



UNIVERSIDADE FEDERAL DE SÃO CARLOS
Centro de Ciências Exatas e Tecnologia - CCET
Programa de Pós Graduação em Física

Alexandre Cesar Ricardo

Estudo de algoritmos quânticos para resolução de equações diferenciais parciais

São Carlos - SP

2022

Alexandre Cesar Ricardo

Estudo de algoritmos quânticos para resolução de equações diferenciais parciais

Dissertação apresentada ao Programa de Pós-Graduação em Física para obtenção do título de Mestre em Física Teórica.

Orientador: Prof. Dr. Celso Jorge Villas-Bôas
Co-Orientador: Prof. Dr. Vivaldo Leiria
Campo Junior

São Carlos - SP

2022



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Física

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Alexandre Cesar Ricardo, realizada em 02/05/2022.

Comissão Julgadora:

Prof. Dr. Celso Jorge Villas Boas (UFSCar)

Prof. Dr. Vivaldo Leiria Campo Junior (UFSCar)

Prof. Dr. Rafael Augusto dos Santos Kapp (UFSCar)

Prof. Dr. Rafael Chaves Souto Araujo (UFRN)

Dedico este trabalho à memória de minha avó, Maria de Lourdes Lima Mendonça.

Agradecimentos

Primeiramente agradeço aos meus orientadores, Prof. Dr. Celso Jorge Villas-Bôas e Prof. Dr. Vivaldo Leiria Campo Junior, pela oportunidade de participar de um projeto tão interessante, pela paciência com meus erros e pelos ensinamentos não somente sobre ciência mas também sobre a vida.

Agradeço a minha namorada, Isabela Turati, por sua companhia, amizade e compreensão. Agradeço à Bruna por todos os momentos de suporte e, mais importante, por sua amizade.

Agradeço aos meus amigos Mateus Borges, Matheus Amaral e Mayk Túlio por todos os momentos juntos e por ajudar a me descontrair mesmo em ocasiões tensas.

Agradeço aos meus amigos Ciro Diniz e Gabriel Pedro, pelas discussões, pelo aprendizado e pelo companheirismo durante toda esta jornada.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

“Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical, and by golly it’s a wonderful problem, because it doesn’t look so easy.”

- Richard Feynman

Resumo

Com grande impacto em várias áreas de engenharia e matemática, a computação quântica é uma tecnologia emergente com uma de suas aplicações na resolução de problemas matemáticos em larga escala, como otimização e simulação. Procurando resolver equações diferenciais com computadores quânticos, estudamos algoritmos quânticos propostos a resolver equações diferenciais e suas implementações em sistemas no estado da arte.

Em particular, estudamos um algoritmo quântico para resolver equações diferenciais lineares não homogêneas da forma $A\psi(\mathbf{r}) = f(\mathbf{r})$ proposta por J. M. Arrazola *et al.* [Phys. Rev. A 100, 032306 (2019)]. Ao inverter o operador diferencial A , é possível obter uma solução particular codificada na função de onda de um sistema de variáveis contínuas, junto com a preparação e medida dos estados de modos auxiliares. Apesar de partir de uma ideia simples, a execução do algoritmo em um computador quântico real requer elementos sofisticados, como criação de um grande número de estados em superposição e detecção de estados com baixa probabilidade de sucesso.

Nesse trabalho, sugerimos modificações na estrutura para reduzir o custo de se preparar o estado inicial nos estados auxiliares, aumentar a probabilidade de sucesso e aprimorar a precisão do algoritmo para um conjunto de entradas mais específico. As modificações propostas permitem uma implementação experimental mais condizente com o atual estado da arte da computação quântica.

Palavras-chave: Computação quântica, algoritmos quânticos, sistemas de variáveis contínuas, sistemas de íons aprisionados, equações diferenciais.

Abstract

With great impact in various areas of engineering and mathematics, quantum computing is an emerging technology with one of its applications in resolving large-scale mathematical problems, such as optimization and simulation. Aiming to solve differential equations with quantum computers, we study quantum algorithms proposed to solve them and their implementations on state-of-the-art systems.

In particular, we study a quantum algorithm for solving nonhomogeneous linear partial differential equations of the form $A\psi(\mathbf{r}) = f(\mathbf{r})$ proposed by J. M. Arrazola *et al.* [Phys. Rev. A 100, 032306 (2019)]. By inverting the differential operator A , it is possible to obtain one particular solution encoded on the wave function of a continuous-variables system, along with the preparation and measurement of special ancillary modes. Despite being a simple idea, running the algorithm on a physical quantum computer requires sophisticated elements, such as creating a large number of states in superposition and detecting states with low probability of success.

In this work we suggest modifications in its structure to reduce the costs of preparing the initial ancillary states, increase the probability of success, and improve the precision of the algorithm for a specific set of inputs. These achievements enable easier experimental implementation of the quantum algorithm based on nowadays technology.

Keywords: Quantum computing, quantum algorithms, continuous variables systems, trapped-ion systems, differential equations.

Lista de ilustrações

Figura 1 – Esfera de Bloch usada para representar um qubit em um estado $ \psi\rangle$ em termos dos ângulos azimutal e polar, φ e θ , respectivamente. Fonte: Elaborada pelo autor.	23
Figura 2 – Representação circuital da proposta inicial do algoritmo. Fonte: Elaborada pelo autor.	30
Figura 3 – Representação circuital da proposta inicial do algoritmo. Fonte: Elaborada pelo autor.	30
Figura 4 – Representação circuital da forma prática do algoritmo original. Fonte: Elaborada pelo autor.	35
Figura 5 – Esquerda: Gráfico dilog de $F(a)$ em função de a com $\Delta = 0$ para diversos valores da dimensão (d) comparados à curva ideal $1/a$. Direita: Comportamento da função $F(a)$ em função de a para diversos valores da precisão na detecção do momento, gerados ao utilizar uma superposição com $d = 140$ estados Fock. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.	37
Figura 6 – Fidelidade em função do tempo efetivo $t' = t\frac{\Omega\eta_a\eta_b}{2}$ da evolução temporal exata da porta entre modos $\hat{U}_{\hat{X}_a\otimes\hat{X}_b} = \exp\{X_a \otimes X_b\}$ em comparação ao hamiltoniano sem aproximações, Eq. (4.9), considerando casos com $\eta = 0, 1$ (esquerda) e $\eta = 0, 2$ (direita). O programa escrito para gerar as curvas de fidelidade pode ser visto no Apêndice C. Fonte: Elaborada pelo autor.	41
Figura 7 – Fidelidade em função do tempo da evolução temporal exata da porta entre modos $H = X$ em comparação ao hamiltoniano sem aproximações Eq. (4.9), considerando o caso com $\eta = 0, 01$. O programa escrito para gerar as curvas de fidelidade pode ser visto no Apêndice C. Fonte: Elaborada pelo autor.	44
Figura 8 – Esquerda: Gráfico dilog de $F(a)$ em função de a considerando $\Delta = 0, 01$, da primeira proposta do algoritmo em comparação com a curva ideal $1/a$. Direita: Gráfico dilog $F(a)$ em função de a considerando $M = 0$, da primeira proposta do algoritmo em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.	50
Figura 9 – Representação circuital do algoritmo modificado. Fonte: Elaborada pelo autor.	50

Figura 10 – Gráfico dílog de $F(a)$ em função de a considerando $M = 0$ da primeira proposta do algoritmo, com diferentes valores de Δ e t em comparação com a curva ideal. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.	51
Figura 11 – Representação circuital da forma prática da segunda proposta de modificação do algoritmo. Fonte: Elaborada pelo autor.	52
Figura 12 – Esquerda: Gráfico dílog de $F(a)$ em função de a , considerando $\Delta = 0,01$, da segunda proposta do algoritmo em comparação com a curva ideal $1/a$. Direita: Gráfico dílog de $F(a)$ da primeira proposta do algoritmo, considerando $M = 0$, em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.	53
Figura 13 – Gráfico dílog de $F(a)$ da segunda proposta do algoritmo, fixando $M = 0$, com diferentes valores de Δ e t em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.	54
Figura 14 – Soluções particulares da Equação de Poisson (6.1), como funções da variável radial r , com função não homogênea $f(\mathbf{r}) = g(4, \mathbf{r})$. As soluções obtidas pelo algoritmo original com dois valores de precisão na detecção $\Delta = \{0,01; 1,0\}$ são comparadas às propostas 1 e 2 para tempos de evolução $t = 1,0$ (a), $t = 10,0$ (b) e $t = 100,0$ (c). A solução exata esperada pelo algoritmo é apresentada para comparação. Fonte: Elaborada pelo autor.	59
Figura 15 – Solução particular da equação não homogênea do oscilador harmônico quântico modificada (6.12) como função da posição x na qual a função não homogênea $f(x)$ é a função de onda do estado coerente com amplitude $\alpha = 2,5$. As soluções obtidas pelo algoritmo original consideram dois valores distintos $\Delta = \{0,01; 1,0\}$ e são comparadas com as dadas pelas propostas 1 e 2 para tempos de evolução $t = 1,0$ (esquerda) e $t = 10,0$ (direita). A solução exata é apresentada para efeitos de comparação. Esse resultado mostra que as versões modificadas do algoritmo tem a evolução temporal como uma ferramenta poderosa de melhorar a fidelidade do algoritmo. Fonte: Elaborada pelo autor.	61

Sumário

1	INTRODUÇÃO	17
2	OBJETIVOS E MÉTODOS	21
3	FUNDAMENTOS TEÓRICOS	23
3.1	Introdução à computação quântica	23
3.1.1	Computação quântica digital	23
3.1.2	Computação quântica sobre variáveis contínuas	24
3.2	Íons aprisionados	25
3.2.1	A armadilha de íons	25
3.2.2	Quantização do movimento de um íon aprisionado	26
3.2.3	Quantização da energia interna do íon	27
3.2.4	Quantização da interação entre estados internos e externos do íon	27
3.3	Algoritmo para resolver equações diferenciais parciais não-homogêneas	28
3.4	O algoritmo	30
3.5	Aproximações necessárias para tornar o algoritmo fisicamente realizável	31
3.6	Consequências diretas do modelo prático do algoritmo	33
4	PORTAS LÓGICAS QUÂNTICAS	39
4.1	Hamiltoniano de interação	39
4.2	Interações elementares e portas quânticas	39
4.2.1	Íons aprisionados em potenciais bi-dimensionais - Interação entre modos	41
4.2.2	A porta $e^{it\hat{X}}$ e o operador deslocamento	42
4.2.3	A porta de Fourier	43
4.2.4	A porta \hat{X}^2 e o operador de compressão	45
4.2.5	A porta \hat{X}^3	46
4.2.6	Interação Kerr	46
4.2.7	Portas lógicas em variáveis contínuas	47
5	PROPOSTAS DE MODIFICAÇÕES	49
5.1	Proposta 1	49
5.2	Proposta 2	51
5.3	Probabilidade de sucesso	53
6	EXEMPLOS	57
6.1	Operador de Laplace	57

6.2	Operador do Oscilador Harmonico Quântico	59
7	CONCLUSÃO	63
	REFERÊNCIAS	65
	APÊNDICES	69
	APÊNDICE A – OBTENDO OS TERMOS DA SUPERPOSIÇÃO NAS PROPOSTAS DE MODIFICAÇÃO 1 E 2.	71
A.1	Primeira proposta de modificação	71
A.2	Segunda proposta de alternativa	71
	APÊNDICE B – PROGRAMAS EM FORTRAN CRIADOS PARA SIMULAR AS CURVAS APRESENTADAS NAS SEÇÕES 3.3 E 5	73
B.1	Programa para gerar as curvas da versão prática do algoritmo original	73
B.2	Algoritmo para gerar as curvas da primeira proposta de modificação.	74
B.3	Algoritmo para gerar as curvas da segunda proposta de modificação.	76
	APÊNDICE C – PROGRAMAS EM PYTHON UTILIZADOS PARA SIMULAR AS DINÂMICAS DE ÍONS APRESEN- TADAS NA SEÇÃO 4	79
C.1	Interação entre modos	79
C.2	Porta \hat{X}	83

1 Introdução

Em 1982, Richard Feynman conjecturou que a simulação de sistemas quânticos complexos requereria o desenvolvimento de um novo tipo de computador, um computador quântico (FEYNMAN, 1982), isto é, uma máquina baseada em sistemas de natureza quântica e, portanto, probabilístico. Em 1994, Peter Shor desenvolveu um algoritmo para computadores quânticos - um algoritmo quântico - capaz de fatorizar grandes números inteiros em números primos em um número de passos que cresce polinomialmente com o número de dígitos (SHOR, 1994). O algoritmo de Shor é considerado por muitos um dos resultados mais expressivos e importantes da computação quântica, uma vez que computadores clássicos podem realizar esta tarefa em um número de passos que cresce exponencialmente com o número de dígitos.

Sistemas de íons aprisionados (CIRAC; ZOLLER, 1995), supercondutores (HUANG et al., 2020) ou ressonância magnética nuclear (NMR) (MARX et al., 2000) podem ser usados como modelos de plataformas para implementação da computação quântica. Nas últimas décadas, o estado da arte da computação quântica foi aprimorado, resultando em diversas aplicações impactantes em áreas como finanças (ORÚS; MUGEL; LIZASO, 2019b; ORÚS; MUGEL; LIZASO, 2019a; COMPUTING; PATENTS, ; MUGEL et al., 2020), equações diferenciais (ARRAZOLA et al., 2019; XIN et al., 2020a), e simulações de sistemas quânticos (BABBUSH; LOVE PETER J.AND ASPURU-GUZIK, 2014; WIEBE et al., 2011).

Computação quântica é o estudo de algoritmos e plataformas que usam sistemas quânticos para manipular informação. Informação quântica é armazenada em *qubits*, que podem existir em uma superposição de estados quânticos, ao contrário dos bits clássicos, que existem apenas em um dado estado (0 ou 1). Computadores quânticos manipulam qubits por portas quânticas. Em particular, Sleator e Weinfurter (SLEATOR; WEINFURTER, 1995) demonstraram que todas as portas lógicas quânticas podem ser decompostas em rotações simples e portas *C-NOT* (NIELSEN; CHUANG, 2000). Dessa maneira, a computação quântica consegue realizar certas tarefas mais eficientemente do que computadores tradicionais, tais como resolver equações diferenciais (XIN et al., 2020b) e fatorizar grandes números inteiros em primos (SHOR, 1994), entre outras aplicações.

Embora o conceito de qubits - superposições de sistemas de dois estados quânticos - esteja intrinsecamente ligado à computação quântica, um conceito mais geral é o de

qudit, uma superposição de d estados quânticos (WANG et al., 2020). Trabalhar com esta forma de armazenamento de informação em *qudits* é computação quântica digital. Além da computação quântica digital, existem outros tipos de computação quântica, como a computação quântica sobre variáveis contínuas (LLOYD; BRAUNSTEIN, 1999; BRAUNSTEIN; LOOCK, 2005; WERBOS; DOLMATOVA, 2016), também chamada de computação quântica analógica, onde os qubits são representados por variáveis contínuas conjugadas como posição e momento dos modos de um campo eletromagnético (TASCA et al., 2011), ou dos modos de um íon aprisionado (ORTIZ-GUTIÉRREZ et al., 2017). Há progresso recente em um novo tipo de computação quântica, que mistura os dois modelos digital e analógico, chamado de computação quântica digital-analógica (Sanz, 2019; PARRA-RODRIGUEZ et al., 2020).

Equações diferenciais são uma das maneiras mais comuns de se representar problemas dinâmicos matematicamente, inclusive problemas de interesse comercial e industrial. Além de suas soluções explícitas, algumas características como unicidade, regularidade e o espaço ao qual as soluções pertencem são relevantes, dependendo do problema. Muitos algoritmos quânticos foram desenvolvidos para resolver diferentes tipos de equações diferenciais como sistemas lineares de equações diferenciais de primeira ordem (XIN et al., 2020a), equações diferenciais parciais com condições periódicas de contorno (GARCÍA-MOLINA; RODRÍGUEZ-MEDIAVILLA; GARCÍA-RIPOLL, 2021), e operadores específicos como a equação de onda (COSTA; JORDAN; OSTRANDER, 2019) ou equação de Poisson (LIU et al., 2020). Em (ARRAZOLA et al., 2019) Arrazola, Kalajdziewski, Weedbrook, e Lloyd desenvolveram um algoritmo quântico utilizando o modelo de computação quântica sobre variáveis contínuas (CQVC) que encontra uma solução particular de uma equação diferencial não homogênea invertendo o operador diferencial. O impacto notável da proposta do último trabalho citado em ramos da física e da engenharia nos inspirou a dedicar nossos estudos de forma a aprimorar seus resultados e fidelidade.

Neste trabalho, estudamos os efeitos de se implementar um algoritmo quântico em uma plataforma física e apresentamos modificações de maneira a reduzir os custos de se preparar os estados iniciais e obter um algoritmo mais preciso para uma classe mais restrita de operadores diferenciais. Também é desenvolvido um conjunto universal de portas lógicas no modelo CQVC na plataforma de íons aprisionados. Na Seção 2, apresentamos os tipos de equações diferenciais que podem ser resolvidas pelo algoritmo quântico proposto por Arrazola *et al.*. Na Seção 3, apresentamos aspectos básicos da computação quântica, da plataforma de íons aprisionados e também é descrito o funcionamento do algoritmo para resolver equações diferenciais parciais não-homogêneas estudado no trabalho. Na Seção 4, estudamos a interação laser-íon em uma armadilha de Paul de modo a ser possível desenvolver um conjunto universal de portas lógicas no modelo CQVC. Na Seção

5, apresentamos propostas de modificação neste algoritmo de maneira que possamos obter resultados com alta fidelidade para a classe de operadores positivo (negativo)-semidefinidos. Na Seção 6, apresentamos exemplos de aplicações do algoritmo comparando os algoritmos propostos com modificações e o algoritmo original. Na Seção 7, são dadas as considerações finais.

2 Objetivos e métodos

Neste projeto investigamos algoritmos quânticos para resolver equações diferenciais e sua implementação em computadores quânticos de íons aprisionados. Muitos problemas importantes na física e engenharia são descritos por equações diferenciais, uma vez que são uma maneira simples de se resolver e representar sistemas. Equações diferenciais conectam uma combinação de derivadas de uma função, que é a solução do nosso problema, a uma função externa, ou não-homogênea. Podemos associar uma equação diferencial como sendo o resultado da ação de um operador diferencial na função solução u ,

$$\hat{A}u = f,$$

onde f é uma função externa que pode depender das variáveis de u . Quando $f \neq 0$ e \hat{A} é linear, $\hat{A}(u + v) = \hat{A}u + \hat{A}v$, a solução pode ser separada em duas partes: Uma solução particular u_p , definida pela função externa (ou não-homogênea) f e uma solução homogênea u_H que satisfaz $\hat{A}u_H = 0$ e é unicamente definida pelo operador diferencial \hat{A} . Como consequência, todo operador diferencial tem uma família de soluções particulares para cada função não-homogênea. É possível construir uma solução geral tendo condições de contorno suficientes, o conjunto de soluções homogêneas e uma solução particular. Neste projeto, iremos trabalhar com equações diferenciais tais que o operador \hat{A} é Hermitiano e a função não-homogênea f é uma função quadrado-integrável.

Partindo do algoritmo quântico baseado em CQVC que encontra uma solução particular de uma equação diferencial parcial ao inverter o operador associado, proposto por Juan Miguel Arrazola *et al.* em (ARRAZOLA *et al.*, 2019), nos propusemos a analisar a possibilidade de desenvolver a implementação em uma plataforma de íons aprisionados, o que de certo alteraria partes estruturais do algoritmo. Baseados na informação obtida e dependendo da relação entre a função não-homogênea f e o operador \hat{A} , pudemos estimar a precisão do algoritmo e sugerir modificações. Para facilitar a compreensão da reestruturação do algoritmo, dividimos o trabalho em três partes: i) Preparação dos estados iniciais do sistema e como eles influenciam na precisão do algoritmo, onde encontramos maneiras de minimizar a dificuldade na criação de tais estados iniciais, sugerindo modificações. ii) Decomposição do operador de evolução em termos de um conjunto universal de operações (3.3) e no desenvolvimento destas interações em sistemas de íons aprisionados, que pode ser visto na seção 3.2, onde também são realizadas simulações para algumas portas do conjunto universal, a fim de analisar a fidelidade das portas obtidas. iii) Analisamos a precisão do algoritmo quando variamos a precisão das medidas projetivas ao final do

algoritmo, onde sugerimos a substituição de detecções homódinas por medidas projetivas no estado de vácuo nos modos auxiliares, aumentando a probabilidade de sucesso do algoritmo, ao custo de reduzir a fidelidade final.

3 Fundamentos teóricos

3.1 Introdução à computação quântica

3.1.1 Computação quântica digital

Na computação quântica baseada em portas lógicas, a informação é armazenada em *qubits*, sistemas de dois níveis que podem existir na superposição de estados $|0\rangle$ e $|1\rangle$. Neste caso, é possível representar qubits como um ponto na superfície de uma esfera com raio unitário cujos polo norte e sul representam os estados $|0\rangle$ e $|1\rangle$, respectivamente. Esta representação geométrica é chamada de esfera de Bloch e as operações realizadas individualmente em um qubit são rotações nesta esfera e podem ser representadas como matrizes de rotação (NIELSEN; CHUANG, 2000). Em geral, um computador quântico terá mais do que apenas um qubit, precisando então de operações entre qubits, também chamadas de operações *controladas*. Cada operação controlada pode ser decomposta em uma combinação de rotações e portas C-NOT (SLEATOR; WEINFURTER, 1995). Ou seja, a partir de operações sobre um único qubit e portas C-NOT entre dois qubits, qualquer operação lógica quântica pode ser implementada, o que nos permite classificar o conjunto de tais operações como conjunto universal, a partir do qual qualquer algoritmo pode ser executado.

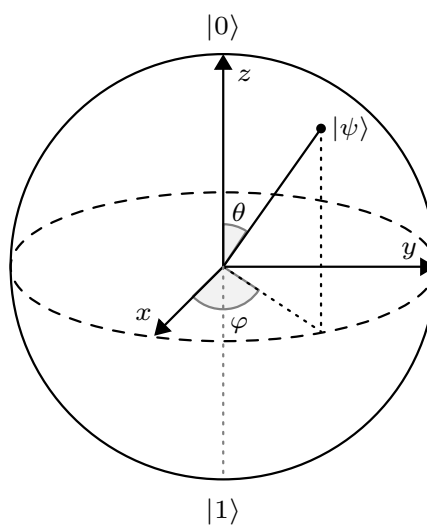


Figura 1 – Esfera de Bloch usada para representar um qubit em um estado $|\psi\rangle$ em termos dos ângulos azimutal e polar, φ e θ , respectivamente. Fonte: Elaborada pelo autor.

3.1.2 Computação quântica sobre variáveis contínuas

As quadraturas de um campo eletromagnético ou os modos vibracionais de íons aprisionados são ótimas alternativas para processar informação em computadores quânticos. No modelo CQVC, a informação pode ser codificada na função de onda ou função de Wigner (ORTIZ-GUTIÉRREZ et al., 2017), por exemplo. Para manipular esta informação apropriadamente, é necessário construir interações específicas de ao menos um e dois modos, de maneira análoga às rotações e operações controladas em qubits, uma vez que a informação agora não é mais dada em um espaço discreto e sim em um espaço contínuo.

Para implementar as portas lógicas necessárias em um sistema de variáveis contínuas, podemos usar decomposições de hamiltonianos em termos de conjuntos universais que irão depender no tipo de hamiltoniano necessário para o problema.

Existem vários tipos de decomposição de portas lógicas em variáveis contínuas, como o modelo de Trotter-Suzuki (TROTTER, 1959; SUZUKI, 1993; DHAND; SANDERS, 2014) que realiza a decomposição de portas lógicas do tipo $\hat{H} = \sum_{j=1}^N \hat{H}_j$ em

$$e^{i\hat{H}t} \approx \prod_{j=1}^N (e^{i\frac{t}{K}\hat{H}_j})^K, \quad (3.1)$$

com erro de ordem $O(t^2/K)$. Utilizando a decomposição de Trotter-Suzuki junto com um modelo de decomposição exato de interações (KALAJDZIEVSKI; ARRAZOLA, 2019), é possível decompor hamiltonianos da forma

$$\hat{H} = \hat{A}\hat{Y}\hat{Z} = (\lambda\mathbb{I} + \sum_{j=1}^N a_j\hat{X}_j + b_j\hat{P}_j + \alpha_j\hat{X}_j^2 + \beta_j\hat{P}_j^2)\hat{X}_{N+1}\hat{X}_{N+2}, \quad (3.2)$$

onde \hat{X}_j são operadores posição de $N + 2$ modos distintos, no conjunto universal:

$$\left\{ e^{i\frac{\pi}{2}(\hat{X}_j^2 + \hat{P}_j^2)}, e^{it_1\hat{X}_j}, e^{it_2\hat{X}_j^2}, e^{it_3\hat{X}_j^3}, e^{i\tau\hat{X}_i \otimes \hat{X}_j} \right\}, \quad (3.3)$$

com $i \neq j$. De forma mais geral, um conjunto universal pode ser gerado por um conjunto de operações não-homogêneas quadráticas nos operadores canônicos de criação e aniquilação e mais um termo que gere dinâmica não linear (MILBURN; WALLS, 1983; BARTLETT; SANDERS, 2002), representado pela porta $\exp\{it_3\hat{X}^3\}$ na Eq. (3.3).

3.2 Íons aprisionados

3.2.1 A armadilha de íons

Em sistemas de íons aprisionados, cadeias de íons são aprisionados por campos eletromagnéticos que nos permitem construir operações lógicas via interação de lasers - luz monocromática coerente - com os íons aprisionados. Atualmente, esta plataforma de computação quântica obtém sucesso graças a sua alta fidelidade nas interações que geram portas lógicas e quantidade de qubits que podem ser manipulados (GRZESIAK et al., 2020). Íons aprisionados são muito úteis para o estudo da interação radiação-matéria e para o desenvolvimento e implementação de computação quântica. As armadilhas de Pauli, tipo mais utilizado atualmente, confinam íons em um potencial da forma (LEIBFRIED et al., 2007)

$$\phi(x, y, z, t) = \frac{U}{2}(ax^2 + by^2 + cz^2) + \frac{\tilde{U}}{2}(\alpha x^2 + \beta y^2 + \gamma z^2) \cos(\omega_{rf}t).$$

O potencial de um campo elétrico deve satisfazer a equação de Laplace $\nabla^2\phi = 0$ para cada tempo t , logo

$$a + b + c = 0, \quad (3.4a)$$

$$\alpha + \beta + \gamma = 0. \quad (3.4b)$$

As Eqs. (3.4a) e (3.4b) expressam condições que nos forçam a escolher α , β ou γ não nulos, tornando impossível aprisionar um íon em um potencial harmônico estático. No entanto, ainda é possível confinar íons em potenciais dependentes do tempo. Em particular, é possível resolver o problema de aprisionar um íon ao encontrar as equações de movimento de uma partícula na vizinhança de um potencial ϕ (LEIBFRIED et al., 2007), considerando campos altamente oscilatórios. Neste caso, a equação de movimento, obtida pelo tratamento clássico da partícula aprisionada na direção axial, se torna

$$\frac{d^2x}{dt^2} = -\frac{Z|e|}{m} [Ua + \tilde{U}\alpha \cos(\omega_{rf}x)], \quad (3.5)$$

onde Z é o nível de ionização do átomo, m a massa do átomo e e a carga elementar do elétron. A Eq. (3.5), pode ser transformada em

$$\frac{d^2x}{d\xi^2} + [a_x - 2q_x \cos(2\xi)] = 0, \quad (3.6)$$

com $\xi = \frac{\omega_{rf}t}{2}$, $a_x = \frac{4Z|e|Ua}{m\omega_{rf}^2}$ e $q_x = -\frac{2Z|e|\tilde{U}\alpha}{m\omega_{rf}^2}$.

Conhecida como Equação de Mathieu, podemos encontrar a solução aproximada da Eq. (3.6), quando $a_x, q_x^2 \ll 1$ (MATHIEU, 1873):

$$x(t) \approx A \cos(\nu_x t) \left[1 - \frac{q_x}{2} \cos(\omega_{rf} t) \right], \quad (3.7)$$

onde o segundo termo dentro dos colchetes é chamado de micromovimento, sendo desprezível no limite de altas frequências, tornando o movimento clássico da partícula praticamente harmônico.

3.2.2 Quantização do movimento de um íon aprisionado

O hamiltoniano de um íon aprisionado em uma armadilha de Pauli unidimensional é descrito aproximadamente pelo hamiltoniano de um oscilador harmônico

$$\hat{H}_m = \frac{\hat{p}^2}{2m} + \frac{m}{2} \nu_x^2 \hat{x}^2, \quad (3.8)$$

onde ν_x é a frequência de oscilação do íon. Os autoestados deste hamiltoniano podem ser obtidos ao definirmos

$$\hat{a} = \sqrt{\frac{m\nu_x}{2\hbar}} \left(\hat{x} + \frac{i}{m\nu_x} \hat{p} \right), \quad (3.9a)$$

$$\hat{a}^\dagger = \sqrt{\frac{m\nu_x}{2\hbar}} \left(\hat{x} - \frac{i}{m\nu_x} \hat{p} \right), \quad (3.9b)$$

respectivamente os operadores de aniquilação e criação de um fônon na base de autoestados do hamiltoniano.

Substituindo (3.9a) e (3.9b) em (3.8), é possível obter

$$\hat{H}_m = \hbar\nu_x \left(\hat{a}^\dagger \hat{a} + \frac{1}{2} \right). \quad (3.10)$$

Definindo o operador número

$$\hat{N} = \hat{a}^\dagger \hat{a}, \quad (3.11)$$

fica claro que \hat{H}_m e \hat{N} compartilham os mesmos autoestados, $|n\rangle$, chamados estados de Fock, que satisfazem

$$\hat{N} |n\rangle = n |n\rangle. \quad (3.12)$$

A ação de \hat{a} e \hat{a}^\dagger é dada por

$$\hat{a} |n\rangle = \sqrt{n} |n-1\rangle, \quad (3.13a)$$

$$\hat{a}^\dagger |n\rangle = \sqrt{n+1} |n+1\rangle. \quad (3.13b)$$

Os estados de Fock formam um conjunto de autoestados do operador Hermitiano \hat{H}_m e representam o número de fônons nos modos vibracionais.

Podemos re-escalar o autovalor de energia de \hat{H}_m em $\hbar\nu_x/2$, obtendo

$$\hat{H}_m = \hbar\nu_x \hat{a}^\dagger \hat{a}, \quad (3.14)$$

como o hamiltoniano do campo do íon aprisionado, cujos autoestados são os estados de Fock.

3.2.3 Quantização da energia interna do íon

De maneira que possamos analisar a interação entre lasers e o íon aprisionado, assumiremos que o íon tem apenas dois níveis eletrônicos, denominados por estados excitado $|e\rangle$ e fundamental $|g\rangle$. Essa consideração é aceitável experimentalmente caso as frequências dos campos eletromagnéticos sejam próximas apenas da ressonância entre estes dois estados internos, de maneira que possamos descartar a interação com outros estados do íon. Definindo as energias dos estados $|g\rangle$ e $|e\rangle$ como sendo E_g e E_e , respectivamente, o hamiltoniano que descreve o sistema de dois níveis pode ser escrito como (LEIBFRIED et al., 2007)

$$\hat{H}_a = E_g \omega_g |g\rangle \langle g| + E_e |e\rangle \langle e| = \frac{E_g + E_e}{2} (|g\rangle \langle g| + |e\rangle \langle e|) + \hbar \frac{\omega_0}{2} (|e\rangle \langle e| - |g\rangle \langle g|). \quad (3.15)$$

onde $\omega_0 = (E_e - E_g)/\hbar$ é chamada de frequência de transição interna do íon. Como as matrizes de Pauli, juntamente com a matriz identidade, formam uma base para todas as matrizes hermitianas 2×2 podemos escrever \hat{H}_a em função das matrizes de Pauli, obtendo

$$\hat{H}_a = \hbar \frac{\omega_0}{2} \sigma_z,$$

onde a energia foi reescalada em $(E_g + E_e)/2$.

3.2.4 Quantização da interação entre estados internos e externos do íon

Restringindo o comprimento de onda do laser a valores muito maiores do que as dimensões do íon, é possível descrever o íon como um dipolo elétrico, cuja interação com um campo eletromagnético pode ser dada por (LEIBFRIED et al., 2007)

$$\hat{V}_I = -\hat{\mathbf{d}} \cdot \mathbf{E}, \quad (3.16)$$

onde $\hat{\mathbf{d}}$ é o operador de dipolo do íon e \mathbf{E} é o campo elétrico, que será tratado como uma onda plana clássica, e pode ser descrito por

$$\mathbf{E} = \mathbf{E}_0 (e^{i(\mathbf{k} \cdot \mathbf{x} - \omega t + \phi)} + h.c.), \quad (3.17)$$

onde \mathbf{k} é o vetor de onda, ω é a frequência, ϕ é a fase da onda eletromagnética e \mathbf{E}_0 é o vetor amplitude da onda. Se assumirmos $\hbar\omega \approx E_e - E_g$, então a interação entre o campo

eletromagnético e todos os estados eletrônicos exceto $|g\rangle$ e $|e\rangle$ pode ser descartado. Dessa maneira, o hamiltoniano de interação entre o íon e o campo pode ser escrito como

$$\hat{V}_I = (|g\rangle \langle e| + |e\rangle \langle g|) \langle g| \hat{V}_I |e\rangle, \quad (3.18)$$

onde assumimos $\langle g| \hat{V}_I |e\rangle$ real. Desta maneira, a interação entre um dipolo com carga e_- e uma onda plana pode ser descrita como a matriz cujos elementos são dados por

$$\langle \alpha| \hat{H}_D |\beta\rangle = e_- \langle \alpha| \hat{\mathbf{d}} \cdot \mathbf{E}_0 |\beta\rangle [e^{i(\mathbf{k} \cdot \hat{\mathbf{x}} - \omega t + \phi)} + h.c.], \quad (3.19)$$

onde $\hat{\mathbf{x}}$ é o operador posição do íon e $h.c.$ é o hermitiano conjugado da expressão entre chaves. O operador deslocamento $\hat{\mathbf{d}}$ pode ser escrito como

$$\hat{\mathbf{d}} = \sum_{\alpha, \beta=g,e} \mathbf{d}_{\alpha\beta} |\alpha\rangle \langle \beta|. \quad (3.20)$$

Assumindo $\hat{\mathbf{d}} = (\hat{d}_{ge}, \hat{d}_{eg})$, podemos definir a frequência de acoplamento de Rabi Ω

$$e_- \langle g| \mathbf{E}_0 \cdot \hat{\mathbf{d}} |e\rangle = \hbar \frac{\Omega}{2}.$$

Podemos escrever

$$\hat{V}_I = \hbar \frac{\Omega}{2} (\hat{\sigma}_+ + \hat{\sigma}_-) (e^{i(\mathbf{k} \cdot \hat{\mathbf{x}} - \omega t + \phi)} + h.c.), \quad (3.21)$$

como o potencial de interação átomo-campo. O produto escalar $\mathbf{k} \cdot \hat{\mathbf{x}}$ pode ser escrito como

$$\mathbf{k} \cdot \hat{\mathbf{x}} = k \sqrt{\frac{\hbar}{2m\omega}} (\hat{a} + \hat{a}^\dagger) \cos \theta = \eta (\hat{a} + \hat{a}^\dagger), \quad (3.22)$$

onde θ é o ângulo entre a direção do laser e a direção de movimento do íon na armadilha. O parâmetro η é chamado parâmetro de Lamb-Dicke.

A partir da Eq. (3.21) é possível derivar Hamiltonianos que descrevem diferentes interações entre os graus de liberdade internos e externos do íon aprisionado, de maneira que seja possível manipular a informação armazenada tanto nos modos vibracionais, quanto nos níveis internos de energia.

3.3 Algoritmo para resolver equações diferenciais parciais não-homogêneas

Ao resolvermos equações diferenciais, é importante saber qual o tipo de equação diferencial estamos lidando. O objetivo desta seção é apresentar o algoritmo desenvolvido em (ARRAZOLA et al., 2019), que funciona para equações diferenciais lineares não-homogêneas.

Uma equação diferencial parcial não homogênea pode ser associada à ação de um operador diferencial \hat{A} sobre uma função solução ψ , resultando na função externa (ou não homogênea) f ,

$$\hat{A}\psi(\mathbf{x}) = f(\mathbf{x}). \quad (3.23)$$

O objetivo do algoritmo em questão é inverter o operador \hat{A} de maneira a obter uma solução particular do problema. Uma vez conhecida uma solução particular, as soluções homogêneas e condições de contorno é possível obter a solução do problema. Para a inversão do operador, notemos que dada qualquer função ímpar $g(x)$ que satisfaça $\int_0^\infty g(x)dx = 1$, temos que $a^{-1} = \int_0^\infty g(ax)dx$ para $a \neq 0$ real. Escolhendo tal função como sendo $g(x) = xe^{-x^2/2}$ e levando em conta o fato de que a transformada de Fourier de g é dada por

$$\mathcal{F}(xe^{-x^2/2}) = -iye^{-y^2/2},$$

podemos encontrar que

$$a^{-1} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Theta(x)e^{iaxy}xe^{-x^2/2}dxdy, \quad (3.24)$$

onde Θ é a função degrau de Heaviside. Este é o primeiro passo para a inversão do operador diferencial \hat{A} . De agora em diante, assumimos que o conjunto $\{|a\rangle\}$ forma a base do operador hermitiano \hat{A} , de maneira que $\hat{A}|a\rangle = a|a\rangle$. Portanto, o conjunto $\{|a\rangle\}$ diagonaliza o operador $e^{i\hat{A}xy}$ e

$$\begin{aligned} \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Theta(x)e^{-i\hat{A}xy}xe^{-x^2/2}|a\rangle dxdy &= \\ \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Theta(x)e^{iaxy}xe^{-x^2/2}|a\rangle dxdy &= a^{-1}|a\rangle. \end{aligned}$$

Escolhendo $|f\rangle$ cuja função de onda na base de posição é $\langle \mathbf{x}|f\rangle = f(\mathbf{x})$, podemos expressar $|f\rangle$ na base $\{|a\rangle\}$, para um operador com espectro discreto, como sendo

$$|f\rangle = \sum_{j \neq 0} \sum_{k=0}^{m_j} f_{a_j,k} |a_j, k\rangle,$$

onde m_j é o grau de multiplicidade do autovalor a_j . Por simplicidade, foi definido $a_0 = 0$.

Quando aplicamos \hat{A} em $|f\rangle$ é possível obter

$$\hat{A}|f\rangle = \sum_{j \neq 0} \sum_{k=0}^{m_j} a_j f_{a_j,k} |a_j, k\rangle.$$

Definindo o estado $|h\rangle = \sum_{j \neq 0} \sum_{k=0}^{m_j} f_{a_j,k}/a_j |a_j, k\rangle$. A ação de \hat{A} em $|h\rangle$ é

$$\hat{A}|h\rangle = \sum_{j \neq 0} \sum_{k=0}^{m_j} f_{a_j,k} |a_j, k\rangle = |f\rangle,$$

portanto, a função de onda de $|h\rangle$ satisfaz a Eq. (3.23), quando projetada na base da posição $\{|x\rangle\}$.

3.4 O algoritmo

O objetivo do algoritmo é obter o estado $|h\rangle$ através de manipulações no estado não homogêneo $|f\rangle$ condicionadas pelos dois modos auxiliares preparados nos estados iniciais

$$|s\rangle = \int_{-\infty}^{\infty} |x\rangle \langle x|s\rangle dx = \int_{-\infty}^{\infty} \Theta(x) |x\rangle dx, \quad (3.25a)$$

$$|\tilde{1}\rangle = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} ye^{-y^2/2} |y\rangle dy. \quad (3.25b)$$

É importante notar que o estado $|\tilde{1}\rangle$ é proporcional ao primeiro estado excitado de Fock. A princípio, assumimos que é possível preparar o estado inicial $|s\rangle_1 |\tilde{1}\rangle_2 |f\rangle_3$. Após a preparação do estado inicial, evoluímos o hamiltoniano $\hat{H} = \hat{A}\hat{X}\hat{Y}$ sobre o sistema, onde \hat{A} age no terceiro modo, relacionado ao estado $|f\rangle$, \hat{X} é o operador posição que atua no modo 1 e \hat{Y} é o operador posição que atua no modo 2. Dessa maneira, o algoritmo pode ser representado pelo circuito da Figura 2 com $|\Psi\rangle$ dado por

$$|\Psi\rangle = \frac{i}{\sqrt{2\pi}} \int_{\mathbb{R}^2} \Theta(x) ye^{-y^2/2} e^{-i\hat{A}xy} |f\rangle |x\rangle |y\rangle dx dy. \quad (3.26)$$

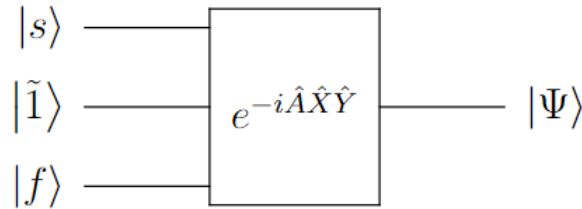


Figura 2 – Representação circuital da proposta inicial do algoritmo. Fonte: Elaborada pelo autor.

Após isso, é necessário fazer uma medida projetiva no estado $|p=0\rangle_1 |p=0\rangle_2$, transformando o circuito anterior de acordo com a representação dada na Figura 3

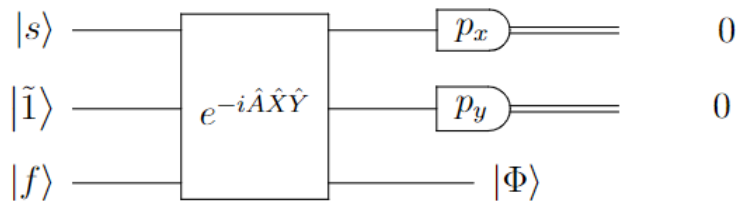


Figura 3 – Representação circuital da proposta inicial do algoritmo. Fonte: Elaborada pelo autor.

Com isso, o estado $|f\rangle$ será projetado no estado

$$|\Phi\rangle = \frac{i}{\sqrt{2\pi}} \int_{\mathbb{R}^2} \Theta(x)y e^{-y^2/2} e^{-i\hat{A}xy} |f\rangle dx dy, \quad (3.27)$$

que representa $|f\rangle$ na base dos autoestados de \hat{A} . Portanto, é possível obter $|\Psi\rangle = |h\rangle$, onde identificamos $|\Phi\rangle = \hat{A}^{-1} |f\rangle$.

3.5 Aproximações necessárias para tornar o algoritmo fisicamente realizável

Na prática, existem algumas restrições que tornam este algoritmo não realizável fisicamente: A primeira é o fato de que o estado degrau $|s\rangle$ não é um estado físico, uma vez que não pode ser representado por uma função de onda contínua e normalizável.

Para contornarmos o problema da função degrau, podemos aproximar $\Theta(x)$ por uma função degrau finita $s_L(x)$,

$$s_L(x) = \begin{cases} 0, & \text{se } x < 0, \\ 1/\sqrt{L}, & \text{se } 0 \leq x \leq L, \\ 0, & \text{se } x > L. \end{cases} \quad (3.28)$$

No entanto, ainda é impossível associar um estado físico à função (3.28), uma vez que $s_L(x)$ é descontínua em $x = 0, L$. Para contornar isso, podemos expandir o estado associado $|s_L\rangle$,

$$|s_L\rangle = \int_0^L \frac{1}{\sqrt{L}} |x\rangle dx, \quad (3.29)$$

em um número finito de estados de Fock. É esperado que, para um número suficientemente grande de estados de Fock, o resultado se aproxime do resultado de $|s_L\rangle$: Para observar a fidelidade dos resultados, o estado $|f\rangle$ pode ser substituído por $|a\rangle$. Dessa maneira, é possível compreender a precisão do algoritmo dado um par \hat{A}, f , uma vez que

$$\begin{aligned} \hat{A}_L |a\rangle |0_x\rangle |0_y\rangle &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{+\infty} dy y e^{-\frac{y^2}{2}} e^{-i\hat{A}xy} |a\rangle |0_x\rangle |0_y\rangle = \\ &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{+\infty} dy y e^{-\frac{y^2}{2}} e^{-iaxy} |a\rangle |0_x\rangle |0_y\rangle = \frac{1}{a} \left(1 - e^{-a^2 L^2/2}\right) |a\rangle |0_x\rangle |0_y\rangle, \end{aligned} \quad (3.30)$$

onde $|0_x\rangle$ e $|0_y\rangle$ são estados com momento nulo nos modos 1 e 2, respectivamente. Identificamos estes modos como sendo os modos auxiliares.

Pela Eq. (3.30) é possível observar que o termo de erro, $e^{-a^2L^2/2}$, se torna desprezível quando o produto aL se torna suficientemente grande. Dados uma função não homogênea f e um operador diferencial \hat{A} é possível expandir $|f\rangle$ na base $\{|a\rangle\}$ e estimar a precisão do algoritmo ao analisar o quão forte é a projeção de $|f\rangle$ no conjunto $\{|a\rangle\}_{|a|\lesssim 1/L}$.

Além da aproximação feita pelo estado $|s_L\rangle$, também deve-se incluir a precisão Δ das medidas projetivas nos modos auxiliares. Para isso, primeiramente consideraremos detecções homódinas seguidas por uma pós-seleção com todos os estados com momento nulo e precisão Δ , dados por $|\Delta\rangle = \sqrt{\frac{\Delta}{\pi^{1/2}}} \int_{-\infty}^{\infty} e^{-x^2\Delta^2/2} |x\rangle dx$. Desta maneira, podemos utilizar a Eq. (3.29) e o estado após a detecção homódina $|\Delta\rangle$ para obter

$$\begin{aligned} \hat{A}_{approx}^{-1} |a\rangle &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{\infty} dy y e^{-y^2/2} e^{-iaxy} e^{-(x^2+y^2)\Delta^2/2} |a\rangle = \\ &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{\infty} dy y e^{-\frac{y^2}{2}(1+\Delta^2)} e^{-iaxy} e^{-x^2\Delta^2/2} |a\rangle = \\ &= \frac{i}{\sqrt{2\pi}} \int_0^L dx \int_{-\infty}^{\infty} dy \frac{y e^{-y^2/2}}{1+\Delta^2} e^{-iy\left(\frac{ax}{\sqrt{1+\Delta^2}}\right)} e^{-x^2\Delta^2/2} |a\rangle = \\ &= \frac{a}{(1+\Delta^2)^{3/2}} \int_0^L x e^{-\frac{x^2}{2}\left[\frac{a^2+\Delta^2+\Delta^4}{1+\Delta^2}\right]} dx |a\rangle = F(a) |a\rangle, \end{aligned} \quad (3.31)$$

onde F é dada por

$$F(a) = \frac{a}{\sqrt{1+\Delta^2}} \frac{1}{a^2 + \Delta^2 + \Delta^4} \left(1 - \exp\left\{ -\frac{L^2}{2} \left(\frac{a^2 + \Delta^2 + \Delta^4}{1 + \Delta^2} \right) \right\} \right). \quad (3.32)$$

Na equação 3.32, foi multiplicado um parâmetro λ proporcional às normas dos estados $|\Delta\rangle$ e $|s_L\rangle$, de maneira que $F(a)$ convirja para $1/a$ quando os parâmetros $L \rightarrow \infty$, $\Delta \rightarrow 0$ e $d \rightarrow \infty$, isto é,

$$\lambda = \sqrt{\frac{L}{\pi}} \frac{1}{2\Delta}. \quad (3.33)$$

É possível obter este parâmetro analiticamente Eq. (3.32) mostra que a precisão da detecção tem grande importância na precisão do algoritmo. É possível calcular como \hat{A}_{approx}^{-1} age sobre $|f\rangle$,

$$\hat{A}_{approx}^{-1} |f\rangle = \sum_j \sum_{k=1}^{m_j} F(a_j) f_{jk} |a_j, k\rangle. \quad (3.34)$$

Por fim, é possível analisar a precisão de uma função externa geral $|f\rangle$ e um operador diferencial \hat{A} , definindo

$$\hat{A}^{-1} = \sum_j \sum_{k=1}^{m_j} \frac{1}{a_j} |a_j, k\rangle \langle a_j, k|,$$

$$\hat{A}_{approx}^{-1} = \sum_j \sum_{k=1}^{m_j} F(a_j) |a_j, k\rangle \langle a_j, k|,$$

tais que

$$|\psi\rangle = \hat{A}^{-1} |f\rangle = \sum_j \sum_{k=1}^{m_j} \frac{f_{jk}}{a_j} |a_j, k\rangle, \quad (3.35)$$

$$|\tilde{\psi}\rangle = \hat{A}_{approx}^{-1} |f\rangle = \sum_j \sum_{k=1}^{m_j} f_{jk} F(a_j) |a_j, k\rangle. \quad (3.36)$$

Calculando a norma da diferença dos estados, é possível obter

$$\| |\psi\rangle - |\tilde{\psi}\rangle \| = \left\| \sum_j \sum_{k=1}^{m_j} f_{jk} \left(\frac{1}{a_j} - F(a_j) \right) |a_j, k\rangle \right\| \leq \sum_j \sum_{k=1}^{m_j} |f_{jk}| \left| \frac{1}{a_j} - F(a_j) \right|. \quad (3.37)$$

Expandindo $F(a)$ em séries de Taylor com $a \geq \Delta\sqrt{1 + \Delta^2}$, é possível obter

$$\left| \frac{1}{a} - F(a) \right| = \frac{2 + a^2}{2a^3} \Delta^3 + O\left(\frac{\Delta^4}{a^4}\right),$$

onde a relação entre a precisão do algoritmo, a precisão da detecção e a projeção de $|f\rangle$ em $\{|a\rangle\}$ se torna explícita.

3.6 Consequências diretas do modelo prático do algoritmo

Uma equação diferencial parcial linear tem infinitas soluções particulares: Qualquer solução particular somada por uma solução homogênea também se torna uma solução da equação. Portanto, a primeira questão se torna descobrir como a construção da solução pela inversão do operador é feita, e se é possível que o algoritmo retorne diferentes soluções particulares para cada execução. Com esse propósito, analisaremos a solução homogênea da Eq. (3.23):

$$\hat{A}\psi_H(\mathbf{x}) = 0, \quad (3.38)$$

escolhendo o estado $|\psi_H\rangle$ com função de onda $\langle \mathbf{x} | \psi_H \rangle = \psi_H(\mathbf{x})$. Podemos decompor $|\psi_H\rangle$ em $\{|a\rangle\}$ e aplicar \hat{A} , obtendo

$$\hat{A} |\psi_H\rangle = \hat{A} \sum_j \sum_{k=1}^{m_j} \psi_{H_{jk}} |a_j, k\rangle = \sum_j \sum_{k=1}^{m_j} a_j \psi_{H_{jk}} |a_j, k\rangle. \quad (3.39)$$

Por hipótese $\hat{A}|\psi_H\rangle = 0$, portanto:

$$\sum_j \sum_{k=1}^{m_j} a_j \psi_{H_{jk}} |a_j, k\rangle = 0. \quad (3.40)$$

Os estados $|a_j, k\rangle$ são linearmente independentes para diferentes valores de a_j , então temos que $a_j = 0$ para cada a_j com $\langle a_j | \psi_H \rangle \neq 0$, tal que

$$|\psi_H\rangle = \sum_{k=1}^{m_0} \psi_{H_k} |0, k\rangle. \quad (3.41)$$

Analisando a ação do operador \hat{A} em $|\psi_p\rangle$, onde $\langle \mathbf{x} | \psi_p \rangle = \psi_p(\mathbf{x})$ é uma solução particular qualquer do problema, é possível obter

$$|f\rangle = \hat{A}|\psi_p\rangle = \sum_{j \neq 0} \sum_{k=1}^{m_j} a_j \psi_{p_{jk}} |a_j, k\rangle. \quad (3.42)$$

A primeira série é tomada com $j \neq 0$, uma vez que essa parte foi anulada pela ação do operador \hat{A} . Realizando a inversão ideal do operador, temos

$$|\psi\rangle = \hat{A}^{-1}|f\rangle = \sum_l \sum_{i=1}^{m_l} \frac{1}{a_l} |a_l, i\rangle \langle a_l, i| \left[\sum_{j \neq 0} \sum_{k=1}^{m_j} a_j \psi_{p_{jk}} |a_j, k\rangle \right] = \sum_{j \neq 0} \psi_{p_{jk}} |a_j, k\rangle. \quad (3.43)$$

Por ortogonalidade dos autoestados com diferentes autovalores, é possível concluir que o estado $|\psi\rangle$ obtido pelo algoritmo é ortogonal a cada estado do conjunto de soluções homogêneas $\{|\psi_H\rangle\}$ do operador \hat{A} . Portanto, $|\psi\rangle$ não é expansível por soluções homogêneas do operador e qualquer projeção do estado $|\psi\rangle$ em qualquer estado $|\psi_H\rangle$ é nula. Como consequência, o algoritmo nos retornará sempre a mesma solução, uma vez que a diferença entre duas soluções particulares do mesmo problema pode ser escrita como uma projeção não-nula.

Os estados $|s_L\rangle$ podem ser aproximados por uma superposição de estados de Fock, dada por

$$|s_{d,L}\rangle = \frac{1}{\Gamma} \sum_{n=0}^d \gamma_n |n\rangle, \quad \Gamma = \sqrt{\sum_{n=0}^d |\gamma_n|^2},$$

onde $\gamma_n = \langle \psi_n | s_L \rangle$. Podemos representar o algoritmo pelo circuito da Figura 4

Na Figura 4, após a medida realizada nos modos 1 e 2, selecionamos os resultados em que tais modos são detectados em estados com momento nulo, dentro de uma precisão Δ como discutido acima, que projeta então o modo de interesse no estado $|\Phi\rangle = \sum_j \sum_k^{m_{a_j}} f_{j,k} F_d(a_j) |a_j, k\rangle$.

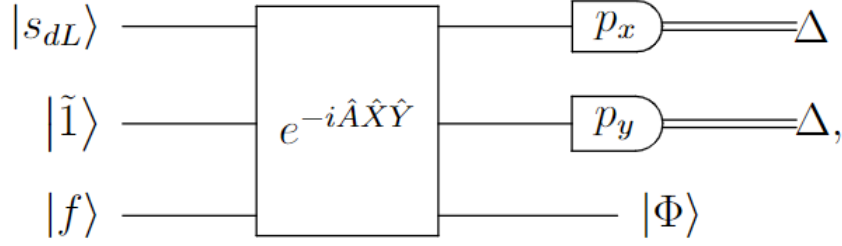


Figura 4 – Representação circuital da forma prática do algoritmo original. Fonte: Elaborada pelo autor.

O novo operador que aproxima \hat{A}^{-1} será chamado de $\hat{A}_{d,L}^{-1}$. A fidelidade do operador $\hat{A}_{d,L}^{-1}$ pode ser calculada ao analisarmos sua ação em um estado $|a\rangle$ e levando-se em conta o efeito da medida $|\Delta\rangle$. Também é possível analisar o limite $\Delta \rightarrow 0$ para caso seja necessário analisar apenas o efeito da expansão em estados de Fock. Para isso, substituímos $|f\rangle$ por $|a\rangle$ no circuito original e obtemos

$$\begin{aligned}
F(a) |\Delta\rangle |\Delta\rangle |a\rangle &= (\mathbb{I} \otimes |\Delta_x\rangle \langle \Delta_x| \otimes |\Delta_y\rangle \langle \Delta_y|) |\Psi\rangle \\
&= \frac{i}{\sqrt{2\pi}} \frac{\Delta}{\sqrt{\pi}} \int_{-\infty}^{\infty} \left(\sum_{k=0}^d N_k \gamma_k \psi_k(x) \right) e^{-x^2 \Delta^2 / 2} \int_{-\infty}^{\infty} N_1 e^{-iaxy} y e^{-y^2 / 2} e^{-y^2 \Delta^2 / 2} dy dx |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\Delta}{\sqrt{\pi}} \sum_{k=0}^d \gamma_k N_k N_1 \int_{-\infty}^{\infty} \psi_k(x) e^{-x^2 \Delta^2 / 2} \left(\frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-iaxy} y e^{-y^2 (1+\Delta^2) / 2} dy \right) dx |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\Delta}{\sqrt{\pi}} \sum_{k=0}^d \frac{\gamma_k N_k N_1}{1 + \Delta^2} \int_{-\infty}^{\infty} \psi_k(x) e^{-x^2 \Delta^2 / 2} \frac{x}{\sqrt{1 + \Delta^2}} e^{-\frac{a^2 x^2}{2(1+\Delta^2)}} dx |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\Delta}{\sqrt{\pi}} \sum_{k=0}^d \frac{\gamma_k N_k N_1}{(1 + \Delta^2)^{3/2}} \int_{-\infty}^{\infty} x H_k(x) e^{-x^2 (1+\Delta^2) / 2} e^{-\frac{a^2 x^2}{2(1+\Delta^2)}} dx |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\Delta}{\sqrt{\pi}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} \frac{\gamma_k N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \int_{-\infty}^{\infty} x H_{2n+1}(x) e^{-x^2 (1+\Delta^2) / 2} e^{-\frac{a^2 x^2}{2(1+\Delta^2)}} dx |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\Delta}{\sqrt{\pi}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \sum_{m=0}^n \frac{(2n+1)! (-1)^{n-m} 2^{2m+1}}{(2m+1)! (n-m)!} \int_{-\infty}^{\infty} x^{2m+2} e^{-x^2 \beta^2} dx |\Delta\rangle |\Delta\rangle |a\rangle
\end{aligned}$$

$$\begin{aligned}
&= 2 \frac{\Delta}{\sqrt{\pi}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \sum_{m=0}^n \frac{(2n+1)! (-1)^{n-m} 2^{2m+1}}{(2m+1)! (n-m)!} \int_0^\infty \beta^{-(2m+3)} u^{m+1/2} e^{-u} du |\Delta\rangle |\Delta\rangle |a\rangle \\
&= 2 \frac{\Delta}{\sqrt{\pi}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \sum_{m=0}^n \frac{(2n+1)! (-1)^{n-m} 2^{2m+1}}{(2m+1)! (n-m)!} \beta^{-(2m+3)} \Gamma(m+3/2) |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \Delta \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \sum_{m=0}^n \frac{(2n+1)! (-1)^{n-m}}{(2m+1)! (n-m)!} \beta^{-(2m+3)} \frac{(2m+2)!}{(m+1)!} |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \Delta \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} (2n+1)! \sum_{m=0}^n \frac{(2m+2) (-1)^{n-m}}{(m+1)! (n-m)!} \beta^{-(2m+3)} |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \Delta \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} a \frac{\gamma_{2n+1} N_{2n+1} N_1}{(1 + \Delta^2)^{3/2}} \frac{(2n+1)!}{n!} \beta^{-3} (\beta^{-2} - 1)^n |\Delta\rangle |\Delta\rangle |a\rangle \\
&= \frac{\sqrt{2} a \Delta}{(a^2 + (1 + \Delta^2)^2)^{3/2}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} \frac{\gamma_{2n+1}}{\sqrt{\pi} \Gamma} h_n(a), \tag{3.44}
\end{aligned}$$

onde

$$h_n(a) = \frac{\sqrt{(2n+1)!}}{2^n} \frac{1}{n!} \left(\frac{1 - a^2 - \Delta^4}{a^2 + (1 + \Delta^2)^2} \right)^n. \tag{3.45}$$

Multiplicando o resultado final, Eq. (3.44), pela constante de convergência λ da Eq. (3.33), obtemos

$$F(a) = 2\sqrt{L\sqrt{\pi}} \frac{a}{(1 + (1 + \Delta^2)^2)^{3/2}} \sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} \gamma_{2n+1} \frac{\sqrt{(2n+1)!}}{2^n} \frac{1}{n!} \left(\frac{1 - a^2 - \Delta^4}{a^2 + (1 + \Delta^2)^2} \right)^n. \tag{3.46}$$

No entanto, quando $|a| > (1 + \Delta^2)$, $F(a)$ se comporta como

$$F(a) = \sum_{k=0}^{\lfloor \frac{d}{2} \rfloor} \sum_{m=0}^{\lfloor \frac{k}{2} \rfloor} \gamma_{2k+1} \frac{a}{|a|^{2m+3}} (2(1 + \Delta^2))^m \Gamma(m+3/2) \left(1 - (m+3/2) \left(\frac{1 + \Delta^2}{a} \right)^2 + O\left(\left(\frac{1 + \Delta^2}{a} \right)^4 \right) \right), \tag{3.47}$$

que significa que o comportamento de $F(a)$ é dado por

$$F(a) \sim \frac{a}{|a|^3}. \tag{3.48}$$

Isto nos leva imediatamente a questionar a fidelidade do algoritmo ao usarmos estados de Fock para aproximar a função degrau. A principal dúvida quanto a essa aproximação se torna: Se não há convergência para $a \gg 1$ ou $a \approx 0$, qual a região onde há alta fidelidade do algoritmo?

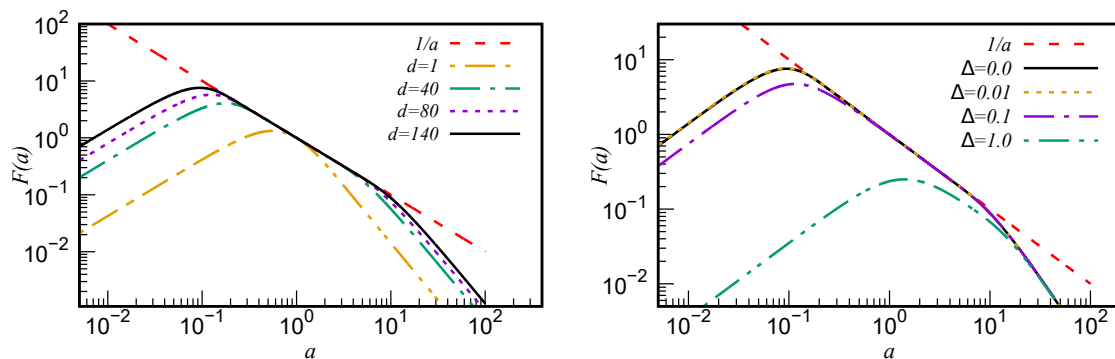


Figura 5 – Esquerda: Gráfico dílog de $F(a)$ em função de a com $\Delta = 0$ para diversos valores da dimensão (d) comparados à curva ideal $1/a$. Direita: Comportamento da função $F(a)$ em função de a para diversos valores da precisão na detecção do momento, gerados ao utilizar uma superposição com $d = 140$ estados Fock. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.

As Figuras 5 mostram como $F(a)$ converge conforme o número de estados de Fock aumenta, nos indicando que para aplicação do algoritmo em um computador real isso poderia ser um problema, uma vez que superposições de muitos estados de Fock são muito difíceis, mesmo que atualmente hajam alternativas, como utilizar redes neurais e aprendizado de máquina para a criação de tais estados iniciais (ARRAZOLA et al., 2019; KILLORAN et al., 2019). É possível trabalhar neste problema ao custo de tornarmos o algoritmo modificado válido apenas para uma classe de operadores. Uma primeira proposta de modificação que recuperaria o aspecto ideal para valores de a grandes é substituir o estado $|f\rangle$ por $\hat{A}|f\rangle$. Com isso o comportamento de $F(a)$ descrito em (3.47) se torna

$$F(a) = \sum_{k=0}^{\lfloor \frac{d}{2} \rfloor} \sum_{m=0}^{\lfloor \frac{k}{2} \rfloor} \gamma_{2k+1} \frac{1}{|a|^{2m+1}} (2(1+\Delta^2))^m \Gamma(m+3/2) \left(1 - (m+3/2) \left(\frac{1+\Delta^2}{a} \right)^2 + O\left(\left(\frac{1+\Delta^2}{a} \right)^4 \right) \right), \quad (3.49)$$

que tem o comportamento assintótico

$$F(a) \sim \frac{1}{|a|}, \quad (3.50)$$

quando multiplicado por uma constante λ tal que

$$\lambda \lim_{a \rightarrow \infty} |a| F(a) = 1. \quad (3.51)$$

Como discutido acima, o algoritmo proposto por Arrazola *et al.* é capaz de inverter o operador diferencial \hat{A} quando se tem uma função degrau ideal e medida ideal dos estados de momento dos modos 1 e 2. No entanto, quando levamos em conta que tanto a representação da função degrau em termos de estados de Fock quanto a medida nos modos 1 e 2 são sempre imperfeitas, o algoritmo é incapaz de inverter o operador \hat{A} com boa precisão. Em um primeiro momento, notamos que uma simples modificação no estado $|f\rangle$ já nos permitiria obter uma inversão do operador \hat{A} mesmo quando temos uma função degrau aproximada e quando as medidas sobre os modos 1 e 2 não são perfeitas. Isso nos motivou a investigarmos quais modificações adicionais no algoritmo poderiam ser introduzidas de modo a simplificá-lo e/ou melhorá-lo. Felizmente conseguimos obter algumas alternativas, que serão expostas no próximo capítulo.

4 Portas lógicas quânticas

4.1 Hamiltoniano de interação

Utilizando da representação de interação (SAKURAI; NAPOLITANO, 2014), podemos desenvolver o estudo da interação radiação-matéria, sendo possível aplicar a aproximação de onda girante, que será explicada à frente.

Na representação de interação, assumimos o hamiltoniano livre $\hat{H}_0 = \hat{H}_a + \hat{H}_m$, de maneira que seja possível calcular o hamiltoniano de interação

$$\hat{H} = e^{i\hat{H}_0 t/\hbar} \hat{V}_I e^{-i\hat{H}_0 t/\hbar} = \hbar \frac{\Omega_0}{2} e^{i\hat{H}_a t} (\hat{\sigma}_+ + \hat{\sigma}_-) e^{-i\hat{H}_a t} e^{i\hat{H}_m t} \left(e^{i(\eta(\hat{a} + \hat{a}^\dagger) - \omega t + \phi)} + h.c. \right) e^{-i\hat{H}_m t}.$$

Sabendo as relações de comutação matrizes de Pauli e também dos operadores de criação e aniquilação, podemos utilizar séries de Taylor e a fórmula de Baker-Campbell-Hausdorff (SAKURAI; NAPOLITANO, 2014) para obter

$$\hat{H} = \hbar \frac{\Omega_0}{2} (\hat{\sigma}_+ e^{-i\delta t} e^{i\hat{\gamma}} e^{i\phi} + \hat{\sigma}_- e^{i\bar{\omega} t} e^{i\hat{\gamma}} e^{i\phi} + h.c.), \quad (4.1)$$

onde o operador $\hat{\gamma} = \eta (\hat{a} e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})$ e a constante $\bar{\omega} = \omega + \omega_0$ foram introduzidos de maneira a simplificar a notação. Os termos cuja dependência temporal são expressidos por exponenciais de $\bar{\omega}$ ou ω_0 podem ser descartados uma vez que, na média, eles não contribuem para a dinâmica do sistema quando comparados aos termos cuja dependência temporal é expressa pela exponencial cuja frequência é $\delta = \omega - \omega_0 \ll \bar{\omega}, \omega_0$. Ao fazermos esta aproximação, chamada de aproximação da onda girante (AOG), \hat{H} é simplificado para (LEIBFRIED et al., 2007)

$$\hat{H} = \frac{\hbar\Omega_0}{2} \left\{ \hat{\sigma}_+ e^{i(\hat{\gamma} + \phi)} e^{-i\delta t} + \hat{\sigma}_- e^{-i(\hat{\gamma}^\dagger + \phi)} e^{i\delta t} \right\}. \quad (4.2)$$

4.2 Interações elementares e portas quânticas

O fator de Lamb-Dicke η quantifica a amplitude de oscilação do íon na armadilha (LEIBFRIED et al., 2007) tornando-se um parâmetro muito importante para a análise do sistema. Se $\eta \ll 1$, dizemos que o sistema está no regime de Lamb-Dicke e, podemos fazer as aproximações

$$e^{i\hat{\gamma}} \approx 1 + i\eta (\hat{a} e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t}), \quad (4.3a)$$

$$e^{i\hat{\gamma}^\dagger} \approx 1 - i\eta (\hat{a}^\dagger e^{i\nu t} + \hat{a} e^{-i\nu t}). \quad (4.3b)$$

Portanto, o hamiltoniano de interação no regime de Lamb-Dicke se torna

$$\hat{H}_{LD} = \frac{\hbar\Omega_0}{2} \left\{ \hat{\sigma}_+ (1 + i\eta (\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t})) e^{i(\phi - \delta t)} + h.c. \right\}. \quad (4.4)$$

Como η quantifica as oscilações atômicas relativas ao comprimento de onda do campo, se $\eta \approx 0$, o vetor de onda se torna praticamente perpendicular à direção de vibração do íon, de maneira que o hamiltoniano de interação se torna

$$\hat{H}_{LD} = \frac{\hbar\Omega_0}{2} \left\{ \hat{\sigma}_+ e^{i(\phi - \delta t)} + h.c. \right\}, \quad (4.5)$$

o que significa que, neste caso, o íon é excitado sem acoplar os estados eletrônico e vibracional.

O regime de Lamb-Dicke é importante, uma vez que podemos explorar novos hamiltonianos que acoplam interações entre os estados internos e externos do íon ao ajustar a dessintonia $\delta = \omega - \omega_0$. É importante notar que as frequências do campo devem ser ressonantes ou quase ressonantes à frequência de transição interna. Isso é feito uma vez que a interação acopla os níveis externos aos internos com dessintonias específicas. Se a frequência do campo é ressonante com a frequência de transição, então $\delta = 0$ e

$$\hat{H}_{\delta=0} = \frac{\hbar\Omega_0}{2} (\hat{\sigma}_+ e^{i\phi} + \hat{\sigma}_- e^{-i\phi}), \quad (4.6)$$

onde a aproximação da onda girante foi aplicada. Eq. (4.6) significa que a interação do campo em ressonância com a frequência de transição do estado de dois níveis induz a transição entre eles sem acoplar os estados vibracionais. Esta é a chamada interação de campo ressonante.

Ajustando-se $\delta = -\nu$ e aplicando a aproximação da onda girante novamente na Eq. (4.4), temos

$$\hat{H}_{rsb} = \frac{\hbar\Omega_0}{2} \eta (\hat{a}\sigma_+ e^{i\phi} + \hat{a}^\dagger \sigma_- e^{-i\phi}), \quad (4.7)$$

chamado de hamiltoniano de Jaynes-Cummings. Este hamiltoniano permite excitações no estado eletrônico do íon ao remover um fônon no modo vibracional.

Ajustando-se $\delta = \nu$ e aplicando a aproximação da onda girante na Eq. (4.4), temos

$$\hat{H}_{bsb} = \frac{\hbar\Omega_0}{2} \eta (\hat{a}^\dagger \hat{\sigma}_+ e^{i\phi} + \hat{a} \hat{\sigma}_- e^{-i\phi}) \quad (4.8)$$

chamado de hamiltoniano de anti-Jaynes-Cummings.

4.2.1 Íons aprisionados em potenciais bi-dimensionais - Interação entre modos

De maneira análoga ao caso de um íon aprisionado em uma dimensão, o hamiltoniano do sistema 2D na representação de Schrödinger é dado por

$$\hat{H} = \hbar\nu_a \hat{a}^\dagger \hat{a} + \hbar\nu_b \hat{b}^\dagger \hat{b} + \frac{\hbar\omega_0}{2} \hat{\sigma}_z + \frac{\hbar\Omega_0}{2} (\hat{\sigma}_+ + h.c.) (e^{i(\mathbf{k}\cdot\mathbf{r}-\omega t+\phi)} + h.c.). \quad (4.9)$$

Na representação de interação, temos

$$\hat{H}_{int}(t) = \frac{\hbar}{2} \Omega_0 \hat{\sigma}_+ \exp\{i\eta_a(\hat{a}e^{-i\nu_a t} + \hat{a}^\dagger e^{i\nu_a t})\} \exp\{i\eta_b(\hat{b}e^{-i\nu_b t} + \hat{b}^\dagger e^{i\nu_b t})\} e^{-i(\delta t - \phi)} + h.c..$$

Podemos obter, após aplicações da aproximação da onda girante, operações controladas entre modos. Em particular, podemos obter a operação controlada entre modos indicada no conjunto universal da Eq. (3.3) ao escolher um laser tetracromático com dessintonias $\delta = \pm(\nu_a + \nu_b)$, fase $\phi = 0$ e no regime de Lamb-Dicke, mantendo termos até ordem η^2 , obtemos o hamiltoniano

$$\hat{H}_{\hat{X}_a \otimes \hat{X}_b} = \frac{\hbar\Omega\eta_a\eta_b}{2} \hat{\sigma}_x (\hat{X}_a \hat{X}_b). \quad (4.10)$$

O operador de evolução temporal associado é da forma

$$\hat{U}_{\hat{X}_a \otimes \hat{X}_b} = \exp\{it' \hat{X}_a \hat{X}_b\}, \quad (4.11)$$

com $t' = t \frac{\Omega\eta_a\eta_b}{2}$. É possível simular a dinâmica de um íon aprisionado utilizando a biblioteca QuTiP (JOHANSSON; NATION; NORI, 2012; JOHANSSON; NATION; NORI, 2013), comparando com a dinâmica esperada à interação ideal.

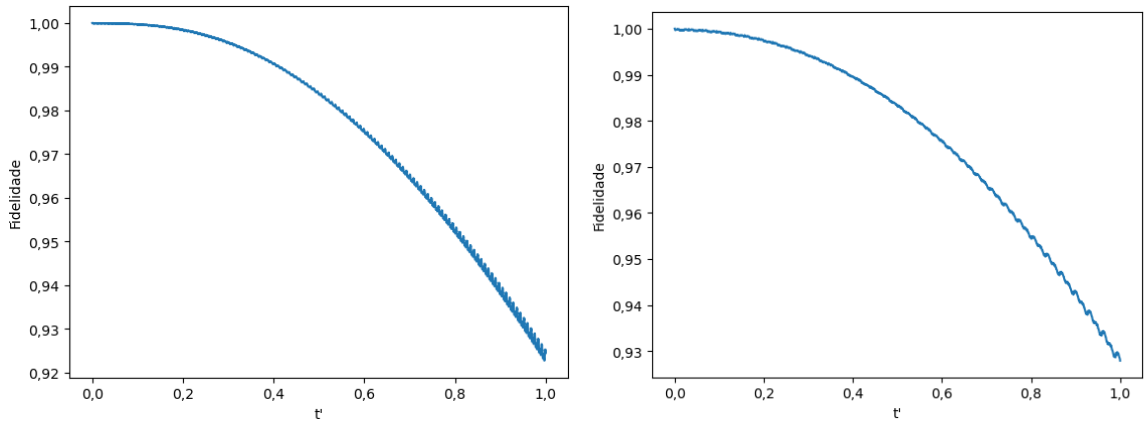


Figura 6 – Fidelidade em função do tempo efetivo $t' = t \frac{\Omega\eta_a\eta_b}{2}$ da evolução temporal exata da porta entre modos $\hat{U}_{\hat{X}_a \otimes \hat{X}_b} = \exp\{X_a \otimes X_b\}$ em comparação ao hamiltoniano sem aproximações, Eq. (4.9), considerando casos com $\eta = 0, 1$ (esquerda) e $\eta = 0, 2$ (direita). O programa escrito para gerar as curvas de fidelidade pode ser visto no Apêndice C. Fonte: Elaborada pelo autor.

É possível observar que a fidelidade dessa porta, definida como sendo a norma da diferença entre o estado final da evolução do operador ideal dado pela Eq. (4.11) e do estado final após a interação sem aproximações gerada pela Eq. de Schrödinger (4.9) para um mesmo estado inicial, decai rapidamente com o tempo. Isso se deve ao fato de que o acoplamento efetivo depende do produto $\eta_a\eta_b$, de ordem menor ou igual a 1.0×10^{-2} . Para portas com acoplamento efetivo da ordem de η_a , por exemplo, o acoplamento se torna mais forte, permitindo fidelidade alta por mais tempo.

4.2.2 A porta $e^{it'\hat{X}}$ e o operador deslocamento

Ajustando-se $\delta = \pm\nu$, mantendo-se termos até ordem η , e aplicando-se a aproximação de onda girante podemos obter os seguintes hamiltonianos

$$\hat{H}_{\delta=-\nu} = \frac{\hbar\Omega_0}{2}\eta (\hat{a}\hat{\sigma}_+e^{i\phi_+} + \hat{a}^\dagger\hat{\sigma}_-e^{-i\phi_+}), \quad (4.12a)$$

$$\hat{H}_{\delta=\nu} = \frac{\hbar\Omega_0}{2}\eta (\hat{a}^\dagger\hat{\sigma}_+e^{i\phi_-} + \hat{a}\hat{\sigma}_-e^{-i\phi_-}). \quad (4.12b)$$

Nestes casos, os estados vibracional e interno são acoplados. O primeiro hamiltoniano é conhecido como o primeiro desvio para o vermelho e produz interações equivalentes ao hamiltoniano de Jaynes-Cummings: o estado interno é excitado ao custo de um fônon do estado vibracional ou o estado interno é de-excitado e um fônon é criado no estado vibracional. Já o segundo hamiltoniano é conhecido como o primeiro desvio para o azul. Nesta interação, tanto o estado interno quanto o externo do íon é excitado ou de-excitado simultaneamente. Em particular, dois casos são de maior interesse: O primeiro caso é a interação entre um laser bicromático com frequências $f = \pm\nu$ e fases relativas $\phi_+ = \phi_- + 2\phi$, resultando no hamiltoniano

$$\hat{H} = \frac{\hbar\Omega\eta}{2}(\hat{\sigma}_+\hat{a}e^{i\phi_+} + \hat{\sigma}_-\hat{a}^\dagger e^{-i\phi_+} + \hat{\sigma}_+\hat{a}^\dagger e^{i\phi_-} + \hat{\sigma}_-\hat{a}e^{-i\phi_-}) = \frac{\hbar\Omega\eta}{2}\hat{\sigma}_{\phi_+\phi_-}(\hat{a}e^{i\phi} + \hat{a}^\dagger e^{-i\phi}), \quad (4.13)$$

onde $\hat{\sigma}_\phi = \sigma_+e^{i\phi} + \sigma_-e^{-i\phi}$. Ao escolhermos $\phi_+ + \phi_- = \varphi - \pi/2$, obtemos

$$\hat{H} = i\frac{\hbar\Omega\eta}{2}(\hat{a}^\dagger e^{-i\phi} - \hat{a}e^{i\phi})\hat{\sigma}_{\phi-\pi/2}, \quad (4.14)$$

cujos operador de evolução temporal pode ser escrito como

$$\hat{D}(\alpha) = \exp\{\hat{\sigma}_{\phi-\pi/2}(\alpha\hat{a}^\dagger - \alpha^*\hat{a})\}, \quad (4.15)$$

onde $\alpha(t, \phi) = \frac{\Omega\eta t}{2}e^{i(\phi-\frac{\pi}{2})}$ é um número complexo. O operador descrito na Eq. (4.15) é chamado de Operador Deslocamento quando o estado interno do íon está no estado $|+\rangle_{\phi-\pi/2}$, descrito como sendo o autoestado do operador $\hat{\sigma}_{\phi-\pi/2}$ com autovalor positivo.

O segundo caso de interesse é a interação de um laser bicromático com frequências $f = \pm\nu$ e fase relativa $\phi = -\pi/2$ durante um tempo t , que resulta no seguinte operador de evolução

$$\hat{U}_{\hat{X}} = \exp\left\{it\frac{\Omega}{2}\eta\hat{\sigma}_x\hat{X}\right\}, \quad (4.16)$$

onde $\hat{X} = (\hat{a} + \hat{a}^\dagger)$, que pode ser visto como o primeiro elemento do conjunto universal (3.3), quando o estado interno do íon está no estado $|+\rangle = \frac{1}{\sqrt{2}}(|g\rangle + |e\rangle)$.

O operador de evolução temporal $\hat{U}_{\hat{X}}$ e suas formas aproximadas pela interação laser-íon podem ser simuladas, permitindo então calcular a fidelidade de um estado calculado via dinâmica completa, sem aproximações, e um estado calculado usando-se o operador U_x . A Figura 4.16, mostra o comportamento da fidelidade em função do tempo efetivo de evolução para o caso $\eta = 0,01$. É possível observar que a amplitude de oscilação da fidelidade é similar ao caso da Figura 6. No entanto, uma vez que o acoplamento entre o estado eletrônico e vibracional é linear com η e utiliza-se um laser bicromático, o comportamento da fidelidade em função do tempo é melhorado.

4.2.3 A porta de Fourier

A porta de Fourier

$$\hat{F} = e^{i\frac{\pi}{2}(\hat{X}^2 + \hat{P}^2)}, \quad (4.17)$$

também pode ser realizada ao expandirmos as Eqs. (4.3a) e (4.3b) até segunda ordem em η :

$$e^{i\hat{\gamma}} \approx 1 + i\eta(\hat{a}e^{-i\nu t} + \hat{a}^\dagger e^{i\nu t}) - \frac{\eta^2}{2}(\hat{a}^{2\dagger}e^{2i\nu t} + \hat{a}^2e^{-2i\nu t} + \hat{a}^\dagger\hat{a} + \hat{a}\hat{a}^\dagger), \quad (4.18a)$$

$$e^{i\hat{\gamma}^\dagger} \approx 1 - i\eta(\hat{a}^\dagger e^{i\nu t} + \hat{a}e^{-i\nu t}) - \frac{\eta^2}{2}(\hat{a}^{2\dagger}e^{2i\nu t} + \hat{a}^2e^{-2i\nu t} + \hat{a}^\dagger\hat{a} + \hat{a}\hat{a}^\dagger). \quad (4.18b)$$

Escolhendo a dessintonia $\delta = 0$ com fase relativa $\phi = \pi$, o hamiltoniano de interação entre o campo e o íon se torna, após aplicação da aproximação da onda girante

$$\hat{H} = -\hbar\frac{\Omega_0}{2}\sigma_x\hat{I} + \hbar\frac{\Omega_0\eta^2}{8}\sigma_x(\hat{X}^2 + \hat{P}^2). \quad (4.19)$$

Resultando no operador de evolução temporal

$$\hat{U}_{\hat{F}} = e^{-it\frac{\Omega_0}{2}\sigma_x\hat{I}}e^{it\frac{\Omega_0\eta^2}{8}\sigma_x(\hat{X}^2 + \hat{P}^2)} \quad (4.20)$$

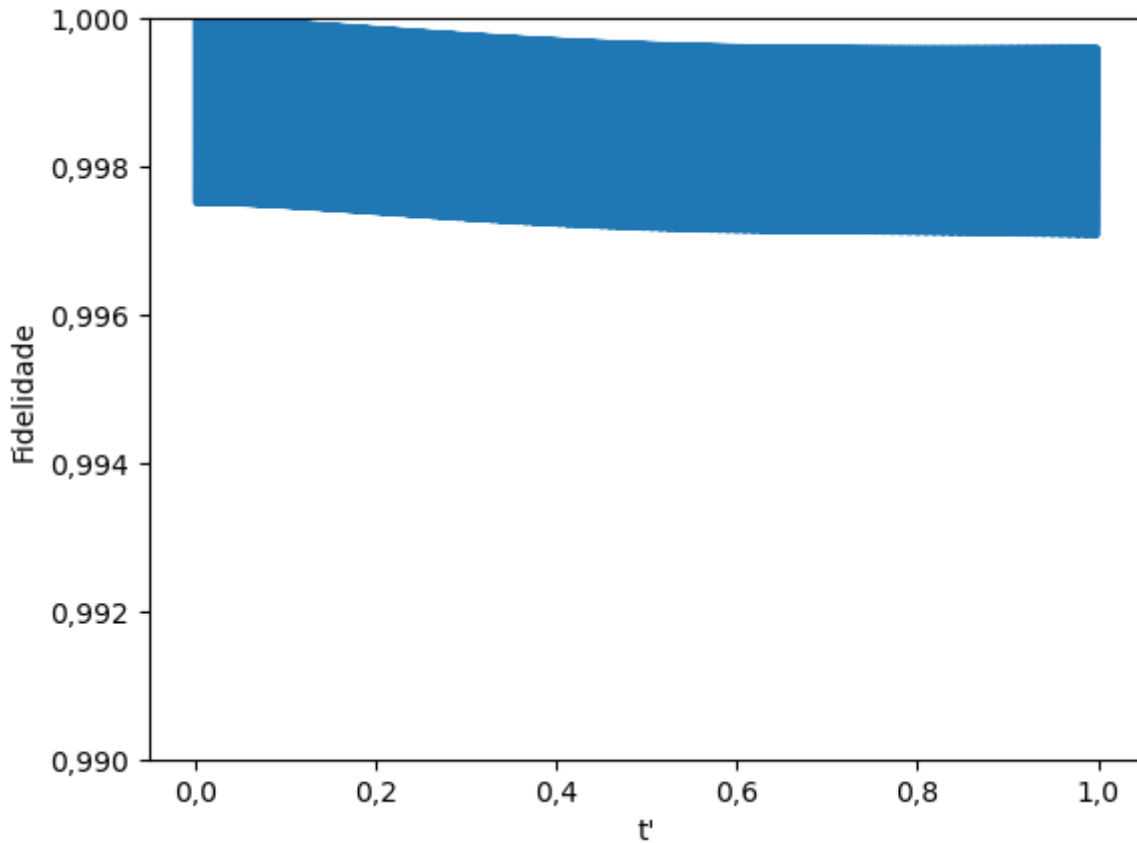


Figura 7 – Fidelidade em função do tempo da evolução temporal exata da porta entre modos $H = X$ em comparação ao hamiltoniano sem aproximações Eq. (4.9), considerando o caso com $\eta = 0,01$. O programa escrito para gerar as curvas de fidelidade pode ser visto no Apêndice C. Fonte: Elaborada pelo autor.

Pela Eq. (4.20), é possível observar que o primeiro termo do hamiltoniano na Eq. (4.19) gera apenas uma fase global quando o estado interno do íon é preparado no autoestado $|+\rangle$. Nesse caso, o estado interno permanece sempre no estado $|+\rangle$ e a dinâmica do estado vibracional fica então descrita pelo operador

$$\hat{U}_{\hat{F}_{eff}} = e^{it'(\hat{X}^2 + \hat{P}^2)}. \quad (4.21)$$

Ajustando $t = \frac{\pi}{2} \frac{8}{\Omega_0 \eta^2}$, obtemos a porta de Fourier

$$\hat{F} = e^{i\frac{\pi}{2}(\hat{X}^2 + \hat{P}^2)}. \quad (4.22)$$

4.2.4 A porta \hat{X}^2 e o operador de compressão

Ao escolhermos $\delta = \pm 2\nu$ com fases ϕ_+ e ϕ_- , expandindo as Eqs. (4.18) e aplicando a aproximação de onda girante na Eq. (4.2), obtemos

$$\hat{H}_{2\nu} = -\frac{\hbar\Omega\eta^2}{4}(\hat{\sigma}_+\hat{a}^2e^{i\phi_+} + \hat{\sigma}_-\hat{a}^{\dagger 2}e^{-i\phi_+}), \quad (4.23a)$$

$$\hat{H}_{-2\nu} = -\frac{\hbar\Omega\eta^2}{4}(\hat{\sigma}_+\hat{a}^{\dagger 2}e^{i\phi_-} + \hat{\sigma}_-\hat{a}^2e^{-i\phi_-}). \quad (4.23b)$$

Dessa maneira, a interação entre um laser bicromático de frequências cujas dessintonias sejam $\delta = \pm 2\nu$ e fases $\phi_+ + \phi_- = 2\phi$ e $\phi_- = -\pi/2$ gera hamiltonianos do tipo

$$\hat{H}_{\pm 2\nu} = \frac{i\hbar\Omega\eta^2}{4}\hat{\sigma}_\phi(e^{-i\phi}\hat{a}^2 - e^{i\phi}\hat{a}^{\dagger 2}), \quad (4.24)$$

de maneira que o operador de evolução temporal se torne

$$\hat{S}(\beta) = \exp\left\{\hat{\sigma}_\phi\left[\frac{1}{2}(\beta\hat{a}^2 - \beta^*\hat{a}^{\dagger 2})\right]\right\}, \quad (4.25)$$

onde $\beta(\phi, t) = \frac{\Omega\eta^2 t}{4}e^{-i\phi}$. Quando o estado interno do íon é preparado em um autoestado do operador $\hat{\sigma}_\phi$, o operador $S(\beta)$ reduz-se exatamente a um operador de compressão.

Ao escolhermos um laser tricromático, com dessintonias $\delta = \pm 2\nu, 0$ e fases $\phi_0 = \phi_+ = \phi_- = \pi$, obtemos o hamiltoniano

$$\hat{H} = -\frac{\hbar\Omega}{2}\hat{\sigma}_x\hat{I} + \frac{\hbar\Omega\eta^2}{4}\hat{\sigma}_x\hat{X}^2. \quad (4.26)$$

O primeiro termo, no entanto, modifica apenas os níveis internos do íon, os quais não estamos interessados nesse contexto, o que nos permite desconsiderar esse termo caso o íon esteja preparado com estado interno $|+\rangle$, isto é, o autoestado de $\hat{\sigma}_x$ com autovalor 1. Dessa maneira, o operador evolução temporal associado ao hamiltoniano dado pela Eq. (4.26) torna-se

$$\hat{U}_{\hat{X}^2} = \exp\{-i\hat{\sigma}_x\hat{X}^2t'\}. \quad (4.27)$$

onde $t' = \frac{\Omega\eta^2}{4}t$.

4.2.5 A porta \hat{X}^3

Escolhendo $\delta = \pm 3\nu$, com respectivas fases φ, θ , e levando-se em conta termos até a ordem η^3 , são geradas as interações

$$\hat{H} = -\frac{i\hbar\Omega\eta^3}{12} (\hat{\sigma}_+ \hat{a}^{\dagger 3} e^{i\varphi} - \hat{\sigma}_- \hat{a}^3 e^{-i\varphi}), \quad (4.28a)$$

$$\hat{H} = -\frac{i\hbar\Omega\eta^3}{12} (\hat{\sigma}_+ \hat{a}^3 e^{i\theta} - \hat{\sigma}_- \hat{a}^{\dagger 3} e^{-i\theta}), \quad (4.28b)$$

de maneira que a interação entre um laser tetracromático e um íon aprisionado com frequências $\delta = \pm\nu, \pm 3\nu$ e fases relativas $\phi_{\pm} = \phi_{\pm} = \varphi = \theta = \pi/2$ é descrita pelo hamiltoniano

$$\hat{H} = -\frac{\hbar\Omega\eta}{2} \hat{X} \hat{\sigma}_x + \frac{\hbar\Omega\eta^3}{12} \hat{X}^3 \hat{\sigma}_x. \quad (4.29)$$

A interação que estamos procurando é dada somente pelo segundo termo do lado direito da equação acima. Desse modo, precisamos encontrar uma forma de eliminar o primeiro termo do lado direito da Eq. (4.29). Felizmente isso é possível pois, como demonstramos acima, com a adição de um novo laser, com parâmetro de Lamb-Dicke η' , muito menor que η de modo a mantermos apenas termos proporcionais a η' , conseguimos derivar um hamiltoniano igual, mas com sinal trocado, ao primeiro termo do lado direito da Eq. (4.29). Com isso, ficamos com o hamiltoniano simplificado

$$\hat{H} = \frac{\hbar\Omega\eta^3}{12} \hat{X}^3 \hat{\sigma}_x. \quad (4.30)$$

4.2.6 Interação Kerr

A interação Kerr, assim como a porta \hat{X}^3 , é uma interação não-gaussiana - um recurso muito importante para computação quântica em variáveis contínuas, como discutido anteriormente - de maneira que sua aplicação em sistemas de variável contínua seja um importante passo para a computação universal. Resumidamente, podemos descrever a interação Kerr como sendo

$$\hat{K} = \exp\{it\hat{a}^{\dagger 2}\hat{a}^2\}. \quad (4.31)$$

Para conseguir simular numericamente como gerar tal interação, Stobínska, Villar e Leuchs (STOBIŃSKA; VILLAR; LEUCHS, 2011) simularam a interação ressonante Eq.

(4.6) com fase nula até quarta ordem. Dessa maneira, o hamiltoniano da interação pode ser escrito como

$$\hat{H}_{Kerr} = \hbar \frac{\Omega}{2} \left[\left(1 - \frac{\eta^2}{2} + \frac{\eta^4}{8} \right) \hat{I} + \left(\frac{\eta^4}{2} - \eta^2 \right) \hat{a}^\dagger \hat{a} + \frac{\eta^4}{2} \hat{a}^{\dagger 2} \hat{a}^2 \right] \hat{\sigma}_x. \quad (4.32)$$

Considerando que o nível interno do íon esteja preparado no autoestado $|+\rangle$ do operador $\hat{\sigma}_x$, o operador de evolução associado ao operador descrito pela Eq. (4.32) torna-se

$$\hat{U}_{Kerr} = e^{-i \frac{\Omega t}{2} \vartheta(\eta) \hat{I}} e^{i \frac{\Omega}{2} \eta^2 t (1 - \eta^2/2) \hat{a}^\dagger \hat{a}} e^{-i \frac{t \Omega \eta^4}{4} \hat{a}^{\dagger 2} \hat{a}^2}, \quad (4.33)$$

onde $\vartheta(\eta) = 1 - \eta^2/2 + \eta^4/8$. O primeiro termo da Eq. (4.33) corresponde a uma fase global, que pode ser desconsiderada na dinâmica efetiva. O segundo termo, relacionado à porta de Fourier, pode ser considerado uma rotação no espaço de fase. Dessa maneira, ao considerarmos um referencial giratório no espaço de fase, a dinâmica efetiva pode ser descrita pelo operador de evolução temporal

$$\hat{K} = \exp\{-it' \hat{a}^{\dagger 2} \hat{a}^2\}, \quad (4.34)$$

com $t' = t \Omega \frac{\eta^4}{4}$.

4.2.7 Portas lógicas em variáveis contínuas

Obtivemos então, para um íon aprisionado, diferentes operações sobre variáveis contínuas que podem ser classificadas de acordo com a Tabela 1.

Tabela 1 – Conjunto de portas em variáveis contínuas apresentado nesta seção. As operações relativas ao conjunto universal Eq. (3.3) foram desenvolvidas pelo autor, com exceção da porta de Fourier. As outras operações podem ser encontradas também em (ORTIZ-GUTIÉRREZ et al., 2017; STOBÍŃSKA; VILLAR; LEUCHS, 2011).

Portas Gaussianas	Operações controladas lineares	$X1 \otimes X2$
	Interações lineares	X
		$\alpha \hat{a}^\dagger - \alpha^* \hat{a}$ (Deslocamento)
	Rotações	$\frac{\pi}{2} (\hat{X}^2 + \hat{P}^2)$ (Fourier)
Interações não-lineares	X^2	
	$\beta \hat{a}^2 - \beta^* \hat{a}^{\dagger 2}$ (Compressão)	
Portas não-Gaussianas	Interações de terceira ordem	X^3
	Interações de quarta ordem	$\hat{a}^{\dagger 2} \hat{a}^2$ (Kerr)

Podemos identificar que todas as operações do conjunto universal apresentado na Eq. (3.3) foram desenvolvidas durante o capítulo. Além disso as operações Gaussianas em

um modo apresentadas neste conjunto universal podem ser decompostas nos operadores de Deslocamento e Compressão, além de rotações no espaço de fase que são portas de Fourier com diferentes tempos de evolução (KALAJDZIEVSKI; QUESADA, 2021), todas apresentados durante esta seção. Dessa maneira, em união com as portas não-Gaussianas é possível realizar diferentes decomposições e conjuntos universais. Em particular, o conjunto dado pela Eq. (3.3) garante vantagem ao decompor sem aproximações operadores do tipo apresentados na Eq. (3.2), que são operadores de interesse para o algoritmo estudado neste trabalho (KALAJDZIEVSKI; ARRAZOLA, 2019).

5 Propostas de modificações

A convergência lenta do algoritmo em função do tamanho da superposição e as limitações quando se leva em conta a imprecisão da medida simultaneamente com a aproximação da função degrau nos levaram a procurar alternativas como a introduzida na seção anterior. Conseguimos então elaborar duas propostas para melhorar o algoritmo proposto por Arrazola *et al.*, cujos detalhes e consequências serão discutidos nas próximas seções.

5.1 Proposta 1

A Eq. (3.49) nos dá o comportamento completo de $F(a)$ para $a > 1 + \Delta^2$, no entanto $F(a)$ acaba sendo insensível à variação do sinal de a . Dessa maneira a proposta de modificação se torna uma boa abordagem para operadores positivo (negativo)-semidefinidos. Além disso, após multiplicarmos a Eq. (3.49) pelo pré-fator da Eq. (3.51), podemos expandir a Eq. (3.49) em séries de potências, de maneira a obter

$$F(a) = \frac{1}{|a|} \left(1 + \frac{\beta_1}{a^2} + \frac{\beta_2}{a^4} + \dots \right). \quad (5.1)$$

Assim, é possível eliminar os termos de erro de ordem a^{-2}, a^{-4}, \dots na expansão da Eq. (5.1) ao alterar apropriadamente os coeficientes da superposição $|s\rangle$. Dessa maneira, o estado $|s\rangle$ agora tem a forma

$$|s_M\rangle = \sum_{n=0}^M \tilde{\gamma}_{2n+1} |2n+1\rangle. \quad (5.2)$$

Para o caso $|s_1\rangle = \tilde{\gamma}_1 |1\rangle + \tilde{\gamma}_3 |3\rangle$, é possível obter analiticamente os coeficientes de superposição

$$\tilde{\gamma}_1 = \frac{\tilde{\gamma}_3}{\sqrt{6}} \left(3 + \frac{4}{1 + \Delta^2} \right), \quad (5.3)$$

$\tilde{\gamma}_3$ pode ser obtido através da normalização do estado $|s\rangle$.

A Figura 8 mostra os diferentes comportamentos da curva gerada pela proposta modificada do algoritmo, mostrando que para $a > 1 + \Delta^2$ o comportamento da curva converge rapidamente para o resultado ideal.

Por outro lado, também é possível ver que, ao definirmos $\Delta = 1,0$ na medida projetiva dos estados auxiliares, há perda consideravelmente menor de fidelidade relativa

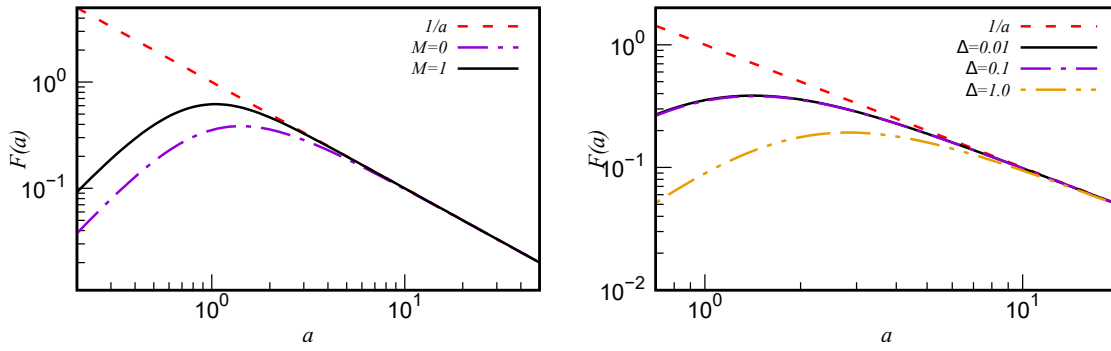


Figura 8 – Esquerda: Gráfico dílog de $F(a)$ em função de a considerando $\Delta = 0,01$, da primeira proposta do algoritmo em comparação com a curva ideal $1/a$. Direita: Gráfico dílog $F(a)$ em função de a considerando $M = 0$, da primeira proposta do algoritmo em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.

ao resultado da proposta prática do algoritmo originalmente desenvolvida. O caso $\Delta = 1,0$, em particular, é muito importante uma vez que pode representar uma medida projetiva no estado de vácuo, resultando em algo mais simples de se realizar em um sistema quântico.

Ao comparar as Figuras 5 e 8, percebe-se que o comportamento para $a < 1$ é muito mais apurado na proposta original para valores de d suficientemente grandes. Mesmo ao aumentarmos a dimensão da superposição na primeira proposta, o resultado apenas ficará mais refinado para $a > 1 + \Delta^2$. Este problema pode ser contornado ao modificarmos o tempo de evolução t do hamiltoniano no algoritmo. Essa modificação pode ser representada pelo circuito da Figura 9.

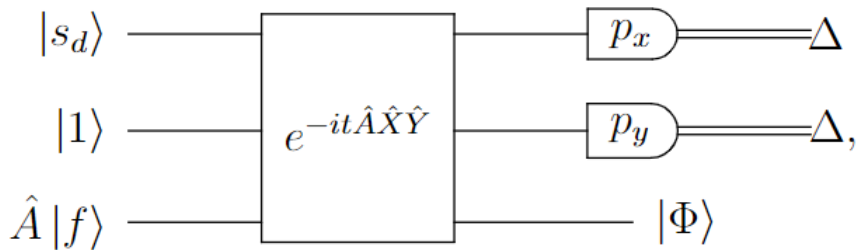


Figura 9 – Representação circuitual do algoritmo modificado. Fonte: Elaborada pelo autor.

Dessa maneira, ao fazermos $\lambda \rightarrow t\lambda$, obtemos para $d = 1$

$$F(a) = \frac{t^2 a^2}{((ta)^2 + (1 + \Delta^2)^2)^{3/2}}, \quad (5.4)$$

uma boa aproximação para $ta > 1 + \Delta^2$. Dessa maneira, podemos melhorar o comportamento ruim para $a < 1 + \Delta^2$ ao aumentarmos o tempo de evolução t . A Figura 10 mostra

diferentes curvas de $F(a)$ para valores diferentes de t .

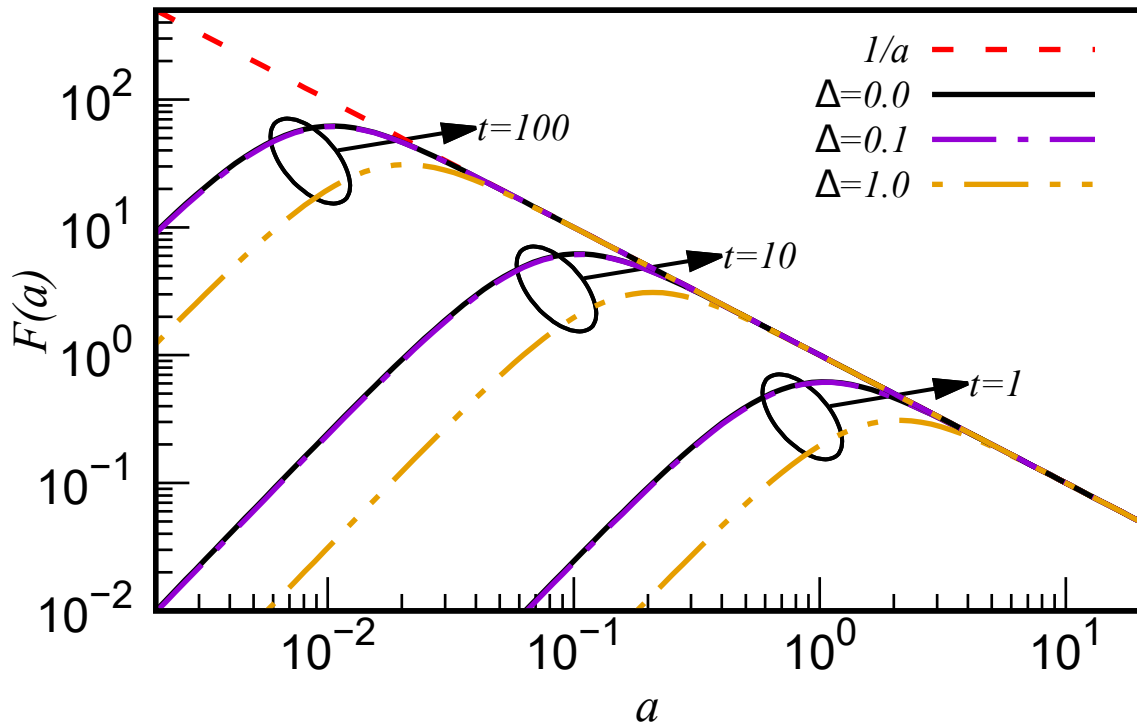


Figura 10 – Gráfico log de $F(a)$ em função de a considerando $M = 0$ da primeira proposta do algoritmo, com diferentes valores de Δ e t em comparação com a curva ideal. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.

Conseqüentemente, é possível aumentar a fidelidade do algoritmo modificado ao aumentarmos o tempo de evolução, cuja limitação pode ser majoritariamente atribuída ao *hardware*, a depender do tempo de decoerência e de alta fidelidade das portas quânticas.

5.2 Proposta 2

No desenvolvimento dos resultados apresentados pelas Eqs. (3.46) e (3.47), é possível notar que duas conseqüências diretas de se utilizar um estado com função de onda ímpar como o estado inicial auxiliar $|\tilde{1}\rangle$ são: (a) a sensibilidade ao sinal do autovalor a e (b) na expansão de $|s_{dL}\rangle$, os estados com função de onda par são desprezados.

Com isso em mente, ao escolhermos uma função suave H par, cuja integral definida seja $\int_{\mathbb{R}} H(x)dx = 1$, temos

$$\int_{-\infty}^{\infty} H(ax)dx = \frac{1}{|a|}. \quad (5.5)$$

Com isso em mente, ao substituirmos o estado inicial $|\tilde{1}\rangle$ por $|0\rangle$, é esperado que $F(a)$ seja uma boa aproximação para $1/|a|$ ao mesmo tempo que apenas estados pares sejam relevantes na expansão $|s_{dL}\rangle$. Dessa maneira, o algoritmo com a segunda modificação se torna aquele representado esquematicamente na Figura 11.

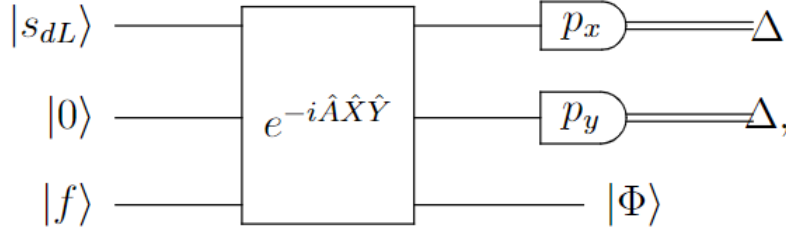


Figura 11 – Representação circuital da forma prática da segunda proposta de modificação do algoritmo. Fonte: Elaborada pelo autor.

Com essa modificação, o resultado do algoritmo, quando substituimos $|f\rangle$ por $|a\rangle$, se torna

$$\langle a|\Phi\rangle = F(a) = \lambda t \frac{2\Delta}{\pi} \frac{1}{\sqrt{a^2 t^2 + (1 + \Delta^2)^2}} \sum_{m=0}^{\lfloor d/2 \rfloor} \frac{\gamma_{2m}}{\Gamma} g_m(at), \quad (5.6)$$

onde

$$g_m(a) = \sqrt{\frac{(2m)!}{2^{2m}}} \frac{(-1)^m}{m!} \left(\frac{a^2 - 1 + \Delta^4}{a^2 + (1 + \Delta^2)^2} \right)^m. \quad (5.7)$$

Consequentemente, temos condições para as quais o novo algoritmo modificado funciona similarmente às da primeira proposta de modificação: operadores positivo (negativo)-semidefinidos e alta precisão para $a > 1 + \Delta^2$, nos retornando uma aproximação de $F(a)$ da forma

$$F(a) = \frac{1}{|a|} \left(1 + \frac{\alpha_1}{|a|^2} + \frac{\alpha_2}{|a|^4} + \dots \right), \quad (5.8)$$

onde podemos remover os termos de erro sistematicamente ao substituirmos o estado $|s\rangle$ por uma superposição

$$|s_M\rangle = \sum_{m=0}^M \tilde{\gamma}_{2m} |2m\rangle, \quad (5.9)$$

de maneira que $\alpha_1 = \alpha_2 = \dots = \alpha_M = 0$. Para $M = 1$, podemos obter analiticamente (Apêndice A)

$$\tilde{\gamma}_0 = \frac{\tilde{\gamma}_2}{\sqrt{2}} \left(1 + \frac{4}{1 + \Delta^2} \right). \quad (5.10)$$

A Figura 12 mostra o comportamento da curva $F(a)$ em relação à curva ideal $1/a$ para diferentes valores de Δ e M , utilizando o resultado obtido na Eq. (5.10).

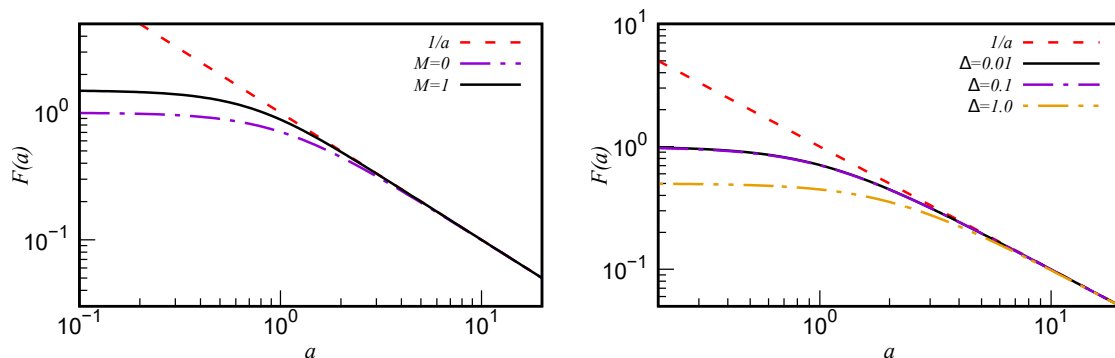


Figura 12 – Esquerda: Gráfico dílog de $F(a)$ em função de a , considerando $\Delta = 0,01$, da segunda proposta do algoritmo em comparação com a curva ideal $1/a$. Direita: Gráfico dílog de $F(a)$ da primeira proposta do algoritmo, considerando $M = 0$, em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.

Com um comportamento similar ao da curva obtida pela primeira modificação, as vantagens obtidas nesse caso ficam na preparação do estado inicial dos modos auxiliares - que pode ser reduzida a inicializar os dois modos auxiliares no estado de vácuo, além de não haver custo computacional para calcular $\hat{A}|f\rangle$ - e um comportamento mais próximo do ideal, apesar de ainda ruim - para $a < 1$.

De maneira análoga à primeira proposta de modificação, é possível melhorar a fidelidade do algoritmo modificado ao aumentarmos o tempo de evolução t . A Figura 13 mostra o comportamento da curva para diferentes valores do tempo de evolução, mostrando que a fidelidade escala com o tempo de maneira similar ao que acontece na primeira proposta de modificação.

5.3 Probabilidade de sucesso

O último passo do algoritmo, selecionar apenas resultados condicionados à detecção de estados com momento nulo nos modos auxiliares, depende de uma probabilidade de sucesso, que por sua vez depende da precisão Δ . A probabilidade de sucesso, $P(\Delta) = \frac{1}{\lambda} \|\langle \Delta, \Delta | \Psi \rangle\|^2$, pode ser calculada para as diferentes formas do algoritmo. No caso do algoritmo original temos $|\Psi\rangle$ dado pela Eq. (3.27), onde a função $\Theta(x)$ é substituída pela

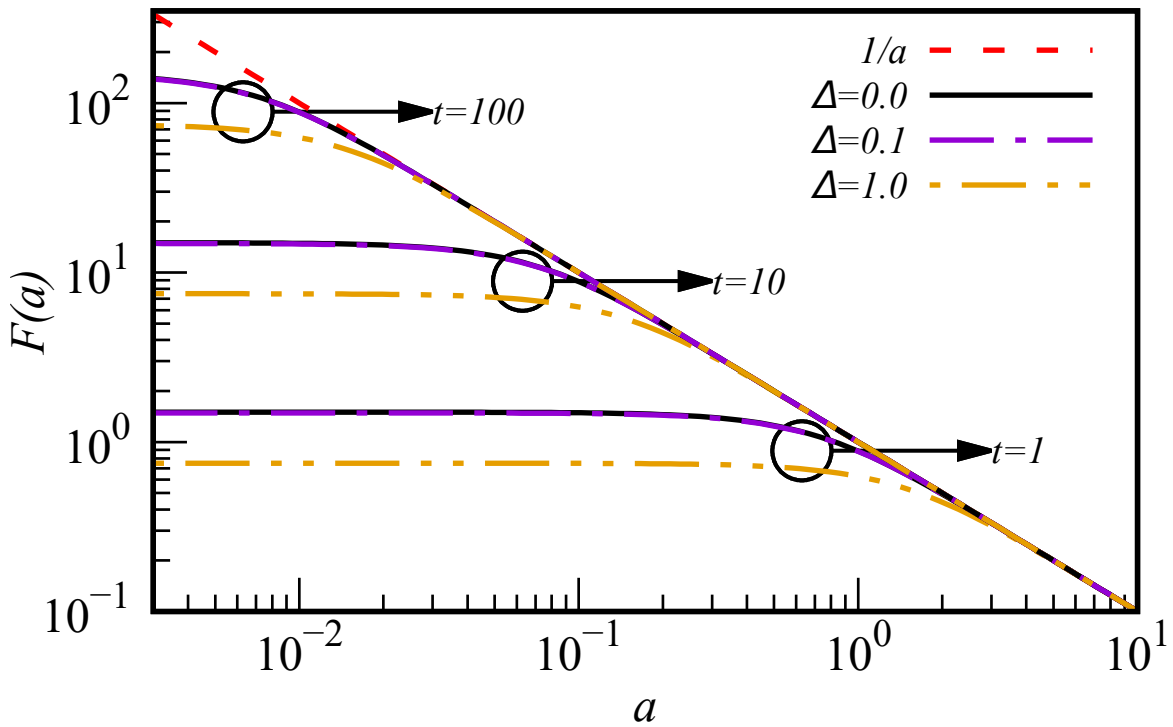


Figura 13 – Gráfico dílog de $F(a)$ da segunda proposta do algoritmo, fixando $M = 0$, com diferentes valores de Δ e t em comparação com a curva ideal $1/a$. O programa que gera as curvas pode ser reproduzido utilizando os códigos no Apêndice B. Fonte: Elaborada pelo autor.

função de onda da superposição $|s\rangle$, previamente discutida. Dessa maneira, é possível obter facilmente

$$P_{Arr}(\Delta) = \frac{4\Delta^2}{\Gamma^2} \sum_a \frac{a^2 |f_a|^2}{(a^2 + (1 + \Delta^2)^2)^3} \left[\sum_{n=0}^{\lfloor \frac{d}{2} \rfloor} \frac{\gamma_{2n+1}}{2^n} \frac{\sqrt{(2n+1)!}}{n!} \left(\frac{1 - a^2 - \Delta^4}{a^2 + (1 + \Delta^2)^2} \right)^n \right]^2, \quad (5.11)$$

onde $f_a = \langle a|f\rangle$.

É importante notar que para valores de Δ próximos a 1, a convergência do algoritmo original se torna muito lenta. Dessa maneira, é seguro dizer que a probabilidade de sucesso do algoritmo original é da ordem $O(\Delta^2)$, valor referente ao caso $\Delta \ll 1$.

A primeira proposta tem probabilidade de sucesso similar à do algoritmo original. Restringindo-se ao caso $d = 1$, a probabilidade de sucesso se torna

$$P_1(\Delta) = \frac{4\Delta^2}{\Omega^2} \sum_a \frac{t^2 a^4 |f_a|^2}{((ta)^2 + (1 + \Delta^2)^2)^3}, \quad (5.12)$$

onde $\Omega = \|\hat{A}|f\rangle\|$ é uma constante de normalização. Já para o caso da segunda proposta, a probabilidade de sucesso se torna

$$P_2(\Delta) = 4\Delta^2 \sum_a \frac{|f_a|^2}{(ta)^2 + (1 + \Delta^2)^2}. \quad (5.13)$$

Garantindo vantagem no uso de recursos, o uso de $\Delta = 1,0$ nas modificações propostas gera uma probabilidade de sucesso muito maior, para t fixado, ao custo de uma ligeira perda na fidelidade do algoritmo, que pode ser compensada mais facilmente com o tempo de evolução.

6 Exemplos

Nesta seção, serão apresentadas comparações entre os diferentes algoritmos apresentados anteriormente, isto é, as propostas de modificação e a versão original do algoritmo. Serão considerados dois exemplos com operadores positivo semidefinidos: O operador de Laplace como o primeiro exemplo e uma versão modificada do oscilador harmônico quântico como o segundo exemplo.

6.1 Operador de Laplace

Como o primeiro exemplo, nosso objetivo será comparar as diferentes propostas considerando o problema de calcular o potencial eletrostático $\psi(\mathbf{r})$ associado a uma densidade de carga $\rho(\mathbf{r})$, o que significa que precisaríamos resolver a equação de Poisson,

$$-\nabla^2\psi(\mathbf{r}) = \rho(\mathbf{r}). \quad (6.1)$$

Iremos considerar que a densidade de cargas é dada por uma superposição de funções Gaussianas da forma

$$g(\sigma, \mathbf{r}) = \frac{\sigma^{D/2}}{\pi^{D/4}} \exp\{-\sigma^2 r^2/2\}, \quad (6.2)$$

onde D é a dimensão espacial e σ determina a largura da função Gaussiana. O operador $A = -\nabla^2$ tem as ondas planas $|\mathbf{k}\rangle$ como autoestados e autovalores $a_{\mathbf{k}} = k^2$,

$$A |\mathbf{k}\rangle = k^2 |\mathbf{k}\rangle, \quad (6.3)$$

de maneira que $\langle \mathbf{r} | \mathbf{k} \rangle = \exp(i\mathbf{k} \cdot \mathbf{r}) / (2\pi)^{D/2}$. O estado $|f\rangle$ tem função de onda $\rho(\mathbf{r})$, e pode ser escrito como

$$|f\rangle = \int d^D k G(\mathbf{k}) |\mathbf{k}\rangle, \quad (6.4)$$

onde

$$G(\mathbf{k}) = \int \frac{d^D r}{(2\pi)^{D/2}} \rho(\mathbf{r}) e^{-i\mathbf{k} \cdot \mathbf{r}}. \quad (6.5)$$

Uma função de densidade gaussiana, $\rho(\mathbf{r}) = g(\sigma, \mathbf{r})$, tem transformada de Fourier,

$$G(\mathbf{k}) = \frac{1}{\pi^{D/4} \sigma^{D/2}} e^{-\frac{k^2}{2\sigma^2}} = g\left(\frac{1}{\sigma}, \mathbf{k}\right), \quad (6.6)$$

com largura $1/\sigma$.

Para o caso do operador inverso ideal A^{-1} , teríamos $A^{-1} |\mathbf{k}\rangle = 1/k^2 |\mathbf{k}\rangle$, enquanto para cada aproximação A_{approx}^{-1} discutidos anteriormente, temos

$$A_{approx}^{-1} |\mathbf{k}\rangle = F(k^2) |\mathbf{k}\rangle, \quad (6.7)$$

com $F(k)$ determinada pela versão do algoritmo. Dessa maneira, a solução esperada pelo algoritmo da Eq. (6.1) será dada por

$$\psi(\mathbf{r}) = \langle \mathbf{r} | A^{-1} | f \rangle = \int d^D k G(\mathbf{k}) \frac{1}{k^2} \langle \mathbf{r} | \mathbf{k} \rangle, \quad (6.8)$$

e as soluções aproximadas serão dadas por

$$\psi(\mathbf{r}) = \langle \mathbf{r} | A_{approx}^{-1} | f \rangle = \int d^D k G(\mathbf{k}) F(k^2) \langle \mathbf{r} | \mathbf{k} \rangle. \quad (6.9)$$

Com as considerações acima, podemos escrever expressões explícitas para as soluções aproximadas e exatas no caso de uma densidade de carga gaussiana, $\rho(\mathbf{r}) = g(\sigma, \mathbf{r})$, em três dimensões. Temos

$$\begin{aligned} \psi_{ex}(\mathbf{r}) &= \frac{1}{\sqrt{2\pi^{5/2}\sigma^3}} \int_0^\infty \frac{2 \sin(kr)}{kr} e^{-k^2/2\sigma^2} dk \\ &= \frac{1}{\sqrt{2\pi^{1/2}\sigma^3}} \frac{erf(\sigma r/\sqrt{2})}{r} \end{aligned} \quad (6.10)$$

para a solução exata, onde $erf(x)$ é a função erro. As soluções aproximadas terão a forma geral

$$\psi_{app}(\mathbf{r}) = \frac{1}{\sqrt{2\pi^{5/2}\sigma^3}} \int_0^\infty \frac{2k \sin(kr)}{r} F(k^2) e^{-k^2/2\sigma^2} dk, \quad (6.11)$$

onde $F(k^2)$ pode ser obtida ao substituirmos $a = k^2$ na Eq. (3.46) para a versão prática do algoritmo original e Eqs. (5.4) e (5.6) para as propostas de modificação 1 e 2, respectivamente.

No caso $\sigma = 1$, Eq. (6.6) nos mostra que teríamos uma projeção forte em $\{\mathbf{a}\}_{|a|<1}$, causando todas as propostas ineficientes. No entanto, ao escolhermos uma função externa composta por duas funções gaussianas, de maneira que os termos de projeção sejam nulos em uma vizinhança de $\mathbf{a} = 0$, é possível obter ótimos resultados. Por exemplo, ao escolher $f = g(\sigma_1, \mathbf{x}) - g(\sigma_2, \mathbf{x})$, onde $\sigma_1 \neq \sigma_2$ podemos obter uma forte projeção fora da origem, o que resultaria em resultados precisos tanto para o algoritmo original e as modificações propostas. Contudo, ao utilizarmos maiores tempos de evolução em nossas modificações propostas do algoritmo, é possível obter bons resultados mesmo para funções com forte projeção na origem, como pode ser visto na Figura 14, onde $f = g(4, \mathbf{x})$. Neste caso, é possível observar que os termos de projeção $\langle a | f \rangle$ são apreciáveis até $k^2 \sim 16$, de maneira que as nossas propostas de modificação consigam cobrir a região $k^2 \in (1/t, \infty)$ com alta fidelidade em contraste ao algoritmo original, que cobre com alta fidelidade $k^2 \in (0, 2; 16)$ de maneira que, para valores do tempo de evolução suficientemente grandes, nossas propostas consigam obter resultados melhores.

É importante notar que, neste exemplo em que descrevemos uma solução particular do potencial elétrico de uma partícula aprisionada, a solução geral para isto pode ser

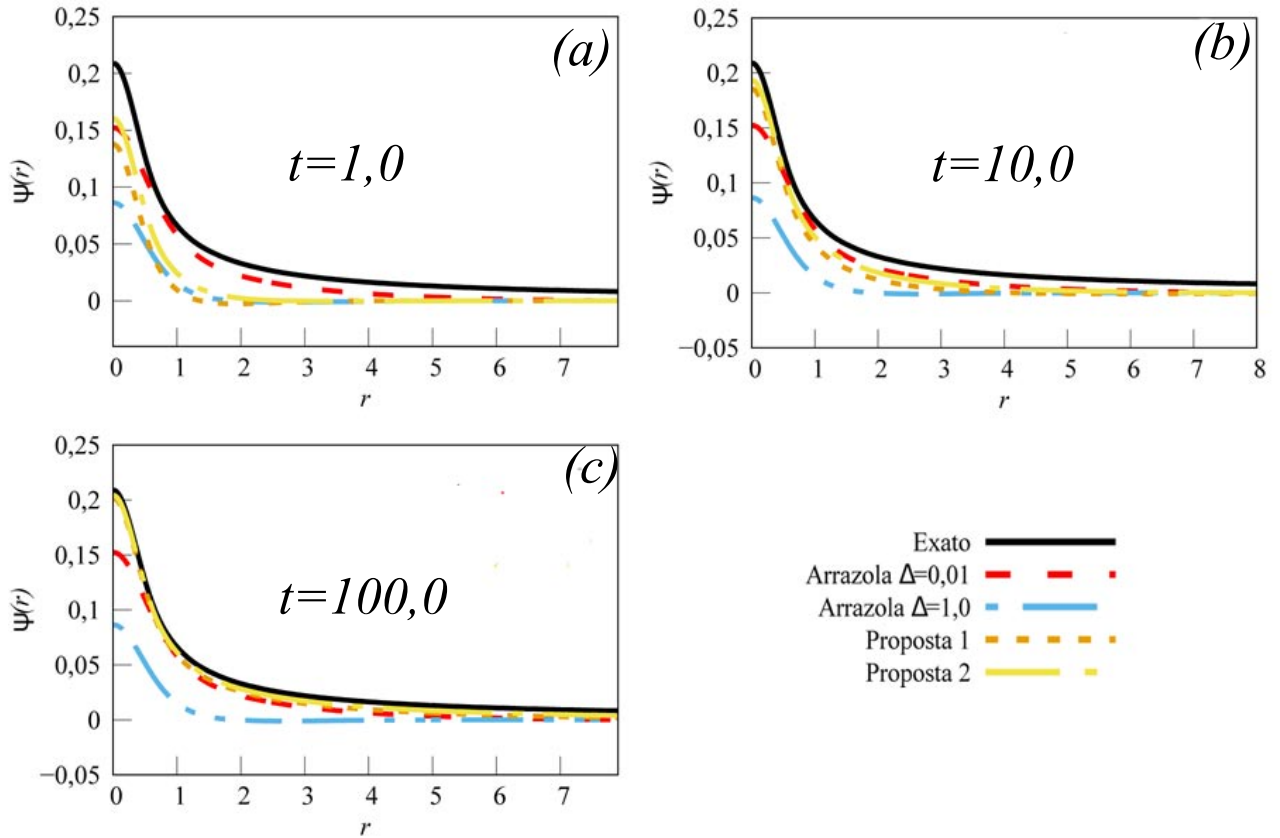


Figura 14 – Soluções particulares da Equação de Poisson (6.1), como funções da variável radial r , com função não homogênea $f(\mathbf{r}) = g(4, \mathbf{r})$. As soluções obtidas pelo algoritmo original com dois valores de precisão na detecção $\Delta = \{0,01; 1,0\}$ são comparadas às propostas 1 e 2 para tempos de evolução $t = 1,0$ (a), $t = 10,0$ (b) e $t = 100,0$ (c). A solução exata esperada pelo algoritmo é apresentada para comparação. Fonte: Elaborada pelo autor.

calculada localmente para regiões de interesse do espaço a partir dos resultados obtidos na Figura 14. Todas as propostas do algoritmo perdem precisão quando r é muito grande, por conta da baixa precisão em $k \ll 1$. No entanto, o comportamento para r grande é conhecido neste caso particular, sendo proporcional a $1/r$.

6.2 Operador do Oscilador Harmonico Quântico

Considerando a equação diferencial

$$(-\nabla^2 + \mathbf{x}^2)\psi(\mathbf{x}) = f(\mathbf{x}), \quad (6.12)$$

para o caso unidimensional, reduzido a uma EDO, sabemos que os autoestados do operador

diferencial são

$$|n\rangle = \int \frac{1}{\sqrt{2^n n!} \pi^{1/2}} H_n(x) e^{-x^2/2} |x\rangle dx, \quad (6.13)$$

onde $H_n(x)$ é o n -ésimo polinômio de Hermite, e com autovalores $2n + 1$, $n = 0, 1, 2, \dots$. Escolhendo o estado coerente $|\alpha\rangle$, com α real, como o estado $|f\rangle$, temos (COHEN-TANNOUJJI; DUI; LALOE, 1991)

$$|f\rangle = |\alpha\rangle = e^{-\alpha^2/2} \sum_{n=0}^{\infty} \frac{\alpha^n}{\sqrt{n!}} |n\rangle \quad (6.14)$$

e

$$f(x) = \frac{1}{\pi^{1/4}} \exp\left\{-\frac{(x - \sqrt{2}\alpha)^2}{2}\right\}. \quad (6.15)$$

Podemos imediatamente escrever a expressão geral para as soluções aproximadas e exata da Eq. (6.12) que podem ser obtidas das diferentes maneiras pelos diferentes algoritmos. Temos então

$$\psi(x) = \langle x | A^{-1} |f(\alpha)\rangle = \frac{e^{-(x^2 + \alpha^2)/2}}{\pi^{1/4}} S(\alpha, x), \quad (6.16)$$

onde

$$S(\alpha, x) = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{\alpha}{\sqrt{2}}\right)^n H_n(x) F(2n + 1). \quad (6.17)$$

Como no exemplo anterior, $F(a)$ é dada pelas Eqs. (3.46), (5.4) e (5.6) para a versão original do algoritmo, propostas 1 e 2, respectivamente. A solução exata esperada é dada por $F(2n + 1) = 1/(2n + 1)$. Na Figura 15 são comparados o algoritmo original com $\Delta = \{0, 01; 1, 0\}$ e suas modificações (propostas 1 e 2) para $\alpha = 2, 5$ considerando dois tempos de evolução distintos $t = 1, 0$ (esquerda) e $t = 10, 0$ (direita). É possível notar que para curtos períodos de evolução ($t = 1, 0$) o algoritmo original tem desempenho melhor do que as propostas de modificação para $x < 1$, enquanto para $x > 1$ propostas 1 e 2 se aproximam melhor da solução exata. Um comportamento similar é observado se o sistema é evoluído durante um tempo de evolução total $t = 10, 0$ com a diferença de que as soluções obtidas pelas propostas 1 e 2 são consideravelmente mais fiéis ao resultado exato esperado.

É importante notar que o uso de $\Delta = 1, 0$ nas diferentes propostas de modificações impacta diretamente na probabilidade de sucesso de se medir o estado $|\Delta, \Delta\rangle$ nos estados auxiliares. Neste exemplo, o algoritmo original tem uma probabilidade de sucesso de $3, 25 \times 10^{-8}$ e $1, 36 \times 10^{-4}$ para $\Delta = 0, 01$ e $\Delta = 1, 0$, respectivamente. Por outro lado, a probabilidade de sucesso da primeira proposta é de $2, 18 \times 10^{-8}$ e para a segunda proposta é de $4, 58 \times 10^{-4}$, ambas considerando $\Delta = 1, 0$ e $t = 10, 0$.

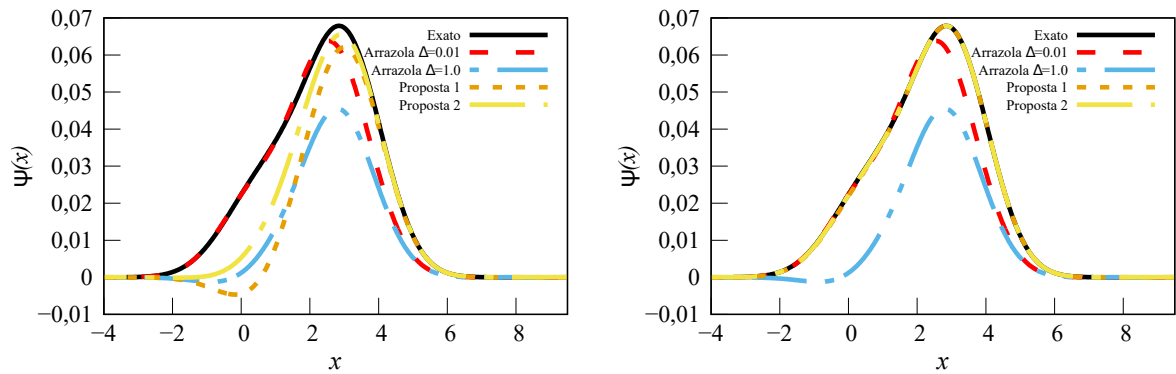


Figura 15 – Solução particular da equação não homogênea do oscilador harmônico quântico modificada (6.12) como função da posição x na qual a função não homogênea $f(x)$ é a função de onda do estado coerente com amplitude $\alpha = 2,5$. As soluções obtidas pelo algoritmo original consideram dois valores distintos $\Delta = \{0,01; 1,0\}$ e são comparadas com as dadas pelas propostas 1 e 2 para tempos de evolução $t = 1,0$ (esquerda) e $t = 10,0$ (direita). A solução exata é apresentada para efeitos de comparação. Esse resultado mostra que as versões modificadas do algoritmo tem a evolução temporal como uma ferramenta poderosa de melhorar a fidelidade do algoritmo. Fonte: Elaborada pelo autor.

7 Conclusão

Este trabalho foi iniciado com o fim de estudar algoritmos quânticos e a possibilidade de implementação em sistemas físicos de maneira a analisar a possibilidade de obter ganho computacional no atual estado da arte da computação quântica ao resolver equações diferenciais parciais e sistemas de equações diferenciais.

Com uma literatura extensa sobre algoritmos quânticos propostos para resolver equações diferenciais, o algoritmo de solução de equações diferenciais parciais não homogêneas (ARRAZOLA *et al.*, 2019) teve sua implementação analisada. A partir dos resultados preliminares que demonstravam grande dificuldade na implementação por conta do custo de preparação do estado inicial e da probabilidade de sucesso ser baixa para casos com alta precisão, foram propostas modificações de maneira a torná-lo um algoritmo com fidelidade maior e com custos computacionais menores, ao custo de restringir os resultados à classe de operadores positivo (negativo)-semidefinidos.

Os resultados sobre as propostas de modificação do algoritmo aqui expressos estão sendo organizados em um artigo, e há a expectativa para a publicação em breve. Além disso, estamos trabalhando na previsão teórica da aplicação do conjunto universal Eq. (3.3) em um sistema com vários íons e também na simulação de todas as portas do conjunto universal em sistemas de íons aprisionados com apenas um e múltiplos íons.

Referências

- ARRAZOLA, J. M. et al. Quantum algorithm for nonhomogeneous linear partial differential equations. *Physical Review A*, American Physical Society (APS), v. 100, n. 3, Sep 2019. ISSN 2469-9934. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.100.032306>>. Citado 6 vezes nas páginas 17, 18, 21, 28, 37 e 63.
- BABBUSH, R.; LOVE PETER J. AND ASPURU-GUZI, A. Adiabatic quantum simulation of quantum chemistry. *Scientific Reports*, v. 4, n. 6603, Oct 2014. ISSN 2045-2322. Disponível em: <<https://doi.org/10.1038/srep06603>>. Citado na página 17.
- BARTLETT, S. D.; SANDERS, B. C. Universal continuous-variable quantum computation: Requirement of optical nonlinearity for photon counting. *Phys. Rev. A*, American Physical Society, v. 65, p. 042304, Mar 2002. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.65.042304>>. Citado na página 24.
- BRAUNSTEIN, S. L.; LOOCK, P. van. Quantum information with continuous variables. *Rev. Mod. Phys.*, American Physical Society, v. 77, p. 513–577, Jun 2005. Disponível em: <<https://link.aps.org/doi/10.1103/RevModPhys.77.513>>. Citado na página 18.
- CIRAC, J. I.; ZOLLER, P. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, American Physical Society, v. 74, p. 4091–4094, May 1995. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.74.4091>>. Citado na página 17.
- COHEN-TANNOUJDI, C.; DUI, B.; LALOE, F. *Quantum Mechanics*. [S.l.]: Wiley-VCH, 1991. Citado na página 60.
- COMPUTING, M.; PATENTS, B. R. . *Multiverse Computing: Optimizing Financial Portfolios with Quantum Computing*. <https://www.dwavesys.com/media/5qahck2o/multiverse_case_study_v7-1.pdf>. Citado na página 17.
- COSTA, P. C. S.; JORDAN, S.; OSTRANDER, A. Quantum algorithm for simulating the wave equation. *Physical Review A*, American Physical Society (APS), v. 99, n. 1, Jan 2019. ISSN 2469-9934. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.99.012323>>. Citado na página 18.
- DHAND, I.; SANDERS, B. C. Stability of the trotter–suzuki decomposition. *Journal of Physics A: Mathematical and Theoretical*, IOP Publishing, v. 47, n. 26, p. 265206, jun 2014. Disponível em: <<https://doi.org/10.1088/1751-8113/47/26/265206>>. Citado na página 24.
- FEYNMAN, R. P. Simulating physics with computers. *International Journal of Theoretical Physics*, Springer Nature Switzerland AG, v. 21, n. 3, May 1982. ISSN 1572-9575. Disponível em: <<https://doi.org/10.1007/BF02650179>>. Citado na página 17.
- GARCÍA-MOLINA, P.; RODRÍGUEZ-MEDIAVILLA, J.; GARCÍA-RIPOLL, J. J. *Solving partial differential equations in quantum computers*. 2021. Citado na página 18.
- GRZESIAK, N. et al. Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer. *Nature Communications*, Springer Science and

- Business Media LLC, v. 11, n. 1, Jun 2020. ISSN 2041-1723. Disponível em: <http://dx.doi.org/10.1038/s41467-020-16790-9>. Citado na página 25.
- HUANG, H.-L. et al. *Superconducting Quantum Computing: A Review*. 2020. Citado na página 17.
- JOHANSSON, J.; NATION, P.; NORI, F. Qutip: An open-source python framework for the dynamics of open quantum systems. *Computer Physics Communications*, v. 183, n. 8, p. 1760–1772, 2012. ISSN 0010-4655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010465512000835>. Citado na página 41.
- JOHANSSON, J.; NATION, P.; NORI, F. Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, v. 184, n. 4, p. 1234–1240, 2013. ISSN 0010-4655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010465512003955>. Citado na página 41.
- KALAJDZIEVSKI, T.; ARRAZOLA, J. M. Exact gate decompositions for photonic quantum computing. *Physical Review A*, American Physical Society (APS), v. 99, n. 2, Feb 2019. ISSN 2469-9934. Disponível em: <http://dx.doi.org/10.1103/PhysRevA.99.022341>. Citado 2 vezes nas páginas 24 e 48.
- KALAJDZIEVSKI, T.; QUESADA, N. Exact and approximate continuous-variable gate decompositions. *Quantum*, Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften, v. 5, p. 394, fev. 2021. ISSN 2521-327X. Disponível em: <https://doi.org/10.22331/q-2021-02-08-394>. Citado na página 48.
- KILLORAN, N. et al. Continuous-variable quantum neural networks. *Physical Review Research*, American Physical Society (APS), v. 1, n. 3, Oct 2019. ISSN 2643-1564. Disponível em: <http://dx.doi.org/10.1103/PhysRevResearch.1.033063>. Citado na página 37.
- LEIBFRIED, D. et al. Quantum dynamics of single trapped ions. *Reviews of Modern Physics*, The American Physical Society, v. 75, n. 1, p. 281–324, 2007. Citado 3 vezes nas páginas 25, 27 e 39.
- LIU, H. et al. *Variational Quantum algorithm for Poisson equation*. 2020. Citado na página 18.
- LLOYD, S.; BRAUNSTEIN, S. L. Quantum computation over continuous variables. *Physical Review Letters*, American Physical Society (APS), v. 82, n. 8, p. 1784–1787, Feb 1999. ISSN 1079-7114. Disponível em: <http://dx.doi.org/10.1103/PhysRevLett.82.1784>. Citado na página 18.
- MARX, R. et al. Approaching five-bit nmr quantum computing. *Phys. Rev. A*, American Physical Society, v. 62, p. 012310, Jun 2000. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevA.62.012310>. Citado na página 17.
- MATHIEU Émile. *Course de physique mathématique*. [S.l.]: Gauthier-Villars, 1873. Citado na página 26.

MILBURN, G. J.; WALLS, D. F. Quantum nondemolition measurements via quadratic coupling. *Phys. Rev. A*, American Physical Society, v. 28, p. 2065–2070, Oct 1983. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.28.2065>>. Citado na página 24.

MUGEL, S. et al. *Dynamic Portfolio Optimization with Real Datasets Using Quantum Processors and Quantum-Inspired Tensor Networks*. 2020. Citado na página 17.

NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. Cambridge: Cambridge University, 2000. Citado 2 vezes nas páginas 17 e 23.

ORTIZ-GUTIÉRREZ, L. et al. Continuous variables quantum computation over the vibrational modes of a single trapped ion. *Optics Communications*, v. 397, p. 166–174, 2017. ISSN 0030-4018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0030401817302821>>. Citado 3 vezes nas páginas 18, 24 e 47.

ORÚS, R.; MUGEL, S.; LIZASO, E. Forecasting financial crashes with quantum computing. *Physical Review A*, American Physical Society (APS), v. 99, n. 6, Jun 2019. ISSN 2469-9934. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.99.060301>>. Citado na página 17.

ORÚS, R.; MUGEL, S.; LIZASO, E. Quantum computing for finance: Overview and prospects. *Reviews in Physics*, v. 4, p. 100028, 2019. ISSN 2405-4283. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405428318300571>>. Citado na página 17.

PARRA-RODRIGUEZ, A. et al. Digital-analog quantum computation. *Physical Review A*, American Physical Society (APS), v. 101, n. 2, Feb 2020. ISSN 2469-9934. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.101.022305>>. Citado na página 18.

SAKURAI, J.; NAPOLITANO, J. Modern quantum mechanics 2nd edition. *Person New International edition*, 2014. Citado na página 39.

Sanz, M. Digital-Analog Quantum Computing. In: *APS March Meeting Abstracts*. [S.l.: s.n.], 2019. (APS Meeting Abstracts, v. 2019), p. Y34.001. Citado na página 18.

SHOR, P. W. Algorithms for quantum computation: discrete logarithms and factoring. p. 124–134, 1994. Citado na página 17.

SLEATOR, T.; WEINFURTER, H. Realizable universal quantum logic gates. *Phys. Rev. Lett.*, American Physical Society, v. 74, p. 4087–4090, May 1995. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.74.4087>>. Citado 2 vezes nas páginas 17 e 23.

STOBIŃSKA, M.; VILLAR, A. S.; LEUCHS, G. Generation of kerr non-gaussian motional states of trapped ions. *EPL (Europhysics Letters)*, IOP Publishing, v. 94, n. 5, p. 54002, may 2011. Disponível em: <<https://doi.org/10.1209/0295-5075/94/54002>>. Citado 2 vezes nas páginas 46 e 47.

SUZUKI, M. General decomposition theory of ordered exponentials. *Proceedings of the Japan Academy, Series B*, v. 69, n. 7, p. 161–166, 1993. Citado na página 24.

TASCA, D. S. et al. Continuous-variable quantum computation with spatial degrees of freedom of photons. *Phys. Rev. A*, American Physical Society, v. 83, p. 052325, May 2011. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.83.052325>>. Citado na página 18.

TROTTER, H. F. On the product of semi-groups of operators. *Proceedings of the American Mathematical Society*, v. 10, p. 545–551, 1959. Disponível em: <<https://doi.org/10.1090/S0002-9939-1959-0108732-6>>. Citado na página 24.

WANG, Y. et al. Qudits and high-dimensional quantum computing. *Frontiers in Physics*, v. 8, p. 479, 2020. ISSN 2296-424X. Disponível em: <<https://www.frontiersin.org/article/10.3389/fphy.2020.589504>>. Citado na página 18.

WERBOS, P. J.; DOLMATOVA, L. Analog quantum computing (aqc) and the need for time-symmetric physics. *Quantum Information Processing*, v. 15, n. 3, p. 1273–1287, Mar 2016. ISSN 1573-1332. Disponível em: <<https://doi.org/10.1007/s11128-015-1146-2>>. Citado na página 18.

WIEBE, N. et al. Simulating quantum dynamics on a quantum computer. IOP Publishing, v. 44, n. 44, p. 445308, oct 2011. Disponível em: <<https://doi.org/10.1088/1751-8113/44/44/445308>>. Citado na página 17.

XIN, T. et al. Quantum algorithm for solving linear differential equations: Theory and experiment. *Physical Review A*, American Physical Society (APS), v. 101, n. 3, Mar 2020. ISSN 2469-9934. Disponível em: <<http://dx.doi.org/10.1103/PhysRevA.101.032307>>. Citado 2 vezes nas páginas 17 e 18.

XIN, T. et al. Quantum algorithm for solving linear differential equations: Theory and experiment. *Phys. Rev. A*, American Physical Society, v. 101, p. 032307, Mar 2020. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevA.101.032307>>. Citado na página 17.

Apêndices

APÊNDICE A – Obtendo os termos da superposição nas Propostas de modificação 1 e 2.

De maneira a calcular os valores apresentados pelas Eqs. (5.3) e (5.10), é necessário expandir as Eqs. (3.46) e (5.6) em primeira ordem para $a > 1 + \Delta^2$.

A.1 Primeira proposta de modificação

Para a primeira proposta de modificação, ao expandirmos a Eq. (3.46) em série de Taylor após definir $M = 1$, obtemos

$$F(a) \propto \frac{1}{|a|} \left\{ \gamma_1 C_1 \left[1 - \frac{3}{2} \left(\frac{1 + \Delta^2}{a} \right)^2 \right] + 12\gamma_3 C_3 \frac{1 + \Delta^2}{a^2} - 3! \gamma_3 C_3 \left[1 - \frac{3}{2} \left(\frac{1 + \Delta^2}{a} \right)^2 \right] \right\} + O(|a|^{-5}).$$

Uma vez que o objetivo principal é obter um resultado assintótico a $|a|^{-1} + O(|a|^{-5})$, é necessário encontrar os valores γ_1, γ_3 de maneira a eliminar os termos proporcionais a $|a|^{-3}$ na soma acima. Dessa maneira, os termos obedecem a equação

$$12\gamma_3 C_3 - \frac{3}{2}\gamma_1 C_1 (1 + \Delta^2) - 9\gamma_3 C_3 (1 + \Delta^2) = 0, \quad (\text{A.1})$$

que é facilmente resolvida por

$$\gamma_1 = \frac{\gamma_3}{\sqrt{6}} \left(3 + \frac{4}{1 + \Delta^2} \right). \quad (\text{A.2})$$

A.2 Segunda proposta de alternativa

Para a segunda proposta de modificação, devemos expandir a Eq. 5.6 em série de Taylor e, definindo $M = 1$ obtemos

$$F(a) \propto \frac{1}{|a|} \left[2\gamma_0 C_0 \left(1 - \frac{1}{2} \left(\frac{1 + \Delta^2}{a} \right)^2 \right) + \frac{8\gamma_2 C_2}{a^2} (1 + \Delta^2) \right]$$

$$-4\gamma_2 C_2 \left(1 - \frac{1}{2} \left(\frac{1 + \Delta^2}{a^2} \right)^2 \right) \Big].$$

Para obter um resultado assintótico a $|a|^{-1} + O(|a|^{-5})$, repetimos o procedimento de cancelar os termos de terceira ordem, de maneira que os termos da superposição devam satisfazer

$$8\gamma_2 C_2 (1 + \Delta^2) + 2\gamma_2 C_2 (1 + \Delta^2)^2 - \gamma_0 C_0 (1 + \Delta^2)^2 = 0, \quad (\text{A.3})$$

que pode ser resolvido por

$$\gamma_0 = \frac{\gamma_2}{\sqrt{2}} \left(1 + \frac{4}{1 + \Delta^2} \right). \quad (\text{A.4})$$

APÊNDICE B – Programas em Fortran criados para simular as curvas apresentadas nas Seções 3.3 e 5 .

Todos os programas dessa seção foram executados utilizando o compilador gnufortran.

B.1 Programa para gerar as curvas da versão prática do algoritmo original

O programa utilizado para gerar as diferentes curvas referentes à Eq. (3.46) é apresentado abaixo.

```

1  program arrazola
2  use iso_fortran_env
3  implicit none
4  integer, parameter :: qp=REAL128
5
6
7  real(kind=qp)::a,da,rpi,delta,F,f1,f2,f3,num,deno,t,at
8  integer,PARAMETER::d=69
9  real(kind=qp),dimension(2*d+2)::gamma,fac,norm
10 integer::m,k,n,j
11
12 open(1,file="arrazola_low.txt")
13 open(2,file="gamma_L20.txt")
14
15 t=1.0
16 delta=0.1
17
18 !Definindo C(n)=norm(n+1)
19 rpi=sqrt(2*asin(1.0))
20 norm(1)=1/sqrt(rpi)
21 do k=2,2*d+2
22 norm(k)=norm(k-1)/sqrt(2.0*(k-1))
23 enddo
24
25 !Definindo fatorial fac(i)=(i-1)!
26 fac(1)=1
27 do k=2,2*d+2

```

```

28     fac(k)=(k-1)*fac(k-1)
29     enddo
30
31     !Ler gamma, j calculado
32     do k=1,2*d+2
33     read(2,*) gamma(k)
34     enddo
35
36
37     a=0.001
38     da=1.005
39
40     do j=1,2309
41     a=a*da
42     at=a/t
43     F=0
44     f1=rpi*(0.5*((a**2) + ( (1 + (delta**2) )**2) ) )**(-1.5)
45     num=1-(a**2)-(delta**4)
46     deno=(a**2)+(1+(delta**2))**2
47
48     do n=1,d+1
49     !Coloquei d como sendo a metade da dimens o total, de maneira que
os gammas corram at 2*d+2
50
51     f2=a*gamma(2*n)*norm(2)*norm(2*n)*sqrt(2.0*rpi)*fac(2*n)/fac(n)
52     f3=(num/deno)**(n-1)
53
54     F=F+t*f1*f2*f3
55     enddo
56     write(1,*) at,F,1/at
57     enddo
58
59
60     end program

```

B.2 Algoritmo para gerar as curvas da primeira proposta de modificação.

O algoritmo utilizado para gerar as diferentes curvas referentes à Eq. (3.49) são expostos abaixo. Os programas calculam apenas os resultados para os casos $M = 0, 1$.

```

1     program prop1
2     use iso_fortran_env
3     implicit none
4     integer, parameter :: qp=REAL128
5

```

```

6
7  real(kind=qp)::a,da,rpi,delta,f1,f2,f3,num,deno,ff,l
8  integer,PARAMETER::d=2
9  real(kind=qp),dimension(2*d+2)::g,fac,norm,F,cte
10 integer::m,k,n,j
11
12 open(1,file="prop1.txt")
13
14 l=1.0
15 delta=1.0
16 g(1)=0
17 g(2)=(3+4/(1+(delta**2)))/sqrt(6.0)
18 g(3)=0
19 g(4)=1.0
20 cte=0
21 !Definindo C(n)=norm(n+1)
22 rpi=sqrt(2*asin(1.0))
23 norm(1)=1/sqrt(rpi)
24 do k=2,2*d+2
25 norm(k)=norm(k-1)/sqrt(2.0*(k-1))
26 enddo
27
28 !Definindo fatorial fac(i)=(i-1)!
29 fac(1)=1
30 do k=2,2*d+2
31 fac(k)=(k-1)*fac(k-1)
32 enddo
33
34 a=-1.0
35 da=2.0
36
37 do j=1,500
38 a=a+da
39 F=0
40 f1=rpi*(0.5*(((l*a)**2) + ( (1 + (delta**2) )**2) ) )**(-1.5)
41 num=1-((l*a)**2)-(delta**4)
42 deno=((l*a)**2)+(1+(delta**2))**2
43
44 do n=1,d+1
45
46 f2=norm(2)*norm(2*n)*sqrt(2.0*rpi)*fac(2*n)/fac(n)
47 f3=l*(a**2)*(num/deno)**(n-1)
48 cte(n)=((-1)**(n-1))*sqrt(rpi)*sqrt(fac(2*n))*(2**(2-n))/fac(n)
49 F(n)=F(n)+(l**2)*f1*f2*f3
50 ff=(g(2)*F(1)+g(4)*F(2))/(g(2)*cte(1)+g(4)*cte(2))
51 enddo
52 write(1,*)a,F(1)/cte(1),ff,1/a

```

```

53     enddo
54
55
56     end program

```

B.3 Algoritmo para gerar as curvas da segunda proposta de modificação.

O algoritmo utilizado para gerar as diferentes curvas referentes à Eq. (5.6) é descrito abaixo. Os programas calculam apenas os resultados para os casos $M = 0, 1$.

```

1     program arrazola
2     use iso_fortran_env
3     implicit none
4     integer, parameter :: qp=REAL128
5
6
7     real(kind=qp) :: r, dr, rpi, delta, f1, f2, f3, num, deno, l, ff, g1, g2
8     real(kind=qp) :: int1, f4, x, dx
9     integer, PARAMETER :: d=2
10    real(kind=qp), dimension(2*d+2) :: fac, norm
11    real(kind=qp), dimension(d+1) :: F, cte
12    integer :: m, k, n, j
13
14    open(1, file="prop2.txt")
15
16    delta=1.0
17    l=1.0
18    g2=1
19    g1=(1+(4/(1+(delta**2))))/sqrt(2.0)
20    !Definindo C(2n+1)=norm(n)
21    rpi=sqrt(2*asin(1.0))
22    norm(1)=1/sqrt(rpi)
23    do k=2, 2*d+2
24    norm(k)=norm(k-1)/sqrt(2.0*(k-1))
25    enddo
26
27    !Definindo fatorial fac(i)=(i-1)!
28    fac(1)=1
29    do k=2, 2*d+2
30    fac(k)=(k-1)*fac(k-1)
31    enddo
32
33    r=-1.0
34    dr=2.0
35    int1=0

```

```
36     do j=1,500
37         r=r+dr
38         cte=0
39         F=0
40         f1=((l*r)**2)+((1+(delta**2))**2)**(-0.5)
41         do n=1,d+1
42             cte(n)=norm(2*n-1)*fac(2*n-1)/fac(n)
43             num=1-((l*r)**2)-(delta**4)
44
45             deno=((l*r)**2)+(1+(delta**2))**2
46             f2=l*cte(n)
47             f3=(num/deno)**(n-1)
48
49
50         F(n)=f1*f2*f3
51     enddo
52     ff=(g1*F(1)+g2*F(2))/(cte(1)*g1-cte(2)*g2)
53
54     write(*,*)r,F(1)/cte(1),ff,1/r
55     write(1,*)r,F(1)/cte(1),ff,1/r
56     enddo
57
58
59     end program
```


APÊNDICE C – Programas em Python utilizados para simular as dinâmicas de íons apresentadas na Seção 4

C.1 Interação entre modos

O programa utilizado para simular a interação entre modos e calcular sua fidelidade pode ser executado utilizando Python.

```

1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # In[1]:
5
6
7 import numpy as np
8 import matplotlib as mpl
9 import scipy as sp
10 from qutip import *
11 from math import *
12 import math
13 import cmath
14 from matplotlib import *
15 import matplotlib.pyplot as plt
16
17
18 # In[2]:
19
20
21 options=Options(atol=1e-10,nsteps=1000,num_cpus=10,store_states=True,
22                 store_final_state=True)
23
24 # In[3]:
25
26
27 N =10; #basis dimension
28 nu=1.0; #field frequency (MHz)
29 nu_1x=1.0; nu_2x=nu; nu_1y=1.2; nu_2y=nu; #individual field frequencies
30 timescaling=0.001 #factor to plot time scale in ms
31 eta_x=0.1; eta_y=0.1;

```

```
32 omega_0=50.0
33 F_omega=20.0; #setup of Rabi frequencis
34 Omega_1x=nu_1x/F_omega; Omega_1y=nu_1y/F_omega;
35 Delta_1=-((nu_1x+nu_1y)+omega_0); Delta_2=-((nu_1x+nu_1y)+omega_0); Delta_3
    =(nu_1x-nu_1y)+omega_0; Delta_4=((nu_1x+nu_1y)+omega_0);
36
37
38 # In[4]:
39
40
41 def f_1(t, args): #ok
42     return cmath.exp(-1j*(Delta_1*(t)-math.pi))
43
44 def f_1d(t, args): #ok
45     return cmath.exp(1j*(Delta_1*(t)-math.pi))
46
47 def f_2(t, args): #ok
48     return cmath.exp(-1j*(Delta_2*(t)-math.pi))
49
50 def f_2d(t, args):
51     return cmath.exp(1j*(Delta_2*(t)-math.pi))
52
53 def f_3(t, args):
54     return cmath.exp(-1j*(Delta_3*(t)-math.pi))
55
56 def f_3d(t, args):
57     return cmath.exp(1j*(Delta_3*(t)-math.pi))
58
59 def f_4(t, args):
60     return cmath.exp(-1j*(Delta_4*(t)-math.pi))
61
62 def f_4d(t, args):
63     return cmath.exp(1j*(Delta_4*(t)-math.pi))
64
65
66 # In[5]:
67
68
69 a = tensor(qeye(2), destroy(N), qeye(N));
70 b = tensor(qeye(2), qeye(N), destroy(N));
71 Na=a.dag()*a;
72 Nb=b.dag()*b;
73 NT=Na+Nb;
74 x_a=a+a.dag();
75 x_b=b+b.dag();
76
77
```

```

78 # In[6]:
79
80
81 g_st = basis(2,0); e_st = basis(2,1);
82 plus_st = (1/np.sqrt(2))*(g_st+e_st); minus_st = (1/np.sqrt(2))*(g_st-
    e_st);
83
84 sig_p=tensor(sigmoid(), qeye(N), qeye(N));
85 sig_m=tensor(sigmam(), qeye(N), qeye(N));
86 sig_x=tensor(sigmoid(), qeye(N), qeye(N));
87 sig_z=tensor(sigmaz(), qeye(N), qeye(N));
88
89
90 # In[7]:
91
92
93 H0=(nu_1x*Na+nu_1y*Nb)+(1/2)*omega_0*sig_z;
94
95 Op_x=(1j*eta_x*x_a);
96 Op_y=(1j*eta_y*x_b);
97 Opd_x=(-1j*eta_x*x_a);
98 Opd_y=(-1j*eta_y*x_b);
99
100
101 # In[8]:
102
103
104 H=(1/2)*Oomega_1x*sig_x*(Op_x.expm())*(Op_y.expm());
105 Hd=(1/2)*Oomega_1x*sig_x*(Opd_x.expm())*(Opd_y.expm());
106
107
108 # In[9]:
109
110
111 H_eff=(1/2)*(Oomega_1x*eta_x*eta_y)*sig_x*x_a*x_b;
112
113
114 # In[10]:
115
116
117 psi0_atom_p=plus_st; #atom initial state
118 psi0_atom_m=minus_st;
119
120 psi0_field=tensor(basis(N,0),basis(N,0)); #field initial state
121
122 psi0_p_full=tensor(psi0_atom_p,psi0_field);
123 psi0_p_eff = psi0_p_full;

```

```
124 psi0_m_full=tensor(psi0_atom_m,psi0_field);
125 psi0_m_eff= psi0_m_full;
126
127
128 # In[11]:
129
130
131 H1t = [H,f_1]
132 H2t = [H,f_2]
133 H3t = [H,f_3]
134 H4t = [H,f_4]
135 H1dt = [Hd,f_1d]
136 H2dt = [Hd,f_2d]
137 H3dt = [Hd,f_3d]
138 H4dt = [Hd,f_4d]
139
140 H_full = [H0,H1t,H2t,H3t,H4t,H1dt,H2dt,H3dt,H4dt]
141
142
143 # In[12]:
144
145
146 Xi=(1/2)*eta_x*eta_y*Omega_1x;
147
148
149 # In[13]:
150
151
152 data_fidelity_p = np.vstack((0,1)).T
153 M=int(math.ceil(1/Xi))
154 t=np.linspace(0,M,M*100)
155 print(M)
156 psi_p_full = mcsolve(H_full,psi0_p_full,t,[],[],progress_bar=True,
    _safe_mode=True,options=options);
157 psi_p_eff = mcsolve(H_eff,psi0_p_eff,t,[],[],progress_bar=True,
    _safe_mode=True,options=options);
158
159
160 # In[14]:
161
162
163 #Fidelity
164 t1=list(t)
165 Fidlist=[]
166 tflist = []
167 a=1
168 iloop=range(0,100*M,a)
```

```

169 for l in iloop:
170     tf=t[l]
171     proj_p_eff = (psi_p_eff.states[l]).proj()
172     F_p=expect(proj_p_eff,(1j*H0*tf).expm()*(psi_p_full.states[l])) #
    Fidelity is the expect value of psi_p in the projector of psi_eff
173     Fidlist.append(F_p)
174     tflist.append(tf*Xi)
175
176
177 # In[15]:
178
179
180 #Plotting
181 plt.xlabel("Time (s)")
182 plt.ylabel("Fidelity")
183 plt.plot(tflist,Fidlist, label="$\eta=0.1$")
184
185
186 # In[ ]:

```

C.2 Porta \hat{X}

```

1 #!/usr/bin/env python
2 # coding: utf-8
3
4 # In[1]:
5
6
7 import numpy as np
8 import matplotlib as mpl
9 import scipy as sp
10 from qutip import *
11 from math import *
12 import math
13 import cmath
14 from matplotlib import *
15 import matplotlib.pyplot as plt
16
17 options=Options(atol=1e-10,nsteps=2000,store_states=True,
    store_final_state=True)
18
19 N =10; #basis dimension
20 nu=1.0; #field frequency (MHz)
21 nu_1x=1.0; nu_2x=nu; nu_1y=1.2; nu_2y=nu; #individual field frequencies
22 omega_0=50.0; #Internal frequency
23 timescaling=0.001 #factor to plot time scale in ms

```

```

24 eta_x=0.01;
25 F_omega=20.0; #setup of Rabi frequencies
26 Omega_1x=nu_1x/F_omega; Omega_1y=nu_1y/F_omega;
27 omega_1=omega_0-nu_1x; omega_2=omega_0+nu_1x;
28
29 def f_1(t, args): #ok
30     return cmath.exp(-1j*(omega_1*(t)+math.pi/2))
31
32 def f_1d(t, args): #ok
33     return cmath.exp(1j*(omega_1*(t)+math.pi/2))
34
35 def f_2(t, args): #ok
36     return cmath.exp(-1j*(omega_2*(t)+math.pi/2))
37
38 def f_2d(t, args):
39     return cmath.exp(1j*(omega_2*(t)+math.pi/2))
40
41 a = tensor(qeye(2), destroy(N));
42 Na=a.dag()*a;
43 NT=Na;
44 x_a=a+a.dag();
45
46 ##### Atom operators
47 # ###QD levels###
48 g_st = basis(2,0); e_st = basis(2,1);
49 plus_st = (1/np.sqrt(2))*(g_st+e_st); minus_st = (1/np.sqrt(2))*(g_st-
    e_st);
50
51 sig_p=tensor(sigmoid(), qeye(N));
52 sig_m=tensor(sigmoid(), qeye(N));
53 sig_x=tensor(sigmoid(), qeye(N));
54 sig_z=tensor(sigmoid(), qeye(N));
55
56 #
    =====
57 #                                     Hamiltonians
58 # =====
59 # Full Hamiltonian
60 # =====
61 # #Free Hamiltonian# #
62 H0=(nu_1x*Na)+(omega_0/2)*sig_z;
63
64 Op_x=(1j*eta_x*x_a);
65 #Op_y=(1j*eta_y*x_b);
66 Opd_x=(-1j*eta_x*x_a);
67 #Opd_y=(-1j*eta_y*x_b);

```

```

68
69 # H=(1/2)*0mega_1x*sig_p*(Op_x.expm())*(Op_y.expm());
70 # Hd=H.dag();
71
72 H=(1/2)*0mega_1x*sig_x*(Op_x.expm());
73 Hd=(1/2)*0mega_1x*sig_x*(Opd_x.expm());
74
75 # =====
76 # Effective sigma_x hamiltonian
77 # =====
78 H_eff=(1/2)*(0mega_1x*eta_x)*sig_x*x_a;
79
80 #
81 # =====
82 #
83 ###initial states of atom and field
84 psi0_atom_p=plus_st; #atom initial state
85 psi0_atom_m=minus_st;
86
87 psi0_field=basis(N,0); #field initial state
88
89 psi0_m_full=tensor(psi0_atom_m,psi0_field);
90 psi0_m_eff = psi0_m_full;
91 psi0_p_full=tensor(psi0_atom_p,psi0_field);
92 psi0_p_eff= psi0_p_full;
93
94 #
95 # =====
96 #
97 list_ops=[NT];
98 #
99 # =====
100 #
101
102 #coupling time dependence to H

```

```

103 H1t = [H,f_1]
104 H2t = [H,f_2]
105 H1dt = [Hd,f_1d]
106 H2dt = [Hd,f_2d]
107
108 H_full = [H0,H1t,H2t,H1dt,H2dt]
109
110
111 # In[2]:
112
113
114
115 Xi=(1/2)*(eta_x)*Omega_1x;
116 data_fidelity_p = np.vstack((0,1)).T
117 M=int(math.ceil(1/Xi))
118 t=np.linspace(0,M,M*100)
119 psi_p_full = mcsolve(H_full,psi0_p_full,t,[],[],progress_bar=True,
    _safe_mode=True,options=options);
120 psi_p_eff = mcsolve(H_eff,psi0_p_eff,t,[],[],progress_bar=True,
    _safe_mode=True,options=options);
121
122
123 # In[3]:
124
125
126 #Fidelity
127 t1=list(t)
128 Fidlist=[]
129 tflist = []
130 a=10
131 iloop=range(0,100*M,a)
132 for l in iloop:
133     tf=t1[l]
134     proj_p_eff = (psi_p_eff.states[l]).proj()
135     F_p=expect(proj_p_eff,(1j*H0*tf).expm()*(psi_p_full.states[l])) #
    Fidelity is the expect value of psi_p in the projector of psi_eff
136     Fidlist.append(F_p)
137     tflist.append(tf*Xi)
138
139
140 # In[4]:
141
142
143 #Plotting
144 plt.xlabel("Time")
145 plt.ylabel("Fidelity")
146 plt.plot(tflist,Fidlist, label="$\eta=0.01$")

```



```
147
148
149 # In[ ]:
150
151
152
153
154
155 # In[ ]:
```