

Comece por Aqui

Nathan Lima

Computação Quântica

Aula 00

Sumário

1	Introdução	2
2	Alguns conhecimentos prévios importantes	2
2.1	Números Complexos	2
2.2	Série e Transformada de Fourier	5
2.3	Solução do poço infinito	7
3	Ferramentas Digitais	10
3.1	Vs Code	10
3.2	Google Collab	12
3.3	GitHub	13
3.4	VMZI	15
4	Próximos passos	15

1 Introdução

O nosso curso de Introdução à Computação Quântica e Teoria Quântica de Informação foi pensado para alunos em nível de pós-graduação. Isto é, espera-se que os alunos tenham passado por disciplinas de Mecânica Quântica, estudado livros como os escritos por Griffiths [3] ou mesmo Eisberg e Resnick [2], que são muito comuns no Brasil.

Por esse motivo, não vamos revisitar tópicos da Antiga Teoria Quântica, ou resolver vários casos da Equação de Schrödinger. Vamos adotar uma proposta fenomenológica, isto é, partindo de experimentos cruciais, para motivar o formalismo da mecânica quântica, a dizer, a **representação de estados físicos por meio de vetores no Espaço de Hilbert. Tanto a medida de valores observáveis bem como a transformação dos estados físicos por meio de operadores lineares.**

Nosso curso tem dois focos complementares: o primeiro é o domínio do formalismo matemático necessário para descrever os processos mobilizados por algoritmos quânticos (como sobreposição, emaranhamento e processamento da informação em sistemas quânticos de uma forma geral) e o segundo é a proficiência na escrita e interpretação de algoritmos quânticos para solução de problemas concretos.

Assim, essa é uma disciplina tanto teórica quanto prática. Esses dois objetivos não serão tratados de forma independentes ao longo de todo curso. Aliás, tentaremos dentro do possível sempre caminhar com (i) fenômeno físico, (ii) descrição matemática, (iii) operacionalização na computação quântica lado a lado.

Neste documento você encontrará alguns tópicos que são importantes de serem revisitados antes de começar o curso. Além disso, você encontrará indicações de ferramentas digitais que serão importantes ao longo do curso. Essa revisão não é exaustiva, mas deve ser encarada como um "mapa" de revisão, dando indícios de caminhos a serem seguidos.

2 Alguns conhecimentos prévios importantes

2.1 Números Complexos

Da forma usual como concebemos a teoria, não é possível ter mecânica quântica sem números complexos [4], embora haja uma intensa literatura recente problematizando o assunto. Dessa forma, o primeiro ponto importante de recordação é justamente a ideia de números complexos.

A ideia original dos números imaginários pode ser associada com o conceito de rotação. Imagine que você tem uma seta apontando do 0 para o número 1 ao longo da reta dos reais. Se você aplicar por um certo valor essa seta gira 180° e aponta para o -1. Se pensarmos nessa operação como sendo composta por duas partes, ao fazer apenas uma, temos uma rotação de 90° , no obrigando a abrir um novo eixo, perpendicular ao primeiro. Assim, se chamarmos de "i" o número que está multiplicando o número 1, temos que $1 \cdot i = i$ e $1 \cdot i^2 = -1$. Esse novo eixo que "abrimos", representa a reta dos números imaginários.

Assim, os números complexos constituem uma extensão dos números reais e desempenham um papel fundamental na Física, na Engenharia e, particularmente, na Mecânica Quântica [1]. Enquanto os números reais podem ser representados por pontos sobre uma reta, os números complexos são representados em um plano bidimensional, denominado **plano complexo**. Um número complexo é escrito na forma

$$z = a + ib, \tag{2.1}$$

em que a e b são números reais e

$$i = \sqrt{-1}, \tag{2.2}$$

é a unidade imaginária, que satisfaz

$$i^2 = -1. \quad (2.3)$$

O número a é denominado **parte real**, enquanto b é denominado **parte imaginária**,

Um número complexo pode ser representado como um vetor no plano complexo, conforme ilustrado na Figura 1. O eixo horizontal representa a parte real e o eixo vertical representa a parte imaginária.

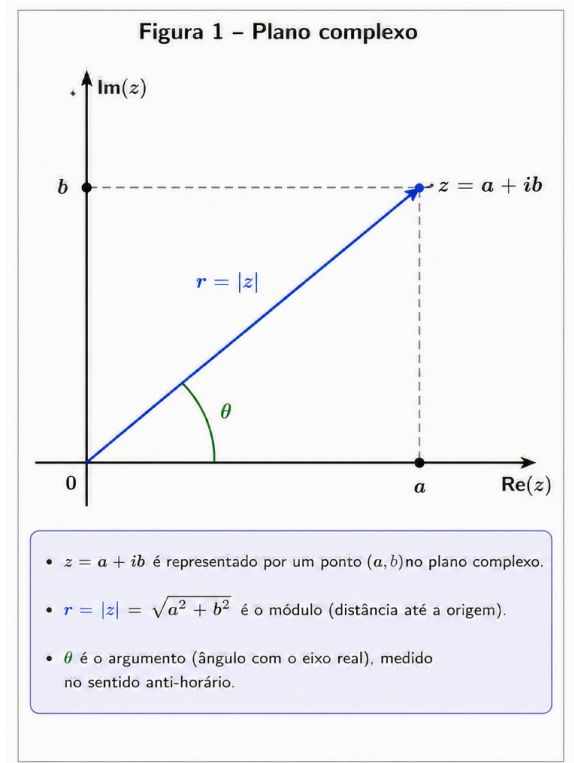


Figura 1: Representação geométrica de um número complexo $z = a + ib$ no plano complexo.

O **complexo conjugado** de um número complexo é obtido invertendo o sinal da parte imaginária. Assim, o conjugado do número complexo z apresentado é

$$z^* = a - ib. \quad (2.4)$$

O produto entre um número complexo e seu conjugado elimina a parte imaginária,

$$zz^* = (a + ib)(a - ib) \quad (2.5)$$

$$= a^2 + b^2. \quad (2.6)$$

Essa propriedade será utilizada recorrentemente ao calcular probabilidades em Mecânica Quântica. A distância entre o número complexo e a origem do plano complexo é denominada **módulo** ou **norma**, sendo definida por

$$|z| = \sqrt{zz^*} = \sqrt{a^2 + b^2}. \quad (2.7)$$

Em vez de utilizar as coordenadas cartesianas (a, b) , também podemos descrever um número complexo utilizando coordenadas polares. Sejam

- $r = |z|$ o módulo;
- θ o ângulo formado com o eixo real.

Da trigonometria,

$$a = r \cos \theta, \quad (2.8)$$

$$b = r \sin \theta. \quad (2.9)$$

Assim,

$$z = r(\cos \theta + i \sin \theta). \quad (2.10)$$

Essa forma evidencia que um número complexo pode ser interpretado como um vetor de comprimento r formando um ângulo θ com o eixo real.

Podemos lembrar, também, que as séries de Taylor do seno e do cosseno são

$$\cos \theta = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \dots \quad (2.11)$$

e

$$\sin \theta = \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots \quad (2.12)$$

Ademais, a função exponencial admite a expansão em série de Taylor

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}. \quad (2.13)$$

Substituindo

$$x = i\theta, \quad (2.14)$$

obtemos

$$e^{i\theta} = 1 + i\theta + \frac{(i\theta)^2}{2!} + \frac{(i\theta)^3}{3!} + \frac{(i\theta)^4}{4!} + \dots \quad (2.15)$$

Como

$$i^2 = -1, \quad (2.16)$$

$$i^3 = -i, \quad (2.17)$$

$$i^4 = 1, \quad (2.18)$$

segue que

$$e^{i\theta} = 1 + i\theta - \frac{\theta^2}{2!} - i\frac{\theta^3}{3!} + \frac{\theta^4}{4!} + i\frac{\theta^5}{5!} - \dots \quad (2.19)$$

Agrupando os termos reais e imaginários,

$$e^{i\theta} = \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \dots\right) \quad (2.20)$$

$$+ i \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots\right). \quad (2.21)$$

$$e^{i\theta} = \cos \theta + i \sin \theta. \quad (2.22)$$

Substituindo essa identidade na representação polar, obtemos

$$\boxed{z = re^{i\theta}}. \quad (2.23)$$

2.2 Série e Transformada de Fourier

A análise de Fourier constitui uma das ferramentas matemáticas mais importantes da Física. Ela permite decompor funções arbitrárias em combinações de ondas senoidais ou, de forma mais elegante, em exponenciais complexas.

Considere uma função periódica $f(x)$ de período L , isto é,

$$f(x + L) = f(x). \quad (2.24)$$

Segundo o teorema de Fourier, sob condições bastante gerais, qualquer função periódica pode ser escrita como uma soma de exponenciais complexas,

$$f(x) = \sum_{n=-\infty}^{+\infty} c_n e^{ik_n x}, \quad (2.25)$$

em que

$$k_n = \frac{2\pi n}{L}, \quad (2.26)$$

é o número de onda permitido pela periodicidade.

Cada termo da soma representa uma onda plana de frequência espacial diferente. Os coeficientes c_n determinam a contribuição de cada uma dessas ondas para reconstruir a função original.

Para encontrar esses coeficientes utiliza-se a propriedade de ortogonalidade das exponenciais complexas,

$$\int_0^L e^{i(k_n - k_m)x} dx = L \delta_{nm}, \quad (2.27)$$

onde δ_{nm} é o delta de Kronecker,

$$\delta_{nm} = \begin{cases} 1, & n = m, \\ 0, & n \neq m. \end{cases}$$

Multiplicando ambos os lados da expansão por $e^{-ik_m x}$ e integrando no intervalo de um período, obtemos

$$\int_0^L f(x) e^{-ik_m x} dx = \sum_n c_n \int_0^L e^{i(k_n - k_m)x} dx \quad (2.28)$$

$$= L c_m. \quad (2.29)$$

Logo,

$$c_n = \frac{1}{L} \int_0^L f(x) e^{-ik_n x} dx. \quad (2.30)$$

Assim, a série de Fourier pode ser escrita como

$$f(x) = \sum_{n=-\infty}^{+\infty} \left[\frac{1}{L} \int_0^L f(x') e^{-ik_n x'} dx' \right] e^{ik_n x}. \quad (2.31)$$

A série de Fourier descreve funções periódicas. Entretanto, muitas funções físicas não possuem período, como uma função de onda localizada ou um pulso luminoso.

Podemos imaginar aumentar o período da função cada vez mais,

$$L \rightarrow \infty.$$

Nesse limite, os valores discretos de número de onda

$$k_n = \frac{2\pi n}{L}$$

ficam infinitamente próximos entre si.

Em vez de uma soma discreta,

$$\sum_n,$$

passamos a trabalhar com uma integral contínua,

$$\sum_n \longrightarrow \frac{L}{2\pi} \int dk.$$

Da mesma forma,

$$c_n \longrightarrow \frac{1}{L} \tilde{f}(k).$$

Substituindo essas relações na série de Fourier, obtemos naturalmente a Transformada de Fourier.

A Transformada de Fourier de uma função $f(x)$ é definida por

$$\tilde{f}(k) = \int_{-\infty}^{+\infty} f(x) e^{-ikx} dx. \quad (2.32)$$

A função $\tilde{f}(k)$ informa quanto da frequência espacial (ou número de onda) k está presente em $f(x)$.

Em Física, costuma-se interpretar $\tilde{f}(k)$ como o espectro da função.

A função original pode ser completamente reconstruída a partir de sua transformada.

A transformada inversa é dada por

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \tilde{f}(k) e^{ikx} dk. \quad (2.33)$$

Portanto, a Transformada de Fourier estabelece uma correspondência entre a função no espaço real e sua representação no espaço dos números de onda.

2.3 Solução do poço infinito

Um dos primeiros problemas exatamente solúveis da Mecânica Quântica é o da partícula confinada em um poço de potencial infinito. Apesar de sua simplicidade, esse modelo ilustra conceitos fundamentais como quantização da energia, autofunções, superposição de estados e evolução temporal.

Considere uma partícula confinada no intervalo

$$0 < x < L,$$

sob a ação do potencial

$$V(x) = \begin{cases} 0, & 0 < x < L, \\ \infty, & x \leq 0 \text{ ou } x \geq L. \end{cases} \quad (2.34)$$

Como o potencial é infinito nas paredes da caixa, a probabilidade de encontrar a partícula fora do intervalo deve ser nula. Consequentemente,

$$\psi(0, t) = 0, \quad \psi(L, t) = 0. \quad (2.35)$$

Essas são as condições de contorno do problema.

A equação de Schrödinger dependente do tempo é

$$i\hbar \frac{\partial \Psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi(x, t)}{\partial x^2} + V(x)\Psi(x, t). \quad (2.36)$$

No interior do poço, onde

$$V(x) = 0,$$

temos

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \Psi}{\partial x^2}. \quad (2.37)$$

Buscando soluções separáveis,

$$\Psi(x, t) = \psi(x)T(t), \quad (2.38)$$

obtemos

$$i\hbar \psi(x) \frac{dT}{dt} = -\frac{\hbar^2}{2m} T(t) \frac{d^2 \psi}{dx^2}. \quad (2.39)$$

Dividindo ambos os lados por ψT ,

$$i\hbar \frac{1}{T} \frac{dT}{dt} = -\frac{\hbar^2}{2m} \frac{1}{\psi} \frac{d^2 \psi}{dx^2} = E, \quad (2.40)$$

onde introduzimos a constante de separação E , que será identificada como a energia do sistema.

A parte espacial satisfaz

$$-\frac{\hbar^2}{2m} \frac{d^2 \psi}{dx^2} = E\psi. \quad (2.41)$$

ou

$$\frac{d^2 \psi}{dx^2} + k^2 \psi = 0, \quad (2.42)$$

onde

$$k^2 = \frac{2mE}{\hbar^2}. \quad (2.43)$$

A solução geral é

$$\psi(x) = A \sin(kx) + B \cos(kx). \quad (2.44)$$

Na parede esquerda,

$$\psi(0) = 0.$$

Como

$$\psi(0) = B,$$

segue que

$$B = 0. \quad (2.45)$$

Logo,

$$\psi(x) = A \sin(kx). \quad (2.46)$$

Na parede direita,

$$\psi(L) = 0,$$

portanto,

$$\sin(kL) = 0. \quad (2.47)$$

Isso somente ocorre quando

$$kL = n\pi, \quad n = 1, 2, 3, \dots \quad (2.48)$$

Assim,

$$k_n = \frac{n\pi}{L}. \quad (2.49)$$

Como

$$k^2 = \frac{2mE}{\hbar^2},$$

obtemos os níveis de energia

$$\boxed{E_n = \frac{n^2 \pi^2 \hbar^2}{2mL^2}}. \quad (2.50)$$

A condição

$$\int_0^L |\psi|^2 dx = 1$$

fornece

$$A = \sqrt{\frac{2}{L}}. \quad (2.51)$$

Portanto,

$$\psi_n(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right). \quad (2.52)$$

A equação temporal é

$$i\hbar \frac{dT}{dt} = ET, \quad (2.53)$$

cujas solução é

$$T(t) = e^{-iEt/\hbar}. \quad (2.54)$$

Logo, os autoestados estacionários são

$$\Psi_n(x, t) = \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right) e^{-iE_n t/\hbar}. \quad (2.55)$$

Como os autoestados formam uma base ortonormal, qualquer estado inicial pode ser escrito como

$$\Psi(x, t) = \sum_{n=1}^{\infty} c_n \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right) e^{-iE_n t/\hbar}. \quad (2.56)$$

Os coeficientes são determinados pela condição inicial,

$$c_n = \sqrt{\frac{2}{L}} \int_0^L \Psi(x, 0) \sin\left(\frac{n\pi x}{L}\right) dx. \quad (2.57)$$

Observe que esta expressão corresponde exatamente aos coeficientes de uma Série de Fourier em senos.

Considere a condição inicial

$$\Psi(x, 0) = A x(L - x), \quad (2.58)$$

uma parábola que se anula nas paredes do poço.

A constante de normalização é obtida impondo

$$\int_0^L |\Psi(x, 0)|^2 dx = 1.$$

Como

$$\int_0^L x^2(L - x)^2 dx = \frac{L^5}{30}, \quad (2.59)$$

segue que

$$A = \frac{\sqrt{30}}{L^{5/2}}. \quad (2.60)$$

Portanto,

$$\Psi(x, 0) = \frac{\sqrt{30}}{L^{5/2}} x(L - x). \quad (2.61)$$

Os coeficientes da expansão são

$$c_n = \sqrt{\frac{2}{L}} \frac{\sqrt{30}}{L^{5/2}} \int_0^L x(L - x) \sin\left(\frac{n\pi x}{L}\right) dx. \quad (2.62)$$

Efetuada a integração, obtém-se

$$c_n = \frac{8\sqrt{15}}{\pi^3 n^3}, \quad n \text{ ímpar}, \quad (2.63)$$

e

$$c_n = 0, \quad n \text{ par}. \quad (2.64)$$

Finalmente, a evolução temporal da função de onda é

$$\Psi(x, t) = \sum_{\substack{n=1 \\ n \text{ ímpar}}}^{\infty} \frac{8\sqrt{15}}{\pi^3 n^3} \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right) e^{-iE_n t/\hbar}. \quad (2.65)$$

Essa solução mostra que uma função de onda arbitrária pode ser construída como uma superposição dos autoestados do poço infinito. Cada autoestado evolui apenas por um fator de fase temporal, enquanto a dinâmica completa surge da interferência entre essas fases.

3 Ferramentas Digitais

3.1 Vs Code

O *Visual Studio Code* (VS Code) é um dos editores de código mais utilizados atualmente para o desenvolvimento em diversas linguagens de programação, incluindo Python. Nesta seção, veremos como instalar o VS Code e executar nosso primeiro programa.

Acesse o site oficial:

<https://code.visualstudio.com>

Clique em **Download** e escolha a versão correspondente ao seu sistema operacional (Windows, Linux ou macOS).

Após o download, execute o instalador e siga as instruções na tela.

Caso ainda não tenha o Python instalado, faça o download em:

<https://www.python.org/downloads/>

Durante a instalação no Windows, marque a opção

Add Python to PATH

antes de clicar em **Install Now**.

Abra o VS Code.

No menu lateral esquerdo, clique no ícone de **Extensions** (ou pressione **Ctrl+Shift+X**).

Pesquise por

Python

e instale a extensão oficial desenvolvida pela Microsoft.
Crie uma nova pasta para armazenar seus projetos.
Em seguida, no VS Code, selecione

File → Open Folder

e abra essa pasta.
Crie um novo arquivo chamado

hello.py

Digite o seguinte código:

```
print("Olá, mundo!")
```

Existem duas maneiras simples de executar o programa.
Clique no botão

Run Python File

(localizado no canto superior direito do VS Code).
A saída aparecerá no terminal integrado:

```
Olá, mundo!
```

Abra o terminal do VS Code selecionando

Terminal → New Terminal

ou utilizando o atalho

Ctrl + `

No terminal, execute

```
python hello.py
```

ou, dependendo da instalação,

```
python3 hello.py
```

Se tudo estiver corretamente configurado, será exibida a mensagem

```
Olá, mundo!
```

A partir deste ponto, o ambiente está pronto para desenvolver programas mais elaborados, incluindo algoritmos para Computação Quântica utilizando bibliotecas como o Qiskit.

3.2 Google Collab

O *Google Colaboratory* (Google Colab) é um ambiente gratuito de programação baseado em *Jupyter Notebook*, executado diretamente no navegador. Sua principal vantagem é não exigir a instalação do Python ou de bibliotecas em seu computador, além de oferecer acesso gratuito a recursos computacionais na nuvem.

Abra seu navegador de internet e acesse o endereço

`https://colab.research.google.com`

Para utilizar o Colab é necessário possuir uma conta Google.

Após fazer o login, será exibida a tela inicial do ambiente.

Na tela inicial, clique em

File → New Notebook

ou em

Novo Notebook

Será criado um novo notebook contendo uma célula de código pronta para ser utilizada.

Na primeira célula, digite

```
print("Olá, mundo!")
```

Em seguida, clique no botão de execução (ícone de **Play**) localizado à esquerda da célula ou pressione

Shift + Enter

A saída será exibida logo abaixo da célula:

Olá, mundo!

Um notebook é composto por diversas células independentes.

Novas células podem ser adicionadas clicando em

+ Code

para inserir uma célula de código ou

+ Text

para inserir uma célula de texto formatado utilizando a linguagem Markdown.

Essa organização permite combinar explicações, equações, figuras e códigos em um único documento.

Caso seja necessário utilizar uma biblioteca que não esteja instalada, basta executar o comando

```
!pip install nome_da_biblioteca
```

Por exemplo, para instalar o Qiskit:

```
!pip install qiskit
```

Após a instalação, a biblioteca poderá ser importada normalmente:

```
from qiskit import QuantumCircuit
```

Os notebooks são salvos automaticamente no Google Drive.

Também é possível renomear o arquivo clicando sobre o nome localizado no canto superior esquerdo da janela.

Caso deseje armazenar uma cópia local, selecione

File → Download

e escolha o formato desejado, como

- Notebook (.ipynb);
- PDF (.pdf);
- Arquivo Python (.py).

O Google Colab é uma excelente ferramenta para o aprendizado de programação e para o desenvolvimento de algoritmos científicos, pois elimina a necessidade de instalação de softwares e permite compartilhar notebooks facilmente entre estudantes e pesquisadores.

3.3 GitHub

O GitHub é uma plataforma utilizada para armazenar, compartilhar e controlar versões de projetos de software. Baseado no sistema de controle de versões Git, ele permite acompanhar todas as modificações realizadas em um projeto, além de facilitar o trabalho colaborativo entre diversos desenvolvedores.

Neste capítulo, veremos como criar uma conta, criar um repositório e enviar um projeto desenvolvido no Visual Studio Code para o GitHub.

Acesse o endereço

<https://github.com>

Clique em

Sign up

e siga as instruções para criar uma conta gratuita.

Após confirmar seu e-mail, faça login na plataforma.

Na página inicial do GitHub, clique no botão

New Repository

ou no símbolo “+” localizado no canto superior direito.

Informe:

- Nome do repositório;
- Descrição (opcional);
- Escolha entre repositório público (*Public*) ou privado (*Private*).

Em seguida, clique em

Create Repository

O repositório será criado e ficará disponível para receber os arquivos do projeto.

Abra o Visual Studio Code.

Selecione

File → Open Folder

e escolha a pasta do projeto que deseja enviar ao GitHub.

Abra o terminal integrado do VS Code através de

Terminal → New Terminal

ou utilizando o atalho

Ctrl + ‘

No terminal execute

```
git init
```

Esse comando transforma a pasta em um repositório Git.

Para adicionar todos os arquivos do projeto execute

```
git add .
```

Em seguida, registre a primeira versão do projeto com

```
git commit -m "Primeiro commit"
```

O termo *commit* representa um registro permanente do estado do projeto naquele momento.

Na página do repositório criado no GitHub será exibido um endereço semelhante a

<https://github.com/usuario/meu-projeto.git>

Conecte o projeto local ao repositório remoto utilizando

```
git remote add origin
```

```
https://github.com/usuario/meu-projeto.git
```

Esse comando precisa ser executado apenas uma única vez para cada projeto.

Para enviar os arquivos ao GitHub execute

```
git branch -M main
```

```
git push -u origin main
```

Na primeira utilização poderá ser solicitada a autenticação da conta GitHub.

Após a conclusão do envio, basta atualizar a página do repositório para visualizar todos os arquivos.

Sempre que novos arquivos forem adicionados ou modificados, execute novamente

```
git add .
```

```
git commit -m "Descrição das alterações"
```

```
git push
```

Esses comandos enviam a versão mais recente do projeto para o GitHub.

O GitHub tornou-se uma das principais plataformas para desenvolvimento de software, pesquisa científica e projetos de Computação Quântica. Além de funcionar como um sistema de backup dos projetos, ele facilita o compartilhamento de códigos, o trabalho colaborativo e o acompanhamento da evolução de um projeto ao longo do tempo.

3.4 VMZI

O Interferômetro Virtual de Mach-Zehnder (IVMZ) é uma simulação computacional desenvolvida pelo Laboratório de Ensino de Física (LIEF) da Universidade Federal do Rio Grande do Sul (UFRGS). A ferramenta permite estudar, de forma interativa, fenômenos como interferência, dualidade onda-partícula e os princípios básicos da Mecânica Quântica.

Abra o navegador de internet e acesse

`https://lief.if.ufrgs.br/~cjhc/vmzi.html`

Faça o download do simulador.

Ao abrir a simulação, observa-se um interferômetro de Mach-Zehnder. Explore os diferentes componentes e funcionalidades.

4 Próximos passos

Após passar por essa revisão, você tem elementos suficientes para acompanhar as próximas aulas. Leia o documento da aula 1 e acompanhe as próximas instruções.

Referências

- [1] James Ward Brown e Ruel V. Churchill. *Complex Variables and Applications*. 9^a ed. New York: McGraw-Hill Education, 2014. ISBN: 9781259072772.
- [2] Robert Eisberg e Robert Resnick. *Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles*. 2^a ed. New York: John Wiley & Sons, 1985.
- [3] David J. Griffiths e Darrell F. Schroeter. *Introduction to Quantum Mechanics*. 3^a ed. Cambridge: Cambridge University Press, 2018.
- [4] Ricardo Avelar Sotomaia Karam. “Why are Complex Numbers Needed in Quantum Mechanics? Some Answers for the Introductory Level”. Em: *American Journal of Physics* 88.1 (2020), pp. 39–45. DOI: 10.1119/10.0000258.