04_functions

October 1, 2019

0.1 Notas de aula: Álgebra Linear, Autor: Jonas Maziero, Departamento de Física, UFSM

```
In [23]: %run init.ipynb
         %run 03_operators.ipynb
```

1 Funções matriciais de matrizes

Já vimos algumas funções escalares de operadores/matrizes, como por exemplo o determinante, o traço, o produto interno e a norma. Aqui veremos como devemos proceder para calcular funções de operadores que nos retornam operadores, como por exemplo $\log(\cdot)$, $\exp(\cdot)$, $\sqrt{\cdot}$, etc.

OBS. A primeira observação a se fazer aqui é que nossa intuição de como aplicar funções matriciais em matrizes está, em geral, errada, ou seja, se $A = (A_{j,k})$, em geral a função da matriz não é obtida aplicando-a em cada um dos elementos da matriz, i.e.,

$$f(A) \neq (f(A_{j,k})). \tag{1}$$

1.1 Definição de $f(A)$

Seja $A : \mathcal{H} \rightarrow \mathcal{H}$ um operador linear e seja

$$f(x) = \sum_{j=0}^{\infty}\left(\frac{1}{j!}\frac{d^j f(x)}{dx^j}\right)x^j \tag{2}$$

a expansão em série de Taylor para a função escalar $f(x)$. A função f tendo operadores lineares como argumento é calculada substituíndo-se, na série de Taylor, o escalar x pelo operador linear correspondente. ### Exemplo Como $e^x = \sum_{j=0}^{\infty} x^j/j! = 1 + x + x^2/2 + x^3/3! + \cdots$, teremos

$$e^A = \mathbb{I}_{\mathcal{H}} + A + \frac{A^2}{2} + \frac{A^3}{3!} + \cdots, \tag{3}$$

onde A^j é a composição $A \circ A \circ \cdots$ por j vezes.

```
In [31]: # para expandir uma função em séries de Taylor no sympy
         x = symbols("x")
         exp(x).series(x, x0=0, n = 7)
```

Out[31]:

$$1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + \frac{x^5}{120} + \frac{x^6}{720} + \mathcal{O}\left(x^7\right)$$

1.1.1 Exemplo: $e^{i\theta\vec{n}\cdot\vec{\sigma}} = \sigma_0 \cos\theta + i\vec{n}\cdot\vec{\sigma}\sin\theta$

Para $\{\sigma_j\}_{j=1}^3$ sendo as matrizes de Pauli e $\vec{n} \in \mathbb{R}^3$, definimos

$$\vec{n}\cdot\vec{\sigma} = \vec{n}^T\vec{\sigma} := \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \end{bmatrix} = \sum_{j=1}^3 n_j\sigma_j. \tag{4}$$

Para o resultado que provaremos na sequência, utilizaremos a seguinte indentidade:

$$\begin{aligned}
(\vec{n}\cdot\vec{\sigma})^2 &= \sum_{j=1}^3 n_j\sigma_j \sum_{k=1}^3 n_k\sigma_k = \sum_{j,k=1}^3 n_j n_k \sigma_j\sigma_k = \sum_{j,k=1}^3 n_j n_k(\delta_{j,k}\sigma_0 + sgn(j,k,l)i\sigma_l) && (5)\\
&= \sum_{j=1}^3 n_j^2\sigma_0 + \sum_{j\neq k} n_j n_k sgn(j,k,l)i\sigma_l && (6)\\
&= ||\vec{n}||^2\sigma_0 + n_1n_2sgn(1,2,3)i\sigma_3 + n_2n_1sgn(2,1,3)i\sigma_3 && (7)\\
&+ n_1n_3sgn(1,3,2)i\sigma_2 + n_3n_1sgn(3,1,2)i\sigma_2 + n_2n_3sgn(2,3,1)i\sigma_1 + n_3n_2sgn(3,2,1)i\sigma_1 && (8)\\
&= ||\vec{n}||^2\sigma_0 + (n_1n_2 - n_2n_1)i\sigma_3 + (-n_1n_3 + n_3n_1)i\sigma_2 + (n_2n_3 - n_3n_2)i\sigma_1 && (9)\\
&= ||\vec{n}||^2\sigma_0. && (10)
\end{aligned}$$

Agora, para $||\vec{n}|| = 1$, $\theta \in \mathbb{R}$ e $\sigma_0 = \mathbb{I}_{\mathbb{C}_2}$, usamos a série de Taylor para a exponencial e $(\vec{n}\cdot\vec{\sigma})^2 = \sigma_0$ para obter:

$$\begin{aligned}
e^{i\theta\vec{n}\cdot\vec{\sigma}} &= \sum_{j=0}^\infty \frac{(i\theta\vec{n}\cdot\vec{\sigma})^j}{j!} && (11)\\
&= \sigma_0 + i\theta\vec{n}\cdot\vec{\sigma} + \frac{(i\theta\vec{n}\cdot\vec{\sigma})^2}{2!} + \frac{(i\theta\vec{n}\cdot\vec{\sigma})^3}{3!} + \frac{(i\theta\vec{n}\cdot\vec{\sigma})^4}{4!} + \frac{(i\theta\vec{n}\cdot\vec{\sigma})^5}{5!} + \cdots && (12)\\
&= \sigma_0 + i\theta\vec{n}\cdot\vec{\sigma} - \frac{\theta^2\sigma_0}{2!} - \frac{i\theta^3\vec{n}\cdot\vec{\sigma}}{3!} + \frac{\theta^4\sigma_0}{4!} + \frac{i\theta^5\vec{n}\cdot\vec{\sigma}}{5!} + \cdots && (13)\\
&= \sigma_0(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \cdots) + i\vec{n}\cdot\vec{\sigma}(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \cdots) && (14)\\
&= \sigma_0\cos\theta + i\vec{n}\cdot\vec{\sigma}\sin\theta. && (15)
\end{aligned}$$

1.2 Fórmula de Baker-Campbell-Hausdorff

Para $c \in \mathbb{C}$ e $A, B : \mathcal{H} \to \mathcal{H}$ segue que

$$e^{cB}Ae^{-cB} = A + c[B,A] + \frac{c^2}{2!}[B,[B,A]] + \frac{c^3}{3!}[B,[B,[B,A]]] + \cdots. \tag{16}$$

Vamos verificar essa igualdade expandindo a exponencial em série de Taylor:

$$e^{cB}Ae^{-cB} = \left(\sum_{j=0}^{\infty}\frac{(cB)^j}{j!}\right) A \left(\sum_{k=0}^{\infty}\frac{(-cB)^k}{k!}\right) \tag{17}$$

$$= \left(\mathbb{I} + cB + \frac{c^2B^2}{2} + \frac{c^3B^3}{3!} + \cdots\right) A \left(\mathbb{I} - cB + \frac{c^2B^2}{2} - \frac{c^3B^3}{3!} + \cdots\right) \tag{18}$$

$$= A + cBA - cAB + \frac{c^2B^2}{2}A + A\frac{c^2B^2}{2} - \frac{2}{2}cBAcB - \frac{c^2B^2}{2}AcB + cBA\frac{c^2B^2}{2} + \frac{c^3B^3}{3!}A - A\frac{c^3B^3}{3!} + \cdots \tag{19}$$

$$= A + c[B,A] + \frac{c^2}{2}(BBA + ABB - 2BAB) + \frac{c^3}{3!}(BBBA - ABBB + 3BABB - 3BBAB) + \cdots \tag{20}$$

$$= A + c[B,A] + \frac{c^2}{2}(B(BA - AB) + (AB - BA)B) + \frac{c^3}{3!}(BB(BA - AB) - (AB - BA)BB + 2B(ABB - B \tag{21}$$

$$= A + c[B,A] + \frac{c^2}{2}(B[B,A] - [B,A]B) + \frac{c^3}{3!}(BB[B,A] + [B,A]BB - 2B[B,A]B)) + \cdots \tag{22}$$

$$= A + c[B,A] + \frac{c^2}{2}[B,[B,A]] + \frac{c^3}{3!}(B(B[B,A] - [B,A]B) + ([B,A]B - B[B,A])B) + \cdots \tag{23}$$

$$= A + c[B,A] + \frac{c^2}{2}[B,[B,A]] + \frac{c^3}{3!}(B[B,[B,A]] - [B,[B,A]B) + \cdots \tag{24}$$

$$= A + c[B,A] + \frac{c^2}{2!}[B,[B,A]] + \frac{c^3}{3!}[B,[B,[B,A]]] + \cdots . \tag{25}$$

1.2.1 Exemplo

Para

$$A = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \text{ e } B = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{26}$$

temos

$$[\sigma_z, \sigma_x] = 2i\sigma_y \text{ e } [\sigma_z, \sigma_y] = -2i\sigma_x. \tag{27}$$

Assim

$$
\begin{aligned}
e^{c\sigma_z}\sigma_x e^{-c\sigma_z} &= \sigma_x + c[\sigma_z,\sigma_x] + \frac{c^2}{2}[\sigma_z,[\sigma_z,\sigma_x]] + \frac{c^3}{3!}[\sigma_z,[\sigma_z,[\sigma_z,\sigma_x]]] && (28)\\
&+ \frac{c^4}{4!}[\sigma_z,[\sigma_z,[\sigma_z,[\sigma_z,\sigma_x]]]] + \frac{c^5}{5!}[\sigma_z,[\sigma_z,[\sigma_z,[\sigma_z,[\sigma_z,\sigma_x]]]]] + \cdots && (29)\\
&= \sigma_x + c2i\sigma_y + \frac{c^2}{2}2i[\sigma_z,\sigma_y] + \frac{c^3}{3!}2i[\sigma_z,[\sigma_z,\sigma_y]] + \frac{c^4}{4!}2i[\sigma_z,[\sigma_z,[\sigma_z,\sigma_y]]] + \frac{c^5}{5!}2i[\sigma_z,[\sigma_z,[\sigma_z,[\sigma_z,\sigma_y]]]] + \cdots && (30)\\
&= \sigma_x + c2i\sigma_y - \frac{c^2}{2}4i^2\sigma_x - \frac{c^3}{3!}4i^2[\sigma_z,\sigma_x] - \frac{c^4}{4!}4i^2[\sigma_z,[\sigma_z,\sigma_x]] - \frac{c^5}{5!}4i^2[\sigma_z,[\sigma_z,[\sigma_z,\sigma_x]]] + \cdots && (31)\\
&= \sigma_x + c2i\sigma_y - \frac{(2ic)^2}{2}\sigma_x - \frac{c^3}{3!}8i^3\sigma_y - \frac{c^4}{4!}8i^3[\sigma_z,\sigma_y] - \frac{c^5}{5!}8i^3[\sigma_z,[\sigma_z,\sigma_y]] + \cdots && (32)\\
&= \sigma_x + c2i\sigma_y - \frac{(2ic)^2}{2}\sigma_x - \frac{c^3}{3!}8i^3\sigma_y + \frac{c^4}{4!}16i^4\sigma_x + \frac{c^5}{5!}16i^4[\sigma_z,\sigma_x] + \cdots && (33)\\
&= \sigma_x + c2i\sigma_y - \frac{(2ic)^2}{2}\sigma_x - \frac{(2ic)^3}{3!}\sigma_y + \frac{(2ic)^4}{4!}\sigma_x + \frac{(2ic)^5}{5!}\sigma_y + \cdots && (34)\\
&= \sigma_x\left(1 - \frac{(2ic)^2}{2!} + \frac{(2ic)^4}{4!} + \cdots\right) + \sigma_y\left(2ci - \frac{(2ic)^3}{3!} + \frac{(2ic)^5}{5!} + \cdots\right) && (35)\\
&= \sigma_x\cos(2ic) + \sigma_y\sin(2ic). && (36)
\end{aligned}
$$

Exercício: Utilize a fórmula de BCH para calcular $e^{c\sigma_x}\sigma_y e^{-c\sigma_x}$.

In [24]:
```
def BCH(c,B,A,order):
    terms = []
    term = A
    co = A
    bch = term
    terms.append(term)
    for j in range(1,order):
        co = comm(B,co)
        term = pow(c,j)*co/fact(j)
        bch += term
        terms.append(term)
    return bch, terms
```

In [26]:
```
#A = Symbol("A", commutative = False)
#B = Symbol("B", commutative = False)
A = pauli(1)
B = pauli(3)
order = 6
bch, terms = BCH(c,B,A,order)
simplify(terms)
```

Out[26]:

$$\left[\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 2.0c\\ -2.0c & 0\end{bmatrix}, \begin{bmatrix}0 & 2.0c^2\\ 2.0c^2 & 0\end{bmatrix}, \begin{bmatrix}0 & 1.33333333333333c^3\\ -1.33333333333333c^3 & 0\end{bmatrix}, \begin{bmatrix}0 \\ 0.666666666666\end{bmatrix}\right.$$

2 Definição de $f(A)$ para operadores normais

Se

$$A = \sum_a a P_a \tag{37}$$

é a decomposição espectral do operador linear normal A, então suas funções podem ser calculadas usando

$$f(A) := \sum_a f(a) P_a. \tag{38}$$

Ou seja, depois de obter a decomposição espectral de A, atuamos a função nos seus autovalores.

2.1 Equivalência entre as definições para $A = A^\dagger$

Vamos verificar que essa definição é equivalente à definição principal via séries de Taylor. Como mostrado por Taylor, se decompomos uma função escalar qualquer em termos de potências de x:

$$f(x) = \sum_{j=0}^{\infty} c_j x^j, \tag{39}$$

os coeficientes são dados como na série de Maclaurin, i.e., $c_j = \frac{1}{j!}\frac{d^j f(x)}{dx^j}$. Assim, como

$$\begin{aligned} A^j = AAA\cdots &= \sum_a a P_a \sum_{a'} a' P_{a'} \sum_{a''} a'' P_{a''} \cdots && (40)\\ &= \sum_{a,a',a'',\cdots} a a' a'' \cdots P_a P_{a'} P_{a''} \cdots = \sum_{a,a',a'',\cdots} a a' a'' \cdots \delta_{aa'} \delta_{a'a''} \cdots P_a && (41)\\ &= \sum_a a^j P_a, && (42) \end{aligned}$$

teremos que

$$\begin{aligned} f(A) &= \sum_{j=0}^{\infty} c_j A^j = \sum_{j=0}^{\infty} c_j \sum_a a^j P_a = \sum_a \left(\sum_{j=0}^{\infty} c_j a^j \right) P_a && (43)\\ &= \sum_a f(a) P_a. && (44) \end{aligned}$$

2.1.1 Exemplo

$$
\begin{aligned}
e^{A} &= \mathbb{I}_{\mathcal{H}} + A + \frac{A^2}{2} + \frac{A^3}{3!} + \cdots & (45)\\
&= \sum_a P_a + \sum_a aP_a + \frac{\sum_a aP_a \sum_{a'} a'P_{a'}}{2} + \frac{\sum_a aP_a \sum_{a'} a'P_{a'} \sum_{a''} a''P_{a''}}{3!} + \cdots & (46)\\
&= \sum_a P_a + \sum_a aP_a + \frac{\sum_{a,a'} aa'P_aP_{a'}}{2} + \frac{\sum_{a,a',a''} aa'a''P_aP_{a'}P_{a''}}{3!} + \cdots & (47)\\
&= \sum_a P_a + \sum_a aP_a + \frac{\sum_a a^2P_a}{2} + \frac{\sum_a a^3P_a}{3!} + \cdots & (48)\\
&= \sum_a (1 + a + \frac{a^2}{2} + \frac{a^3}{3!} + \cdots)P_a & (49)\\
&= \sum_a e^a P_a. & (50)
\end{aligned}
$$

Sobre $\sqrt{A}$ Em geral $\sqrt{A}$ não é bem ou unicamente definida. Para operadores normais podemos ver que definimos $\sqrt{A} = \sum_a \sqrt{a}P_a$ então $\sqrt{A}\sqrt{A} = \sum_a \sqrt{a}P_a \sum_{a'} \sqrt{a'}P_{a'} = \sum_{a,a'} \sqrt{a}\sqrt{a'}P_aP_{a'} = \sum_{a,a'} \sqrt{a}\sqrt{a'}\delta_{a,a'}P_a = \sum_a aP_a = A$.

Exercício: Verifique que qualquer operador Hermitiano e unitário é a raiz quadrada do operador identidade.

Exercício: Calcule $\sqrt{\sigma_y}$.

```
In [29]: def mat_func(A,func): # retorna exp(A) ou sqrt(A) de um operador normal
             eig = A.eigenvects()
             d = A.shape[0]
             A_func = zeros(d,d)
             ne = 0
             j = 0
             lk = 0
             while ne < d:
                 mult = eig[j][1] # multiplicidade
                 ne += mult # número de autovalores considerados até então
                 for k in range(0,mult):
                     Proj = proj(eig[j][2][k])
                     if func == 'sqrt':
                         A_func += sqrt(eig[j][0])*(Proj/trace(Proj))
                     elif func == 'exp':
                         A_func += exp(eig[j][0])*(Proj/trace(Proj))
                 j += 1
             return A_func
```

```
In [30]: A = pauli(1)
         #a,b,c,d = symbols("a b c d")
         #A = Matrix([[a,b],[c,d]])
         mat_func(A,'sqrt')
```

Out[30]:

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

2.2 Exemplo de aplicação: Entropia de von Neumann

O estado de uma sistema física pode ser descrito por uma matriz positiva com traço igual a um, a chamada matriz densidade. Considere a decomposição espectral de $\rho = \sum_r rP_r$, com P_r sendo projetores 1D. A entropia de von Neumann dessa matriz é definida e dada por:

$$\begin{aligned}
S(\rho) &= -Tr(\rho\log_2(\rho)) = -Tr\left(\rho\log_2\left(\sum_r rP_r\right)\right) = -Tr\left(\left(\sum_{r'} r'P_{r'}\right)\left(\sum_r \log_2(r)P_r\right)\right) && (51)\\
&= -Tr\left(\sum_{r',r} r'\log_2(r)P_{r'}P_r\right) = -Tr\left(\sum_{r',r} r'\log_2(r)\delta_{rr'}P_r\right) = -\sum_r r\log_2(r)Tr(P_r) && (52)\\
&= -\sum_r r\log_2(r)\sum_{r'}(|r'\rangle, P_r(|r'\rangle)) = -\sum_r r\log_2(r)\sum_{r'}(|r'\rangle, \langle r|r'\rangle|r\rangle) && (53)\\
&= -\sum_r r\log_2(r)\sum_{r'}\delta_{r,r'}(|r'\rangle, |r\rangle) = -\sum_r r\log_2(r)\langle r|r\rangle && (54)\\
&= -\sum_r r\log_2(r), && (55)
\end{aligned}$$

que é a chamada entropia de Shannon da distribuição de probabilidades $\{r\}$ (os autovalores de ρ).

Exercício: Verifique que a entropia de von Neumann de $\rho = |\psi\rangle\langle\psi|$ é nula. Dica: Considere que a base de autovetores de ρ é constituída por $|\psi\rangle$ e por estados ortogonais e este.

Exercício: Verifique que a entropia de von Neumann de $\rho = d^{-1}\mathbb{I}_d$ é $\log_2(d)$.

In [96]:
```
# Outside this function, initialize: evals = zeros(d,1)
def eVals(A):
    d = A.shape[0]
    eig = A.eigenvects()
    ne = 0
    j = 0
    lk = 0
    while ne < d:
        mult = eig[j][1]
        ne += mult
        nk = lk + mult
        for k in range(lk,nk):
            evals[k] = eig[j][0]
        lk = nk
        j += 1
    return evals
```

In [97]:
```
def shannon(pv):
    d = pv.shape[0]
```

```
        H = 0
        for j in range(d):
            if isinstance(pv[j], sympy.Expr)==False and pv[j] < 1.e-15 and pv[j] > (1.0-1.e
                continue
            else:
                H -= pv[j]*log(pv[j],2)
        return H
```

```
In [98]: def von_neumann(rho):
             d = rho.shape[0]
             evals = zeros(d,1)
             evals = eVals(rho)
             return shannon(evals)
```

```
In [108]: a, b, c, d = symbols("a b c d"); rho = Matrix([[a,c],[d,b]])
          #rho = Matrix([[1/2,0],[0,1/2]])
          evals = zeros(rho.shape[0],1); von_neumann(rho)
```

Out[108]:

$$-\frac{1}{\log(2)}\left(\frac{a}{2}+\frac{b}{2}-\frac{1}{2}\sqrt{a^2-2ab+b^2+4cd}\right)\log\left(\frac{a}{2}+\frac{b}{2}-\frac{1}{2}\sqrt{a^2-2ab+b^2+4cd}\right)-\frac{1}{\log(2)}\left(\frac{a}{2}+\frac{b}{2}+\frac{1}{2}\sqrt{a^2-\right.$$

Exercício: Para $H=\sum_h hP_h$, com P_h um projetor 1D e $\beta\in\mathbb{R}$, calcule a expressão (decomposição espectral) para o estado térmico de Gibbs:

$$\rho_g=\frac{e^{-\beta H}}{Z}. \tag{56}$$

Use $Tr(\rho_g):=1$ para obter a expressão da função de partição, que é o escalar Z nessa equação.

```
In [120]: def rho_gibbs(beta, H): # retorna o estado de Gibbs
              expmbH = mat_func(-beta*H,'exp')
              pf = partition_function(expmbH)
              return expmbH/pf
```

```
In [121]: def partition_function(expmbH):
              return trace(expmbH)
```

```
In [126]: w, b, hb = symbols('omega beta hbar')
          H = -(hb*w/2)*pauli(1)
          rho_gibbs(b, H)
```

Out[126]:

$$\begin{bmatrix}\frac{\frac{1}{2}e^{\frac{\beta\hbar}{2}\omega}+\frac{1}{2}e^{-\frac{\beta\hbar}{2}\omega}}{e^{\frac{\beta\hbar}{2}\omega}+e^{-\frac{\beta\hbar}{2}\omega}} & \frac{\frac{1}{2}e^{\frac{\beta\hbar}{2}\omega}-\frac{1}{2}e^{-\frac{\beta\hbar}{2}\omega}}{e^{\frac{\beta\hbar}{2}\omega}+e^{-\frac{\beta\hbar}{2}\omega}}\\ \frac{\frac{1}{2}e^{\frac{\beta\hbar}{2}\omega}-\frac{1}{2}e^{-\frac{\beta\hbar}{2}\omega}}{e^{\frac{\beta\hbar}{2}\omega}+e^{-\frac{\beta\hbar}{2}\omega}} & \frac{\frac{1}{2}e^{\frac{\beta\hbar}{2}\omega}+\frac{1}{2}e^{-\frac{\beta\hbar}{2}\omega}}{e^{\frac{\beta\hbar}{2}\omega}+e^{-\frac{\beta\hbar}{2}\omega}}\end{bmatrix}$$