'U es unitaria'
array_to_latex(np.dot(U,U.getH()))
```

```
<>:17: SyntaxWarning: invalid escape sequence '\d'
<>:17: SyntaxWarning: invalid escape sequence '\d'
/tmp/ipykernel_19338/3458884779.py:17: SyntaxWarning: invalid escape sequence '\d'
array_to_latex(np.dot(U.getH(),np.dot(A,U)),prefix='A_{diag} = U^{\dagger} A U
↪= ')
```

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

```
valprop = [1.+1.j 1.-1.j]
```

$$A|\lambda_m\rangle - \lambda_m|\lambda_m\rangle = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

## Subespacio Propio

Decimos que un autovalor  $\lambda_k$  es  $d_k$  veces degenerado si existen  $d_k$  autovectores linealmente independientes,  $|\lambda_k^a\rangle$  con  $a = 1, \dots, d_k$  asociados al **mismo** autovalor

$$A|\lambda_k^a\rangle = \lambda_k|\lambda_k^a\rangle$$

Dichos autovectores generan un subespacio propio  $S(\lambda_k) \subset H$ .

Por ejemplo, sea  $|u\rangle = \sum_{a=1}^{d_k} c_a |\lambda_k^a\rangle$  una combinación de dichos vectores propios, entonces

$$A|u\rangle = \sum_{a=1}^{d_k} c_a A|\lambda_k^a\rangle = \sum_{a=1}^{d_k} c_a \lambda_k |\lambda_k^a\rangle = \lambda_k \sum_{a=1}^{d_k} c_a |\lambda_k^a\rangle = \lambda_k |u\rangle \quad (1.10)$$

Por tanto  $|u\rangle \in S(\lambda_k)$ .

- El teorema de Gramm-Schmidt garantiza que podemos elegir (mediante un cambio adecuado) el conjunto  $\{|\lambda_k^a\rangle\} \in (\lambda_k), a = 1, \dots, d_k$  de forma que sea una base ortonormal

$$\langle \lambda_k^a | \lambda_k^b \rangle = \delta_{ab}$$

- El **proyector ortogonal** sobre el subespacio propio  $S(\lambda_k)$  será

$$P_k = \sum_{a=1}^{d_k} |\lambda_k^a\rangle \langle \lambda_k^a|$$

### Ejemplo

- Llamemos  $R_z(\theta)$  el operador que efectúa una rotación en el plano  $(x, y)$  de ángulo  $\theta$ . Cuando  $\theta = \pi$  encontramos la siguiente acción sobre los tres elementos  $\{\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}}\}$  de la base cartesiana

$$R_z(\pi)\hat{\mathbf{x}} = -\hat{\mathbf{x}} \quad (1.11)$$

$$R_z(\pi)\hat{\mathbf{y}} = -\hat{\mathbf{y}} \quad (1.12)$$

$$R_z(\pi)\hat{\mathbf{z}} = +\hat{\mathbf{z}} \quad (1.13)$$

- Vemos que hay un autovector  $\hat{\mathbf{z}}$  con autovalor  $+1$  y dos autovectores  $\hat{\mathbf{x}}$  y  $\hat{\mathbf{y}}$  con autovalor  $-1$ .
- El espacio  $\mathbb{R}^3$  se divide en dos subespacios propios de  $R_z(\pi)$ , uno de dimensión 1 (a lo largo del eje  $\hat{\mathbf{z}}$ ) y otro de dimensión 2 (en el plano  $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ ).
- Los proyectores asociados serán

$$P_{\hat{\mathbf{z}}} = |\hat{\mathbf{z}}\rangle \langle \hat{\mathbf{z}}| = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix}, \quad P_{\hat{\mathbf{x}}\hat{\mathbf{y}}} = |\hat{\mathbf{x}}\rangle \langle \hat{\mathbf{x}}| + |\hat{\mathbf{y}}\rangle \langle \hat{\mathbf{y}}| = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix},$$

## Espectro de Operadores

### Espectro de Operadores Normales

Recordemos la definición de un operador normal.  $N$  será un operador normal si conmuta con su adjunto

$$NN^\dagger = N^\dagger N$$

La importancia de los operadores normales radica en el siguiente lema

Teorema:

los autovectores de un operador normal asociados a dos autovalores distintos son ortogonales

$$\lambda_i \neq \lambda_j \iff \langle \lambda_i | \lambda_j \rangle = 0$$

### Prueba:

De la ecuación de autovalores  $N|\lambda_j\rangle = \lambda_j|\lambda_j\rangle$ , y de  $NN^\dagger = N^\dagger N$ , se sigue que

$$\langle \lambda_j | (N^\dagger - \lambda_j^*)(N - \lambda_j) | \lambda_j \rangle = \langle \lambda_j | (N - \lambda_j)(N^\dagger - \lambda_j^*) | \lambda_j \rangle = 0$$

de donde obtenemos  $(N^\dagger - \lambda_j^*)|\lambda_j\rangle = 0 \Rightarrow \langle \lambda_j | N = \langle \lambda_j | \lambda_j$ . Entonces

$$\langle \lambda_j | N | \lambda_i \rangle = \lambda_j \langle \lambda_j | \lambda_i \rangle = \lambda_i \langle \lambda_j | \lambda_i \rangle,$$

de donde se sigue que, para  $\lambda_i \neq \lambda_j \Rightarrow \langle \lambda_i | \lambda_j \rangle = 0$ .

En general, cada autovalor  $\lambda_k$  será  $d_k \geq 1$  veces degenerado.

En ese caso hay  $\{|\lambda_k^a\rangle\}$ ,  $a = 1, \dots, d_k$  autovectores que generan el subespacio propio,  $S(\lambda_k) \subset H$ , de dimensión  $d_k$ .

Subespacios  $S(\lambda_k) \perp S(\lambda_j)$  son ortogonales para  $k \neq j$  según el lema.

En resumen: siempre podemos encontrar una base ortonormal de  $H$ , formada por autovectores de un operador normal  $N$

$$I = \sum_k \sum_{a=1}^{d_k} |\lambda_k^a\rangle \langle \lambda_k^a| \quad ; \quad \langle \lambda_j^a | \lambda_k^b \rangle = \delta_{ab} \delta_{jk}$$

El proyector sobre el subespacio propio  $S(\lambda_k)$  será

$$P_k = \sum_{a=1}^{d_k} |\lambda_k^a\rangle \langle \lambda_k^a|$$

## Espectro de Operadores Hermíticos

El espectro de un operador hermítico  $A = A^\dagger$ , tiene dos propiedades importantes:

- 1. los autovalores de un operador hermíticos son reales  $\lambda_i \in \mathbb{R}$ .
- 2. los autovectores  $|\lambda_i\rangle$  de un operador hermítico asociados a autovalores distintos son ortogonales

$$\lambda_i \neq \lambda_j \iff \langle \lambda_i | \lambda_j \rangle = 0.$$

### Prueba:

1. Tomemos un autovector normalizado de  $A$ ,  $|\lambda\rangle$  de autovalor  $\lambda$ .

$$\lambda = \langle \lambda | A | \lambda \rangle = (\langle \lambda | A^\dagger | \lambda \rangle)^* = (\langle \lambda | A | \lambda \rangle)^* = \lambda^*.$$

2. De hecho esta propiedad es cierta para operadores  $N$ , normales. Los operadores Hermíticos son normales. De la ecuación de autovalores  $N|\lambda_j\rangle = \lambda_j|\lambda_j\rangle$ , y de  $NN^\dagger = N^\dagger N$ , se sigue que

$$\langle \lambda_j | (N^\dagger - \lambda_j^*)(N - \lambda_j) | \lambda_j \rangle = \langle \lambda_j | (N - \lambda_j)(N^\dagger - \lambda_j^*) | \lambda_j \rangle = 0$$

de donde obtenemos  $(N^\dagger - \lambda_j^*)|\lambda_j\rangle = 0 \Rightarrow \langle \lambda_j | N = \langle \lambda_j | \lambda_j$ . Entonces

$$\langle \lambda_j | N | \lambda_i \rangle = \lambda_j \langle \lambda_j | \lambda_i \rangle = \lambda_i \langle \lambda_j | \lambda_i \rangle,$$

de donde se sigue que, para  $\lambda_i \neq \lambda_j \Rightarrow \langle \lambda_i | \lambda_j \rangle = 0$ . }

El conjunto de autovectores  $|\lambda_i\rangle$  de un operador hermítico forma una base ortogonal. Puede normalizarse para formar una base ortonormal

$$\langle \lambda_i | \lambda_j \rangle = \delta_{ij}$$

---

### Ejercicio

Escribe una función, *random\_hermitian*, que genere una matriz hermítica de dimensión  $d$ .

Comprueba en distintos casos que el espectro es real.

---

## Espectro de Operadores Unitarios

Los autovalores de un operador unitario son fases puras

$$U^\dagger = U^{-1} \iff \lambda_i = e^{i\phi_i}$$

### Prueba

aquí tu prueba

## Espectro de Proyectores

Lema:

los autovalores de un proyector sólo pueden ser 0 ó 1

$$P^2 = P \iff \lambda_i \in \{0, 1\}$$

### Prueba

La ecuación

$$P^2 = P \implies P^2|u\rangle = P|u\rangle$$

sólo tiene dos soluciones consistentes

$$P|u\rangle = |u\rangle \quad \text{y} \quad P|u\rangle = 0$$

## Operadores que conmutan

Cuando dos operadores conmutan se dan ciertas propiedades algebraicas que son muy ventajosas. En cierto modo se parecen más a c-números. Veamos la primera.

Teorema:

Dados dos operadores  $A$  y  $B$  que conmutan, existe una base  $\{|\lambda_i\rangle\}$  de autovalores simultáneos de ambos operadores, es decir

$$A = \lambda_i^A |\lambda_i\rangle\langle\lambda_i| \quad , \quad B = \lambda_i^B |\lambda_i\rangle\langle\lambda_i|$$

## Prueba

Supongamos que  $A$  y  $B$  conmutan. Entonces la acción de  $A$  estabiliza los subespacios propios de  $B$ .

Es decir, si  $|\lambda\rangle$  es autoestado de  $B$ , entonces  $B|\lambda\rangle = |\mu\rangle$  también es autoestado con idéntico autovalor. Se comprueba fácilmente

$$A(B|\lambda\rangle) = B(A|\lambda\rangle) = B(\lambda|\lambda\rangle) = \lambda(B|\lambda\rangle)$$

Por tanto  $|\lambda\rangle$  y  $B|\lambda\rangle$  pertenecen al mismo subespacio propio. Esto es lo que se entiende por estabilizar el subespacio.

Si  $\lambda$  es degenerado esto sólo asegura que  $B|\lambda\rangle = |\lambda'\rangle$  pertenece al subespacio propio del mismo autovalor  $\lambda$ .

Esto quiere decir que, dentro de cada subespacio propio de  $B$ , podemos escoger la base que queramos. En particular podemos escoger una base que diagonalice  $A$  dentro de dicho subespacio.

En otras palabras, dos operadores que conmutan son diagonalizables simultáneamente. Su matriz en la base  $\{|\lambda_i\rangle\}$  es

$$A = \begin{bmatrix} \lambda_1^A & & & \\ & \lambda_2^A & & \\ & & \ddots & \\ & & & \lambda_n^A \end{bmatrix}, \quad B = \begin{bmatrix} \lambda_1^B & & & \\ & \lambda_2^B & & \\ & & \ddots & \\ & & & \lambda_n^B \end{bmatrix}.$$

## Valores singulares

Sea  $A \in \text{Lin}(H)$  un operador general. Entonces  $A^\dagger A$  es un operador hermítico que tiene autovalores positivos o cero  $\lambda_i \geq 0$

$$A^\dagger A|\lambda_i\rangle = \lambda_i|\lambda_i\rangle \Rightarrow \lambda_i = \langle \lambda_i | A^\dagger A | \lambda_i \rangle = \|A|\lambda_i\rangle\|^2 \geq 0$$

Por tanto, podemos tomar su raíz cuadrada  $s_i = \sqrt{\lambda_i} \geq 0$

Definición:

Se definen los valores singulares de  $A \in \text{Lin}(H)$  como  $s_i = \sqrt{\lambda_i}$  donde  $\lambda_i$  son los autovalores del operador  $A^\dagger A$ .

Los valores singulares son muy importantes para caracterizar la diferencia entre operadores, como veremos más adelante.

## Representación espectral de proyectores

Sea  $A$  una matrix normal con autovalores  $\lambda_i$  y autovectores  $|\lambda_{i,p}\rangle, p = 1, \dots, g_i = \text{deg}(\lambda_i)$ ,

$$P_i = \sum_{p=1}^{g_i} |\lambda_{i,p}\rangle \langle \lambda_{i,p}|$$

es un proyector sobre el subespacio propio asociado a  $\lambda_i$ .

Teorema:

Supongamos que sólo conocemos el espectro de autovalores  $\{\lambda_i\}$  de  $A$  y no sus autovectores. Aun así el proyector  $P_i$  se puede escribir como

$$P_i = \prod_{k \neq i} \frac{A - \lambda_k I}{\lambda_i - \lambda_k}$$



Por construcción, se cumple que  $P_i |\lambda_j\rangle = \delta_{ij} |\lambda_j\rangle$  ,  $AP_i = \lambda_i A$  .

Notar que el rango de  $P_i$  está incluido por construcción sin necesidad de conocer la base que genera el subespacio  $\{|\lambda_{i,p}\rangle\}$ .

## Descomposición de Operadores

### Descomposición espectral

Teorema: Descomposición espectral

Para todo operador normal  $N$  existe una base de autovectores ortonormales,  $\{|\lambda_k^a\rangle\}$ , tales que  $A$  admite la siguiente descomposición espectral

$$N = \sum_{k=1}^d \lambda_k P_k .$$

Aquí  $d = \dim(H)$  y  $P_k = \sum_{a=1}^{g_k} |\lambda_k^a\rangle \langle \lambda_k^a|$  es el proyector sobre el subespacio propio  $S(\lambda_k)$ , donde  $\lambda_k$  es  $g_k$  veces degenerado.

```
A = np.array([[1, 1], [-1, 1]])
array_to_latex(A)
```

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

```
' Realizamos la descomposición espectral'
eigvals, eigvecs = np.linalg.eig(A)

eigvec0 = eigvecs[:,0]
P0 = tQ.ket_bra(eigvec0,eigvec0)
display(array_to_latex(P0,prefix='P_0='))

eigvec1 = eigvecs[:,1]
P1 = tQ.ket_bra(eigvec1,eigvec1)
display(array_to_latex(P1,prefix='P_1='))

'verificamos completitud'
array_to_latex(P0+P1,prefix='P_0 + P_1=')
```

$$P_0 = \begin{bmatrix} \frac{1}{2} & -\frac{i}{2} \\ \frac{i}{2} & \frac{1}{2} \end{bmatrix}$$

$$P_1 = \begin{bmatrix} \frac{1}{2} & \frac{i}{2} \\ -\frac{i}{2} & \frac{1}{2} \end{bmatrix}$$

$$P_0 + P_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

```
A_descomp_espect = eigvals[0]*P0+eigvals[1]*P1
array_to_latex(A_descomp_espect)
```

$$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

La matriz  $A_{ij}$  que expresa  $A$  en la base  $|\lambda_i\rangle$  es diagonal

$$A_{ij} = \langle \lambda_i^a | A | \lambda_j^b \rangle = \lambda_k \delta_{kj} \delta_{ab} = \begin{bmatrix} \lambda_1 & & & \\ & \ddots & & \\ & & \lambda_2 & \\ & & & \ddots \\ & & & & \lambda_N \end{bmatrix}$$

donde  $\lambda_k$  aparecerá  $d_k$  veces repetido.

---

### Notar

El operador identidad tiene a cualquier vector por autovector  $I|v\rangle = |v\rangle$ , con autovalores  $\lambda_i = 1$ . Por tanto, en cualquier base, la matriz asociada a  $I$  tiene la forma diagonal

$$I_{ij} = \delta_{ij} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

La descomposición espectral de  $I$  no es otra que la relación de completitud, que es cierta *para cualquier base*, ya que todas las bases son bases de autoestados de  $I$

$$I = \sum_{i=1}^N |\lambda_i\rangle\langle\lambda_i| = \sum_{i=1}^N |e_i\rangle\langle e_i|$$


---

### Ejercicio

Escribe una función `spectral_decomp` en python, que devuelva las dos listas  $\lambda_i$  y  $P_i$  asociadas a la descomposición espectral de un operador diagonalizable  $A = \sum_i \lambda_i P_i$ .

---

## Descomposición Polar (PD)

Teorema:

Todo operador  $A \in \text{Lin}(H)$  admite la descomposición polar  $A = UR$  donde  $U$  es un operador unitario, y  $R$  es un operador semi-definido positivo (sólo tiene autovalores positivos o cero)

La descomposición polar es *única* y generaliza la representación polar de números complejos  $z = re^{i\phi}$  a operadores.

El hecho de que  $r \geq 0$  es la contrapartida a que  $R$  sea semi-definida positiva.

El factor  $e^{i\phi}$  es análogo al hecho de que un operador unitario, como veremos, sólo tiene autovalores que son fases puras.

### Ejercicio

Escribe una función `random_unitary` que genere una matriz unitaria de dimensión  $d$ . Comprueba en distintos casos que el espectro está formado por fases

```
'''Método para construir una matriz unitaria arbitraria usando la descomposición
polar'''
d = 3
A = np.matrix(np.random.rand(d,d)+ np.random.rand(d,d) * 1j)

#u, s, vh = linalg.svd(A, full_matrices=False)
u,r = la.polar(A)

R = np.matrix(r)
' verificamos que R sólo tiene autovalores no-negativos '
Reigval, Reigvec = la.eig(R)
print(np.round(Reigval,3))

U=np.matrix(u)
display(Latex('$$U = %s$$'%MatrixToLatex(U)))

''' Verifiquemos unitariedad '''
display(Latex('$$U^{\dagger}U = %s$$'%MatrixToLatex(np.dot(U.getH(),U))))

''' verificamos que los autovalores de U son fases'''
np.round([la.eig(U)[0][i]*la.eig(U)[0][i].conjugate() for i in range(d)],5)
```

```
[2.047-0.j 0.753+0.j 0.217+0.j]
```

```
<>:17: SyntaxWarning: invalid escape sequence '\d'
<>:17: SyntaxWarning: invalid escape sequence '\d'
/tmp/ipykernel_19338/208356260.py:17: SyntaxWarning: invalid escape sequence '\d'
display(Latex('$$U^{\dagger}U = %s$$'%MatrixToLatex(np.dot(U.getH(),U))))
```

$$U = \begin{bmatrix} 0.13 - 0.44j & 0.05 + 0.83j & 0.27 + 0.16j \\ 0.21 + 0.32j & 0.30 - 0.16j & 0.45 + 0.73j \\ 0.24 + 0.77j & 0.13 + 0.41j & 0.07 - 0.40j \end{bmatrix}$$

$$U^\dagger U = \begin{bmatrix} 1.00 + 0.00j & -0.00 - 0.00j & -0.00 - 0.00j \\ -0.00 + 0.00j & 1.00 + 0.00j & 0.00 + 0.00j \\ -0.00 + 0.00j & 0.00 - 0.00j & 1.00 + 0.00j \end{bmatrix}$$

```
array([1.+0.j, 1.+0.j, 1.+0.j])
```

## Descomposición en Valores Singulares (SVD)

Vamos a enunciar este teorema para matrices. Concretamente el teorema habla de una matriz  $m \times n$ .

Este tipo de matrices se corresponden con operadores  $O \in \text{Lin}(H_A, H_B)$  entre espacios de dimensiones  $m$  y  $n$ .

Teorema:

Sea  $A$  una matriz compleja  $m \times n$ . Entonces admite la siguiente forma (descomposición en valores singulares)

$$A = U \Sigma V^\dagger,$$

donde  $U \in U(m)$ ,  $V \in U(n)$  son matrices unitarias cuadradas y  $\Sigma$  es una matriz rectangular  $m \times n$  con los  $s_1, \dots, s_r$  valores singulares no nulos de  $A$  en la diagonal, donde  $r \leq \min(m, n)$ .

*Prueba:* Tomemos  $m > n$  sin pérdida de generalidad.

Sean  $|\lambda_i\rangle$  los  $n$  autovectores de  $A^\dagger A \in H(n)$ , hermítica de dimensión  $n$ .

Supongamos que hay  $r$  valores singulares  $s_i = \sqrt{\lambda_i}$  organicémoslos en orden decreciente  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$  y  $\lambda_{r+1} = \dots = \lambda_n = 0$ .

Con la base ortonormal  $|\lambda_i\rangle$  podemos formar las siguientes matrices

$$V(n \times n) = (|\lambda_1\rangle, |\lambda_2\rangle, \dots, |\lambda_r\rangle, |\lambda_{r+1}\rangle, \dots, |\lambda_n\rangle)$$

$$\Sigma(m \times n) = \begin{pmatrix} \sqrt{\lambda_1} & & & & 0 \\ & \ddots & & & \vdots \\ & & \sqrt{\lambda_r} & & \\ & & & 0 & \\ & & & & \ddots \\ 0 & & \dots & & 0 \\ \vdots & & & & \vdots \\ 0 & & \dots & & 0 \end{pmatrix}$$

$$U(m \times m) = \left( \frac{1}{\sqrt{\lambda_1}} A |\lambda_1\rangle, \frac{1}{\sqrt{\lambda_2}} A |\lambda_2\rangle, \dots, \frac{1}{\sqrt{\lambda_r}} A |\lambda_r\rangle, |\mu_{r+1}\rangle, \dots, |\mu_m\rangle \right).$$

donde  $|\mu_{r+1}\rangle, \dots, |\mu_m\rangle$  completa la base de vectores ortonormales. Ahora podemos operar

$$U \Sigma V^\dagger = \left( \frac{1}{\sqrt{\lambda_1}} A |\lambda_1\rangle, \frac{1}{\sqrt{\lambda_2}} A |\lambda_2\rangle, \dots, \frac{1}{\sqrt{\lambda_r}} A |\lambda_r\rangle, |\mu_{r+1}\rangle, \dots, |\mu_m\rangle \right) \begin{pmatrix} \sqrt{\lambda_1} \langle \lambda_1 | \\ \sqrt{\lambda_2} \langle \lambda_2 | \\ \vdots \\ \sqrt{\lambda_r} \langle \lambda_r | \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$= A \sum_{i=1}^r |\lambda_i\rangle \langle \lambda_i| = A \sum_{i=1}^n |\lambda_i\rangle \langle \lambda_i| = A,$$

donde usamos que  $A |\lambda_i\rangle = 0$  para  $i > r$ .

' numpy tiene la función svd para descomponer en valores singulares'

```
A = np.random.randn(3,2)+ 1j*np.random.randn(3,2)
display(array_to_latex(A,prefix='A='))
print( 'the shape of A is :', A.shape)

u, s, vh = la.svd(A, full_matrices=True)

print( 'the shape of u =',u.shape, ' s =', s.shape, ' v =', vh.shape)
```

$$A = \begin{bmatrix} -0.0244200116 + 0.1041895795i & 0.9349803126 + 0.3421874814i \\ -1.5492338807 - 0.1724318703i & 1.2856329068 - 0.2683114857i \\ 0.9646564048 - 0.0099417172i & -0.366942065 + 1.0589036124i \end{bmatrix}$$

```
the shape of A is : (3, 2)
the shape of u = (3, 3)  s = (2,)  v = (2, 2)
```

```
U=np.matrix(u)
S=np.matrix(np.diag(s))
V=np.matrix(vh).getH()

display(array_to_latex(U,prefix='U='))
display(array_to_latex(S,prefix='S='))
display(array_to_latex(V,prefix='V='))

'''Verifiquemos unitariedad'''
display(array_to_latex(np.dot(U.getH(),U),prefix='U^{\dagger}U ='))
display(array_to_latex(np.dot(V.getH(),V),prefix='V^{\dagger}V ='))
```

```
<>:10: SyntaxWarning: invalid escape sequence '\d'
<>:11: SyntaxWarning: invalid escape sequence '\d'
<>:10: SyntaxWarning: invalid escape sequence '\d'
<>:11: SyntaxWarning: invalid escape sequence '\d'
/tmp/ipykernel_19338/582162267.py:10: SyntaxWarning: invalid escape sequence '\d'
display(array_to_latex(np.dot(U.getH(),U),prefix='U^{\dagger}U ='))
/tmp/ipykernel_19338/582162267.py:11: SyntaxWarning: invalid escape sequence '\d'
display(array_to_latex(np.dot(V.getH(),V),prefix='V^{\dagger}V ='))
```

$$U = \begin{bmatrix} -0.1773651733 - 0.2087808279i & -0.3952061246 - 0.6550921243i & -0.2332610853 + 0.5340485512i \\ -0.7632131121 - 0.1906945229i & 0.3622448006 - 0.2161066755i & 0.4293049613 - 0.1375325561i \\ 0.5160011558 - 0.1995903629i & -0.1266866838 - 0.4697802904i & 0.5980410426 - 0.3154526409i \end{bmatrix}$$

$$S = \begin{bmatrix} 2.5414687533 & 0 \\ 0 & 0.9407201426 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.6679617784 & -0.7441955808 \\ -0.6169703939 - 0.4161425184i & -0.5537692673 - 0.3735137695i \end{bmatrix}$$

$$U^\dagger U = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$V^\dagger V = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

## Traza

### Traza de un operador

Definición:

La traza de un operador  $A$  se define como la suma

$$\text{tr} A = \sum_i \langle e_i | A | e_i \rangle = \sum_i A_{ii}$$

de sus elementos de matriz diagonales en cualquier base

Para ser consistente esta definición es necesario probar que no importa la base utilizada

Lema:

La traza de un operador es independiente de la base en la que se calcule

### Prueba

$$\begin{aligned} \text{tr} A &= \sum_i A_{ii} = \sum_i \langle i | A | i \rangle = \sum_i \langle i | A \left( \sum_j |\tilde{j}\rangle \langle \tilde{j}| \right) | i \rangle \\ &= \sum_{ij} \langle i | A | \tilde{j} \rangle \langle \tilde{j} | i \rangle = \sum_{ij} \langle \tilde{j} | i \rangle \langle i | A | \tilde{j} \rangle \\ &= \sum_j \langle \tilde{j} | \left( \sum_i | i \rangle \langle i | \right) A | \tilde{j} \rangle = \sum_j \langle \tilde{j} | A | \tilde{j} \rangle \\ &= \sum_j \tilde{A}_{jj} \end{aligned} \tag{1.14}$$

Por esta importante propiedad la traza de  $A$  es la suma de sus autovalores

$$\text{tr} A = \sum_i \langle \lambda_i | A | \lambda_i \rangle = \sum_i \lambda_i$$

siempre que  $A$  sea diagonalizable en la base  $|\lambda_i\rangle$

$$A|\lambda_j\rangle = \lambda_j|\lambda_j\rangle$$

La traza es una operación *lineal*

$$\text{tr}(A + B) = \text{tr} A + \text{tr} B$$

La traza de un producto de operadores tiene la propiedad de *cíclicidad*: es invariante bajo permutaciones cíclicas de los operadores en su argumento. Por ejemplo, para tres operadores  $A, B$  y  $C$

$$\text{tr}(ABC) = \text{tr}(BCA)$$

Sea el operador  $A = |u\rangle\langle v|$  entonces

$$\text{tr} \left( |u\rangle\langle v| \right) = \sum_i \langle e_i | u \rangle \langle v | e_i \rangle = \langle v | \left( \sum_i | e_i \rangle \langle e_i | \right) | u \rangle = \langle v | u \rangle$$

---

### Ejercicio

Demuestra este resultado

Para un producto de dos operadores, el anterior resultado implica que la *traza de un conmutador es cero*. Dicho de otra forma

$$\text{tr}(AB) = \text{tr}(BA) \Rightarrow \text{tr}([A, B]) = 0.$$

### $\text{Lin}(H)$ como un espacio de Hilbert

Para transformar  $\text{Lin}(H)$  en un espacio de Hilbert sólo es necesario definir un producto escalar hermítico

Definición: producto escalar

Dados dos operadores  $A, B \in \text{Lin}(H)$  definimos su producto escalar  $(A, B) \in \mathbb{C}$

$$(A, B) = \text{tr}(A^\dagger B)$$

En una base tenemos que

$$(A, B) = \sum_{ij} A_{ij}^\dagger B_{ji} = \sum_{ij} A_{ji}^* B_{ji}$$

mientras que

$$(B, A) = \sum_{ij} B_{ij}^\dagger A_{ji} = \sum_{ij} B_{ji}^* A_{ji}$$

Se sigue que  $(B, A) = (A, B)^*$ . Además es trivial comprobar que  $(A, B + C) = (A, B) + (A, C)$ , por lo que se trata de un producto escalar sesquilineal o hermítico.

### $\text{Lin}(H)$ como un espacio normado

Una **norma** definida sobre  $\text{Lin}(H)$  es una función real  $A \rightarrow \|A\| \in \mathbb{R}$  con las propiedades que se han definido en una sección anterior

Definición:  $p$ -norma Shatten

Dado un operador  $A \in \text{Lin}(H)$  la función

$$\|A\|_p = \left( \text{tr}(A^\dagger A)^{p/2} \right)^{1/p}$$

define una norma, denominada  $p$ -norma de Shatten.

Los tres casos más frecuentes son

- $p = 1$  **norma de la traza**  $\Rightarrow \|A\|_1 = \text{tr} \sqrt{A^\dagger A}$

Esta norma es igual a la suma de los valores singulares de  $A \Rightarrow \|A\|_1 = \sum_i s_i$ , donde  $s_i^2$  son los autovalores de  $A^\dagger A$

- $p = 2$  **norma de Frobenius**  $\Rightarrow \|A\|_2 = \sqrt{\text{tr} A^\dagger A}$

La norma de Frobenius es la que se obtiene a partir del producto escalar  $\|A\|_2 = (A, A)$

- $p = \infty$  **norma espectral**  $\Rightarrow \|A\|_\infty = \lim_{p \rightarrow \infty} \|A\|_p$

Puede demostrarse que la norma espectral es equivalente a la siguiente definición

$$\|A\|_\infty = \max_{|u\rangle \in H} \{\|A|u\rangle\| \text{ con } \|u\rangle = 1\}$$

---

### Ejercicio

Escribe, en python, una función  $\text{trace\_norm}(A)$ , que calcule la norma de la traza de un operador.

---

### Distancia de traza

Cualquier norma permite definir una noción de *distancia* o *diferencia* entre dos operadores.

Definición: Distancia de traza

Se define la distancia de traza entre dos operadores  $A$  y  $B$  como la norma de la diferencia

$$d(A, B) = \|A - B\|_1$$

## 1.3.6 Funciones de Operadores

### Funciones analíticas

Estamos acostumbrados a escribir funciones *de una variable real o compleja*. Por ejemplo  $f(x) = x^2$ , ó,  $f(z) = e^z$ .

Queríamos dar sentido a una función *de un operador*  $A \rightarrow f(A)$

En el caso de que  $f(z)$  sea una función analítica expresable como una serie de Taylor en torno a  $x = 0$

$$f(z) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) z^n$$

tomaremos como **definición** la *misma serie* cambiando el argumento  $x \rightarrow A$

$$f(A) = \sum_{n=0}^{\infty} \frac{1}{n!} f^{(n)}(0) A^n$$

---

### Notar

De la misma forma que, para funciones analíticas  $f(z)^* = f(z^*)$ , también la definición anterior asegura que  $f(A)^\dagger = f(A^\dagger)$

---



## Exponencial de un operador

la exponencial de un operador será

$$\exp(A) = e^A = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \dots$$

Una propiedad importante de la función exponencial es  $e^x e^y = e^{x+y}$ .

La propiedad análoga para operadores *sólo es cierta cuando conmutan entre sí*.

Para el caso genérico tenemos dos opciones

Teorema:

Sea  $A$  un operador que verifique  $A^2 = I$ , demuestra que se cumple la generalización de la ecuación de Euler

$$e^{i\alpha A} = \cos \alpha I + i \sin \alpha A$$

---

### Ejercicio

Demuestra el teorema anterior

---

Teorema: Baker-Campbell-Hausdorff

Sean  $A, B \in \mathcal{L}(H)$  dos operadores lineales genéricos. Entonces

$$e^A e^B = e^{(A+B+\frac{1}{2}[A,B]+\frac{1}{12}[A,[A,B]]+\frac{1}{12}[B,[B,A]]+\dots)}$$

Vemos que

- Si  $A$  y  $B$  conmutan,

$$[A, B] = 0 \Leftrightarrow e^A e^B = e^{A+B}$$

- Si el conmutador de  $A$  y  $B$  es un c-número

$$[A, B] = cI \Leftrightarrow e^A e^B = e^{A+B+\frac{c}{2}}$$

- El inverso de  $e^A$  es  $e^{-A}$ . Efectivamente, como

$$[A, A] = 0 \Rightarrow e^A e^{-A} = e^{A-A} = e^0 = I$$

Teorema: Lie-Suzuki-Trotter

Sean  $A, B \in \mathcal{L}(H)$  dos operadores lineales genéricos. Entonces

$$e^{A+B} = \lim_{n \rightarrow \infty} (e^{A/n} e^{B/n})^n$$

Esta segunda opción es de uso muy frecuente en el contexto de la *simulación cuántica*.

## Operadores unitarios a partir de hermíticos

Todo operador unitario  $U$  se puede expresar como la exponencial imaginaria de un operador hermítico  $H$

$$U = e^{iH}$$

Efectivamente,

$$U^\dagger = (e^{iH})^\dagger = e^{-iH^\dagger} = e^{-iH} = U^{-1}$$

por tanto,  $U$  es unitario si y sólo si  $H$  es hermítico.

## Prueba

$$U^\dagger = (e^{iH})^\dagger = \left(1 + iH + \frac{1}{2}(iH)^2 + \dots\right)^\dagger \quad (1.15)$$

$$= 1 - iH^\dagger + \frac{1}{2}(-i)^2(H^2)^\dagger + \dots \quad (1.16)$$

$$= 1 - iH + \frac{1}{2}H^2 - \dots \quad (1.17)$$

$$= e^{-iH} \quad (1.18)$$

$$= U^{-1} \quad (1.19)$$

## Funciones generales

No siempre  $f(z)$  es analítica en torno a  $z=0$ . Es el caso, por ejemplo  $f(z) = \exp(1/z)$ . Por tanto, esta función no admite una expansión en serie de Taylor.

Aun así, si la función  $f(z)$  existe, el operador  $f(A)$  también. Para construirlo es necesario recurrir a la forma *forma diagonalizada*

Teorema:

Sea  $A$  un operador diagonalizable, y sea  $A = \sum_i \lambda_i |\lambda_i\rangle\langle\lambda_i|$  su representación espectral. Entonces el operador  $f(A)$  tiene la representación espectral siguiente

$$f(A) = \sum_i f(\lambda_i) |\lambda_i\rangle\langle\lambda_i|$$

En otras palabras, si la matriz  $A_{ij}^{(D)} = \lambda_i \delta_{ij}$  es diagonal, cualquier función de la misma es, trivialmente, la matriz diagonal que se obtiene evaluando  $f(\lambda_i)$

$$f(A_{ij}^{(D)}) = \begin{bmatrix} f(\lambda_1) & & & \\ & f(\lambda_2) & & \\ & & \ddots & \\ & & & f(\lambda_n) \end{bmatrix}$$

---

### Ejemplo 1

$$e^{1/A} = \sum_i e^{1/\lambda_i} |\lambda_i\rangle\langle\lambda_i|$$

### Ejemplo 2

$$\begin{aligned} \text{tr}(A \log A) &= \text{tr} \left[ \left( \sum_j \lambda_j |\lambda_j\rangle \langle \lambda_j| \right) \left( \sum_k \log \lambda_k |\lambda_k\rangle \langle \lambda_k| \right) \right] = \text{tr} \left[ \sum_k \lambda_k \log \lambda_k |\lambda_k\rangle \langle \lambda_k| \right] \\ &= \text{tr} \begin{bmatrix} \lambda_1 \log \lambda_1 & & & \\ & \lambda_2 \log \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \log \lambda_n \end{bmatrix} = \sum_k \lambda_k \log \lambda_k \end{aligned}$$

### 1.3.7 Matrices de Pauli

Las matrices hermíticas forman un subespacio vectorial  $\text{Her}(H) \subset \text{Lin}(H)$  que admite una base de matrices hermíticas

Definición:

Se definen las matrices de Pauli

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

### Notar

También se usan los subíndices enteros  $\sigma_1 = \sigma_x$ ,  $\sigma_2 = \sigma_y$  y  $\sigma_3 = \sigma_z$ .

Si añadimos la matriz identidad  $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  el conjunto  $\{I, \sigma_x, \sigma_y, \sigma_z\}$  forma una *base* para el espacio de *matrices hermíticas*  $2 \times 2$ .

$$A = a_0 I + \mathbf{a} \cdot \boldsymbol{\sigma} = a_0 I + a_1 \sigma_1 + a_2 \sigma_2 + a_3 \sigma_3 = \begin{bmatrix} a_0 + a_3 & a_1 - ia_2 \\ a_1 + ia_2 & a_0 - a_3 \end{bmatrix} = A^\dagger$$

con  $a_i \in \mathbb{R}$  cuatro números reales.

En total, las matrices de Pauli poseen tres propiedades que las hacen únicas

- son hermíticas  $\sigma_i^\dagger = \sigma_i$
- son unitarias  $\sigma_i^\dagger = \sigma_i^{-1} = \sigma_i$
- tienen traza nula  $\text{tr}(\sigma_i) = 0$

### Composición

La composición de dos matrices de Pauli es otra matriz de Pauli que cumple la siguiente identidad

$$\sigma_i \sigma_j = \delta_{ij} I + i \epsilon_{ijk} \sigma_k$$

donde

$$\epsilon_{123} = \epsilon_{231} = \epsilon_{312} = 1 \quad , \quad \epsilon_{213} = \epsilon_{132} = \epsilon_{321} = -1$$

Ejemplos:

$$\begin{aligned}\sigma_1 \sigma_2 &= i \sigma_3 \\ \sigma_2 \sigma_1 &= -i \sigma_3 \\ \sigma_2 \sigma_1 \sigma_2 &= \sigma_2 (i \sigma_3) = i^2 \sigma_1 = -\sigma_1 \\ \sigma_2 \sigma_2 &= I\end{aligned}$$

```
s0 = np.matrix([[1,0],[0,1]])
s1 = np.matrix([[0,1],[1,0]])
s2 = np.matrix([[0,-1j],[1j,0]])
s3 = np.matrix([[1,0],[0,-1]])
```

```
'verifica todas las opciones'
```

```
print(s1*s1==s0)
print(s1*s2==1j*s3)
print(s2*s1== -1j*s3)
print(s2*s1*s2== -s1)
print(s1*s1==s0)
```

```
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
[[ True  True]
 [ True  True]]
```

---

### Nota

De las relaciones de composición anteriores se siguen inmediatamente las siguientes reglas de (anti) conmutación

$$\{\sigma_i, \sigma_j\} = 2\delta_{ij} \quad , \quad [\sigma_i, \sigma_j] = 2i\epsilon_{ijk}\sigma_k$$

---

Las matrices de Pauli tienen traza nula

$$\text{tr } \sigma_i = 0$$

Tomando la traza de la relación de composición obtenemos que las matrices de Pauli *son ortogonales* en el sentido siguiente

$$(\sigma_i, \sigma_j) = \text{tr}(\sigma_i \sigma_j) = \text{tr}(\delta_{ij} I + i\epsilon_{ijk}\sigma_k) = 2\delta_{ij}$$

## Exponenciación

Consideremos la matriz hermítica siguiente

$$A = \mathbf{a} \cdot \boldsymbol{\sigma}$$

con  $a_i \in \mathbb{R}$  tres números reales.

Separemos  $\mathbf{a}$  en un módulo y una dirección unitaria

$$\mathbf{a} = (a_1, a_2, a_3) = a \left( \frac{a_1}{a}, \frac{a_2}{a}, \frac{a_3}{a} \right) = a \hat{\mathbf{n}}$$

donde  $a = |\mathbf{a}| = \sqrt{a_1^2 + a_2^2 + a_3^2}$  y  $\hat{\mathbf{n}}$  es unitario

Entonces

$$\mathbf{a} \cdot \boldsymbol{\sigma} = a \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} = aA$$

Queremos hallar la exponencial  $e^{i\mathbf{a} \cdot \boldsymbol{\sigma}}$ .

Teorema:

$$\exp(i\mathbf{a} \cdot \boldsymbol{\sigma}) = (\cos a) I + i(\sin a) \hat{\mathbf{n}} \cdot \boldsymbol{\sigma}$$

## Ejercicio

- Demuestra que  $(\hat{\mathbf{n}} \cdot \boldsymbol{\sigma})^2 = I$ . Esto convierte al teorema anterior en un corolario.
- Demuestra el teorema a partir de la descomposición espectral de las tres matrices de Pauli,  $\sigma_x, \sigma_y$  y  $\sigma_z$ .

```
from scipy.linalg import expm

avec = np.random.rand(3)
a = np.linalg.norm(avec)
nvec = avec/a
print('a=',a)
display(array_to_latex(nvec))

sigvec = np.array([s1,s2,s3])

adots= sum(list(avec[i]*sigvec[i] for i in range(3)))

'exponentiating'
e1 = expm(1j*adots)

'using the Euler-like formula'
ndots= sum(list(nvec[i]*sigvec[i] for i in range(3)))
e2 = np.cos(a)*s0 + 1j*np.
    sin(a)*(nvec[0]*sigvec[0]+nvec[1]*sigvec[1]+nvec[2]*sigvec[2])
```

(continues on next page)

(continued from previous page)

```
'verify'
display(array_to_latex(e1,prefix='exp(s1)='))
display(array_to_latex(e2,prefix='exp(s1)='))
```

```
a= 0.8196453748381084
```

$$\begin{aligned} & [0.2089228489 \quad 0.2741500062 \quad 0.938718817] \\ \exp(s1) &= \begin{bmatrix} 0.6824804471 + 0.6861131983i & 0.2003772952 + 0.1527025148i \\ -0.2003772952 + 0.1527025148i & 0.6824804471 - 0.6861131983i \end{bmatrix} \\ \exp(s1) &= \begin{bmatrix} 0.6824804471 + 0.6861131983i & 0.2003772952 + 0.1527025148i \\ -0.2003772952 + 0.1527025148i & 0.6824804471 - 0.6861131983i \end{bmatrix} \end{aligned}$$

Feb 13, 2024 | 8 min read

## 1.4 Tensores

```
%run ../../macro_tQ.py
import sys
sys.path.append('../../')
import macro_tQ as tQ

import numpy as np
import scipy.linalg as la
from IPython.display import display,Markdown,Latex
import matplotlib.pyplot as plt
from qiskit.tools.visualization import array_to_latex
```

### 1.4.1 Producto Tensorial

Un computador clásico es capaz de tratar grandes cantidades de bits a base de acumular dispositivos de un solo bit. Cuánticamente consideraríamos una colección de sistemas que, cada uno, implemente un cúbit.

La herramienta matemática para describir un sistema compuesto es el *producto tensorial* de espacios de Hilbert

Definición:

dados dos vectores  $|u\rangle_1 \in H_1$  y  $|v\rangle_2 \in H_2$ , denominamos producto tensorial al par ordenado

$$|u\rangle_1 \otimes |v\rangle_2 \equiv |uv\rangle$$

con la propiedad *bilineal*. Es decir, como el producto ordinario, satisface la propiedad distributiva.

$$(|u\rangle + |v\rangle) \otimes (|y\rangle + |z\rangle) \equiv |u\rangle \otimes |y\rangle + |u\rangle \otimes |z\rangle + |v\rangle \otimes |y\rangle + |v\rangle \otimes |z\rangle \quad (1.20)$$

$$= |uy\rangle + |uz\rangle + |vy\rangle + |vz\rangle \quad (1.21)$$

A continuación construimos un espacio vectorial, que incluya todas las posibles combinaciones lineales de pares ordenados.

Definición:

El espacio producto tensorial  $H = H_1 \otimes H_2$  está formado por todas las combinaciones lineales posibles de pares ordenados

$$|w\rangle = a|u\rangle_1 \otimes |u\rangle_2 + b|v\rangle_1 \otimes |v\rangle_2 + \dots$$

donde  $|u\rangle_1, |v\rangle_1, \dots \in H_1$  y  $|u\rangle_2, |v\rangle_2, \dots \in H_2$ , y  $a, b, \dots \in \mathbb{C}$  son coeficientes complejos.

## Base y Dimensión

Sea  $|i_1\rangle$  una base de  $H_1$  y  $|i_2\rangle$  una base de  $H_2$ . Entonces, una base de  $H_1 \otimes H_2$  se obtiene a partir de *todos* los emparejamientos

$$|i_1 i_2\rangle = |i_1\rangle \otimes |i_2\rangle$$

con  $i_1 = 1, \dots, d_1$  y  $i_2 = 1, \dots, d_2$ . Cada elemento así formado es linealmente independiente de los demás.

El número de parejas posibles es  $d_1 d_2$ , que coincide con la **dimensión** de  $H_1 \otimes H_2$ .

Vemos que las **etiquetas** de los vectores de la base forman un *bi-índice*  $\rightarrow i_1 i_2$  que asume  $d_1 d_2$  parejas de valores distintos

Un *vector* se escribirá igualmente usando  $d_1 d_2$  componentes complejas  $w_{i_1 i_2}$ , etiquetadas mediante un bi-índice en lugar de un índice

$$|\omega\rangle = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} w_{i_1 i_2} |i_1 i_2\rangle = w_{11}|11\rangle + w_{12}|12\rangle + \dots + w_{21}|21\rangle + \dots + w_{d_1 d_2}|d_1 d_2\rangle$$

## Ejemplo

Supongamos que  $d_1 = 2$  y  $d_2 = 3$ , entonces

$$|w\rangle = w_{11}|11\rangle + w_{12}|12\rangle + w_{13}|13\rangle + w_{21}|21\rangle + w_{22}|22\rangle + w_{23}|23\rangle$$

o, en notación matricial

$$|w\rangle \sim \begin{pmatrix} w_{11} \\ w_{12} \\ w_{13} \\ w_{21} \\ w_{22} \\ w_{23} \end{pmatrix}$$

## Notar

- Prescindiremos del subíndice  $|u\rangle_1 \otimes |y\rangle_2 = |u\rangle \otimes |y\rangle \equiv |uy\rangle$  que estará implícito en el orden.
- Para computación cuántica con cúbits (cúbits), el valor relevante es  $d = 2$  ( $d \geq 3$ ).

### Indexación equivalente

Podemos etiquetar las componentes (o los elementos de la base) con índices, en lugar de bi-índices

Para ello basta con definir un mapa entre dos bi-índices  $i_1, i_2$  y un índice  $a$

$$|w\rangle = \sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} w_{i_1 i_2} |i_1 i_2\rangle = \sum_{a=1}^{d_1 d_2} w_a |a\rangle$$

donde hemos querido resaltar que las componente y los vectores son los mismos, etiquetados de forma diferente.

En el caso genera el mapa es

$$i_1 i_2 \rightarrow a = d_2 * (i_1 - 1) + i_2$$

Claramente  $a = 1, \dots, d_1 d_2$  tiene  $d_1 d_2$  valores distintos

### Producto de Kronecker

Como sabemos, cualquier vector admite, en una base, una representación como un vector columna con sus coeficientes como entradas.

$$|w\rangle = \sum_{i,j=1}^2 w_{ij} |e_{ij}\rangle \sim \begin{pmatrix} w_{11} \\ w_{12} \\ w_{21} \\ w_{22} \end{pmatrix}$$

La matriz columna asociada  $|uv\rangle = |u\rangle \otimes |v\rangle$  se forma a partir de las matrices columna de  $|u\rangle$  y  $|v\rangle$  mediante el denominado *producto de Kronecker* o, también *producto tensorial*.

$$|uv\rangle = |u\rangle \otimes |v\rangle \sim \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \otimes \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \equiv \begin{pmatrix} u_1 \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ u_2 \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} u_1 v_1 \\ u_1 v_2 \\ u_2 v_1 \\ u_2 v_2 \end{pmatrix}$$

---

### Notar

Con dos vectores  $|u\rangle = \sum_i u_i |e_i\rangle$  y  $|v\rangle = \sum_i v_i |e_i\rangle \in H$ , hay dos objetos muy parecidos que podemos formar

1. un operador  $\Omega = |u\rangle\langle v| = \sum_{i,j} u_i v_j^* |e_i\rangle\langle e_j| \in \text{Lin}(H)$ .
2. un vector  $|uv\rangle = |u\rangle \otimes |v\rangle = \sum_{i,j} u_i v_j |e_i\rangle |e_j\rangle \in H \otimes H$ .

Nota la conjugación compleja relativa entre las dos matrices de componentes  $\Omega_{ij} = u_i^* v_j$  y  $\omega_{ij} = u_i v_j^*$ .

---

### Ejercicio

Escribe una función `kronecker(u,v)` que tome dos kets (como vectores columna) y devuelva su producto de Kronecker. Verifica el resultado con la función `kron` de `numpy`.

---



## 1.4.2 Factorización y Entrelazamiento

Definición: Vector entrelazado

Decimos que, un vector  $|w\rangle \in H \otimes H$  es factorizable cuando es posible encontrar vectores  $|u\rangle, |v\rangle \in H$  tales que  $|w\rangle = |u\rangle \otimes |v\rangle$ .

Cuando esto no sea posible, decimos que  $|w\rangle$  es un vector entrelazado.

Ya hemos visto que, dada una base  $|e_i\rangle$  de  $H$ , el vector más general que pertenece al espacio producto admite una descomposición

$$|w\rangle = \sum_{i,j=1}^d w_{ij} |e_i\rangle \otimes |e_j\rangle = w_{11} |e_1\rangle \otimes |e_1\rangle + w_{12} |e_1\rangle \otimes |e_2\rangle + \dots$$

Podría ocurrir que en otra base  $|w\rangle = \tilde{w}_{11} |f_1\rangle \otimes |f_1\rangle$  sólo tuviese un término y fuese factorizable.

Discernir si un vector es factorizable o entrelazado no es algo que se pueda hacer a primera vista.

### Criterio de factorizabilidad

El estado es  $|w\rangle$  es factorizable si y sólo si las componentes  $w_{ij}$  son factorizables en la forma  $w_{ij} = u_i v_j$  con  $i = 1, \dots, d_1$  y  $j = 1, \dots, d_2$ .

### Prueba

$$|w\rangle = \sum_{i,j=1}^d w_{ij} |e_{ij}\rangle = \sum_{i,j} u_i v_j |e_i\rangle \otimes |e_j\rangle = \sum_{i,j} u_i |e_i\rangle \otimes v_j |e_j\rangle = \sum_i u_i |e_i\rangle \otimes \sum_j v_j |e_j\rangle = |u\rangle \otimes |v\rangle$$

identidad que se puede leer en ambos sentidos

El carácter entrelazado de un vector es genérico, mientras que el carácter factorizable es accidental.

Esto se sigue de un sencillo conteo:

- como función de  $d$ ,  $\{w_{ij}\}$  forma un conjunto de  $d_1 d_2$  parámetros complejos (grados de libertad).
- sin embargo en  $\{u_i v_j\}$  sólo hay  $d_1 + d_2$  números independientes. Es evidente que  $d_1 d_2 \gg d_1 + d_2$ .

En el caso  $d_1 = d_2 = 2$  la condición de *factorizabilidad*  $\Rightarrow w_{ij} = u_i v_j$  es equivalente a verificar la anulación del determinante de la matriz  $2 \times 2$  formada por las componentes

$$\det w_{ij} = w_{11} w_{22} - w_{12} w_{21} = u_1 v_1 u_2 v_2 - u_1 v_2 u_2 v_1 = 0$$

En el caso general  $d_1, d_2 \geq 2$  la búsqueda de un criterio para detectar si  $w_{ij}$  es factorizable o entrelazado pasa por la descomposición de Schmidt.

## Descomposición de Schmidt

Supongamos que, en sendas bases arbitrarias  $\{|e_{1,i}\rangle, i = 1, \dots, d_1\}$  de  $H_1$  y  $\{|e_{2,a}\rangle, a = 1, \dots, d_2\}$  de  $H_2$  nuestro vector se escribe

$$|w\rangle = \sum_{i=1}^{d_1} \sum_{a=1}^{d_2} w_{ia} |e_{1,i}\rangle \otimes |e_{2,a}\rangle$$

Los valores de las *componentes*  $w_{ia}$  **dependen de las bases escogida**. En *otras* base  $|\tilde{e}_{1,i}\rangle \otimes |\tilde{e}_{2,a}\rangle$  encontraremos *otras* componentes  $\tilde{w}_{ia}$  para el *mismo* vector

Si existe una base en la que  $\tilde{w}_{ia} = 0$  para todos los  $i, a$  menos para uno (por ejemplo  $\tilde{w}_{11} \neq 0$ ), entonces

$$|w\rangle = \tilde{w}_{11} |\tilde{e}_{1,1}\rangle \otimes |\tilde{e}_{2,1}\rangle$$

y, secretamente, el vector  $|w\rangle$  era factorizable.

El siguiente teorema nos permite averiguar **cuánto nos podemos acercar a esta situación**

Teorema: de Schmidt

para cada vector  $|w\rangle \in H_1 \otimes H_2$ , existen sendas bases  $|f_{1,i}\rangle$  de  $H_1$  y  $|f_{2,a}\rangle$  de  $H_2$ , tales que, podemos expresar

$$|w\rangle = \sum_{i=1}^r s_i |f_{1,i}\rangle \otimes |f_{2,i}\rangle,$$

donde  $s_i > 0$ , y la suma involucra el mínimo número,  $r$ , de términos.

El número  $1 \leq r \leq \min(d_1, d_2)$  es rango de  $w$  y se denomina Número de Schmidt. Es la información relevante porque

- cuando  $r = 1$  el estado  $|w\rangle$  será *factorizable*.
- si  $r \geq 2$  el estado será *entrelazado*.

La *demonstración del Teorema de Schmidt* es interesante porque nos da un **método constructivo** para encontrar la descomposición.

Supongamos que nuestro vector se escribe

$$|w\rangle = \sum_{i=1}^{d_1} \sum_{a=1}^{d_2} w_{ia} |e_{1,i}\rangle \otimes |e_{2,a}\rangle$$

La matriz de coeficientes  $w_{ia}$  tiene dimension  $d_1 \times d_2$ . Asumiremos  $d_1 \geq d_2$  sin pérdida de generalidad

El **teorema SVD** de descomposición en valores singulares, nos garantiza que podemos expresar dicha matriz en la forma siguiente

$$w = U \Sigma V^\dagger \Rightarrow w_{ia} = \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} U_{ij} \Sigma_{jb} V_{ab}^*$$

donde  $U$  y  $V$  son unitarias ( $d_1 \times d_1$ ) y ( $d_2 \times d_2$ ) respectivamente, mientras que  $\Sigma$  es diagonal

$$\Sigma_{jb} = \left[ \begin{array}{cccc} \overbrace{\begin{matrix} s_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ & s_r & \\ & 0 & \ddots \end{matrix}}^{d_2} & & \\ & 0 & \cdots & 0 \\ & \vdots & & \vdots \\ 0 & \cdots & & 0 \end{array} \right]_{d_1} \Rightarrow \Sigma_{jb} = s_j \delta_{jb}$$

- Los números  $s_1, \dots, s_r > 0$  son los autovalores de la matriz  $w^\dagger w$  y se denominan valores principales de la matriz  $w_{ij}$

Esto quiere decir que podemos escribir

$$\begin{aligned} |w\rangle &= \sum_{i=1}^{d_1} \sum_{a=1}^{d_2} \left( \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} U_{ij} \Sigma_{jb} V_{ab}^* \right) |e_{1,i}\rangle \otimes |e_{2,a}\rangle \\ &= \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} \Sigma_{jb} \left( \sum_{i=1}^{d_1} U_{ij} |e_{1,i}\rangle \right) \otimes \left( \sum_{a=1}^{d_2} V_{ab}^* |e_{2,a}\rangle \right) \\ &= \sum_{j=1}^{d_1} \sum_{b=1}^{d_2} s_j \delta_{jb} |f_{1,j}\rangle \otimes |f_{2,b}\rangle \\ &= \sum_{j=1}^r s_j |f_{1,j}\rangle \otimes |f_{2,j}\rangle \end{aligned}$$

Por tanto, podemos saber si un *estado bipartito* es entrelazado calculando la descomposición en valores singulares de su matriz de coeficientes en cualquier base.

```
'Veamos el caso genérico'

d1=4 # Dimensión de H1
d2=3 # Dimensión de H2

' generate a random complex matrix '
w = np.random.randn(d1,
    d2)+ np.random.randn(d1,d2) * 1j # coeficientes w_{ia} de un estado genérico
display(array_to_latex(w))

' perform the SVD decomposition'
u, s, vh = np.linalg.svd(w, full_matrices=True)

np.round(s,3)

print('principal values s_i = ',np.round(s,3))
print('The Schmidt number is r =', np.count_nonzero(s))
```

$$\begin{bmatrix} -0.6771133129 - 1.7592191532i & -0.169279762 + 0.9459507926i & 0.9021837268 + 0.1822725284i \\ 0.1260322326 + 0.0055995793i & 0.6940500089 - 0.2605170943i & 2.4442163068 - 0.5115800371i \\ 0.3871659959 + 0.3934163266i & -0.5397614987 - 2.1503665821i & 0.4082058626 + 0.9059317522i \\ 0.0480247357 - 0.1089298156i & -0.4562253407 + 0.8137884951i & 0.7089854565 + 0.4507804966i \end{bmatrix}$$

```
principal values s_i = [3.274 2.697 1.388]
The Schmidt number is r = 3
```

Corre la celda anterior muchas veces y mira si consigues encontrar algún caso en que  $r < \min(d_1, d_2)$

```
'Veamos ahora el caso particular de un estado factorizable'

d1=5
d2=3

' create two random vectors '
u = tQ.random_ket(d1)
v = tQ.random_ket(d2)

w = np.outer(u,v)
display(array_to_latex(w))

' SVG decomposition '
u, s, vh = np.linalg.svd(w, full_matrices=True)

print('principal values s_i = ', np.round(s,3))
print('The Schmidt number is p =', np.count_nonzero(np.round(s,3)))
```

$$\begin{bmatrix} 0.245547712 + 0.2501984358i & -0.2719892208 + 0.0244247214i & -0.1188884336 - 0.2626364848i \\ -0.293928596 - 0.1611076509i & 0.2421947252 - 0.0975627948i & 0.1814381513 + 0.2075143255i \\ -0.0446425495 - 0.0377016084i & 0.0447581119 - 0.00828499i & 0.0238162545 + 0.0417363197i \\ -0.1655612383 + 0.1912009731i & -0.0334657752 - 0.1941590325i & 0.1819106228 - 0.1008476749i \\ 0.3671469454 - 0.023954469i & -0.1668359102 + 0.2330500354i & -0.2903014488 - 0.0853003486i \end{bmatrix}$$

```
principal values s_i = [1. 0. 0.]
The Schmidt number is p = 1
```

Podemos correr la celda anterior varias veces comprobar que nunca obtendremos  $r > 1$

### 1.4.3 Producto Tensorial Múltiple

El producto tensorial se puede generalizar a más de un factor.

El espacio  $H_1 \otimes H_2 \dots \otimes H_n$  formado por todas las  $n$ -tuplas ordenadas de vectores

$$|u\rangle = |u_1 u_2 \dots u_n\rangle \equiv |u_1\rangle \otimes |u_2\rangle \otimes \dots \otimes |u_n\rangle$$

donde  $|u_i\rangle \in H_i$  y sus combinaciones lineales  $\{a|u\rangle + b|v\rangle + \dots\}$ .

Salvo mención expresa, asumiremos que todos los  $H_j = H$  son iguales y de dimension  $d$ . En el contexto de la computación cuántica usual con cúbits  $\Rightarrow d = 2$

### Ejercicio

Escribe una función  $\text{kronecker}(u_1, u_2, \dots, u_n)$  que tome  $n$  kets (como vectores columna) y devuelva su producto de Kronecker múltiple.

### Base de $H^{\otimes n}$

Una base de  $H^{\otimes n}$  se obtiene a partir de cadenas

$$|i_1 i_2 \dots i_n\rangle = |i_1\rangle |i_2\rangle \dots |i_n\rangle$$

donde  $i_1, \dots, i_n = 0, \dots, d-1$ .

El número de posibles cadenas es  $d^n$  que es la dimensión de  $H^{\otimes n}$

$$\dim_{\mathbb{C}} H^{\otimes n} = d^n$$

Podemos cambiar de etiqueta

$$|i_1 \dots i_n\rangle \rightarrow |a\rangle$$

con

$$a = i_n + di_{n-1} + d^2 i_{n-2} + \dots + d^{n-1} i_1$$

Claramente  $a \in (0, d^n - 1)$ .

Si cada base  $\{|i\rangle\}$  es ortonormal, tendremos que la *base producto* también lo será

$$\langle i_1 i_2 \dots i_n | j_1 j_2 \dots j_n \rangle = \delta_{i_1 j_1} \delta_{i_2 j_2} \dots \delta_{i_n j_n} \quad \leftrightarrow \quad \langle a | b \rangle = \delta_{ab}$$

### Estado entrelazado general

Un *vector general* admitirá una expansión en esta base mediante  $d^n$  *componentes complejas*  $u_{i_1 i_2 \dots i_n}$  en la forma

$$|u\rangle = \sum_{i_1, \dots, i_n=0}^{d-1} u_{i_1 i_2 \dots i_n} |i_1 i_2 \dots i_n\rangle = \sum_{a=0}^{d^n-1} u_a |a\rangle.$$

Podemos obtener cualquier componente compleja proyectando sobre el elemento correspondiente de la base

$$u_{i_1 i_2 \dots i_n} = \langle i_1 i_2 \dots i_n | u \rangle \quad \leftrightarrow \quad u_a = \langle a | u \rangle$$

Decimos que  $u_{i_1 i_2 \dots i_n}$  forman las componentes de un *tensor de rango  $n$*

Un estado genérico depende de un conjunto de  $d * d * \dots * d = d^n$  componentes independiente  $u_{i_1 i_2 \dots i_n}$ .

Ya hemos visto los casos de rango  $n = 1$  (vector) y  $n = 2$  (operador)

## Estado factorizable

Al igual que antes, *sólo en casos muy particulares*, un vector de  $H^{\otimes n}$  se podrá escribir en forma factorizada

$$|w\rangle = |v_1\rangle|v_2\rangle \dots |v_n\rangle \equiv |v_1 v_2 \dots v_n\rangle$$

Escribiendo  $|v_k\rangle = \sum_{i_k=1}^d v_{i_k} |i_k\rangle$  vemos que un *vector factorizable* admite una expansión general en la que los coeficientes son factorizables

$$v_{i_1 i_2 \dots i_n} = v_{i_1} v_{i_2} \dots v_{i_n}$$

El conjunto de coeficientes está parametrizado por  $d + d + \dots + d = nd$  cantidades  $v_{i_k}$ ,  $i_k = 1, \dots, d$ ,  $k = 1, \dots, n$ .

## Estados producto de matrices (MPS)

En medio de los dos casos extremos anteriores encontramos la posibilidad de que las componentes del tensor se puedan escribir como productos de matrices.

Definición:

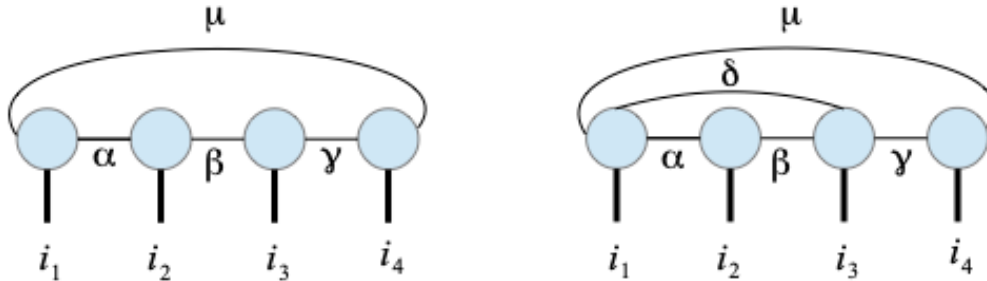
Un estado  $|v\rangle$  es un MPS (estado producto de matrices) si sus componentes en cualquier base pueden escribirse como la traza total de un producto de matrices

$$v_{i_1 i_2 i_3 \dots i_n} = \text{tr}(A_{i_1}^{(1)} \cdot A_{i_2}^{(2)} \cdot A_{i_3}^{(3)} \dots A_{i_n}^{(n)})$$

- Si  $A_i^{(a)}$  con  $a = 1, \dots, n$  y  $i = 1, \dots, d$  es un conjunto de  $nd$  matrices de dimensión  $D \times D$ , con  $D$  la (*dimensión local de enlace*), el número de parámetros independientes es  $ndD^2$ . Los MPS tienen entrelazamiento no nulo, que crece con  $D$ .
- Si  $A_i^{(a)} = A_i$  para todo  $a = 1, \dots, n$  decimos que el MPS formado es *invariante traslacional*

Por ejemplo, con  $n = 4$ ,  $|v\rangle = v_{i_1 i_2 i_3 i_4} |e_{i_1 i_2 i_3 i_4}\rangle$

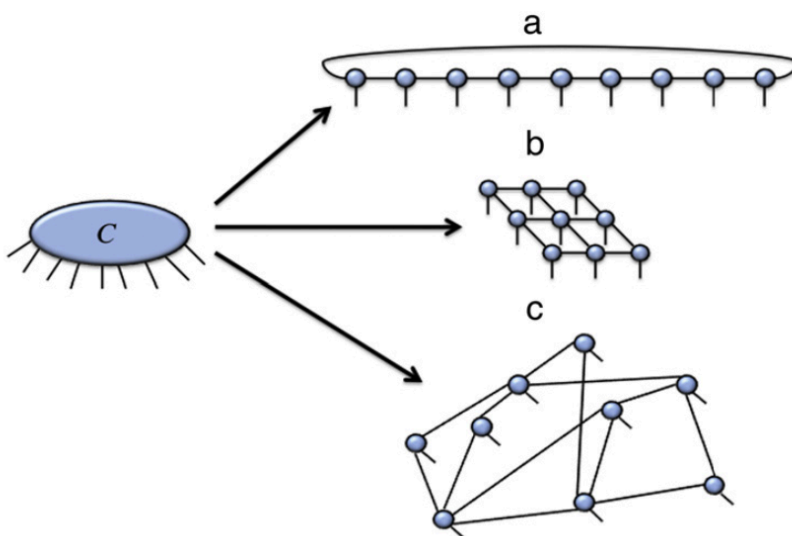
$$v_{i_1 i_2 i_3 i_4} = \sum_{\alpha, \beta, \gamma, \mu=1}^D A_{i_1}^{\mu\alpha} A_{i_2}^{\alpha\beta} A_{i_3}^{\beta\gamma} A_{i_4}^{\gamma\mu}$$



## Redes de tensores

Si queremos aumentar el entrelazamiento del estado podemos, además de aumentar  $D$ , recurrir a contracciones de índices de tensores de mayor rango

$$v_{i_1 i_2 i_3 i_4} = \sum_{\alpha \beta \gamma \mu \delta = 1}^D A_{i_1}^{\mu \alpha \delta} A_{i_2}^{\alpha \beta} A_{i_3}^{\beta \delta \gamma} A_{i_4}^{\gamma \mu}$$



Los estados MPS, y los estados TN para una dimensión de enlace local  $D$  finita, no son suficientemente expresivos para capturar el máximo entrelazamiento posible en un estado.

### Notar

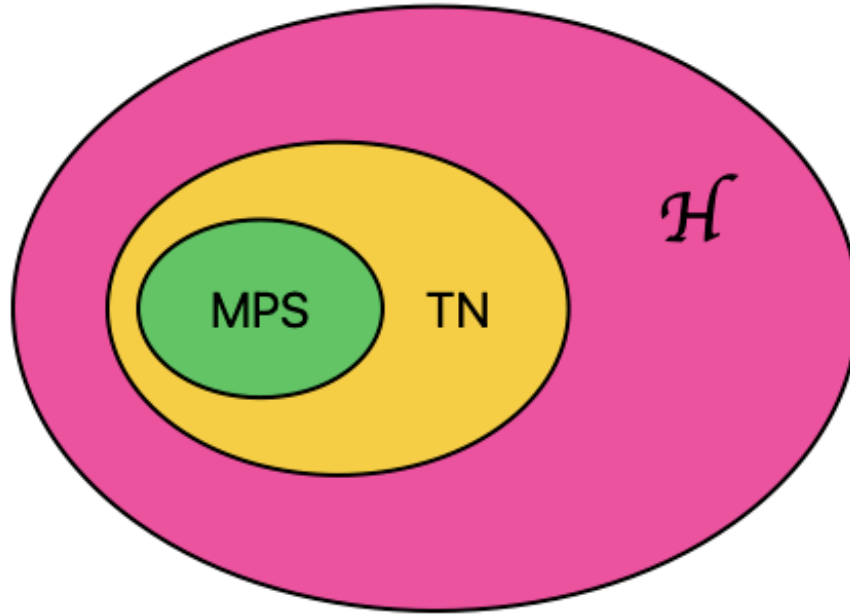
- $nd \ll d^n$ . El crecimiento exponencial del número de estados entrelazados es el ingrediente crucial para la computación cuántica. Observar que  $d^n$  es el número de enteros alcanzables por  $n$  bits. Pero en computación cuántica es el número de dimensiones en la que podemos poner  $d^n$  amplitudes complejas.
- No existe un criterio general para saber si un estado es, a priori, factorizable o entrelazado.
- Además, hay formas de caracterizar matemáticamente el nivel de entrelazamiento (*entanglement witnesses*, *entanglement monotones* etc.) desde nulo (estado factorizable) hasta maximal.

### 1.4.4 Operadores sobre $H^{\otimes n}$

El espacio  $H^{\otimes n}$  admite, como cualquier espacio vectorial, la acción de *operadores lineales*  $A : H^{\otimes n} \rightarrow H^{\otimes n}$  donde

$$A : |u\rangle \rightarrow |v\rangle \equiv A|u\rangle$$

El conjunto de todos los operadores lineales forman el espacio vectorial  $\text{Lin}(H^{\otimes n})$ .



## Matrices

- A cada operador,  $A$ , le podemos asociar una *matriz*, una vez elijamos nuestra base  $\{|i_1 i_2 \dots i_n\rangle\}$  donde,  $i_k = 1, \dots, d$ .
- Los *elementos de matriz* ahora vendrán etiquetados por dos *multi-índices*

$$A_{i_1 \dots i_n, j_1 \dots j_n} = \langle i_1 \dots i_n | A | j_1 \dots j_n \rangle \quad \leftrightarrow \quad A_{ab} = \langle a | A | b \rangle$$

- Con la matriz, el operador se reconstruye en la base canónica de productos externos

$$A = \sum_{i_1, \dots, i_n, j_1, \dots, j_n=0}^{d-1} A_{i_1 \dots i_n, j_1 \dots j_n} |i_1 \dots i_n\rangle \langle j_1 \dots j_n| = \sum_{a, b=0}^{d^n-1} A_{ab} |a\rangle \langle b| \quad (1.22)$$

- En  $A_{i_1 \dots i_n, j_1 \dots j_n} = A_{ab}$  hay  $d^n \times d^n = d^{2n}$  grados de libertad. Esta sería la dimensión del espacio  $\text{Lin}(H^{\otimes n})$ .

## Producto tensorial de operadores

En  $\text{Lin}(H^{\otimes n})$  hay un análogo de los vectores factorizables de  $H^{\otimes n}$ : los *operadores factorizables*.

Supongamos que existen  $n$  operadores lineales  $A^{(a)}$ ,  $a = 1, \dots, n$  definidos sobre cada espacio factor  $H$ .

Definición:

La acción del producto tensorial de operadores  $A = A^{(1)} \otimes A^{(2)} \otimes \dots \otimes A^{(n)}$  sobre un vector  $|v\rangle = |v\rangle_1 \otimes \dots \otimes |v_n\rangle \in H$  factoriza

$$A|v\rangle = A^{(1)}|v_1\rangle \otimes \dots \otimes A^{(n)}|v_n\rangle.$$



La acción sobre vectores generales se sigue imponiendo linealidad

$$A(|v\rangle + |w\rangle) = A|v\rangle + A|w\rangle.$$

- El adjunto de un producto tensorial de operadores es el producto de los adjuntos (no se permuta el orden)

$$A^\dagger = A^{(1)\dagger} \otimes \dots \otimes A^{(n)\dagger}$$

- El producto tensorial de operadores hermíticos es hermítico

$$A^{(a)\dagger} = A^{(a)} \implies A^\dagger = A$$

- El producto tensorial de operadores unitarios, es unitario

$$A^{(a)\dagger} = A^{(a)-1} \implies A^\dagger = A^{-1}$$

### Producto de Kronecker de matrices

¿Cómo será la matriz  $A_{i_1 \dots i_n, j_1 \dots j_n}$  de un operador factorizable,  $A = A^{(1)} \otimes A^{(2)} \otimes \dots \otimes A^{(n)}$ , en términos de las matrices  $A_{ij}^{(a)}$  de sus factores?

Vamos a tomar  $n = 2$  por simplicidad

$$\begin{aligned} A = A^{(1)} \otimes A^{(2)} &= \left( \sum_{i_1 i_2} A_{i_1 j_1}^{(1)} |i_1\rangle \langle j_1| \right) \left( \sum_{i_2 j_2} A_{i_2 j_2}^{(2)} |i_2\rangle \langle j_2| \right) \\ &= \sum_{i_1 i_2, j_1 j_2} A_{i_1 j_1}^{(1)} A_{i_2 j_2}^{(2)} |i_1 i_2\rangle \langle j_1 j_2| \\ &= \sum_{i_1 i_2, j_1 j_2} A_{i_1 i_2, j_1 j_2} |i_1 i_2\rangle \langle j_1 j_2| \end{aligned}$$

Vemos que la matriz asociada a  $A$  se obtiene a partir de las matrices de  $A^{(a)}$  mediante el *producto exterior de las matrices*, o *producto de Kronecker*.

$$A_{i_1 i_2, j_1 j_2} = A_{i_1 j_1}^{(1)} A_{i_2 j_2}^{(2)}$$

- El método para de **representar** matricialmente el producto de Kronecker de dos matrices  $A \otimes B$  es sencillo. Supongamos que  $d = 2$  y tenemos un operador producto  $A \otimes B$ . Entonces su matriz

$$(A \otimes B)_{ab} = \begin{pmatrix} A_{00}B & A_{01}B \\ A_{10}B & A_{11}B \end{pmatrix} = \begin{pmatrix} A_{00}B_{00} & A_{00}B_{01} & A_{01}B_{00} & A_{01}B_{01} \\ A_{00}B_{10} & A_{00}B_{11} & A_{01}B_{10} & A_{01}B_{11} \\ A_{10}B_{00} & A_{10}B_{01} & A_{11}B_{00} & A_{11}B_{01} \\ A_{10}B_{10} & A_{10}B_{11} & A_{11}B_{10} & A_{11}B_{11} \end{pmatrix}.$$

- El producto de Kronecker verifica las siguientes propiedades para dos matrices  $A$  y  $B$  de dimensiones  $d_A$  y  $d_B$ .

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$$

$$\text{tr}(A \otimes B) = (\text{tr}A)(\text{tr}B)$$

$$A \otimes (B + D) = A \otimes B + A \otimes D$$

$$(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$$

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

$$\det(A \otimes B) = (\det A)^{d_B} (\det B)^{d_A}$$

donde  $AC$  significa el producto de matrices  $A$  y  $C$

---

### Ejercicio

1. Demuestra estos resultados.
2. Rescribe la función *kron* para que acepte dos matrices  $A$  y  $B$  de dimensiones  $d_A$  y  $d_B$  y devuelva su producto de Kronecker  $A \otimes B$ . Verifica el resultado con la función *kron* de *numpy*. Verifica las propiedades anteriores.

---

La generalización a todo  $n$  es obvia. El producto de Kronecker de  $n$  matrices  $A_{i_a j_a}^{(a)}$  asociadas a operadores  $A^{(a)}$  es

$$A_{i_1 \dots i_n, j_1 \dots j_n} = A_{i_1 j_1}^{(1)} \dots A_{i_n j_n}^{(n)}$$

---

### Ejercicio

Calcula  $\sigma_1 \otimes \sigma_2 \otimes \sigma_3$

---

---

### Notar

Observar que en un operador general, la matriz  $A_{i_1 \dots i_n, j_1 \dots j_n}$  tiene  $d^n \times d^n = d^{2n}$  entradas independientes.

Sin embargo en un producto de Kronecker  $A_{i_1 j_1}^{(1)} \dots A_{i_n j_n}^{(n)}$  sólo hay  $nd^2$ .

Por tanto, los *operadores factorizables* forman un subconjunto muy pequeño dentro del conjunto de los operadores generales.

---

## Generación de entrelazamiento

Supongamos que  $|u\rangle = |u_1\rangle \otimes |u_2\rangle$  es factorizable.

- Si  $A = A_1 \otimes A_2$  es un operador factorizable entonces

$$|v\rangle = A|u\rangle = A_1|u_1\rangle \otimes A_2|u_2\rangle = |v_1\rangle \otimes |v_2\rangle$$

también es factorizable.

- Inversamente, la acción de un operador no factorizable  $A \neq A_1 \otimes A_2$  genera estados entrelazados

$$|v\rangle = A|u\rangle \neq |v_1\rangle \otimes |v_2\rangle$$

En lenguaje físico, lo que esto quiere decir es que para generar entrelazamiento debe haber *interacción* entre los grados de libertad que residen en  $H_1$  y  $H_2$ . Si no hay interacción, estados factorizados seguirán siendo factorizables en el futuro.

Feb 13, 2024 | 3 min read

## 1.5 Probabilidades

```
%run ../../macro_tQ.py
import sys
sys.path.append('../../')
import macro_tQ as tQ

import numpy as np
import scipy.linalg as la
from IPython.display import display,Markdown,Latex
import matplotlib.pyplot as plt
from qiskit.tools.visualization import array_to_latex
import copy
```

### 1.5.1 Variables aleatorias

Denotamos con  $(X, p(X))$  una *variable aleatoria* donde

- $X$  es el espacio muestral de valores  $\{x_1, x_2, \dots, x_n\}$  que pueden aparecer en una *consulta* la variable aleatoria
- $p(X)$  es la *distribución de probabilidad*

#### Distribución de probabilidad

Una distribución de probabilidad es una función real  $x \rightarrow p(x)$  que debe verificar las dos condiciones siguientes

$$p(x) \in [0, 1] \quad , \quad \sum_{x \in X} p(x) = 1$$

Es decir, la suma de probabilidades de todos los sucesos posibles debe ser la unidad.

Media: la media de una variable aleatoria viene dada por la expresión

$$\bar{X} = \sum_i x_i p(x_i)$$

Varianza: la varianza,  $\sigma_X^2$ , es la media de la desviación cuadrática  $\overline{(x_i - \bar{X})^2}$

$$\sigma_X^2 = \sum_j (x_j - \bar{X})^2 p(x_j) = \overline{X^2} - \bar{X}^2$$

La cantidad  $\sigma_X$  se denomina desviación estándar

$$\sigma_X = \sqrt{\overline{X^2} - \bar{X}^2}$$

## 1.5.2 La conexión estadística

- Nuestro conocimiento del mundo se basa en la realización de experimentos, el resultado de los cuales es (empíricamente) aleatorio.
- Podemos pensar en el hecho de medir un sistema como la consulta de una variable aleatoria  $(X, p(X))$  donde la distribución de probabilidad incorpora todo nuestro conocimiento acerca del sistema

### Frecuencias e Histogramas

Cualquier consulta o medida da lugar a una *muestra* finita de valores  $A_N = (a_1, a_2, \dots, a_N)$  donde  $a_i \in \{x_1, \dots, x_n\}$  pueden repetirse en la muestra, con números de aparición  $n(x_i)$  tales que  $n(x_1) + \dots + n(x_p) = N$ .

Estos datos se pueden agrupar en intervalos o *bins* que eliminen cierta precisión numérica.

Por ejemplo, si truncamos nuestra precisión a las décimas de unidad, 13.10 y 13.19 pertenecerán al mismo *bin*.

Un histograma es un diagrama en el que, por cada *bin*, hay una columna, cuya altura representa el número de sucesos que pertenecen a dicho *bin*

En el siguiente ejemplo, puedes ver cómo, la misma lista de datos conduce a distintos histogramas si cambias la anchura de los bins

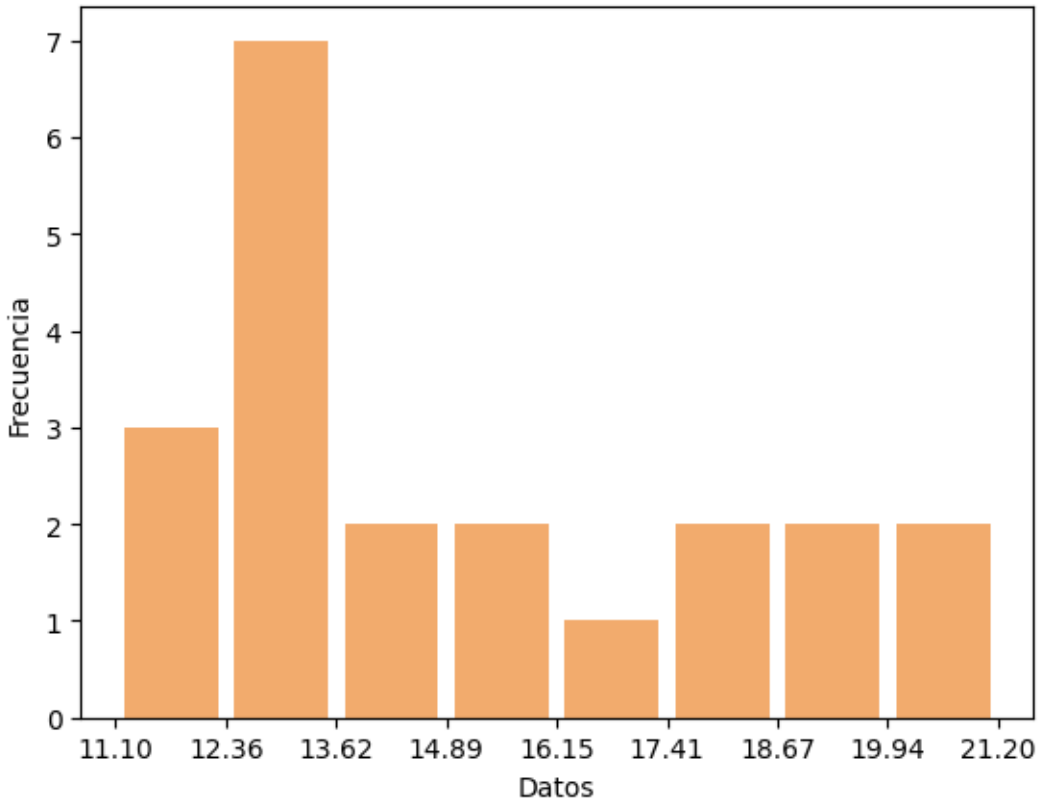
```
import numpy as np
import matplotlib.pyplot as plt

datos1 = [12.2, 15.3, 13, 12.8, 18.1, 13, 20, 19.5, 20.2, 12.1, 12.4, 13.2, 17.9,
↪14.1, 15, 13.4, 11.1, 14.6, 13, 17.4, 19.2]
datos2 = np.linspace(0.,10.,20)
datos = datos1

anchura_bins = 1. # para modificar la anchura de los bins
num_bins = int((max(datos)-min(datos))/anchura_bins) #número de bins
bins_list = np.linspace(min(datos), max(datos) + 1 ,num_bins) #calculamos los ↪
↪extremos de los intervalos

counts, bins, ignore = plt.hist(x=datos, bins = bins_list, color='#F2AB6D', ↪
↪rwidth=0.85)
plt.xlabel('Datos')
plt.ylabel('Frecuencia')
plt.xticks(bins_list)

plt.show()
```



### Ley de los grandes números

Las fracciones relativas tienden en el límite  $N \rightarrow \infty$  a la probabilidad relativa de aparición de  $x_i$  en una sola tirada

$$f_N(x_i) = \frac{n(x_i)}{N} \xrightarrow{N \rightarrow \infty} p(x_i)$$

Experimentalmente sólo tenemos acceso a las frecuencias relativas  $f_N(x_i)$  para un  $N$  grande aunque finito.

Igualmente, nuestro conocimiento de la media  $\bar{X}$  y la varianza  $\sigma_X^2$  siempre es aproximado, y se realiza a través de las medias y varianzas muestrales

$$\begin{aligned} \bar{A}_N &= \sum_i x_i f_N(x_i) \xrightarrow{N \rightarrow \infty} \bar{X} \\ \sigma_{A_N}^2 &= \sum_i (x_i - \bar{A}_N)^2 f_N(x_i) \xrightarrow{N \rightarrow \infty} \sigma_X^2 \end{aligned}$$

### 1.5.3 Distribuciones frecuentes

#### La distribución de Bernoulli

Una variable aleatoria de Bernoulli  $X = (x, p(x))$  tiene dos posibles resultados

- *éxito*  $\rightarrow x = 1$  con probabilidad  $p(1) = p$
- *fracaso*  $\rightarrow x = 0$  con probabilidad  $p(0) = 1 - p$

Podemos calcular fácilmente

$$\bar{X} = \sum_i x_i p_i = 1 \cdot p + 0 \cdot (1 - p) = p \quad (1.23)$$

$$\sigma^2 = \sum_i (x - \bar{X})^2 p_i = (1 - p)^2 p + (0 - p)^2 (1 - p) = p(1 - p) \quad (1.24)$$

#### Distribución binomial

La variable aleatoria binomial  $X = (x, p(x))$  se define como

$x$  = número de *xitos* obtenidos en  $n$  pruebas de Bernoulli sucesivas

Claramente  $x \in (0, 1, 2, \dots, n)$ .

Ahora es muy sencillo obtener la probabilidad de un suceso con  $x$  éxitos

$$p(x) = \binom{n}{x} p^x (1 - p)^{n-x}$$

donde el primer factor tiene en cuenta las posibles ordenaciones en que aparecen  $x$  éxitos en  $n$  intentos.

Un cálculo un poco más largo permite ver que, ahora

$$\bar{X} = np \quad (1.25)$$

$$\sigma^2 = np(1 - p) \quad (1.26)$$

#### La distribución normal

Vamos a estudiar la distribución normal centrada en  $\mu$  y con anchura  $\sigma$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Nos encontramos ante una variable aleatoria con un espacio muestral continuo  $x \in (-\infty, +\infty)$ .

$$\bar{X} = \int_{-\infty}^{+\infty} xp(x)dx = \mu \quad (1.27)$$

$$\overline{(x - \bar{X})^2} = \int_{-\infty}^{+\infty} (x - \mu)^2 p(x)dx = \sigma^2 \quad (1.28)$$

```
'generamos una instancia de la distribución normal'
mu, sigma = 0, 0.1
N = 10000
s = np.random.normal(mu, sigma, N)
```

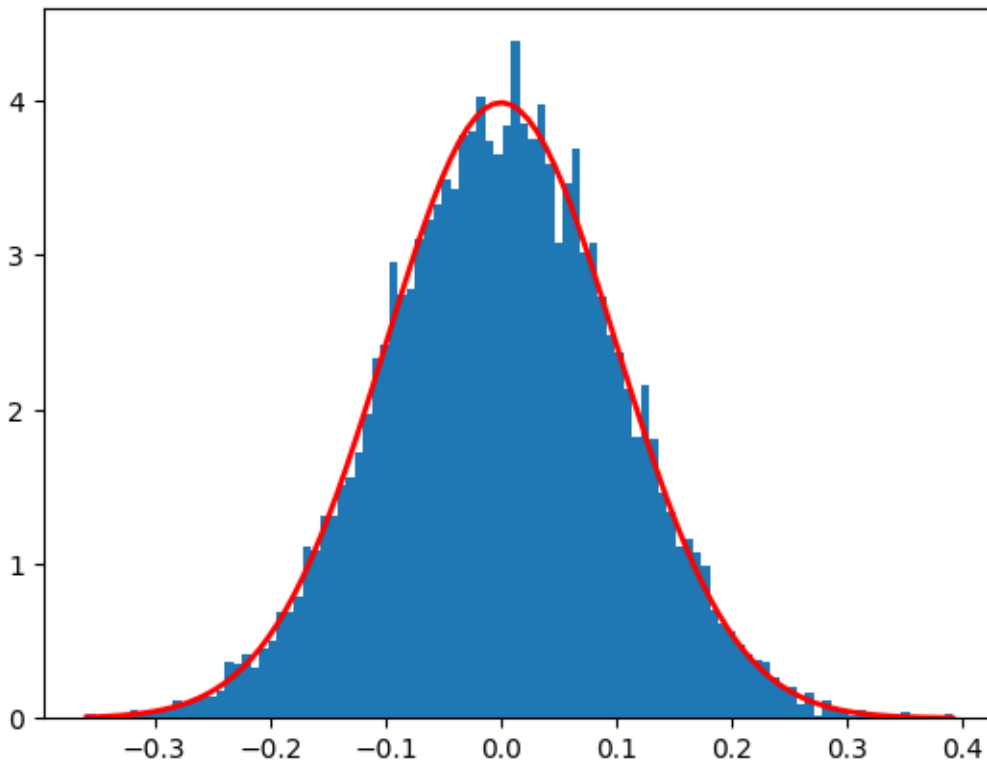
```
import matplotlib.pyplot as plt

'generamos una instancia de histograma con un número de bins'
num_bins=100
counts, bins, ignore = plt.hist(s, num_bins, density=True)
'extraemos los bins para hacer el plot de la funcion analítica'

'dibujamos la función analítica para comparar'
def gaussian(x,mu,sigma):
    return 1/(sigma * np.sqrt(2 * np.pi))*np.exp( - (x - mu)**2 / (2 * sigma**2) )

plt.plot(bins, gaussian(bins,mu,sigma),linewidth=2, color='r')

plt.show()
```



En el límite de número de eventos  $N \rightarrow \infty$ , el histograma y la curva matemática confluyen.

### 1.5.4 Probabilidades combinadas

Las probabilidades combinadas son *la base de las correlaciones*. Es aquí donde la Mecánica Cuántica *predice* resultados diferentes a la Mecánica Clásica.

Ahora vamos a examinar variables aleatorias formadas por dos espacios muestrales  $X$  e  $Y$ . Dependiendo de la forma en que combinemos la observación de cada una tendremos distintas distribuciones de probabilidad

#### Probabilidad combinada

La *probabilidad combinada*  $p(X, Y)$  asocia un número  $p(x, y)$  a la probabilidad de observación conjunta de  $x$  e  $y$ .

Hay que tratar las parejas de eventos como un solo evento  $a = (x, y)$ . Por eso, la condición de normalización ahora es

$$\sum_a p(a) = \sum_{xy} p(x, y) = 1.$$

La suma parcial sobre una de las dos variables conduce a sendas **distribuciones marginales**

$$q(x) = \sum_y p(x, y) \quad \tilde{q}(y) = \sum_x p(x, y)$$

Si la probabilidad combinada es el producto de las probabilidades de los miembros del par, decimos que  $X$  e  $Y$  son **variables aleatorias independientes**

$$p(x, y) = p(x)p(y)$$

La distribución de cada variable coincide con la que se deduce de marginalizar la otra

$$\sum_y p(x, y) = p(x) \quad , \quad \sum_x p(x, y) = p(y)$$

#### Probabilidad condicionada

La distribución de **probabilidad condicionada**  $p(X|Y)$  asigna un número  $p(x|y)$  a la probabilidad de encontrar un suceso  $X = x$  una vez *sabemos* que  $Y = y$  ha sido el resultado de consultar  $Y$ .

La manera de acceder experimentalmente a estas distribuciones, es efectuar un muestreo  $(a_i, b_i)$ ,  $i = 1, \dots, N$  de valores de  $(X, Y)$  y *seleccionar* sólo aquellos sucesos donde  $b_i = y$  un valor concreto de  $Y$ .

#### Teorema de Bayes

Las probabilidades *condicionales* y *combinadas* no son independientes. Se relacionan de la forma siguiente

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x) = p(y, x)$$

La segunda igualdad conduce al *teorema de Bayes*

Teorema: de Bayes

$$p(x|y) = p(y|x) \frac{p(x)}{p(y)}$$



Feb 13, 2024 | 7 min read

Talent3



## FUNDAMENTOS DE MECÁNICA CUÁNTICA

```
%run ../macro_tQ.py
import sys
sys.path.append('../')
import macro_tQ as tQ

import numpy as np
import scipy.linalg as la
from IPython.display import display, Markdown, Latex
import matplotlib.pyplot as plt
from qiskit.tools.visualization import array_to_latex
import copy
```

### 2.1 Axiomas

La Mecánica Cuántica es una teoría fundamentada en unos pilares axiomáticos cuya selección admite cierta flexibilidad. La más aceptada constituye lo que se denomina la interpretación de Copenhagen.

Postulado I: Vector de estado

En un instante,  $t$ , la máxima información accesible de un sistema cuántico aislado está asociada a un vector  $|\psi\rangle$ , de norma unidad,  $\langle\psi|\psi\rangle = 1$ , perteneciente a un espacio de Hilbert  $H$ , cuya dimensión compleja  $N$  equivale al número de grados de libertad del sistema.

- El vector de estado es el objeto matemático que contiene la máxima información accesible. La palabra *estado* es engañosa por cuanto parece indicar que el sistema *está* en alguna estado concreto. El vector de estado recibe también el nombre de *función de onda*.
- La dimensión de  $H$  está relacionada con el número de grados de libertad del sistema.
- La *fase global* del vector de estado no contiene información: dos vectores que difieran en una fase global representan al mismo estado ningún experimento permite distinguirlos

$$|\psi\rangle \sim |\psi'\rangle = e^{i\varphi}|\psi\rangle,$$

• **Postulado II: Evolución con el tiempo**

Lo que caracteriza y distingue a un sistema cuántico de otro está contenido en un *operador hermitico*  $H = H^\dagger$  que se denomina **operador Hamiltoniano**

El operador Hamiltoniano gobierna su evolución. Supongamos que el sistema se prepara a  $t = 0$  en un cierto estado  $|\psi(0)\rangle$

Postulado II: Evolución con el tiempo

El cambio continuo en el tiempo de  $|\psi(t)\rangle$  de un sistema cuya dinámica está gobernada por un Hamiltoniano  $H(t)$ , es determinista y es solución de la Ecuación de Schrödinger

$$\frac{\partial |\psi(t)\rangle}{\partial t} = -\frac{i}{\hbar} H(t) |\psi(t)\rangle$$

La ecuación de Schrödinger es una ecuación diferencial ordinaria (una sola variable) para un vector.

Postulado III: Medidas de Von Neumann

- A una magnitud física medible le está asociado un operador hermítico  $A = A^\dagger$  que denominamos observable.
- Los resultados de una medición no pueden dar como resultado nada más que uno de los valores propios de  $A \Rightarrow \lambda_n$ .
- Regla de Born: La probabilidad de obtener el autovalor  $\lambda_k$  como resultado de una cierta medición, viene dada por la expresión

$$p(\lambda_k) = |\langle \lambda_k | \psi \rangle|^2$$

donde  $|\lambda_k\rangle$  es el autovector asociado.

Las magnitudes medibles son números reales, de ahí la exigencia de que  $A$  deba ser un operador hermítico

---

### Nota

- La información contenida en  $|\psi\rangle$  es intrínsecamente probabilística:  $\psi$  codifica una distribución de probabilidad,  $p(\lambda_m)$ , para cada posible resultado  $\lambda_m$ .
  - *Muy importante:*  $(\lambda_m, p(\lambda_m))$  no es una pseudo-probabilidad sino, probablemente, la única verdadera variable aleatoria en el sentido puro de la palabra que existe en la naturaleza.
- 

La fórmula para  $p(\lambda_k)$  es cierta cuando el autovalor en cuestión es *no degenerado*.

Cuando  $\lambda_k$  tiene degeneración  $d_k$  debe generalizarse:

$$\begin{aligned} p(\lambda_k) &= \sum_{a=1}^{d_k} |\langle \lambda_k^a | \psi \rangle|^2 = \langle \psi | \left( \sum_{a=1}^{d_k} |\lambda_k^a\rangle \langle \lambda_k^a| \right) | \psi \rangle \\ &= \langle \psi | P_k | \psi \rangle \end{aligned} \tag{2.1}$$

(2.2)

La última expresión revela que  $p(\lambda_k)$  no es más que el *valor esperado del proyector*  $P_k = P_k^2$ .

Esta es la razón por la que las medidas a las que se refiere este postulado se denominan *proyectivas*

Veremos en el capítulo 5 que son un caso especial del tipo más general posible de medidas.

Postulado IV: Colapso de la función de onda

Si el resultado de una medida ha sido  $\lambda_n$ , el estado del sistema, inmediatamente después de la medida,

viene dado por el vector propio

$|\lambda_n\rangle \in H$ , normalizado  $|\langle \lambda_n | \lambda_n \rangle| = 1$

$$|\psi\rangle \xrightarrow{\lambda_n} |\lambda_n\rangle$$

## 2.2 Medidas y valores esperados

- En Mecánica Cuántica especificamos un estado en las componentes de la expansión de un estado en una base ortonormal. Existen infinitas bases posibles para expresar un vector. ¿Qué base es la mejor? La respuesta es que depende del proceso que estudiemos.
- Por ejemplo, si el proceso es la medida de un cierto observable  $A = A^\dagger$ , el resultado de nuestra medición sólo puede ser uno de los valores propios  $\lambda_i$

$$A|\lambda_k\rangle = \lambda_k|\lambda_k\rangle$$

donde  $|\lambda_k\rangle$  son los autovectores  $k = 1, 2, \dots$  que forman una base ortonormal.

- Ésta es la base natural adaptada a procesos de medida de  $A$ . En particular para conocer *con qué probabilidad* se producirá cada resultado, expandiremos el estado

$$|\psi\rangle = \sum_{i=1}^N a_i |\lambda_i\rangle.$$

El axioma IV (regla de Born) afirma que la probabilidad de aparición del resultado  $\lambda_i$  es precisamente

$$p(\lambda_i) = |\langle \lambda_i | \psi \rangle|^2 = |a_i|^2.$$

- La forma de tener acceso experimental a los números  $p(\lambda_i)$  es mediante la **repetición estadística**. Si efectuamos la medida un número  $n$  de veces con  $n \rightarrow \infty$ , y contamos la frecuencia de aparición de los distintos  $\lambda_i$ , esto es

$$\frac{n(\lambda_i)}{n}.$$

en el límite  $n \rightarrow \infty$  dicha frecuencia experimental convergerá a la probabilidad teórica

$$\lim_{n \rightarrow \infty} \frac{n(\lambda_i)}{n} = p(\lambda_i) = |a_i|^2.$$

Vemos que, mediante repetición estadística, solamente podemos recuperar el *módulo de las amplitudes*

$$|a_i| = \lim_{n \rightarrow \infty} \sqrt{\frac{n(\lambda_i)}{n}}$$

---

### Notar

- En Computación Cuántica, almacenamos la información en las amplitudes  $a_i$  en una base dada (la base computacional). Vemos que sólo su módulo  $|a_i|$  es recuperable a través de medidas.
  - Cada vez que seleccionamos un estado  $|\psi\rangle$ , y un observable  $A$ , se genera una variable aleatoria  $(\lambda, p(\lambda)) \rightarrow$  en cada medida de  $A$  sobre  $|\psi\rangle$  obtenemos un autovalor  $\lambda_i$  con una probabilidad  $p(\lambda_i)$ . La repetición del experimento permite reconstruir estimadores estadísticos.
- 

### Ejercicio

Generaliza las expresiones anteriores al caso en que los autovalores  $\lambda_k$  puedan ser  $d_k$  veces degenerados.

---

## 2.2.1 Valores esperados

Rescribamos la probabilidad de encontrar  $\lambda_k$  en el caso de que sea  $d_k$  veces degenerado

$$\begin{aligned} p(\lambda_k) &= \sum_{a=1}^{d_k} |\langle \lambda_k^a | \psi \rangle|^2 = \sum_{a=1}^{d_k} \langle \psi | \lambda_k^a \rangle \langle \lambda_k^a | \psi \rangle \\ &= \langle \psi | \left( \sum_{a=1}^{d_k} |\lambda_k^a \rangle \langle \lambda_k^a| \right) | \psi \rangle = \langle \psi | P_k | \psi \rangle \end{aligned}$$

donde

$$P_k = \sum_{a=1}^{d_k} |\lambda_k^a \rangle \langle \lambda_k^a|$$

es el proyector sobre el subespacio propio

Por el *valor esperado*  $\langle A \rangle$  del *observable*  $A$  en el estado  $|\psi\rangle$ , entendemos el *valor medio* de la variable aleatoria  $\lambda_n$  obtenida por el método descrito.

$$\langle A \rangle = \sum_k \lambda_k p(\lambda_k) = \sum_k \lambda_k \langle \psi | P_k | \psi \rangle = \langle \psi | \left( \sum_k \lambda_k P_k \right) | \psi \rangle \quad (2.3)$$

Reconocemos entre paréntesis la descomposición espectral de  $A$ . Llegamos así a la siguiente expresión para el valor esperado de un observable  $A$  en un estado  $|\psi\rangle$

Teorema:

$$\langle A \rangle = \langle \psi | A | \psi \rangle$$

En particular, las probabilidades mismas se pueden expresar como *valores esperados del proyector asociado*

$$p(\lambda_k) = \langle \psi | P_k | \psi \rangle = \langle P_k \rangle$$

El espectro de un observable está formado por números reales que podemos ordenar  $\lambda_{min} < \dots < \lambda_{max}$ .

El valor esperado de un observable en uno de sus autoestados coincide con el autovalor asociado

$$\langle \lambda_i | A | \lambda_i \rangle = \langle \lambda_i | \lambda_i | \lambda_i \rangle = \lambda_i \langle \lambda_i | \lambda_i \rangle = \lambda_i$$

Lema:

El valor esperado de un observable  $A$  está acotado entre sus valores propios mínimo y máximo

$$\lambda_{min} \leq \langle \psi | A | \psi \rangle \leq \lambda_{max}$$

y las desigualdades anteriores se saturan en los autoestados correspondientes

## Desviación estándar

La *varianza* es la media de la desviación cuadrática de la variable aleatoria  $(\lambda, p(\lambda))$  es decir

$$\sigma^2 = \overline{(\lambda_i - \bar{\lambda})^2} = \overline{\lambda^2} - \bar{\lambda}^2$$

de aquí se sigue, para la desviación estándar

Teorema:

$$\sigma = \sqrt{\langle \psi | A^2 | \psi \rangle - \langle \psi | A | \psi \rangle^2}$$

Evaluada en un estado propio  $|\lambda_i\rangle$ , la varianza de un operador es cero

$$\sigma^2 = \langle \lambda_i | A^2 | \lambda_i \rangle - \langle \lambda_i | A | \lambda_i \rangle^2 = \langle \lambda_i | \lambda_i^2 | \lambda_i \rangle - \langle \lambda_i | \lambda_i | \lambda_i \rangle^2 = 0$$

```
def vev_sig(A, ket):

    assert np.round(np.sqrt(np.dot(ket.conj().T, ket)[0,0]).real,5) == 1. # chequeamos que la
    norma es unidad
    bra = ket.conj().T

    'el valor esperado'
    vev = np.dot(bra, np.dot(A, ket))[0,0].real

    'la varianza'
    A2 = np.dot(A, A)
    var = np.dot(bra, np.dot(A2, ket))[0,0] - vev**2
    sig = np.sqrt(np.abs(var).real)

    return(np.round(vev,5), np.round(sig,10))
```

```
' definamos una matriz hermitica '
#np.random.seed(10)

A = np.matrix(np.random.randn(3,3) + np.random.randn(3,3)*1j)
A += A.getH()

eigvals, eigvecs = np.linalg.eigh(A)

np.round(eigvals,4)
```

```
array([-2.2561,  0.8435,  4.7955])
```

A continuación evaluamos el valor esperado en un vector aleatorio, y en dos autovectores de  $A$

```
'definamos un ket'
vec1 = np.random.randn(len(A),1) # este es un autovector aleatorio
vec2 = eigvecs[:,0] # este es el autovector de autovalor mínimo
vec3 = eigvecs[:,-1] # éste es un autovector de autovalor máximo

ket = vec2 # elegimos uno
```

(continues on next page)

(continued from previous page)

```
'normalizado'
norma=np.sqrt(np.dot(ket.conj().T,ket)[0,0]).real
uket = ket/norma

vev, sig = vev_sig(A,uket)

print('<A>= ',vev, ' sigma = ', sig)
```

```
<A>= -2.2561 sigma = 2.98e-08
```

## 2.3 Evolución temporal

### 2.3.1 Ecuación de Schrödinger

El estado de un sistema  $|\psi(t)\rangle$  posee la máxima información instantánea accesible de un sistema, es decir, a un tiempo dado  $t$ .

Dado  $|\psi(t_0)\rangle$  en un instante inicial  $t_0$ , la evolución posterior obedece a la ecuación diferencial de Schrödinger

$$\frac{d}{dt}|\psi(t)\rangle = -\frac{i}{\hbar}H(t)|\psi(t)\rangle.$$

Hemos enfatizado el hecho de que, en el caso más general,  $H(t)$  puede depender del tiempo.

---

#### Notar

Las cantidades que entran en la ecuación de Schrödinger *tienen unidades*:

- $H$  tiene unidades de *energía*. En el SI (Sistema Internacional)  $\rightarrow$  Julios = J
- $t$  tiene unidades de *tiempo*, en el SI  $\rightarrow$  segundos = s
- $\hbar$  es la constante de Planck, y tiene unidades de *acción*= energía  $\times$  tiempo. En el SI  $\rightarrow$  Julios  $\cdot$  segundo = J  $\cdot$  s.

Se trata de una *cantidad universal* \$  $\hbar = 1.054 \times 10^{-34}$  J  $\cdot$  s \$

---

#### Conservación de la probabilidad.

La *hermiticidad* de  $H$  se relaciona directamente con la interpretación probabilística de  $|\psi\rangle$ .

Teorema:

En la evolución temporal se conserva la norma del vector de estado



## Prueba

$$\begin{aligned}
 \frac{d}{dt} \|\psi\|^2 &= \frac{d}{dt} \langle \psi(t) | \psi(t) \rangle = \left( \frac{d}{dt} \langle \psi(t) | \right) | \psi(t) \rangle + \langle \psi(t) | \left( \frac{d}{dt} | \psi(t) \rangle \right) \\
 &= \left( \langle \psi(t) | \frac{i}{\hbar} H^\dagger \right) | \psi(t) \rangle + \langle \psi(t) | \left( -\frac{i}{\hbar} H | \psi(t) \rangle \right) \\
 &= \frac{i}{\hbar} \langle \psi(t) | (H^\dagger - H) | \psi(t) \rangle = 0
 \end{aligned}$$

Es decir:  $\|\psi\|$  no cambia con el tiempo, y esto implica que la norma de  $|\psi(t)\rangle$  se conserva  $\Rightarrow 1 = \|\psi(0)\| = \|\psi(t)\|$

## 2.3.2 Operador de Evolución

La conservación de la norma implica que la evolución  $\|\psi(0)\| = \|\psi(t)\|$  es un *proceso unitario*. En otras palabras, debe existir un operador unitario

$$U(t, t_0)^{-1} = U(t, t_0)^\dagger$$

que lleve el estado inicial al actual a tiempo  $t$

$$U(t, t_0) : |\psi(t_0)\rangle \rightarrow |\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle$$

### Notar

El operador de evolución satisface las siguientes propiedades

- $U(t_0, t_0) = I$  , .
- transitividad:  $U(t, t_1)U(t_1, t_0) = U(t, t_0)$
- invertibilidad:  $U(t, t_0)^{-1} = U(t_0, t)$

Evidentemente, el operador de evolución debe estar relacionado con el Hamiltoniano, que es quien gobierna la dinámica. Veamos a ver que hay una ecuación de Schrödinger también para el operador de evolución

Teorema:

El operador de evolución satisface la siguiente ecuación de evolución

$$\frac{d}{dt} U(t, t_0) = -\frac{i}{\hbar} H(t) U(t, t_0)$$

### Demostración

Tomando la derivada temporal de la ecuación  $|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle$ , que define  $U(t, t_0)$ , tenemos para cada miembro

$$\begin{aligned}
 \frac{d}{dt} |\psi(t)\rangle &= -\frac{i}{\hbar} H(t) |\psi(t)\rangle = -\frac{i}{\hbar} H(t) U(t, t_0) |\psi(t_0)\rangle \\
 \frac{d}{dt} U(t, t_0) |\psi(t_0)\rangle &= \left( \frac{d}{dt} U(t, t_0) \right) |\psi(t_0)\rangle
 \end{aligned} \tag{2.4}$$

Igualando ambas expresiones, y teniendo en cuenta que  $|\psi_0\rangle$  es arbitrario, obtenemos la ecuación deseada.

## Caso de $H$ independiente del tiempo

Cuando  $H$  no depende del tiempo podemos dar una expresión analítica para el operador de evolución

Teorema:

Para un hamiltoniano  $H$  independiente del tiempo, el operador de evolución  $U(t, t_0)$  es

$$U(t, t_0) = \exp\left(-\frac{i}{\hbar}(t - t_0)H\right)$$

## Demostración

Basta con demostrar que se satisface la ecuación de evolución y la condición de contorno

$$\begin{aligned} \frac{d}{dt}U(t, t_0) &= \frac{d}{dt} \exp\left(-\frac{i}{\hbar}(t - t_0)H\right) \\ &= \frac{d}{dt} \left( I + (t - t_0) \left(-\frac{i}{\hbar}H\right) + \frac{1}{2!}(t - t_0)^2 \left(-\frac{i}{\hbar}H\right)^2 + \dots \right) \\ &= 0 + \left(-\frac{i}{\hbar}H\right) + (t - t_0) \left(-\frac{i}{\hbar}H\right)^2 + \dots \\ &= \left(-\frac{i}{\hbar}H\right) \left( I + (t - t_0) \left(-\frac{i}{\hbar}H\right) + \dots \right) \\ &= \left(-\frac{i}{\hbar}H\right) \exp\left(-\frac{i}{\hbar}(t - t_0)H\right) \\ &= -\frac{i}{\hbar}H U(t, t_0) \end{aligned} \tag{2.5}$$

Además  $U(t_0, t_0) = \exp\left(-\frac{i}{\hbar}(t_0 - t_0)H\right) = \exp(0) = I$

---

## Notar

Que la cantidad  $-\frac{i}{\hbar}(t - t_0)H$  es adimensional  $\frac{1}{\text{J}\cdot\text{s}} \cdot \text{J} = 1$

Es decir, es un número puro. Por eso podemos exponenciarla. El operador de evolución  $U(t, t_0)$  es, por tanto, también, una magnitud adimensional.

---



---

## Ejercicio

Escribe una función `psi(t, psi0, H)` que reciba como argumentos el array `psi0` de componentes de  $|\psi(0)\rangle$ , una matriz hermitica `H`, y un valor de `t`, y devuelva un vector columna correspondiente a  $|\psi(t)\rangle$ . Considera  $\hbar = 1$ .

---

```
def psi(t, psi0, H):
    lpsi = len(psi0)
    lH = H.shape[0]
    assert lpsi == lH

    psi0ket = np.array(psi0).reshape(lpsi, 1)
    psitket = np.dot(expm(- 1j*t*H), psi0ket)
    return psitket
```

## Ejercicio

Crea un Hamiltoniano aleatorio  $2 \times 2$  y considera que la base en la que está escrito es  $\{|0\rangle, |1\rangle\}$ . Partiendo de cualquiera de ellos, grafica las probabilidades de permanencia  $p(|0\rangle \rightarrow |0\rangle)$  y de transición  $p(|0\rangle \rightarrow |1\rangle)$  como funciones del tiempo (considera que  $\hbar = 1$  por simplicidad). Grafica también el valor esperado de  $H$  en ambos estados como función del tiempo. Observa el resultado e intenta explicar por qué es así.

```
from matplotlib import pyplot as plt
from scipy.linalg import expm

' definamos un Hamiltoniano '
A = np.random.randn(2,2) + np.random.randn(2,2)*1j
A = np.asmatrix(A)
H = A + A.getH()

'estado inicial |psi(0)>'
psi0 = [1,0]

tend = 2
steps = 100
tlist= np.linspace(0,tend, steps + 1)

'estado a tiempo t |psi(t)>'
psitketlist = [psi(tlist[i],psi0,H) for i in range(len(tlist))]

'probabilidades'
p00list = [np.abs(psitketlist[i][0,0]) for i in range(len(tlist))]
p01list = [np.abs(psitketlist[i][1,0]) for i in range(len(tlist))]

'valores esperados <H>(t)'
psitbralist= [psitketlist[i].conj().reshape(1,2) for i in range(len(tlist))]
Hvevtlist = [np.dot(psitbralist[i],np.dot(H,psitketlist[i])).real[0,0] for i in range(len(tlist))]
```

Graficamos

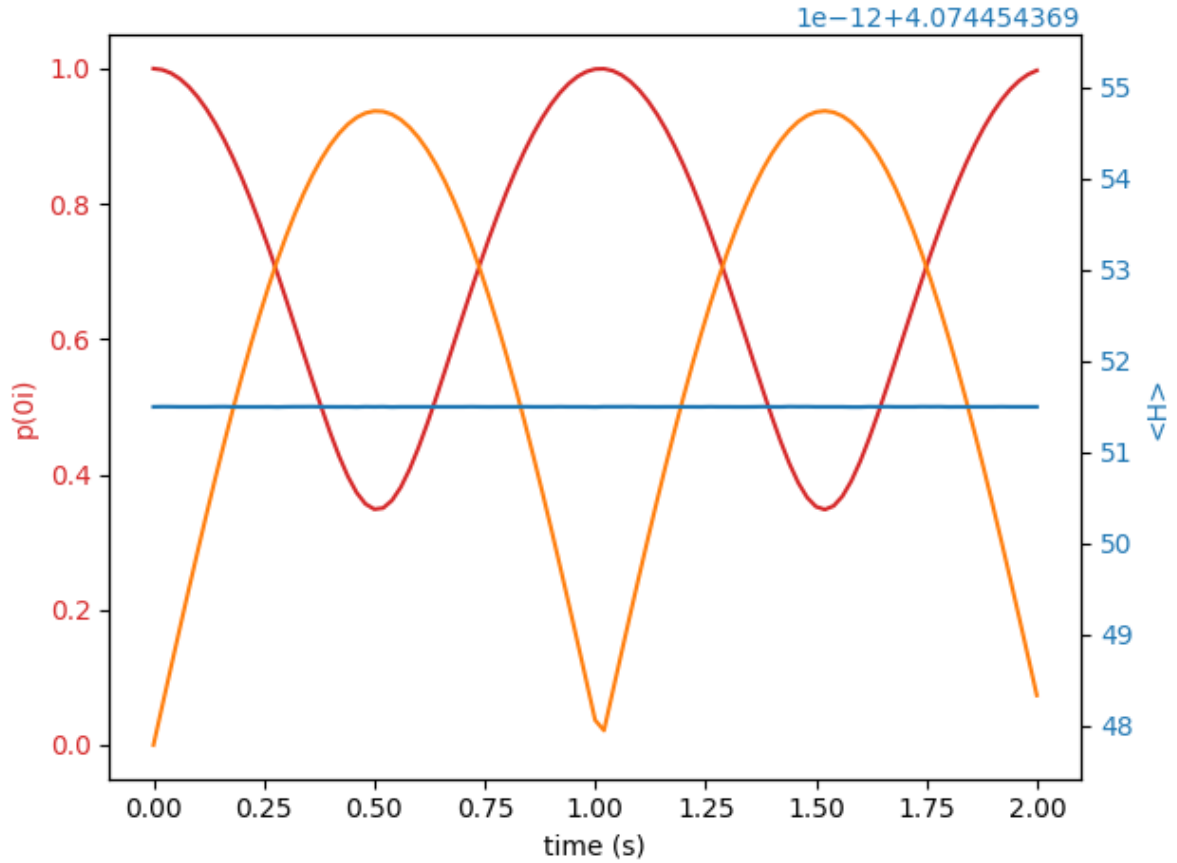
```
fig, ax1 = plt.subplots()

color = 'tab:red'
ax1.set_xlabel('time (s)')
ax1.set_ylabel('p(0i)', color= 'tab:red')
ax1.plot(tlist,p00list, color= 'tab:red')
ax1.plot(tlist,p01list, color= 'tab:orange')
ax1.tick_params(axis='y', labelcolor='tab:red')

ax2 = ax1.twinx() # instantiate a second axes that shares the same x-axis

color = 'tab:blue'
ax2.set_ylabel('<H>', color='tab:blue') # we already handled the x-label with ax1
ax2.plot(tlist,Hvevtlist, color='tab:blue')
ax2.tick_params(axis='y', labelcolor='tab:blue')

fig.tight_layout() # otherwise the right y-label is slightly clipped
plt.show()
```



### Evolución en la base autoestados de H

Sea  $\{|n\rangle\}$  una base de autoestados de  $H \Rightarrow H|n\rangle = E_n|n\rangle \quad n = 1, \dots, N$ .

En esta base  $H$  es una matriz  $H_{mn}$  diagonal

$$H_{mn} = \begin{bmatrix} E_0 & & \\ & E_1 & \\ & & \ddots \end{bmatrix} = E_m \delta_{mn}.$$

Entonces la matriz de evolución es también diagonal

$$U_{mn} = \exp\left(-\frac{i}{\hbar}tH_{mn}\right) = \begin{bmatrix} e^{-\frac{i}{\hbar}tE_0} & & \\ & e^{-\frac{i}{\hbar}tE_1} & \\ & & \ddots \end{bmatrix} = e^{-\frac{i}{\hbar}tE_m} \delta_{mn}.$$

### En resumen

- La evolución temporal de un autoestado de la energía es trivial (es una fase global)

$$U(t, 0)|n\rangle = e^{-\frac{i}{\hbar}tE_n}|n\rangle.$$

- Esta fase deja de ser trivial cuando afecta a una combinación lineal. La forma más eficaz de calcular la evolución de un estado arbitrario es expresarla en la base  $\{|n\rangle\}$ . Es decir, si a  $t = 0$

$$|\psi(t=0)\rangle = \sum_n c_n |n\rangle$$

entonces, a tiempo  $t$

$$|\psi(t)\rangle = \sum_n c_n e^{-\frac{i}{\hbar} E_n} |n\rangle .$$

---