

▼ Modelo: Sistema 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, e não apenas "átomos".

Para introduzir esses conceitos de uma forma didática (com o mínimo de "distracões"), vou usar como base o modelo que já discutimos nas aulas anteriores, que é o sistema de 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 -- porém, neste caso a interação será do tipo **dipolo elétrico**. Do ponto de vista formal, porém, **veremos que não há grandes mudanças na descrição** (matemática) desses sistemas, que aparentemente podem parecer inicialmente bastante distintos.

▼ Representação de um átomo de dois níveis

Seja um sistema 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

$$H_0 = E_1 |1\rangle \langle 1| + E_2 |2\rangle \langle 2|.$$

Em um espaço de dimensão 2, temos dois vetores na base e podemos, portanto, 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 particular 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 de Pauli $\sigma_{x,y}$, como já vimos antes ao escrever a matriz densidade em termos dos operadores de Pauli e a identidade, mas veremos que escolha indicada acima será mais conveniente aqui. De todo modo, sabemos escrever a relação entre eles:

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

Para entender o significado de cada um desses operadores, 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, portanto, que o operador σ^+ promove uma transição do estado estado fundamental para o estado excitado, enquanto o op. σ^- faz o inverso disso. 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.

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 (nível de referência) entre os dois níveis com energias (anteriormente) 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 vimos nas últimas aulas.

▼ Definindo o problema (analiticamente) no Sympy

```
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
```

```
sm*e == g # testa se sigma_ em |e> leva a |g>
```

```
True
```

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

```
True
```

```
sp*g # mostra o resultado de sigma+ em |g>
```

```

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

```

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

```

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

```

```
psi
```

```

$$\begin{bmatrix} c_2 \\ c_1 \end{bmatrix}$$

```

```
psi.T*sz*psi # valor esperado de sigma_z
```

```

$$[-c_1^2 + c_2^2]$$

```

```
Ha # Hamiltoniano do átomo
```

```

$$\begin{bmatrix} 0.5\hbar\omega_{21} & 0 \\ 0 & -0.5\hbar\omega_{21} \end{bmatrix}$$

```

```
Ho # Hamiltoniano original (note o ordenamento!)
```

```

$$\begin{bmatrix} E_2 & 0 \\ 0 & E_1 \end{bmatrix}$$

```

▼ Definindo o problema (numericamente) no [QuTiP](#)

Uma das principais vantagens do QuTiP é poder resolver numericamente a dinâmica (eq. de 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 externos (problema da lista), alinhado na direção \hat{y} (ou \hat{x} , ou ambas), na condição de ressonância.

```
!pip install qutip
import qutip
import numpy as np
import matplotlib.pyplot as plt
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
```

```
Collecting qutip
```

```
  Downloading qutip-4.7.0-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_12_x86_64.manylinux2010_x86_64.whl (14.7 MB)
    14.7 MB 3.6 MB/s
```

```
Requirement already satisfied: numpy>=1.16.6 in /usr/local/lib/python3.7/dist-packages (from qutip) (1.21.6)
```

```
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-packages (from qutip) (1.4.1)
```

```
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from qutip) (21.3)
```

```
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->qutip) (3)
```

```
Installing collected packages: qutip
```

```
Successfully installed qutip-4.7.0
```

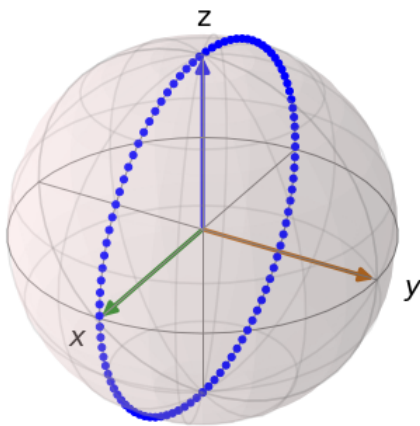
```
w = 1
w21 = 2*np.pi
q1 = qutip.Qobj([[0],[1]]) # estado |1>
q2 = qutip.Qobj([[1],[0]]) # estado |2>
tempos = np.linspace(0.0, 2.0, 100)
```

```
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()])
```



```
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.    ]]
```

```
# 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.render()
```



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