



QUTIP COMO RECURSO DIDÁTICO PARA APLICAÇÃO DE CONCEITOS DE MECÂNICA QUÂNTICA

QUTIP as a Teaching Resource for Applying Concepts of Quantum Mechanics

Gabriella Gonçalves Damas¹

Ângelo Antônio Santos de Oliveira²

Resumo: Este trabalho destaca a importância de integrar conceitos da mecânica quântica em cursos de graduação vinculados às áreas das ciências exatas e de tecnologias. Utilizamos o artigo de Jesus *et al.* (2021) como ponto de partida, que detalha o uso de uma plataforma de código aberto para trabalhar com algoritmos quânticos e aplicar conceitos vistos antes em sala de aula de computação quântica em circuitos quânticos, utilizando um *software* computacional. A partir deste trabalho surge uma nova proposta de recurso pedagógico para o ensino de física, agora apresentando uma biblioteca de *software* livre criada por Johansson, Nation e Nori em 2012, o QuTip, que possibilita a simulação de sistemas quânticos e operações matemáticas com a notação de Dirac. Destacamos as principais características do QuTiP, incluindo sua integração com o ambiente Jupyter Notebook e a capacidade de descrever sistemas quânticos e executar operações com operadores e vetores de estado. Além disso, oferecemos uma revisão dos conceitos fundamentais da mecânica quântica que podem ser explorados com esta plataforma. Os resultados obtidos e apresentados neste trabalho são consistentes com as previsões teóricas, sugerindo o potencial do QuTiP como uma ferramenta de trabalho em sala de aula.

Palavras-chave: Mecânica Quântica; Simulação; Python.

Abstract: This work emphasizes the importance of integrating quantum mechanics concepts into undergraduate courses related to the fields of exact sciences and technology. We use the article by Jesus *et al.* (2021) as a starting point, which details the use of an open-source platform for working with quantum algorithms and applying previously learned concepts from quantum computing in quantum circuit classrooms, using computational software. This work gives rise to a new pedagogical resource proposal for physics education, now presenting a freely available software library created by Johansson, Nation, and Nori in 2012, called QuTip, which enables the simulation of quantum systems and mathematical operations using Dirac notation. We highlight the main features of QuTiP, including its integration with the Jupyter Notebook environment and its ability to describe quantum systems and perform operations with operators and state vectors. Additionally, we provide an overview of fundamental quantum mechanics concepts that can be explored using this platform. The results obtained and presented in this work align with theoretical predictions, suggesting the potential of QuTiP as a classroom tool.

Keywords: Quantum Mechanics; Simulation; Python.

¹Mestra em Física. UFG. <https://orcid.org/0000-0003-3376-9281>. gabriella.damasg@gmail.com.

²Mestre em Educação em Ciências e Matemática. UFG. <https://orcid.org/0000-0001-9272-2569>. angelosantos66@gmail.com.



1 Introdução

A compreensão dos estudantes em relação aos conceitos de Mecânica Quântica (MQ) é um desafio significativo (MONTEIRO; NARDI; BASTOS FILHOS, 2009), e aprimorar o ensino desse tópico no nível universitário e sua inclusão no currículo do ensino médio são áreas de grande relevância na pesquisa em Ensino de Ciências (MOTA, 2000; PINTO; ZANETIC, 1999; TERRAZZAN, 1994; 1992; FARMELO, 1992). Os princípios fundamentais dessa teoria desafiam nossa visão de mundo, enraizada no paradigma clássico, muitas vezes levando a conclusões que podem parecer “antinaturais”. É evidente a necessidade de iniciativas voltadas para o aprimoramento do ensino da MQ, visando tornar esse campo de estudo mais acessível e compreensível para os estudantes e, assim, promover uma compreensão mais profunda e eficaz dos princípios quânticos (GRECA; MOREIRA; VICTORIA, 2001).

Na literatura, existem diversos trabalhos que destacam as aplicações de diferentes tecnologias no ensino de Física (OLIVEIRA, 2022) em especial no Ensino de Física Moderna (FM) e Mecânica Quântica (MQ) (PIETROCOLA; BROCKINGTON, 2003). Apesar de tais trabalhos buscarem alternativas válidas para o processo de ensino-aprendizagem desses conteúdos, que frequentemente envolvem conceitos e definições complexas, encontram-se poucas informações sobre a introdução desses recursos em cursos de graduação voltados para as áreas de Ciências Exatas e Tecnologia. É notável que a pesquisa existente dá uma ênfase significativa ao Ensino de Física Moderna no contexto do ensino médio, enquanto há uma relativamente menor atenção dada ao ensino superior (OSTERMANN; MOREIRA, 2016; REZENDE JUNIOR; CRUZ, 2009). A carência de trabalhos dedicados a essa transição e adaptação indica uma oportunidade para futuras pesquisas e desenvolvimento curricular nesta área, com potencial para enriquecer significativamente a experiência de aprendizado no nível superior (PERFOLL; REZENDE JUNIOR, 2006).

Entender adequadamente a MQ desempenha um papel crucial durante o período de graduação, uma vez que essa teoria é fundamental para a compreensão de fenômenos físicos complexos que desempenham um papel central em diversas áreas tecnológicas, incluindo engenharia, produção e computação. Nesse sentido, Perfoll e Rezende Junior (2006), apontam que

[...] uma formação excluindo esses conceitos pode estar contribuindo para formar consumidores profissionais de tecnologia, ou seja, profissionais que não estarão capacitados para o desenvolvimento dessas tecnologias e que o torna inevitavelmente dependente de grandes centros de desenvolvimento tecnológico, que se encontram principalmente em países com tradição científica.

Da estrutura atômica aos fundamentos da própria MQ, o domínio dessa teoria possibilita *insights* profundos sobre a natureza da matéria e da energia. Além disso, à medida que a pesquisa nesse campo avança, o estudo da MQ torna-se cada vez mais relevante. Suas aplicações práticas, como a computação quântica e a criptografia quântica, estão emergindo e têm potencial de revolucionar diversas áreas (FEDORTCHENKO, 2016).

Em 2012, Johansson, Nation e Nori³ apresentaram o QuTip (*Quantum Toolbox*), uma biblioteca de código aberto para Física Computacional. Essa ferramenta permite a simulação de sistemas quânticos, a realização de operações matemáticas usando a notação de Dirac e inclui recursos visuais para a representação gráfica e outros tipos de conteúdo. O QuTip pode ser acessado por meio de uma interface Python, como o Jupyter Notebook, por exemplo.

Nesse sentido, o QuTip é uma valiosa ferramenta para o ensino de Mecânica Quântica na graduação. Ele é amplamente utilizado em pesquisas e simulações computacionais (DAMAS; ASSIS; ALMEIDA, 2023), sendo um software livre e acessível. O ensino do processamento de informação quântica com o QuTip é uma abordagem excelente para introduzir os conceitos fundamentais da MQ, permitindo uma compreensão prática e interativa.

Adicionalmente, o QuTip não requer um grande poder de processamento e pode ser utilizado com computadores domésticos, tornando-o ainda mais acessível aos estudantes. Assim, oferece uma solução eficiente e viável para o ensino de MQ, facilitando o aprendizado em um ambiente computacional amigável.

No seu trabalho, Jesus *et al.* (2021) apresentam uma abordagem para o ensino de computação utilizando um software de Física Computacional. Nesta pesquisa, os autores disponibilizam códigos ao longo do texto, possibilitando que mesmo aqueles com pouca experiência em computação científica possam reproduzi-los e aplicar os métodos discutidos em seus próprios projetos de computação quântica.

Neste artigo, seguindo uma abordagem semelhante, explora-se o potencial didático do QuTip como uma ferramenta para o Ensino de Física. Nosso objetivo é apresentar o QuTip como um recurso para o ensino de MQ, permitindo a aplicação prática e interativa de conceitos fundamentais em um ambiente educacional, tornando o aprendizado mais intuitivo e envolvente em sala de aula.

2 Computação Quântica e Ensino

As abordagens convencionais adotadas nos cursos não oferecem um ambiente propício para que os estudantes assimilem a nova perspectiva necessária para compreender os fenômenos regidos pelos princípios da MQ (JOHNSTON; CRAWFORD; FLETCHER, 1998). Essa lacuna na compreensão dos fundamentos da MQ leva à necessidade de desenvolver estratégias didáticas que promovam uma aprendizagem mais profunda e significativa (OSTERMANN; MOREIRA, 2016; GRECA; MOREIRA; VICTORIA, 2001).

É nesse contexto que surge a proposta de introduzir uma ferramenta de simulação como um recurso valioso para o ensino de MQ (OSTERMANN; PRADO; RICCI, 2008). Essa abordagem visa preencher a lacuna entre a teoria e a compreensão prática, permitindo que os alunos visualizem e experimentem os conceitos quânticos de uma maneira mais tangível. Trabalhos recentes têm explorado o uso de simulação virtual no ensino de Física (OLIVEIRA, 2022), especialmente no escopo deste trabalho, enfocando experimentos que abordam conceitos de MQ em cursos de graduação. Um exemplo relevante é o trabalho de Fedortchenko (2016), que descreve um experimento de teleportação quântica utilizando o processador quântico de 5 qubits da IBM. No experimento, os autores utilizaram a interface gráfica do IBM Quantum Experience para implementar o protocolo de teleportação quântica nos qubits do processador

³ O projeto QuTip foi concebido em 2010 por Paul Nation, um estudante de doutorado, que mais tarde colaborou com Robert Johansson na criação do pacote. Publicado em 2012 (JOHANSSON; NATION; NORI, 2012) sob uma licença de código aberto, o QuTip é continuamente expandido por uma comunidade de colaboradores.

supercondutor. Outro trabalho relevante é o de Rabelo e Costa (2018), que exploraram o IBM Q Experience (IBM-Q) como recurso pedagógico na aplicação de conceitos de computação quântica. Eles realizaram uma análise detalhada do IBM-Q, examinando diversos aspectos práticos e metodológicos. Da mesma forma, Alves *et al.* (2020), discutiram a implementação remota de sistemas quânticos em um processador quântico supercondutor, utilizando a plataforma IBM Quantum Experience.

2.1 QuTip: caixa de ferramentas quântica em Python

Johansson, Nation e Nori (2012), apresentam uma estrutura de código aberto orientada a objetos para resolver a dinâmica de sistemas quânticos abertos escritos em Python, o QuTip: The Quantum Toolbox in Python. É uma biblioteca cuja linguagem de programação é Python e que é compatível nos sistemas operacionais Linux, Mac OS e Windows. Para rodar o QuTip algumas bibliotecas adicionais são necessárias: Cython, Jupyter Notebook, Matplotlib, Nose, Numpy, SciPy e Spyder.

O QuTip, assim como as outras bibliotecas citadas, é acessível e gratuito, podendo ser encontrado facilmente na internet em forma de download. O software livre Jupyter Notebook é uma boa ferramenta enquanto instrumento de aprendizagem de computação como aborda Cardoso, Leitão e Teixeira (2018) em seu trabalho. O ambiente Jupyter permite programar em Python e utilizar o QuTip.

Como gerenciador de pacotes, recomenda-se o uso da plataforma Anaconda, que já contém todas as bibliotecas necessárias para modelar sistemas físicos, além de 150 pacotes previamente instalados e centenas de pacotes de código aberto que podem ser adicionados, dentre eles incluso o QuTip. O guia de instalação detalhado se encontra no apêndice. Os pacotes são compartimentos onde se congrega classes, interfaces, entre outros.

2.2 Fundamentos da Mecânica Quântica

A compreensão dos sistemas quânticos requer uma distinção entre sistemas abertos, isolados e fechados, que também são conceitos presentes na mecânica clássica. Inicialmente, é importante destacar a diferença entre essas categorias no contexto clássico antes de abordarmos sua aplicação na mecânica quântica.

Na mecânica clássica, um sistema isolado é aquele que não sofre interações com o ambiente externo (CALLEN, 1985). Ele é completamente independente e não há trocas de energia, matéria ou informação com o ambiente. Por outro lado, um sistema aberto é aquele que interage com o ambiente e recebe influências externas, como forças externas ou fluxo de calor. Essas interações podem afetar a dinâmica e as propriedades do sistema, tornando necessário considerar as interações para uma descrição precisa.

Ao transitar para a MQ, esses conceitos são reinterpretados no contexto dos sistemas quânticos. Todo sistema quântico encontrado na natureza é, essencialmente, um sistema quântico aberto. Mesmo ao idealizar um modelo, é inevitável que influências externas e interações com o ambiente estejam presentes em um sistema real. Portanto, é fundamental desenvolver ferramentas teóricas e numéricas que levem em consideração essas interações para uma compreensão adequada da dinâmica dos sistemas quânticos.

A noção de sistemas quânticos abertos surge do acoplamento entre o sistema e seu ambiente, frequentemente referido como banho ou reservatório (BREUER; PETRUCCIONE, 2002). Quando o acoplamento entre sistema e ambiente é fraco, existe uma dinâmica específica e conveniente para tratar esse tipo de problema, onde é possível distinguir o sistema do

ambiente. Isso permite que a dinâmica do ambiente seja "rastreada" separadamente, resultando em uma matriz densidade reduzida que descreve apenas o sistema isolado. A equação mestre de Lindblad é uma das equações gerais que governam a evolução dessa matriz densidade reduzida. Essa equação descreve a evolução média de um conjunto de um grande número de realizações de sistemas idênticos, em que considera-se um sistema e a média é tomada sobre infinitas cópias desse sistema. Além disso, existem outros métodos utilizados para representar a dinâmica de sistemas quânticos, como o estudo das trajetórias quânticas, também conhecido como Monte Carlo (CARMICHAEL, 2002).

Os sistemas mencionados anteriormente geralmente não são acompanhados por uma descrição analítica, levando à necessidade do emprego de simulações numéricas de equações diferenciais. Ao estudarmos sistemas quânticos, pode parecer inicialmente que um computador quântico seja absolutamente necessário para essa tarefa. No entanto, enfatiza-se que é possível realizar simulações desses sistemas em computadores convencionais de uso doméstico, o que constitui a vantagem e a proposta central deste trabalho. Johansson, Nation e Nori (2012) destacam algumas vantagens de sua ferramenta como: a estrutura é baseada inteiramente em software de código aberto; desenvolvimento de código rápido e fácil de ler usando a linguagem de programação Python; suporte para Hamiltonianos arbitrários e dependentes do tempo; faz uso de vários núcleos de processamento encontrados em computadores modernos; infraestrutura baseada na comunidade, permitindo contribuições de usuários para a base de código.

Alguns fundamentos de mecânica quântica são necessários para que se possa criar códigos e resolver problemas quânticos. A unidade de informação na computação clássica é o *bit*, que é a um estado que assume um dentre dois valores, 0 ou 1. A teoria da computação quântica é fundamentada em uma unidade de informação conhecida como *qubit*, que se diferencia de um *bit* clássico em sua natureza (NIELSEN; CHUANG, 2010). Enquanto um *bit* clássico pode assumir apenas um valor, seja 0 ou 1, o *qubit* tem a capacidade de ser representado como uma superposição de dois estados quânticos, introduzindo uma nova notação $\{|0\rangle, |1\rangle\}$. Esses estados são os *kets* e podem ser representados de forma matricial:

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (1)$$

Essa base é comumente chamada de base computacional. Para cada *ket* existe um bra $\langle\psi|$, tal que $\langle\psi| = (|\psi\rangle)^+$, que denota o conjugado transposto ou adjunto. Para a base computacional, $\langle 0| := \begin{pmatrix} 1 & 0 \end{pmatrix}$, $\langle 1| := \begin{pmatrix} 0 & 1 \end{pmatrix}$.

A representação mais geral para um *qubit* é um vetor de estado $|\psi\rangle$ definido no espaço de Hilbert, que é um espaço vetorial de dimensão n no corpo dos números complexos. A forma geral de um *qubit* é dada por:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2)$$

onde α e β são amplitudes de probabilidade complexas. Em MQ, uma amplitude de probabilidade é um número complexo cujo módulo ao quadrado representa uma probabilidade ou densidade de probabilidade. Logo, as constantes obedecem a condição:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (3)$$

na qual, $|\alpha|^2$ é a probabilidade de obter o estado $|0\rangle$ e $|\beta|^2$ a probabilidade de obter o estado $|1\rangle$ dado uma medida no estado $|\psi\rangle$.

Um postulado fundamental da MQ se trata da evolução temporal de um sistema quântico. A evolução de um sistema fechado é regida pela equação de Schroedinger (SAKURAI, 1994):

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = H(t)|\psi\rangle \tag{4}$$

onde $H(t)$ é um operador denominado hamiltoniano, responsável por representar a energia total do sistema. Além disso, é possível estabelecer um novo operador que descreve o estado quântico de um sistema físico, chamado de operador densidade ou matriz densidade ρ . Esse operador é definido em função da probabilidade P da seguinte forma:

$$\rho = \sum_{\psi} P_{\psi} |\psi\rangle\langle\psi| \tag{5}$$

No caso em que todos os valores de P_{ψ} são nulos exceto para $|\psi\rangle$, a equação (5) se torna

$$\rho = |\psi\rangle\langle\psi| \tag{6}$$

e o estado é considerado um estado puro. Assim como para o vetor de estado, também é possível obter uma equação de movimento para a matriz densidade, reescrevendo a equação (4) de forma que se isole a derivada do *ket* como se segue:

$$|\dot{\psi}\rangle = -\frac{i}{\hbar} H(t)|\psi\rangle \tag{7}$$

Tomando adequadamente a derivada da equação (5)

$$\dot{\rho} = \sum_{\psi} P_{\psi} (|\dot{\psi}\rangle\langle\psi| + |\psi\rangle\langle\dot{\psi}|) \tag{8}$$

e substituindo a equação (7) na equação (8), obtém-se finalmente:

$$\dot{\rho} = -\frac{i}{\hbar} [H(t), \rho] \tag{9}$$

Essa equação é a equação de Von Neumann de movimento para uma matriz densidade (SCULLY; ZUBAIRY, 1997).

3 *QuTip*: Potencial didático

Com os conceitos fundamentais da mecânica quântica estabelecidos, é possível criar algoritmos e resolver problemas utilizando o *QuTip*. O primeiro passo é importar os pacotes necessários, conforme exemplificado no Quadro 1:



Quadro 1 - Comandos para importar pacotes.

```
Caixa 1: Importando os Pacotes4  
  
%matplotlib inline  
import matplotlib.pyplot as plt  
import numpy as np  
from IPython.display import Image  
from qutip import *
```

Fonte: Os autores (2023).

O comando `%matplotlib inline` configura o ambiente interno do `matplotlib`, permitindo que as saídas dos gráficos sejam exibidas diretamente na interface do *Jupyter Notebook*, logo abaixo da célula onde o comando é executado. Por sua vez, o `numpy` é uma biblioteca matemática fundamental que abriga diversas funções amplamente conhecidas.

Para usar o *QuTip*, é necessário incluir o seu pacote, que corresponde ao comando `from qutip import *`. Este comando torna as funções e classes disponíveis em todo o programa. Antes de trabalhar com as funções é importante executar um teste que garante que todas as funções e classes contidas no *QuTip* estejam acessíveis. O comando a seguir, apresentado no Quadro 2, é responsável por realizar esse teste, o qual pode levar alguns minutos para ser concluído.

Quadro 2 - Comando para teste de verificação da importação dos pacotes.

```
import qutip.testing  
qutip.testing.run()
```

Fonte: Os autores (2023).

Para criar e modelar problemas, primeiro é preciso definir a classe de objetos quânticos, ou “Qobj” no jargão do programa. Essa classe contém todas as informações necessárias para a modelagem de um sistema quântico. O Quadro 3 mostra como um objeto quântico pode ser definido:

Quadro 3 - Comando para definição de um objeto quântico (Qobj).

```
Caixa 2: Definindo um objeto quântico  
  
q = Qobj([[1], [0]])  
q
```

```
Saída da Caixa 2:  
  
Quantum object: dims = [[2], [1]], shape = [2, 1], type = ket  
      (1.0)  
      (0.0)
```

Fonte: Os autores (2023).

Esse objeto é caracterizado por dois atributos essenciais: sua dimensão no espaço de Hilbert e seu tamanho. Normalmente, quando desejamos definir estados e operadores (para definir um vetor de estado ou *ket*), é necessário especificar, no comando correspondente, o

⁴ Adotamos um sistema de caixas numeradas para organizar e referenciar seções específicas do texto. Cada caixa é identificada pelo número correspondente. Utilizamos a notação 'Saída da Caixa X' para indicar a saída resultante da entrada correspondente à Caixa X.



número de estados no espaço de Hilbert e qual estado será ocupado, como exemplificado no Quadro 4 abaixo:

Quadro 4 - Comando para definição do espaço de Hilbert.

```
Caixa 3: Definindo as propriedades do ket
N = 2 #número de estados no espaço de Hilbert
n = 1 #estado que será ocupado
basis(N, n)
```

```
Saída da Caixa 3:
Quantum object: dims = [[2], [1]], shape = [2, 1], type = ket
(0.0)
(1.0)
```

Fonte: Os autores (2023).

Ao chamar um vetor de uma variável, supondo “ket1” e adicionar *.dag()*, têm-se o hermitiano conjugado, o *bra*, mostrado no Quadro 5:

Quadro 5 – Hermitiano conjugado.

```
Caixa 4: Chamando o ket transposto conjugado.
ket1 = basis(2, 0)
ket1.dag()
```

```
Saída da Caixa 4:
Quantum object: dims = [[1], [2]], shape = [1, 2], type = bra
(1.0)
(0.0)
```

Fonte: Os autores (2023).

Uma representação útil de um qubit pode ser obtida em função de α e β da equação (2), que pode ser reescrita de modo a considerar a fase global:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \quad (10)$$

onde o par (θ, ϕ) define então um ponto em uma esfera de raio 1 que é conhecida como a Esfera de Bloch. Esta esfera proporciona uma representação geométrica para o espaço de Hilbert de um estado quântico. Para plotar o estado na esfera, primeiro se define a base computacional e o *ket* $|\psi\rangle$, visto no Quadro 6:

Quadro 6 - Comando para implementar o *ket* para uma esfera de Bloch.

```
Caixa 5: Implementação do ket.
g = basis(2,0)
e = basis(2,1)
psi = 1/sqrt(2)*(e + g)
```

Fonte: Os autores (2023).



O comando abaixo, cria uma Esfera de Bloch, com um vetor $|0\rangle$ de estado definido como no Quadro 7:

Quadro 7 - Implementação e adição de um vetor a esfera de Bloch.

```
Caixa 6: Implementando uma esfera de Bloch  
b = Bloch()  
b.show()
```

```
Caixa 7: Adicionando um vetor  $|0\rangle$  a esfera de Bloch  
b.add_states(g)  
b.show()
```

Fonte: Os autores (2023).

Também é possível plotar vários pontos, vetores e estados ao mesmo tempo passando listas ou matrizes em vez de elementos individuais. Antes de adicionar mais um vetor a esfera o comando `clear()` remove os dados atuais da esfera Bloch, o segundo exemplo é com $|1\rangle$:

Quadro 8 - Adição de um vetor $|1\rangle$ a esfera de Bloch.

```
Caixa 8: Adicionando um vetor  $|1\rangle$  a esfera de Bloch  
b.clear()  
b.add_states(e)  
b.show()
```

Fonte: Os autores (2023).

E por último, uma soma de vetores, multiplicado por uma constante de normalização pode ser visto no Quadro 9:

Quadro 9 - Adição de vetores multiplicado pela constante de normalização.

```
Caixa 9: Adicionando um vetor a esfera de Bloch  
b.clear()  
b.add_states(psi)  
b.show()
```

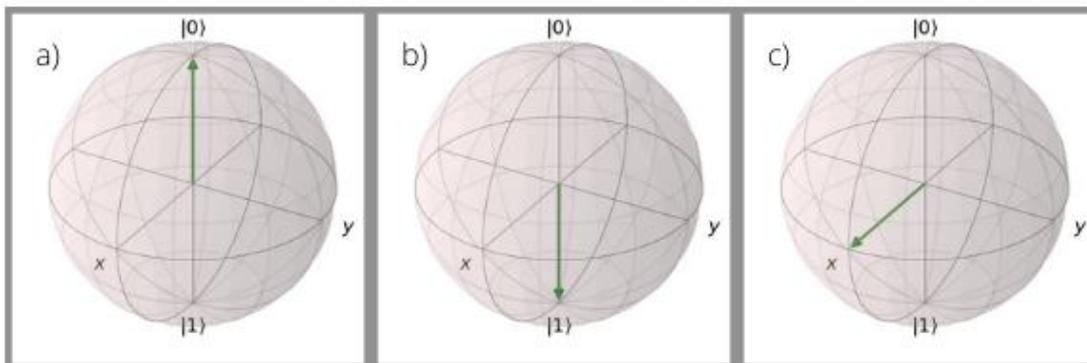
Fonte: Os autores (2023).

A Figura 1 mostra a representação da esfera de Bloch de um qubit em três situações distintas. As imagens (a), (b) e (c) são saídas das caixas 7, 8 e 9 respectivamente.



Figura 1 - Representação da esfera de Bloch de um qubit em três situações

a) $\theta = 0^\circ$ na equação (10), $|\psi\rangle = |0\rangle$; b) $\theta = 180^\circ$, $|\psi\rangle = |1\rangle$ c) $\theta = 90^\circ$, $\phi = 0^\circ$, $|\psi\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle)$



Fonte: Autores (2022).

É possível solucionar a equação de Schroedinger, conforme apresentada na equação (4), ou seja, calcular a evolução de um estado inicial ao longo do tempo com apenas alguns comandos. Para realizar isso, é necessário definir o operador Hamiltoniano H , um *ket* inicial $|\psi\rangle$ e uma lista de tempos. No exemplo mostrado no Quadro 10, considera-se $H = \sigma_z$ e o estado inicial como $|0\rangle$. A função utilizada para evoluir o vetor de estado é denominada *mesolve*.

Quadro 10 - Exemplo de aplicação na solução da equação de Schroedinger.

```
Caixa 10: Implementando a evolução de um estado quântico para um determinado psi

#Definindo os parâmetros
H = sigmaz()

psi0 = basis(2, 0)

tlist = np.linspace(0, 10, 100)

#função que evolui o estado
result = mesolve(H, psi0, tlist, [], [])

#estado final
result.states[-1]
```

```
Saída da Caixa 10:

Quantum object: dims = [[2], [1]], shape = (2, 1), type = ket
( 0.0 (-0.839 + 0.544j)
  0.0 0.0)
```

Fonte: Os autores (2023).

Quando se trata não mais de um *ket*, e sim de uma matriz densidade ρ , o comando permanece, os parâmetros são os mesmos, exceto a matriz densidade substituindo o $psi0$ definido acima. Supondo, $\rho = |0\rangle\langle 1|$ os comandos seguem abaixo:



Quadro 11 - Implementação da evolução de um estado quântico para ρ .

Caixa 11: Implementando a evolução de um estado quântico para um determinado rho
<pre>#Definindo os parâmetros H = sigmaz() psi0 = basis(2, 0) psi1 = basis(2, 1).dag() rho=psi0*psi1 tlist = np.linspace(0, 10, 100) #função que evolui o estado result = mesolve(H, rho, tlist, [], []) #estado final result.states[-1]</pre>
Saída da Caixa 11:
<pre>Quantum object: dims = [[2], [2]], shape = (2, 2), type = oper, isherm = False (0.0 (0.408 - 0.913j) 0.0 0.0)</pre>

Fonte: Os autores (2023).

No Quadro 11, foi implementada a evolução de um estado quântico para um determinado rho. A transição de um estado quântico representado por um *ket* para uma matriz densidade no QuTiP reflete a necessidade de incorporar considerações mais realistas e abrangentes em modelos quânticos. Enquanto um *ket* descreve um estado puro específico, a matriz densidade permite a representação de estados mistos, incorporando incertezas, interações com o ambiente e complexidades que podem surgir em sistemas quânticos do mundo real.

4 Considerações finais

O trabalho de Jesus *et al.* (2021) trouxe importantes percepções acerca do uso de plataforma de algoritmos quânticos para a aplicação de conceitos. Neste trabalho foi apresentado o pacote *QuTip*, uma biblioteca de linguagem Python que pode ser acessada facilmente pelo Jupyter Notebook, onde é possível descrever sistemas quânticos e realizar operações com operadores e vetores de estado.

O trabalho foi estruturado de forma a demonstrar a instalação e os pacotes adicionais necessários para a utilização do *QuTip*. Logo após, contém uma breve introdução de conceitos fundamentais de MQ, como vetores de estado (*kets* e *bra*), a equação de Schroedinger, o operador densidade e a equação de Von Neumann.

Foram destacadas as principais condições para a construção de um programa, como a implementação de bibliotecas, definição de parâmetros e variáveis, e a solução de problemas. Todos os problemas e exemplos que foram implementados e resolvidos coincidem com as previsões teóricas, o que destaca a eficácia do *QuTip* como uma ferramenta de ensino aprendizagem no contexto do ensino de FQ. O *QuTip* apresenta limitações (KRÄMER *et al.*, 2018), sendo notável a necessidade de recorrer a linguagens de programação de baixo nível,

GRECA, I. M.; MOREIRA, M. A.; HERSCOVITZ, V. E. Uma proposta para o ensino de mecânica quântica. **Revista Brasileira de Ensino de Física**, v. 23, n. 4, p. 444–457, dez. 2001. DOI: <https://doi.org/10.1590/S0102-47442001000400010>. Disponível em: <https://doi.org/10.1590/S0102-47442001000400010>. Acesso em: 12 out. 2023.

JOHNSTON, I.; CRAWFORD, K.; FLETCHER, P. Student difficulties in learning quantum mechanics. **International Journal of Science Education**, v. 20, n. 4, p. 427-446, 1998. DOI: <https://doi.org/10.1080/0950069980200404>. Acesso em: 15 out. 2023.

JOSÉ, M. A.; PIQUEIRA, J. R. C.; LOPES, R. D. Introdução à programação quântica. **Revista Brasileira de Ensino de Física**, v. 35, n. 1, p. 1306, 2013. DOI: 10.1590/S1806-11172013000100006. Disponível em: <https://doi.org/10.1590/S1806-11172013000100006>. Acesso em: 15 out. 2023.

JESUS, G. F. D. *et al.* Computação quântica: uma abordagem para a graduação usando o Qiskit. **Revista Brasileira de Ensino de Física**, v. 43, p. e20210033, 2021. DOI: 10.1590/1806-9126-RBEF-2021-0033. Disponível em: <https://doi.org/10.1590/1806-9126-RBEF-2021-0033>. Acesso em: 15 jul. 2023.

JOHANSSON, J. R.; NATION, P. D.; NORI, Franco. QuTiP: An open-source Python framework for the dynamics of open quantum systems. **Computer Physics Communications**, v. 183, n. 8, p. 1760-1772, 2012. DOI: 10.1016/j.cpc.2012.02.021. Disponível em: <https://doi.org/10.1016/j.cpc.2012.02.021>. Acesso em: 10 jun. 2023.

JOHANSSON, J. R.; NATION, P. D.; NORI, Franco. QuTiP 2: An open-source Python framework for the dynamics of open quantum systems. **Computer Physics Communications**, v. 184, n. 4, p. 1234-1240, 2013. DOI: 10.1016/j.cpc.2012.11.019. Disponível em: <https://doi.org/10.1016/j.cpc.2012.11.019>. Acesso em: 10 jun. 2023.

KRÄMER, S.; PLANKENSTEINER, D.; OSTERMANN, L.; RITSCH, H. QuantumOptics.jl: A Julia framework for simulating open quantum systems. **Computer Physics Communications**, v. 227, p. 109-116, 2018. DOI: <https://doi.org/10.1016/j.cpc.2018.02.004>. Acesso em: 15 out. 2023.

MONTEIRO, M. A.; NARDI, R.; BASTOS FILHO, J. B. A sistemática incompreensão da teoria quântica e as dificuldades dos professores na introdução da Física Moderna e Contemporânea no Ensino Médio. **Ciência & Educação**, Bauru, v. 15, n. 3, p. 557–580, 2009. Disponível em: <https://doi.org/10.1590/S1516-73132009000300007>. Acesso em: 12 out. 2023.

MOTA, L. M. **As controvérsias sobre a interpretação da mecânica quântica e a formação dos licenciados em física**. Dissertação (Mestrado em Educação) - Universidade Federal de Santa Catarina Florianópolis, 2000. Disponível em: <http://repositorio.ufsc.br/xmlui/handle/123456789/79096>. Acesso em: 20 out. 2023.

NIELSEN, M. A.; CHUANG, I. L. **Quantum computation and quantum information**: 10th anniversary edition. Cambridge: Cambridge University Press Inc., 2010.

SCULLY, M. O.; ZUBAIRY, M. S. **Quantum optics**. New York: Cambridge University Press Inc., 1997.

SOUZA, A. C. L.; GONÇALVES, C. B. O uso de Tecnologias na Educação e no Ensino de Ciências a partir de uma pesquisa bibliográfica. **Revista da Rede Amazônica de Educação em Ciências e Matemática**, v. 7, n. 3, p. 256-276, 2020. DOI: 10.26571/reamec.v7i3.9256. Disponível em: <https://periodicoscientificos.ufmt.br/ojs/index.php/reamec/article/view/9256>. Acesso em: 14 out. 2023.

TERRAZZAN, E. A. A inserção da física moderna e contemporânea no ensino de física na escola de 2º grau. **Caderno Catarinense de Ensino de Física**, Florianópolis, v. 9, n. 3, p. 209-214, dez. 1992. Disponível em: <https://periodicos.ufsc.br/index.php/fisica/article/view/7392>. Acesso em: 12 out. 2023.

TERRAZZAN, E. A. **Perspectivas para a inserção da física moderna na escola média**. 1994. Tese (Doutorado) – Universidade de São Paulo, São Paulo, 1994. Disponível em: <https://repositorio.usp.br/item/000742484>. Acesso em: 08 out. 2023.

Recebido em setembro de 2023.

Aprovado em outubro de 2023.



APÊNDICE A - GUIA DE INSTALAÇÃO

INSTALAÇÃO ANACONDA

- WINDOWS

1. Faça o download da plataforma Anaconda. Link: <https://www.anaconda.com/>
2. Após a instalação, clique em "I Agree" para aceitar os termos de licença, clique em "Next" e "Next" novamente. Em seguida, proceda com a instalação.
3. Após a conclusão da instalação do Anaconda, abra o menu pesquisar e digite “Anaconda Prompt” e pressione “Enter”. Abra o terminal do Anaconda e digite:

```
condaconfig--add channels defaults  
condaconfig--add channels conda-forge  
condaupdate--all
```

- Mac OS X

1. Faça o download da plataforma Anaconda.
2. Após a instalação abra o menu pesquisar e digite “Anaconda Prompt” e pressione “Enter”. Abra o terminal do Anaconda e digite:

```
condaconfig--add channels defaults  
condaconfig--add channels conda-forge  
condaupdate--all
```

- Linux

1. Faça o download da plataforma Anaconda. Após a conclusão do download, vá para a pasta onde o arquivo foi salvo pelo terminal (cd ~/nomeDaPasta).
2. Em seguida, execute: bash Anaconda3-2019.07-Linux-x86_64.sh
3. Após o aceite do termo de licença, pressione “Enter” para confirmar o local de instalação (ou insira um novo caminho) e, por fim, confirme a adição do Anaconda no PATH.
4. Após a conclusão da instalação do Anaconda, abra o menu pesquisar e digite “Anaconda Prompt” e pressione “Enter”. Abra o terminal do Anaconda e digite:

```
condaconfig--add channels defaults  
condaconfig--add channels conda-forge  
condaupdate--all
```

INSTALAÇÃO QUTIP



Abra o menu pesquisar e digite “Anaconda Prompt” e pressione “Enter”. Abra o terminal do Anaconda e digite:

```
condaconfig--add channels defaults  
condaconfig--add channels conda-forge  
conda install qutip  
condaupdate--all
```

Instruções de instalação:

<https://qutip.org/docs/latest/installation.html>

Documentação:

<https://qutip.org/documentation.html>

Comando para testar todos os recursos do qutip:

```
import qutip.testing  
qutip.testing.run()
```