

Lab 5 - Átomo de 2 níveis + Campos externos – pós-aula (Aula 11)

April 30, 2021

1 O Modelo de 2 níveis

O modelo quântico de interação da luz com a matéria mais simples que podemos fazer é o de um átomo de dois níveis interagindo com um modo do campo eletromagnético. O modelo, em princípio, pode ser usado para representar “virtualmente” qualquer sistema de dois níveis, com pequenas modificações.

Para introduzir esses conceitos, de uma forma didática (com o mínimo de “distrações”), vou usar como base o modelo que já discutimos nas aulas anteriores, que é o sistema de (um) spin-1/2 em campo magnético externo. Veremos que esse modelo (mínimo) já contém a maioria dos elementos necessários para descrever sistemas de dois níveis mais gerais, o que faremos mais adiante no curso. Nesta primeira aula, vamos introduzir os elementos básicos, mapeando o problema no formalismo que discutimos nas últimas aulas.

Mais adiante, veremos que exceto por alguns detalhes –que são importante na prática, basicamente o mesmo modelo (i.e., o mesmo formalismo) pode ser usado para descrever um átomo de dois níveis interagindo com um laser, por exemplo. Do ponto de vista formal, porém, **veremos que não há grandes mudanças na descrição** (matemática) desses sistemas, aparentemente bastante distintos.

1.1 Representação de um átomo de dois níveis

Seja um átomo com dois estados não degenerados $|1\rangle$ e $|2\rangle$, representando o estado fundamental (mais baixa energia) e o estado excitado, com respectivas energias E_1 e E_2 .

O operador Hamiltoniano (diagonalizado) desse sistema é

$$H_a = E_1 |1\rangle \langle 1| + E_2 |2\rangle \langle 2|.$$

Em um espaço de dimensão 2, temos dois vetores na base e podemos ter $2 \times 2 = 4$ operadores linearmente independentes, construídos a partir deles. Uma possível escolha é a seguinte

$$\begin{aligned}\mathbb{1} &= |1\rangle \langle 1| + |2\rangle \langle 2| \\ \sigma_z &= |2\rangle \langle 2| - |1\rangle \langle 1| \\ \sigma^+ &= |2\rangle \langle 1| \\ \sigma^- &= |1\rangle \langle 2|.\end{aligned}$$

A escolha desses operadores ficará evidente quando se observar o significado de cada um deles. Note, porém, que os **dois últimos não são Hermitianos**. Eles poderiam ser substituídos pelos operadores $\sigma_{x,y}$, como já vimos antes ao escrever a matriz densidade em termos dos operadores de Pauli e a identidade, mas veremos que conveniente usá-los aqui.

$$\begin{aligned}\sigma_x &= \sigma^- + \sigma^+ \\ \sigma_y &= i(\sigma^- - \sigma^+).\end{aligned}$$

Para entender o significado de cada um deles, basta observar sua ação em um estado genérico

$$|\psi\rangle = c_1 |1\rangle + c_2 |2\rangle$$

obtendo

$$\begin{aligned}\sigma^+ |\psi\rangle &= c_1 |2\rangle \\ \sigma^- |\psi\rangle &= c_2 |1\rangle \\ \sigma_z |\psi\rangle &= c_2 |2\rangle - c_1 |1\rangle.\end{aligned}$$

Pode-se observar que o operador σ^+ promove uma transição do estado estado fundamental para o estado excitado e vice-versa. O operador σ_z , por outro lado, permite-nos calcular a diferença de população, como pode ser observado pelo seu **valor esperado** (como é Hermitiano, define um observável físico para o qual podemos calcular valores esperados físicos)

$$\langle \sigma_z \rangle = \langle \psi | \sigma_z | \psi \rangle = |c_2|^2 - |c_1|^2.$$

Se tivéssemos um *ensemble* de N átomos, por exemplo, a quantidade $N\langle \sigma_z \rangle$ corresponderia à população invertida (no estado excitado). No caso de um único átomo, podemos interpretar isso em termos das probabilidades indicadas acima.

1.1.1 Reescrevendo o Hamiltoniano

É conveniente reescrever o operador Hamiltoniano numa forma mais favorável e familiar, usando os operadores definidos acima. Podemos fazer isso deslocando os níveis de energia (redefinindo o nível de referência) para ficarem simétricos em torno de $E=0$, de modo a colocá-lo na forma

$$H_A = \frac{1}{2} \hbar \omega_{21} \sigma_z$$

com a definição da frequência de transição

$$\omega_{21} = \frac{1}{\hbar} (E_2 - E_1)$$

O efeito nos níveis de energia pode ser entendido ao observar a matriz do Hamiltoniano

$$\begin{pmatrix} E_2 & 0 \\ 0 & E_1 \end{pmatrix} \rightarrow \frac{\hbar}{2} \begin{pmatrix} \omega_{21} & 0 \\ 0 & -\omega_{21} \end{pmatrix}$$

o que equivale a colocar o nível *zero* de energia entre os dois níveis os níveis E_1 e E_2 .

É conveniente, também, lembrar a forma matricial dos operadores

$$\begin{aligned} \mathbb{1} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \sigma_z &= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ \sigma^+ &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\ \sigma^- &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \end{aligned}$$

e dos vetores

$$\begin{aligned} |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ |2\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ |\psi\rangle &= c_1 |1\rangle + c_2 |2\rangle = \begin{pmatrix} c_2 \\ c_1 \end{pmatrix} \end{aligned}$$

Com essas definições, o problema está posto e temos os elementos básicos para começar a discutir aplicações (e consequências) da interação desse sistema de dois níveis com campos externos. Antes, porém, vamos definir o problema computacionalmente, usando as ferramentas que aprendemos nas últimas aulas (semana T3).

1.2 Definindo o problema (analiticamente) no Sympy

```
[1]: import sympy as sy
import numpy as np
import sympy.physics as phys
from sympy import symbols, Matrix, sin, cos, exp, diff, integrate

sx = phys.matrices.msigma(1) # sigma_x
sy = phys.matrices.msigma(2) # sigma_y
sz = phys.matrices.msigma(3) # sigma_z
sp = Matrix([[0,1],[0,0]]) # sigma^+
sm = Matrix([[0,0],[1,0]]) # sigma^-
I = Matrix([[1,0],[0,1]]) # unit matrix

g = k1 = Matrix([[0],[1]]) # ground-state
e = k2 = Matrix([[1],[0]]) # excited-state
```

```

c1, c2 = symbols('c1 c2')
E1, E2 = symbols('E1 E2')
hbar, omega21, omega = symbols('hbar omega21 omega')

psi = c1*k1 + c2*k2

Ho = E1*k1*k1.T + E2*k2*k2.T
Ha = 1/2*(hbar*omega21)*sz

```

```
[2]: sm*e == g # testa se sigma_ em |e> leva a |g>
```

```
[2]: True
```

```
[3]: sp*g == e # testa se sigma+ em |g> leva a |e>
```

```
[3]: True
```

```
[4]: sp*g # mostra o resultado de sigma+ em |g>
```

```
[4]:  $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 
```

```
[5]: sp*e # mostra o resultado de sigma+ em |e> --> null (zero) result!
```

```
[5]:  $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ 
```

```
[6]: psi
```

```
[6]:  $\begin{bmatrix} c_2 \\ c_1 \end{bmatrix}$ 
```

```
[7]: psi.T*sz*psi # valor esperado de sigma_z
```

```
[7]:  $[-c_1^2 + c_2^2]$ 
```

```
[8]: Ha # Hamiltoniano do átomo
```

```
[8]:  $\begin{bmatrix} 0.5\hbar\omega_{21} & 0 \\ 0 & -0.5\hbar\omega_{21} \end{bmatrix}$ 
```

```
[9]: Ho # Hamiltoniano original (note o ordenamento!)
```

```
[9]:  $\begin{bmatrix} E_2 & 0 \\ 0 & E_1 \end{bmatrix}$ 
```

```
[ ]:
```

1.3 Definindo o problema (numericamente) no QuTiP

Uma das principais vantagens do QuTiP é que poder usá-lo para resolver numericamente a dinâmica (evolução temporal) quântica. Primeiro vamos considerar a evolução unitária (sistema fechado) do sistema, na presença do campo externo. Por simplicidade, usaremos o exemplo de um spin-1/2 num campo magnético externo (problema da lista), alinhado na direção \hat{x} .

```
[10]: import qutip
import numpy as np
import matplotlib.pyplot as plt
```

```
[11]: w = 1
w21 = 2*np.pi
q1 = qutip.Qobj([[0],[1]]) # estado |1>
q2 = qutip.Qobj([[1],[0]]) # estado |2>
tempos = np.linspace(0.0, 2.0, 100)

H = 0.5*w21*qutip.sigmay()
#H = 0.5*w21*(qutip.sigmax()+qutip.sigmay())
result = qutip.mesolve(H, q1, tempos, [],
                      [qutip.sigmax(), qutip.sigmay(),qutip.sigmaz()])

# plota os resultados
fig, ax = plt.subplots(figsize=(10,5))
ax.plot(result.times, result.expect[2])
ax.plot(result.times, result.expect[1])
ax.plot(result.times, result.expect[0])
ax.set_xlabel('Tempo',fontsize=16)
ax.set_ylabel('Valores esperados',fontsize=16)
#ax.legend(("$\sigma_z$", "$\sigma_y$"),fontsize=16)
ax.legend(("$\sigma_z$", "$\sigma_y$", "$\sigma_x$"),fontsize=16)
ax.grid(True)
plt.show()
```



```

0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
array([-1.          , -0.99195482, -0.96794872, -0.92836798, -0.8738495 ,
-0.80527048, -0.72373441, -0.63055324, -0.5272263 , -0.41541614,
-0.29692184, -0.17365   , -0.0475841 ,  0.07924744,  0.20480386,
 0.32706494,  0.44406347,  0.55391691,  0.65485768,  0.74526161,
 0.82367406,  0.88883334,  0.939691   ,  0.97542872,  0.99547146,
 0.9994967 ,  0.98743967,  0.95949438,  0.91611049,  0.85798608,
 0.78605639,  0.70147881,  0.60561424,  0.50000516,  0.38635087,
 0.2664801 ,  0.14232162,  0.01587315, -0.11083071, -0.23575129,
-0.35687856, -0.47226358, -0.58004976, -0.67850279, -0.76603851,
-0.84124845, -0.90292243, -0.9500681 , -0.98192685, -0.99798605,
-0.9979873 , -0.98193057, -0.95007422, -0.90293085, -0.84125902,
-0.76605106, -0.67851711, -0.58006562, -0.47228072, -0.3568967 ,
-0.23577013, -0.11084998,  0.01585377,  0.14230243,  0.26646141,
 0.38633296,  0.49998832,  0.60559874,  0.70146491,  0.78604432,
 0.85797603,  0.91610264,  0.95948886,  0.98743657,  0.99949607,
 0.99547333,  0.97543305,  0.93969771,  0.88884231,  0.82368514,
 0.74527462,  0.65487241,  0.55393311,  0.44408088,  0.32708328,
 0.20482284,  0.07926676, -0.04756474, -0.17363091, -0.29690331,
-0.41539847, -0.52720977, -0.63053812, -0.72372095, -0.8052589 ,
-0.87383998, -0.9283607 , -0.96794379, -0.99195233, -1.          ])]

```

[13]: *# plotando os pontos calculados na esfera de Bloch*

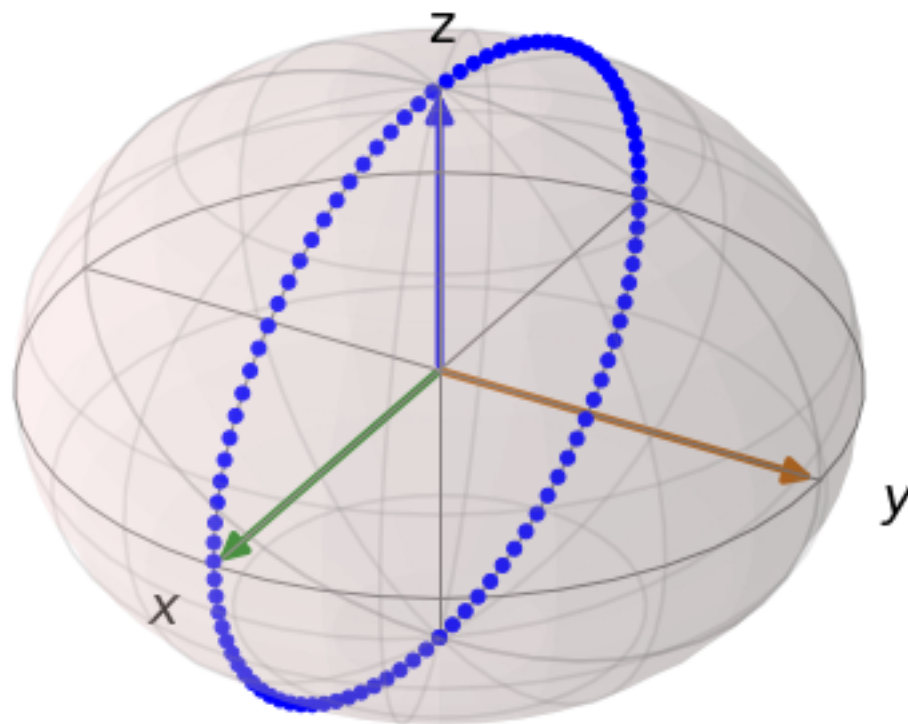
```

x = np.array([1, 0, 0]) # eixo x
y = np.array([0, 1, 0]) # eixo y
z = np.array([0, 0, 1]) # eixo z

# vetores de Bloch (calculados)
rx = result.expect[0]
ry = result.expect[1]
rz = result.expect[2]
r = [rx, ry, rz]

# Pontos mostram a evolução (rotação) na esfera de Bloch
b = qutip.Bloch()
b.zlabel = ("z", "")
b.add_vectors([x, y, z])
b.add_points(r)
b.show()

```



Os pontos mostram a “trajetória” da dinâmica (evolução temporal) dos valores esperados dos operadores no estado (inicialmente $|1\rangle$), representados na esfera de Bloch.

[]: