

基于非酉量子线路的量子算法研究

(申请清华大学理学博士学位论文)

培养单位：物理系

学 科：物理学

研 究 生：谢 浩 楠

指 导 教 师：龙 桂 鲁 教 授

二〇二四年六月

Research on Quantum Algorithms Based on Nonunitary Quantum Circuits

Dissertation submitted to
Tsinghua University
in partial fulfillment of the requirement
for the degree of
Doctor of Philosophy
in
Physics
by
Xie Haonan

Dissertation Supervisor: Professor Long Guilu

June, 2024

学位论文公开评阅人和答辩委员会名单

公开评阅人名单

无（全隐名评阅）

答辩委员会名单

主席	何珂	教授	清华大学
委员	龙桂鲁	教授	清华大学
	阮东	教授	清华大学
	王川	教授	北京师范大学
	王铁军	教授	北京邮电大学
秘书	潘兴博	助理研究员	清华大学

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》及上级教育主管部门具体要求，向国家图书馆报送相应的学位论文。

本人保证遵守上述规定。

作者签名：_____

导师签名：_____

日 期：_____

日 期：_____

摘要

量子计算作为一种新兴计算技术，有望在后摩尔定律时代满足人类急速增长的算力需求，在无序搜索、质数分解、线性系统求解等问题中的优越表现引起了人们的广泛关注。量子计算的发展，特别是其应用，既需要高性能的量子硬件和实验技术，也需要高效的量子算法作为理论基础。因此，量子算法的设计与研究是量子计算走向实用化、发挥量子优越性的重要保证。

量子多体系统的模拟是物理学中的重要问题，在量子物理、量子化学、生物学及医学、组合优化等领域中都有重要应用。在经典计算机中，模拟量子系统的复杂度与体系中的粒子数呈指数关系，对于稍大的量子系统，如由几百或几千个原子构成的生物大分子，经典计算将变得极为困难。因此，针对量子多体系统模拟开发高效的量子算法成为重要的研究方向。另一方面，稀疏成像是根据对被观测物体的采样数据，对具有稀疏性的物体图像进行重建。大规模数据的稀疏成像存在计算量大、噪声与误差、参数设定等困难。因此，针对稀疏成像问题开展高效的量子算法研究同样具有重要意义。

已有的量子算法在模拟量子多体系统和实现稀疏成像问题中面临多种困难，其中几个突出的困难是需要大量的经典计算资源辅助、难以在现有条件下实验实现、无法给出普适的和具体的量子线路构造方式、稀疏成像结果通常不具备稀疏性。针对以上问题，我们开展了系统性研究，取得以下三项成果：

1. 我们设计了一种基于非酉量子线路的量子算法实现量子系统的虚时演化模拟，并将其应用于基态求解问题中。该算法在求解精度为 ϵ 时，时间复杂度为 $\mathcal{O}(\text{poly}(1/\epsilon))$ 。我们在超导和离子阱量子计算硬件上完成了演示实验，并且在经典计算机上进行了数值模拟，验证了算法的有效性。我们的算法提供了一种使用基本量子门构建量子线路的系统性方法，适合在实验上实现。

2. 我们将基于非酉量子线路的虚时演化算法应用于非厄米边缘突起效应的计算，并通过数值模拟验证了算法的有效性。我们的算法具有普适性，可以针对任意非厄米体系的动力学演化提供系统的量子线路构造方法。此外，我们还进一步研究了变分量子算法在非厄米体系模拟中的应用。

3. 我们利用非酉量子线路，设计了一种迭代收缩阈值量子稀疏成像算法。对于 N 维稀疏成像问题，在稀疏度为 s 且求解精度为 ϵ 的情况下，该算法的空间复杂度为 $\mathcal{O}(\log(N)/\epsilon^3)$ ，时间复杂度为 $\mathcal{O}(\log(N)s^2/\epsilon^3)$ 。我们通过数值模拟验证了算法的有效性，并在超导量子计算硬件上完成了原理演示。我们的算法可以优化

摘要

矩阵的一范数，从而在成像结果上体现稀疏性。

小结论文工作，本文构造的基于非酉线路的量子算法，可以应用于虚时演化、非厄米体系模拟、稀疏成像问题，算法的有效性通过演示实验与数值模拟得到了验证。这些算法补充了量子多体系统模拟问题与稀疏成像问题的相关量子算法研究，在未来的量子计算机中将发挥重要的实用价值。

关键词：非幺正量子算法；量子模拟；虚时演化；非厄米量子系统；稀疏成像

Abstract

Quantum computation, as a new emerging computational technology, is expected to meet the rapidly growing computational demands of humanity in the post-Moore's Law era. Its superior performance in problems such as unstructured database searching, prime factorization, and solving linear systems has garnered widespread attention. The development, especially the application of quantum computation requires not only high-performance quantum hardware and experimental techniques, but also efficient quantum algorithms as theoretical foundations. Therefore, the design and research of quantum algorithms are crucial guarantees for the practicality of quantum computation and the realization of quantum advantage.

The simulation of quantum many-body systems is a common problem in physics, with significant applications in fields such as quantum physics, quantum chemistry, biology and medicine, and combinatorial optimization. In classical computers, the complexity of simulating quantum systems grows exponentially with the number of particles in the system. For moderately large quantum systems, such as biomolecules composed of hundreds or thousands of atoms, classical computation becomes extremely challenging. Therefore, developing efficient quantum algorithm for simulating quantum many-body systems has become an important research direction. On the other hand, sparse imaging involves reconstructing images of objects with sparsity based on sampled data. Large-scale sparse imaging encounters difficulties such as large amount of computation, noise or error, parameter setting and so on. Therefore, it is also of great significance to conduct research on efficient quantum algorithms for sparse imaging problems.

The existing quantum algorithms for simulating quantum systems and sparse imaging face various challenges in practical applications, among which several prominent ones include the requirement for substantial classical computing resources, difficulties in experimental implementation under existing conditions, the inability to provide universal and specific quantum circuit construction methods, lack of sparsity in sparsifying imaging results. In response to these issues, we have conducted systematic research and achieved the following three outcomes:

1. We develop a quantum algorithm based on nonunitary quantum circuits to simulate the imaginary-time evolution of quantum systems, and apply it to ground state solving

Abstract

problems. The algorithm achieves a time complexity of $\mathcal{O}(\text{poly}(1/\epsilon))$ when solving to the precision ϵ . We validate the feasibility of the algorithm through demonstration experiments on superconducting and trapped-ion quantum hardwares and numerical simulations on classical computers. Our algorithm provides a systematic approach to construct the required quantum circuits using universal quantum gates, making it suitable for experimental implementation.

2. We apply the imaginary-time evolution algorithm based on nonunitary quantum circuits to the calculation of non-Hermitian edge burst, and verify its feasibility through numerical simulation. Our algorithm is universal, and provides a systematic method to construct the quantum circuits for the dynamical simulation of arbitrary non-Hermitian systems. Additionally, we further investigate the application of variational quantum algorithm in simulating non-Hermitian systems.

3. We design a quantum sparse imaging algorithm based on iterative shrinkage thresholding method using nonunitary quantum circuits. For an N -dimensional sparse imaging problem with sparsity s and accuracy ϵ , the algorithm achieves a space complexity of $\mathcal{O}(\log(N)/\epsilon^3)$ and a time complexity of $\mathcal{O}(\log(N)s^2/\epsilon^3)$. We verify the feasibility of the algorithm through numerical simulations and conduct a theoretical demonstration on a superconducting quantum hardware. Our algorithm optimizes the one-norm of the matrix, thus reflecting sparsity in the imaging results.

In conclusion, the quantum algorithm constructed in this paper based on nonunitary circuits can be applied to imaginary-time evolution, simulation of non-Hermitian systems, and sparse imaging problems. The feasibility of the algorithms has been demonstrated through experimental demonstrations and numerical simulations. These algorithms complement the research on quantum algorithms related to simulating quantum many-body systems and sparse imaging problems, and will play an important practical role in future quantum computers.

Keywords: non-unitary quantum algorithms; quantum simulation; imaginary-time evolution; non-Hermitian quantum systems; sparse imaging

目 录

摘要	I
Abstract	III
目录	V
插图清单	VIII
附表清单	XI
符号和缩略语说明	XII
第 1 章 绪论	1
1.1 量子力学简介	2
1.1.1 量子态与密度矩阵	2
1.1.2 量子态的演化与矩阵函数	3
1.1.3 量子测量	5
1.2 量子计算基本概念	6
1.2.1 量子比特	6
1.2.2 量子门与量子线路	7
1.3 量子计算的物理实现与云平台简介	12
1.4 本章小结	14
第 2 章 量子计算与常用量子算法介绍	17
2.1 算法的复杂度	17
2.2 几种基础的量子算法	18
2.2.1 量子搜索算法	18
2.2.2 量子傅立叶变换与相位估计算法	20
2.2.3 哈密顿量模拟算法	23
2.3 参数化量子线路与变分量子算法	24
2.3.1 变分量子本征求解器	26
2.3.2 量子近似优化算法	27
2.3.3 变分量子模拟	28
2.4 幺正算符线性组合及非幺正量子算法	30
2.4.1 幺正算符线性组合方法	31

目 录

2.4.2 全量子本征求解算法	32
2.5 线性系统的量子算法	33
2.5.1 量子线性方程组求解算法与矩阵函数的实现	33
2.5.2 量子主成分分析与量子奇异值阈值算法	35
2.6 本章小结	37
第 3 章 基于非幺正量子线路的概率性虚时演化算法	38
3.1 虚时演化及其相关算法	38
3.2 适用于泡利形式哈密顿量的概率性虚时演化算法	41
3.3 概率性虚时演化算法在求解基态问题中的应用	46
3.3.1 量子计算云平台中的氢分子基态求解演示实验	46
3.3.2 氢化锂分子与量子伊辛链基态求解的数值模拟	51
3.4 改进的概率性虚时演化算法及其应用	54
3.5 概率性虚时演化算法的复杂度讨论	59
3.6 本章小结	61
第 4 章 非厄米体系动力学演化的量子模拟	62
4.1 非厄米体系及其动力学演化	62
4.2 基于概率性虚时演化的非厄米体系动力学模拟	65
4.3 基于变分量子算法的非厄米体系动力学模拟	70
4.4 本章小结	75
第 5 章 量子稀疏成像算法	77
5.1 稀疏成像问题及相关算法	77
5.2 基于阈值迭代收缩算法的量子稀疏成像算法	79
5.3 量子稀疏成像算法在成像问题中的应用	88
5.4 本章小结	94
第 6 章 总结与展望	96
参考文献	98
致 谢	113
声 明	115
个人简历、在学期间完成的相关学术成果	116
指导教师评语	117

答辩委员会决议书	118
----------------	-----

插图清单

图 1.1 不同量子比特的状态在 Bloch 球中的表示.....	7
图 1.2 常见的单比特量子门示意图	9
图 1.3 常见的两比特量子门示意图及其构造方法	10
图 1.4 三比特 Toffoli 门及其构造方法.....	10
图 1.5 量子线路示意图中的元素	11
图 1.6 幺正量子线路流程图	11
图 1.7 非幺正量子线路流程图	12
图 1.8 Quafu 平台 10 比特超导量子处理器 P-10 的拓扑结构	14
图 1.9 IonQ 平台 11 比特离子阱量子处理器的拓扑结构	15
图 2.1 量子傅立叶变换算法的线路图	21
图 2.2 量子相位估计算法的线路图	22
图 2.3 实现两比特泡利项演化算符的量子线路	24
图 2.4 变分量子算法的基本框架	25
图 2.5 参数化量子线路的一层的例子	25
图 2.6 最大割问题的图的例子	28
图 2.7 QAOA 的参数化量子线路结构	28
图 2.8 测量两个量子态的内积实部的量子线路	30
图 2.9 实现幺正算符线性组合的量子线路	31
图 2.10 HHL 算法的量子线路图	33
图 2.11 QSVT 算法的量子线路图	37
图 3.1 实现虚时演化算符 \tilde{T}_k 的量子线路	41
图 3.2 在 5 比特量子线路中利用单比特门和 CNOT 门构造 U_k 的两个例子.....	42
图 3.3 在量子计算云平台上，利用 PITE 算法计算氢分子基态与基态能量的演示实验的量子线路图	50
图 3.4 在 Quafu 和 IonQ 云平台上，利用 PITE 算法求解氢分子基态能量的演示实验结果	51
图 3.5 在数值模拟中，利用 PITE 算法计算氢化锂分子与量子伊辛链的基态与基态能量（采用量子信道噪声模型）	53
图 3.6 在数值模拟中，利用 PITE 算法求解氢化锂分子与量子伊辛链的基态的累积成功概率（采用量子信道噪声模型）	54

图 3.7 在数值模拟中, 利用 PITE 算法计算氢化锂分子与量子伊辛链的基态与基态能量 (采用旋转角涨落噪声模型)	55
图 3.8 在数值模拟中, 利用 PITE 算法求解氢化锂分子与量子伊辛链的基态的累积成功概率 (采用旋转角涨落噪声模型)	56
图 3.9 利用 GPITE 算法求解氢化锂分子的基态时, 实现 H_{18} 项的虚时演化的量子线路图	58
图 3.10 利用 GPITE 算法求解量子伊辛链的基态时, 实现 H_k 项的虚时演化的量子线路图	59
图 3.11 在数值模拟中, 利用 GPITE 算法和高阶 Trotter 展开的 PITE 算法求解氢化锂分子与量子伊辛链的基态与基态能量, 以及算法的累积成功概率	60
图 4.1 非厄米格点体系中存在的边缘突起现象, 图片来自参考文献 ^[264]	64
图 4.2 链长 $L = 2$ 的非厄米格点模型	65
图 4.3 利用 Trotter 分解与 PITE 算法模拟 $L = 2$ 的非厄米格点体系所需要的量子线路	66
图 4.4 利用 Trotter 分解和 PITE 算法模拟 $L = 2$ 的非厄米格点体系演化及边缘突起现象	68
图 4.5 利用 Trotter 分解和 PITE 算法模拟 $L = 2$ 的非厄米格点体系演化的累积成功概率与格点体系中真实波函数的密度变化	69
图 4.6 利用 Trotter 分解和 PITE 算法模拟 $L = 4$ 的非厄米格点体系演化及边缘突起现象	70
图 4.7 利用 Trotter 分解和 PITE 算法模拟 $L = 4$ 的非厄米格点体系演化的累积成功概率与格点体系中真实波函数的密度变化	71
图 4.8 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系所使用的参数化量子线路	73
图 4.9 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系时, 测量 A_{jk} 和 B_j 所需要的量子线路	74
图 4.10 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系演化及边缘突起现象	75
图 4.11 利用 PITE 算法与利用变分量子算法模拟 $L = 2$ 的非厄米格点体系得到演化态的误差	76
图 5.1 利用各类算法对 Yak-42 飞机进行稀疏成像的结果, 图片来自参考文献 ^[272]	80
图 5.2 基于相位估计方法实现量子梯度下降的线路图	82
图 5.3 基于 LCU 方法实现量子梯度下降的线路图	83

插图清单

图 5.4 基于 LCU 方法，利用梯度算符的幺正演化实现量子梯度下降的线路图	84
图 5.5 根据量子态 $ X\rangle$ 制备密度矩阵 ρ_{XX^\dagger} 的量子线路图（以三比特为例）	85
图 5.6 实现阈值收缩操作的量子线路图	85
图 5.7 “THU”字样采样前的原始图像	88
图 5.8 不同方差的噪声下，利用采样矩阵的逆直接恢复“THU”字样的图像	89
图 5.9 不同方差的噪声下，利用 QISTA 对“THU”字样进行稀疏成像的结果	90
图 5.10 利用 QISTA 对“THU”字样进行稀疏成像迭代过程中的损失函数值与单步成功概率	91
图 5.11 利用 QISTA 对 Yak-42 飞机进行稀疏成像的结果	92
图 5.12 利用 QISTA 对 Yak-42 飞机进行稀疏成像迭代过程中的损失函数值与单步成功概率	92
图 5.13 在 QISTA 演示实验中所使用的量子线路	94
图 5.14 QISTA 演示实验得到的成像结果与原始图像对比	95

附表清单

表 1.1 Quafu 云平台中超导量子芯片 P-10 的性能参数	16
表 3.1 不同键长 R 下, 氢分子哈密顿量中各泡利项系数, 数据来自文献 ^[229]	47
表 3.2 最稳定键长 $R = 1.5\text{\AA}$ 下, 氢化锂分子哈密顿量中各泡利项系数, 以及在 GPITE 算法中泡利项的分组方式	48

符号和缩略语说明

ALB	近似下界
CNOT	受控翻转门
EP	异常点
FFT	快速傅立叶变换
FISTA	快速迭代阈值收缩算法
FQE	全量子本征求解器
GPITE	推广的概率性虚时演化
HF	Hartree-Fock
HHL	Harrow-Hassidim-Lloyd (算法)
I	单位矩阵
ISTA	迭代阈值收缩算法
ITE	虚时演化
NHEB	非厄米边缘突起
LASSO	最小完全收缩和选择算符
LCU	幺正算符线性组合
log	对数 (复杂度)
NISQ	含噪声中等规模量子
\mathcal{O}	渐进上界/算法复杂度
OMP	正交匹配追踪
PCA	主成分分析
PITE	概率性虚时演化
poly	多项式 (复杂度)
POVM	正算符值测量
PQC	参数化量子线路
PVM	投影值测量
QAOA	量子近似优化算法
QFT	量子傅立叶变换
QISTA	量子迭代阈值收缩算法
QITE	量子虚时演化
QPCA	量子主成分分析
QPE	量子相位估计

符号和缩略语说明

QRA	量子增强重构算法
QSVT	量子奇异值阈值
RD	距离-多普勒
RLB	严格下界
RSA	Rivest-Shamir-Adleman (加密协议)
R_x	单比特 x 轴旋转门
R_y	单比特 y 轴旋转门
R_z	单比特 z 轴旋转门
SWAP	两比特交换门
σ_0	二维单位矩阵
σ_x	泡利 x 矩阵
σ_y	泡利 y 矩阵
σ_z	泡利 z 矩阵
Toffoli	多比特受控翻转门
U	幺正算符/量子门
VFF	变分快进
VQA	变分量子算法
VQE	变分量子本征求解器
VQS	变分量子模拟
VQSL	变分量子态学习

第1章 绪论

量子力学的建立和发展是 20 世纪人类最伟大的科学成果之一，不仅推动了人类对微观世界的认知，也引发了工业和信息的科技革命。量子力学在发展的过程中逐渐与其它学科相融合，许多新领域与学科分支就此诞生，其中就包括量子信息科学。

在当今信息时代，计算机是人们的生产生活中不可或缺的工具。自 20 世纪 40 年代第一代计算机问世以来，计算机经历了日新月异的飞速发展，其成本越来越低、体积越来越小、运算速度越来越快、性能越来越优越。如今世界上运算速度最快的超级计算机峰值运算速度已达每秒百亿亿次的量级。英特尔（Intel）创始人之一戈登·摩尔（Gordon Moore）总结了信息技术进步的经验规律：当价格不变时，集成电路上可容纳的晶体管数目每隔约 18 到 24 个月翻一番，性能提升一倍。这一规律被称为“摩尔定律”，其并非自然科学定律，但在过去的几十年间很好地预测了信息技术的发展。然而随着集成电路元器件尺寸的减小，经典物理学逐渐失效，量子效应开始显现。摩尔定律已逐渐失效。另一方面，随着大数据时代到来和人工智能的发展，人类社会对算力的需求仍在快速增长，因此急需发展新的计算技术来突破传统计算技术的瓶颈。而新兴计算技术中，最有前景的发展方向之一就是量子计算。

量子计算和量子模拟的概念在上世纪 80 年代由 Benioff、Manin、Feynman 等人分别提出^[1-5]。受限于当时的实验技术，早期的量子计算只存在于理论构想中。几十年来，人们在量子计算的理论方面取得了众多重要成果，其中诞生了一系列可以解决重要基础问题的量子算法^[6-11]。这些量子算法所消耗的计算资源在数量级上都远低于解决同问题的经典算法，使得人们普遍相信量子计算具有超越经典计算的优越性，量子计算研究也因此成为持续的国际研究前沿。时至今日，量子计算研究已涵盖了众多领域，包括基础数学^[12-15]、密码学^[16-17]、量子物理与量子化学^[18-22]、量子机器学习^[23-24]、量子优化算法^[25-26]等。除了各类量子算法之外，量子计算的理论体系还包括量子纠错^[27-31]、复杂度理论^[32-34]等问题。这些领域的研究持续受到广泛关注，也为量子计算的大规模应用提供了重要的理论基础。

随着实验技术的发展，量子计算的实验实现也已取得诸多进展。人们已经在超导电路、离子阱、光学系统等众多物理体系中实现了量子计算^[35-40]，许多量子算法也陆续得到了实验验证^[41-44]。2019 年以来，研究人员相继在超导量子计算机与光量子计算机中验证了量子计算在特定问题上的计算效率上对经典计算的超

越^[45-50]，被称为“量子霸权”(quantum supremacy)。此外，国内外的一系列实验成果正在逐步提升量子计算与量子模拟的硬件性能与实验技术^[51-59]。其他实验成果还覆盖了量子存储^[60-63]、量子控制^[64-66]、量子传感^[67-69]等领域。这些成果使得量子计算领域持续成为关注热点，也为量子计算的实用化奠定了坚实的技术基础。

目前，量子计算领域发展到了含噪声中等规模量子（Noisy Intermediate Scale Quantum, NISQ）时代^[70]，基于 NISQ 设备的量子算法（如变分量子算法等）近年来受到广泛关注^[71-72]，有望能在 NISQ 时代利用量子计算机帮助解决有实际价值的问题。尽管量子计算已经能在特定问题上实现对经典计算的超越，但若要真正实现生产生活中的大规模应用、完全发挥量子计算优势，在硬件和软件方面仍需要继续探索和研究。

1.1 量子力学简介

量子力学的诞生与发展源于 19 世纪末经典物理学在某些问题上的失效。人们发现对于氢原子结构、黑体辐射等问题，经典物理理论无法对实验现象进行准确描述。20 世纪初，随着玻尔氢原子模型、光的量子化、德布罗意波等理论的提出，量子力学逐渐成型，并迅速发展成熟。尽管量子力学的很多结论与人们的日常生活经验不符，但它能很好地解决许多经典物理学无法解决的问题，如黑体辐射、光电效应、氢原子结构等，给人类对微观领域的认知带来了革命性的突破。在本节中，我们不针对量子力学进行系统的详细讲述，仅对与本论文内容相关的部分进行简要介绍^[73-75]。

1.1.1 量子态与密度矩阵

一个量子系统在某一时刻所处的状态可以用希尔伯特空间中的一个态来表示，利用 Dirac 符号将其记为 $|\psi\rangle$ ，其数学形式等价于一个 N 维复线性空间中的列向量，其中 N 是量子系统的维度或自由度。Dirac 符号 $|\cdot\rangle$ 称为右矢，代表列向量，同理也存在左矢 $\langle\cdot|$ ，代表行向量。相互对应的左式与右矢之间互为共轭转置，即 $\langle\psi| = |\psi\rangle^\dagger$, $|\psi\rangle = \langle\psi|^\dagger$ 。左矢和右矢之间具有两种运算： $|\psi_1\rangle\langle\psi_2|$ 叫做外积，运算结果是一个矩阵或算符； $\langle\psi_1|\psi_2\rangle$ 叫做内积，运算结果是一个复数。

任意一个量子态都可以表示为一组基矢的叠加：

$$|\psi\rangle = \sum_{i=0}^{N-1} c_i |\phi_i\rangle, \quad (1.1)$$

其中 c_i 为展开系数， $\{|\phi_i\rangle\}$ 是希尔伯特空间中的一组完备的正交归一基，其完备

性与正交归一性分别对应如下两个式子：

$$\sum_{i=0}^{N-1} |\phi_i\rangle\langle\phi_i| = \mathbf{I}_N, \quad (1.2)$$

$$\langle\phi_i|\phi_j\rangle = \delta_{ij}. \quad (1.3)$$

基组 $\{|\phi_i\rangle\}$ 可以人为选取，但一般都尽量使计算和表达更简便。此外，系数 c_i 的模平方等于在基组 $\{|\phi_i\rangle\}$ 上测量量子态得到结果为 $|\phi_i\rangle$ 的概率，由所有结果概率之和为 1，可得系数应满足归一化条件 $\sum_i |c_i|^2 = 1$ 。

对于多个量子系统构成的组合系统，其量子态的维数是所有子系统量子态维数的乘积，基矢是所有子系统基矢的直积。例如 N_A 维系统 A 的基矢为 $|\phi_i^A\rangle$ ($i = 0, 1, \dots, N_A - 1$)， N_B 维系统 B 的基矢为 $|\phi_j^B\rangle$ ($j = 0, 1, \dots, N_B - 1$)，则 A 和 B 构成的组合系统的基矢为 $|\phi_i^A\rangle \otimes |\phi_j^B\rangle$ ，通常简写为 $|\phi_i^A\rangle|\phi_j^B\rangle$ 或 $|\phi_i^A\phi_j^B\rangle$ ，且组合系统的任意量子态都可以写成如下形式：

$$|\psi^{AB}\rangle = \sum_{i=0}^{N_A-1} \sum_{j=0}^{N_B-1} c_{ij} |\phi_i^A\rangle|\phi_j^B\rangle. \quad (1.4)$$

以上提到的量子态都是纯态，用一个复向量就可以描述。然而当我们面对量子系统或与环境存在相互作用的量子系统时，纯态并不能准确描述系统的性质。这时需要引入混态和密度矩阵这两个概念。

假设一个量子系统，它有 p_k 的概率处在 $|\psi_k\rangle$ 量子态 ($k = 0, 1, \dots, M - 1$)，则称该量子系统的状态为混态。定义该量子系统的密度矩阵为：

$$\rho = \sum_{k=1}^{M-1} p_k |\psi_k\rangle\langle\psi_k|. \quad (1.5)$$

根据此定义，纯态 $|\psi\rangle$ 的密度矩阵为 $\rho = |\psi\rangle\langle\psi|$ 。

密度矩阵有如下三个性质：(1) 厄米性： $\rho = \rho^\dagger$ ；(2) 半正定性：对任意 $|\phi\rangle$ 都有 $\langle\phi|\rho|\phi\rangle \geq 0$ ；(3) 矫迹性： $\text{tr}(\rho) = 1$ 。此外，纯态的密度矩阵还具有一个额外性质——幂等性： $\rho^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = |\psi\rangle\langle\psi| = \rho$ ，因此 $\text{tr}(\rho^2) = 1$ ；而对于混态， $\text{tr}(\rho^2) < 1$ ，因此可以通过密度矩阵平方的迹是否为 1 来判断量子系统是否处于纯态。

1.1.2 量子态的演化与矩阵函数

一个孤立的量子系统，其动力学性质由系统的哈密顿量 (Hamiltonian) 决定，哈密顿量记为 \mathbf{H} ，是一个厄米算符。孤立系统的量子态随时间的演化满足薛定谔

方程^[76]:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = \mathbf{H}(t) |\psi(t)\rangle, \quad (1.6)$$

其中 \hbar 为约化普朗克常数, 以后将其记为 1。

若已知 t_1 时刻量子系统的量子态 $|\psi(t_1)\rangle$, 可根据薛定谔方程得到此后 t_2 时刻的量子态:

$$|\psi(t_2)\rangle = \exp \left[-i \int_{t_1}^{t_2} \mathbf{H}(\tau) d\tau \right] |\psi(t_1)\rangle, \quad (1.7)$$

其中 $\exp \left[-i \int_{t_1}^{t_2} \mathbf{H}(\tau) d\tau \right] \equiv \mathbf{U}(t_1, t_2)$ 称为时间演化算符, 或简称演化算符, 它是一个么正算符, 满足 $\mathbf{U}^\dagger \mathbf{U} = \mathbf{U} \mathbf{U}^\dagger = \mathbf{I}$ 。若采用密度矩阵描述量子系统, 已知 t_1 时刻的密度矩阵 $\rho(t_1)$, 则密度矩阵的演化方程以及 t_2 时刻的密度矩阵为

$$\frac{d\rho(t)}{dt} = -i [\mathbf{H}(t), \rho(t)], \quad \rho(t_2) = \mathbf{U}(t_1, t_2) \rho(t_1) \mathbf{U}^\dagger(t_1, t_2). \quad (1.8)$$

对于哈密顿量恒定的系统, 演化算符为 $\mathbf{U}(t_1, t_2) = e^{-i\mathbf{H}(t_2-t_1)} = e^{-i\mathbf{H}\Delta t}$, 它只与时间间隔 Δt 有关, 因此可以定义 $\mathbf{U}(\Delta t) = e^{-i\mathbf{H}\Delta t}$ 。

从形式上看, 时间演化算符是将哈密顿量放在指数上, 是哈密顿量的指数函数, 我们需要给出哈密顿量指数的严格定义。由于哈密顿量为厄米算符, 其本征值为实数, 且不同本征值对应的本征态相互正交, 因此可以写出如下公式:

$$\mathbf{H} = \sum_{j=0}^{N-1} E_j |E_j\rangle \langle E_j|, \quad (1.9)$$

其中 E_j 是哈密顿量的本征值, $|E_j\rangle$ 是相应的本征态。这一式子被称为谱分解公式。根据哈密顿量的谱分解, 我们将时间演化算符定义为:

$$\mathbf{U}(\Delta t) = e^{-i\mathbf{H}\Delta t} = \sum_{j=0}^{N-1} e^{-iE_j\Delta t} |E_j\rangle \langle E_j|. \quad (1.10)$$

进一步, 我们可以将哈密顿量的指数推广至更广泛的矩阵函数^[77-78]。矩阵函数是将我们平常所接触的复数域上的函数 $f : \mathbb{C} \rightarrow \mathbb{C}$ 推广为从矩阵到矩阵的映射。对于可以写成谱分解的矩阵 $\mathbf{A} = \sum_{i=0}^{N-1} \lambda_i |\lambda_i\rangle \langle \lambda_i|$, 定义矩阵函数为: $f(\mathbf{A}) = \sum_{i=0}^{N-1} f(\lambda_i) |\lambda_i\rangle \langle \lambda_i|$ 。由于厄米矩阵和么正矩阵都具有谱分解, 因此都可以根据这一定义给出矩阵函数。

对于不能写出谱分解的矩阵, 通常采用另一种方式定义矩阵函数: 设函数 f 在 x_0 的某个邻域内有多项式展开 $f(x) = \sum_{n=0}^{\infty} c_n (x - x_0)^n$, 则矩阵 \mathbf{A} 的函数定义为 $f(\mathbf{A}) = \sum_{n=0}^{\infty} c_n (\mathbf{A} - x_0 \mathbf{I})^n$ 。这种定义方式的缺点在于较难判断矩阵多项式的收敛性, 但对矩阵 \mathbf{A} 的性质没有特殊要求。可以证明, 对于具有谱分解的矩阵, 如

厄米矩阵或么正矩阵，基于谱分解定义的矩阵函数和基于多项式展开定义的矩阵函数是等价的。

除此之外，还可以根据矩阵的奇异值分解来定义矩阵函数，这里不再详细介绍。常用的矩阵函数的形式包括多项式、反比例函数、指数函数、对数函数、三角函数等，它们在量子系统动力学演化^[79]、虚时演化^[80]、本征值求解^[81]、线性方程组求解^[82]等问题中都有重要应用。

1.1.3 量子测量

量子力学的基本公设认为，量子系统的可观测量 \mathbf{O} 是厄米算符，满足 $\mathbf{O} = \sum_i \lambda_i \mathbf{P}_i$ ，其中 $\mathbf{P}_i = |\lambda_i\rangle\langle\lambda_i|$ 为投影算符，满足正交性 $\mathbf{P}_i \mathbf{P}_j = \delta_{ij} \mathbf{P}_i$ 和完备性 $\sum_i \mathbf{P}_i = \mathbf{I}$ 。对量子系统的投影值测量（projection-valued measure, PVM）可以用投影算符集 $\{\mathbf{P}_i\}$ 表示：如果对纯态 $|\psi\rangle$ 或密度矩阵 ρ 进行投影值测量，则得到观测值 λ_k 的概率 $\text{Prob}(\lambda_k)$ 以及测量引起的量子态变化分别为：

$$\begin{aligned} \text{纯态: } \text{Prob}(\lambda_k) &= \langle\psi|\mathbf{P}_k|\psi\rangle, \quad |\psi\rangle \xrightarrow{\text{PVM}} \frac{\mathbf{P}_k|\psi\rangle}{\sqrt{\langle\psi|\mathbf{P}_k|\psi\rangle}} = |\lambda_k\rangle, \\ \text{密度矩阵: } \text{Prob}(\lambda_k) &= \text{tr}(\mathbf{P}_k \rho), \quad \rho \xrightarrow{\text{PVM}} \frac{\mathbf{P}_k \rho \mathbf{P}_k}{\text{tr}(\mathbf{P}_k \rho)}. \end{aligned} \quad (1.11)$$

由测量而引发的量子态变化被称为量子态的坍缩，这一过程是不可逆的。

通常我们关心可观测量的测量期望值 $\langle\mathbf{O}\rangle$ 。经验上，期望值可通过对量子态的多次重复测量得到。然而量子系统一旦被测量，其状态就会坍缩，因此一般我们需要制备大量完全相同的独立的量子态，然后在每个独立的态上分别进行可观测量 \mathbf{O} 的测量，再统计测量值的平均值即可得到 $\langle\mathbf{O}\rangle$ 。对纯态 $|\psi\rangle$ 或密度矩阵 ρ 进行测量得到的期望值分别为

$$\begin{aligned} \text{纯态: } \langle\mathbf{O}\rangle &= \sum_{k=0}^{N-1} \lambda_k \langle\psi|\mathbf{P}_k|\psi\rangle = \langle\psi|\mathbf{O}|\psi\rangle, \\ \text{密度矩阵: } \langle\mathbf{O}\rangle &= \sum_{k=0}^{N-1} \lambda_k \text{tr}(\mathbf{P}_k \rho) = \text{tr}(\mathbf{O} \rho). \end{aligned} \quad (1.12)$$

投影值测量是理想条件下的测量。对投影值测量的一种常用的推广被称为正算符值测量（positive operator-valued measure, POVM）。有限维量子系统的 POVM 可以用一组半正定算符集 $\{\mathbf{M}_i\}$ 表示，满足完备性 $\sum_i \mathbf{M}_i = \mathbf{I}$ ，但对正交性无要求。线性代数中的结论告诉我们，半正定算符一定可以写成 $\mathbf{M}_i = \mathbf{E}_i^\dagger \mathbf{E}_i$ 的形式，其中 \mathbf{E}_i 称为 Kraus 算符。如果对密度矩阵 ρ 进行 POVM，则得到 \mathbf{M}_k 所对应的观测值

的概率以及测量引起的量子态变化分别为：

$$\text{Prob}(\mathbf{M}_k) = \text{tr}(\mathbf{M}_k \rho), \quad \rho \xrightarrow{\text{POVM}} \frac{\mathbf{E}_k \rho \mathbf{E}_k^\dagger}{\text{tr}(\mathbf{M}_k \rho)}. \quad (1.13)$$

当测量算符 \mathbf{M}_i 限定为投影算符 \mathbf{P}_i 时, Kraus 算符为 $\mathbf{E}_i = \mathbf{P}_i$, 则 POVM 完全退化为 PVM。

1.2 量子计算基本概念

计算机进行运算的本质是利用一个人工设计的系统 A 去模拟另一个系统 B。通过某种方式, 人们将 B 中已知的信息编码在 A 中, 对 A 进行特定操作并对操作后的结果进行测量, 从而得到 B 中未知的信息。对于经典计算而言, 这种人工设计的系统是经典系统, 而量子计算则是将其替换为量子系统。由于量子系统具有不同于经典系统的特殊性质, 量子计算也被赋予了更强的能力。

对于一台计算机而言, 其基本要素包括比特、逻辑门、线路、测量等。在本节中, 我们分别介绍量子计算中如何利用量子力学原理实现这些要素^[78]。

1.2.1 量子比特

经典比特 (bit) 是经典计算机中最基本的数据存储单元。一个比特具有两种可能的稳定物理状态, 例如电平高低、磁矩方向等。我们将这两种状态分别记为二进制字符 0 和 1。一个由 n 个经典比特构成的系统具有 2^n 种可能的状态, 可以用一个长度为 n 的二进制字符串来表示。

量子比特 (qubit) 是量子计算机中最基本的数据存储单元。一个量子比特通常是一个二能级物理系统, 或者一个多能级物理系统中的两个特定能级, 如原子轨道、自旋等。我们将两个能级分别记为 $|0\rangle$ 和 $|1\rangle$, 分别对应经典计算中的二进制字符 0 和 1。在矩阵形式下, 两个能级通常表示为

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (1.14)$$

一个任意量子比特的状态可以表示成 $|0\rangle$ 和 $|1\rangle$ 的如下线性组合:

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) = e^{i\gamma} \begin{bmatrix} \cos \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} \end{bmatrix}, \quad (1.15)$$

其中 $0 \leq \theta \leq \pi$ 且 $0 \leq \phi, \gamma < 2\pi$ 。该式中的 $e^{i\gamma}$ 叫做全局相位因子, γ 叫做全局相位, 其取值不会影响任何测量结果, 因此通常略去因子 $e^{i\gamma}$ 。

通常, 我们利用 Bloch 球来更直观地观察单个量子比特的状态。对于如 1.15 式

所示的量子态，略去 $e^{i\gamma}$ 后，可以将其中的参数 θ 和 φ 与单位球面上的一个点对应起来，如图1.1所示。 θ 和 φ 分别对应球坐标的极角和方位角，球的“北极点”和“南极点”对应的量子态分别是 $|0\rangle$ 和 $|1\rangle$ 。

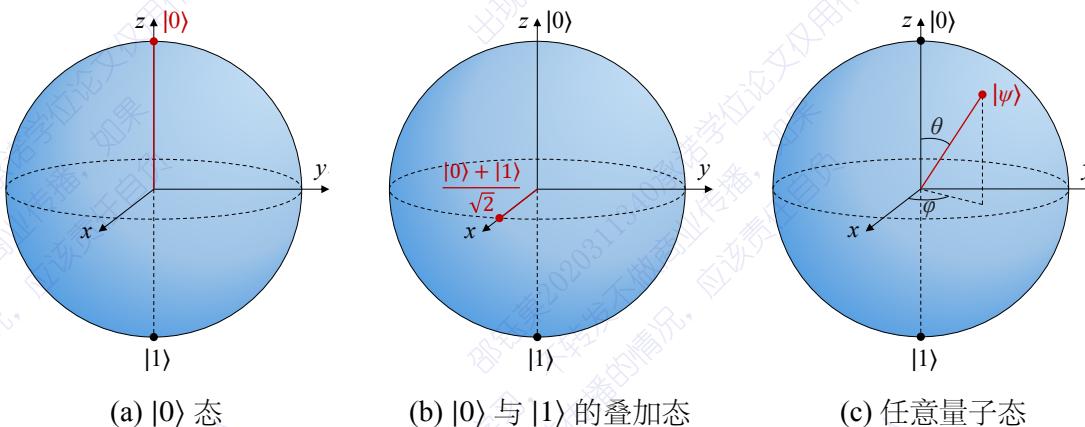


图 1.1 不同量子比特的状态在 Bloch 球中的表示

一个由 n 个量子比特构成的系统具有 2^n 个本征态。用长度为 n 的二进制字符串表示这些态，分别记为 $|00 \dots 00\rangle, |00 \dots 01\rangle, |00 \dots 10\rangle, \dots, |11 \dots 11\rangle$ 。有时我们也将这些二进制字符串用十进制数字表示，则分别记为 $|0\rangle, |1\rangle, |2\rangle, \dots, |2^n - 1\rangle$ 。二进制和十进制这两种表示方法都经常使用，有时还经常互换使用。此后若不加说明，量子比特的量子态都选取二进制态或对应的十进制态为基矢，这样的基矢称为计算基。

1.2.2 量子门与量子线路

逻辑门是对比特中的二进制信息进行逻辑转换的操作。在经典计算机中，基本的逻辑门包括非门、与门、或门、异或门、与非门、或非门等，而实际问题中所用的经典逻辑运算十分复杂，需要利用基本逻辑门构建结构更大、功能更复杂的逻辑线路，从而实现所需要的逻辑运算。

量子计算中的逻辑门称为量子门，它是对量子比特所进行的操作。量子门的数学形式是幺正矩阵，其可以保证作用前后的量子态模长始终为 1，也可以保证量子计算具有可逆性。

幺正算符有无穷多个，在量子计算机中直接实现任意的幺正算符是不可能的。因此就像经典逻辑门一样，我们需要找到一些基本的量子门，使得任意的幺正算符都可以通过将这些基本量子门按一定次序操作来近似实现。这样的一组基本量子门就叫做普适量子门。

最简单的一类量子门是单比特量子门，常见的单比特门包括 X 门、Y 门、Z

门、H门，矩阵形式分别如下：

$$\mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (1.16)$$

其中，X门与经典非门一样，将0和1互换，也称为比特翻转门；Z门则翻转了 $|1\rangle$ 态的相位；H门全称Hadamard门，是最常用的单比特门之一，它经常用来产生均匀叠加态。上面的X门、Y门、Z门的矩阵形式分别是泡利矩阵 $\sigma_x, \sigma_y, \sigma_z$ ，它们既是幺正矩阵也是厄米矩阵，既是量子门也是可观测量。另外还有两个单比特门有时也会用到，分别是S门和T门：

$$\mathbf{S} = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \quad (1.17)$$

此外还有一类常用的含参数的单比特门，叫做旋转门，它们是：

$$\begin{aligned} \mathbf{R}_x(\theta) &= e^{-i\theta\sigma_x/2} = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \\ \mathbf{R}_y(\theta) &= e^{-i\theta\sigma_y/2} = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix}, \\ \mathbf{R}_z(\theta) &= e^{-i\theta\sigma_z/2} = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}. \end{aligned} \quad (1.18)$$

从几何角度理解，上面三个量子门的作用是使Bloch球上的点分别绕 x 轴、 y 轴、 z 轴的正方向旋转 θ 角度。实际上，任意的单比特门（即任意的二维幺正矩阵）都可以理解为在Bloch球上绕某个轴的旋转，有如下形式：

$$\mathbf{U} = e^{i\alpha} \mathbf{R}_n(\theta), \quad (1.19)$$

其中 $e^{i\alpha}$ 是全局相位因子， $\mathbf{R}_n(\theta)$ 表示在Bloch球上绕单位向量 n 旋转 θ 角度，满足：

$$\mathbf{R}_n(\theta) = e^{-i\theta(n_x\sigma_x + n_y\sigma_y + n_z\sigma_z)/2} = \cos \frac{\theta}{2} \mathbf{I} - i \sin \frac{\theta}{2} (n_x\sigma_x + n_y\sigma_y + n_z\sigma_z). \quad (1.20)$$

在实际的物理系统中，往往无法实现绕任意轴的旋转，因此常将绕任意轴的旋转拆分为若干个绕固定轴的旋转。例如，在欧拉角表示下， $\mathbf{R}_n(\theta)$ 可以写成如下形式：

$$\mathbf{R}_n(\theta) = \mathbf{R}_z(\beta) \mathbf{R}_y(\gamma) \mathbf{R}_z(\delta), \quad (1.21)$$

其中 β, γ, δ 就是欧拉角。这一结论表明，只要能够实现两种绕固定轴的旋转操作，就可以构建出任意的单比特量子门。

在量子计算中，能够产生和消除量子纠缠的操作主要是两比特量子门，这也

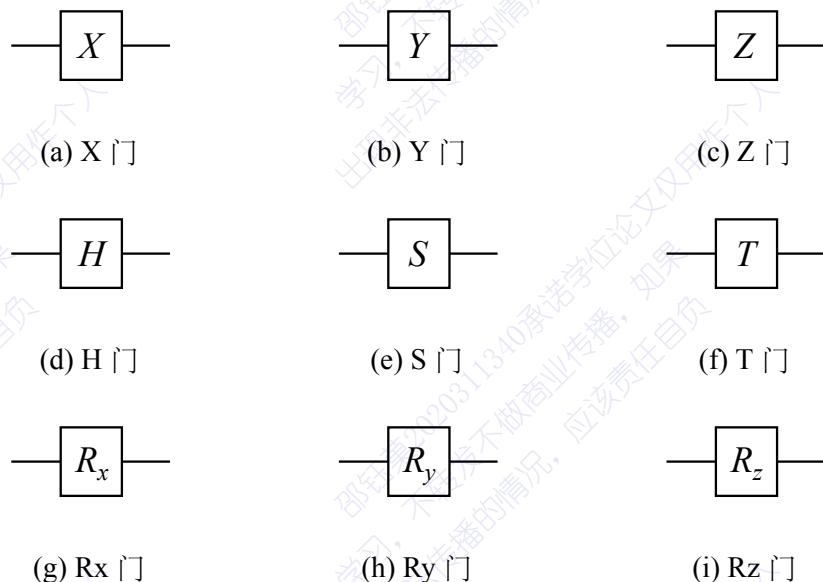


图 1.2 常见的单比特量子门示意图

是量子计算能产生加速的主要原因。最常用的两比特量子门是受控翻转门，称为 CNOT 门，它包含一个控制比特和一个目标比特，当控制比特为 0 时，目标比特保持不变；当控制比特为 1 时，目标比特被翻转（即被施加 \mathbf{X} 操作）。CNOT 门的矩阵形式为：

$$\text{CNOT} = |0\rangle\langle 0| \otimes \mathbf{I} + |1\rangle\langle 1| \otimes \mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1.22)$$

量子线路中的 CNOT 门示意图如图 1.3(a) 所示，其中上面的比特是控制比特，下面的比特是目标比特。

利用 CNOT 门和单比特门，我们可以构造出更多类型的两比特量子门。例如：0 控 NOT 门，它与 CNOT 门相反，当控制比特为 0 时目标比特翻转，当控制比特为 1 时目标比特不变；CZ 门，即受控 Z 门，当控制比特为 0 时目标比特保持不变，当控制比特为 1 时目标比特被施加 \mathbf{Z} 操作；CR x 门，即受控 Rx 门，当控制比特为 0 时目标比特保持不变，当控制比特为 1 时目标比特被施加旋转操作 $R_x(\theta)$ ，与此同理的还有 CR y 门和 CR z 门；SWAP 门，使两个比特的状态互换。图 1.3 展示了部分两比特门示意图的例子，以及如何利用 CNOT 门和单比特门构造它们。

除此之外，还有一种多比特受控量子门叫做 Toffoli 门，它具有至少两个控制比特，以及一个目标比特。当所有控制比特都是 1 时，目标比特被翻转，否则目标比特不变。Toffoli 门也可以由 CNOT 门和单比特门构造出来，一个三比特 Toffoli

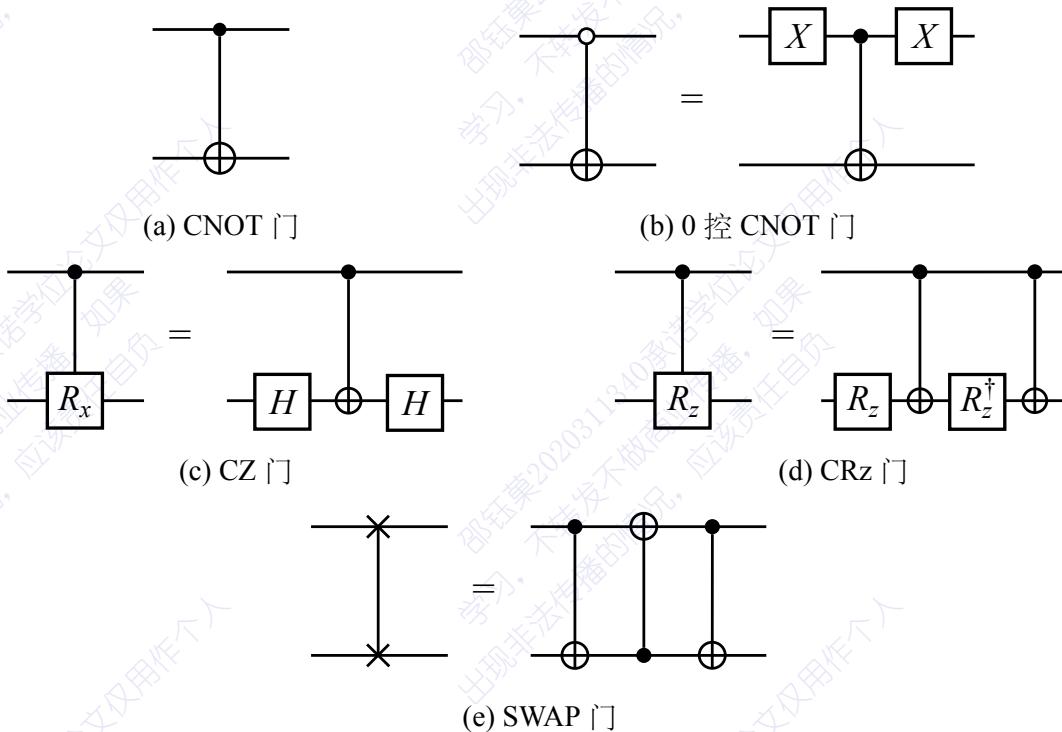


图 1.3 常见的两比特量子门示意图及其构造方法

门的示意图和构造方法如图1.4所示，其中上面的两个比特控制着最下面的比特的翻转。

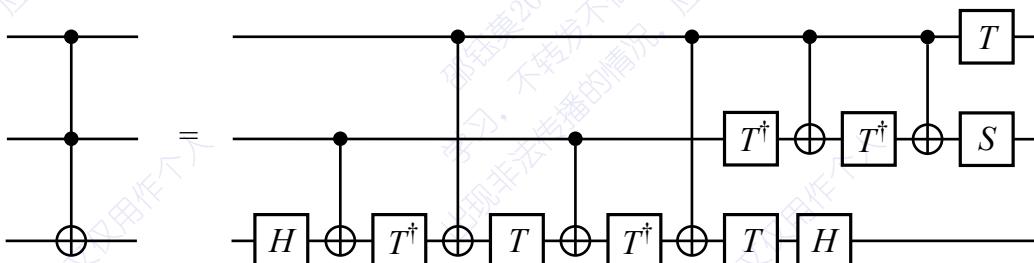


图 1.4 三比特 Toffoli 门及其构造方法

在一个实现量子计算的物理系统中，如果能够实现任意单比特门和 CNOT 门，那么就可以利用它们实现出任意的幺正操作。但在一些系统中，任意角度的单比特旋转操作较难实现，且含有连续变量的旋转操作会使量子纠错变得困难，因此通常用一组离散的量子门作为一个可纠错的完备量子门集合，使得任意一个幺正算符都可以利用这组量子门以任意精度实现。Hadamard 门、S 门、CNOT 门和 T 门就是满足这一条件的量子门集合。

将量子门按照一定次序作用在量子比特上，即可得到量子线路。量子线路的示意图中包含单量子比特线、多量子比特线、投影测量等元素，它们的示意图如图1.5所示。

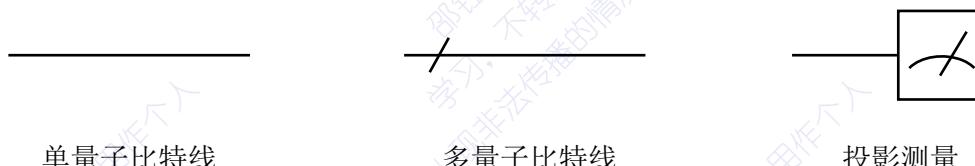


图 1.5 量子线路示意图中的元素

一般来说，一个完整的量子线路通常如图1.6所示，其中量子比特分为两类：一类是工作比特，用于储存初始数据、参与运算过程并储存运算结果，另一类是辅助比特，并不携带与运算直接相关的数据，仅对运算过程起辅助作用。有的量子线路中并不需要辅助比特。量子比特从量子线路图左侧开始，向右依次经过各个操作，包括量子比特的初始化、初始态制备、量子算法过程以及读出操作，最终从线路最右端输出。后面将介绍的 Grover 算法、量子傅立叶变换算法等都是这样的线路模式。在这样的量子线路中，除了测量之外的所有操作都是幺正操作，因此其实现的是幺正的量子算法。

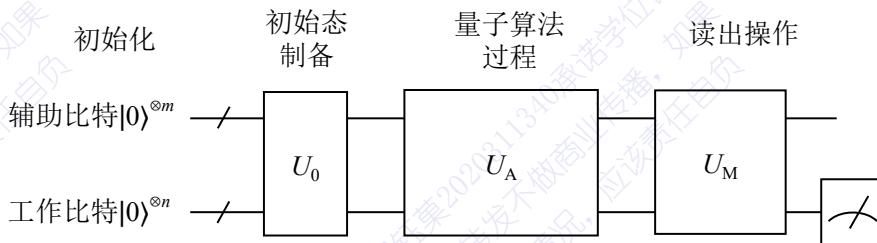


图 1.6 幺正量子线路流程图

然而随着理论的发展，人们越来越多地需要在量子线路中实现非幺正操作。这时就需要使用非幺正量子线路^[83]。一个完整的非幺正量子线路通常如图1.7所示，其中在量子算法过程结束后，需要先对辅助比特进行测量；如果测量得到某个或某些特定的结果，则继续对工作比特进行后续的读出操作；而如果测量没有得到特定的结果，则需要重新运行该线路，直到对辅助比特的测量得到特定结果为止。这一过程通常也称测量后选择。这样的线路就可以等效地实现非幺正运算，后面将介绍的幺正算符线性组合采用的就是这样的量子线路。以幺正算符线性组合为代表的非幺正量子线路，拓展了量子计算所能实现的功能，为量子算法的设计和构造提供了更多可能性。

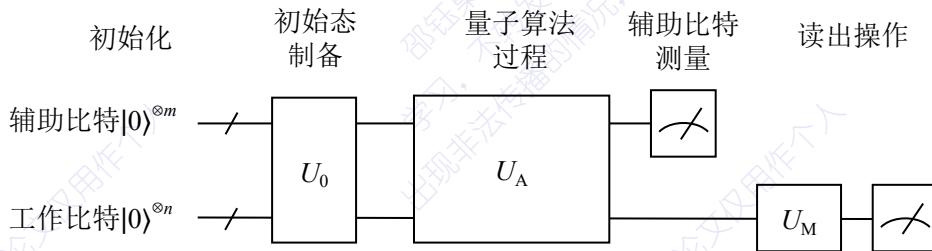


图 1.7 非幺正量子线路流程图

1.3 量子计算的物理实现与云平台简介

在真实的物理系统中实现量子计算需要考虑很多因素。DiVincenzo 于 2000 年提出了在实际的物理系统中实现量子计算需要考虑的准则^[84]，包括以下五点：

- 可以构建特征良好的量子比特，且具有可扩展性；
- 可以将量子比特的态初始化为特定的初态；
- 退相干时间足够长，至少要长于量子门操作的时间；
- 可以实现完备的量子门集合；
- 可以对量子比特的态进行读出。

除此之外，我们有时还要考虑系统的纠错能力。总而言之，我们希望用来实现量子计算的物理系统具有足够好的鲁棒性和可操作性。目前，人们已经找到了众多适合实现量子计算的物理系统，它们具有不同的优缺点，适用于不同场景，当前也处在不同的发展阶段。这些物理实现方法包括超导约瑟夫森结^[35,85-92]、囚禁离子阱^[36,93-96]、半导体量子点^[37,97-98]、光学系统^[38,49-50,99-100]、金刚石-氮空位中心^[39,101-102]、核磁共振^[40,103-105]、拓扑系统^[106-108]等。下面简单介绍几种较为重要的量子计算物理实现方法。

利用超导体实现量子计算是目前发展最为迅速、也是目前最受关注的通用量子计算实现方法之一。在两块超导体之间夹一层很薄的半导体或绝缘体，就会形成一种叫做约瑟夫森结（Josephson junction）的结构^[109]。由于超导体的宏观量子性质，约瑟夫森结中的电子会产生不同的能级^[110]，利用其中特定的两个能级就可以构建量子比特。在实际的应用中，人们把约瑟夫森结与超导电路相结合，构成复杂的电路元件，并通过电学原理来实现超导量子比特的初始化、量子门、测量等操作^[86,89-90]。超导量子计算的门操作速度快，量子比特的构建、控制和测量都是利用电路实现，其硬件加工和操控技术更接近传统半导体工艺，易于集成和封装。也正是因为这一点，超导量子计算是目前最具有商用前景的一种量子计算机方案，近几年来发展迅速，代表性成果包括：2019 年，谷歌公司（Google）在可编程超导量子芯片“Sycamore”上运行了随机量子线路采样，200 秒内采样百万次，而预估

当时最快的超级计算机需要一万年才能完成同样的工作，因此宣称实现了“量子霸权”^[45]，尽管后续工作大大压缩了经典计算机预估完成任务的时间，但这一成果仍代表着量子计算首次实现了在特定问题上对经典计算的超越；2021年，中国科学技术大学团队发布了分别具有62和66比特的可编程超导量子芯片“祖冲之号”和“祖冲之二号”，其中在“祖冲之号”上完成了高保真度量子随机行走模拟^[46]，在“祖冲之二号”耗时1.2-4.2小时实现了上千万次随机量子线路采样^[47-48]，预估当时的超级计算机完成同样的工作需要数百至数万年；2023年，IBM公司发布最新的超导量子芯片“Condor”^[111]，具有1121个超导量子比特，超导量子芯片比特数达到四位数。

囚禁离子阱（trapped ion）方法也是目前备受关注的量子计算实现方法之一。某些金属离子如¹⁷¹Yb⁺，其最外层电子具有类氢原子的能级，只要选取合适的两个能级就可以构建量子比特。离子阱利用交变电磁场，将多个相同离子囚禁并悬浮在容器中央，通过对离子施加激光脉冲，就可以控制电子在能级之间的跃迁以及离子之间的相互作用，从而实现量子门^[112]。量子态的初态制备和读取则利用光学手段。目前IonQ、Quantinuum等公司均在离子阱量子计算方面持续取得进展^[113-114]。

光学系统方法利用光的偏振状态构建量子比特，并通过多模干涉等光学方式实现量子操作，既可以构造通用量子计算机，也可以构造解决特定问题的专用量子模拟机。代表性成果包括，2020至2021年中国科学技术大学团队成功构造“九章”和“九章二号”光量子计算机^[49-50]，利用双模压缩态和多模干涉仪实现了高斯玻色采样，采样速率分别比当时的超级计算机快10¹⁴和10²⁴倍。

其他量子计算物理实现方法中，半导体量子点方法发展较晚，但其与传统半导体工艺兼容性高，十分具有发展潜力；金刚石-氮空位中心与核磁共振方法的可扩展性不强，难以实现可实用化的大规模量子计算，目前常用于算法演示、教学展示、量子测量等场景；拓扑方法目前仅存于理论中，实验上可用来构建量子比特的物理特征尚未被证实。这里我们对这些方法不再具体介绍。

近年来，量子计算云平台发展迅速，使得不具备量子计算实验条件的研究者也可以通过网络连接到有相应条件的量子计算实验设备的服务器中，远程操控量子计算设备进行相应的实验操作。用户无需了解实验设备的具体操作方式，只需要提供量子线路图或对应的代码，就可以在量子硬件中实现算法的运行并获得结果。这些云平台各自由不同的研究机构或公司开发和运营。目前已较为成熟的量子计算云平台包括Quafu^[115-117]、IBM Quantum Experience^[118-119]、IonQ^[120]等，其后端量子芯片所使用的物理体系主要包括超导和离子阱。本论文所使用的云平台是由北京量子信息科学研究院开发的Quafu平台和IonQ公司开发的IonQ平台，其

中 Quafu 平台后端为超导量子芯片，直接通过 Quafu 平台官网^[115]访问；IonQ 平台后端为离子阱量子芯片，通过亚马逊公司的 Amazon Braket 服务^[121]进行访问。

Quafu 量子计算云平台目前提供了五个超导量子计算芯片，比特数量从 10 个到 136 个不等，可以通过基于 Quafu 云平台的开源 python 软件开发工具 PyQuafu^[116]进行访问。本论文使用的是 P-10 芯片，其比特之间连接的拓扑结构如图1.8所示，该芯片包含 10 个 transmon 超导量子比特（Q₁-Q₁₀），呈线状排列，每对相邻的比特之间通过电容进行耦合。每个量子比特的频率可以在 4 到 5.7GHz 之间进行调节。每个比特都具有读出谐振器，连接在同一个传输线路中，所有比特的测量都可以通过这种方式进行。各个量子比特的性能参数和相干时间如表1.1所示。

IonQ 量子计算云平台提供 11 比特的离子阱量子芯片，其中具有 11 个被囚禁的 ¹⁷¹Yb⁺ 离子线状排列，通过在真空腔中制造交变电极势阱对量子比特进行空间上的限制。本论文所使用的量子处理器的拓扑结构如图1.9所示，其中每个量子比特之间都可以进行一对一对偶合。囚禁离子排列成线性晶体，悬浮在径向赝势频率约为 3.1 MHz 的势阱中。量子门是通过双光子拉曼跃迁实现的。有关该设备的更多信息可以查阅参考文献^[122]。

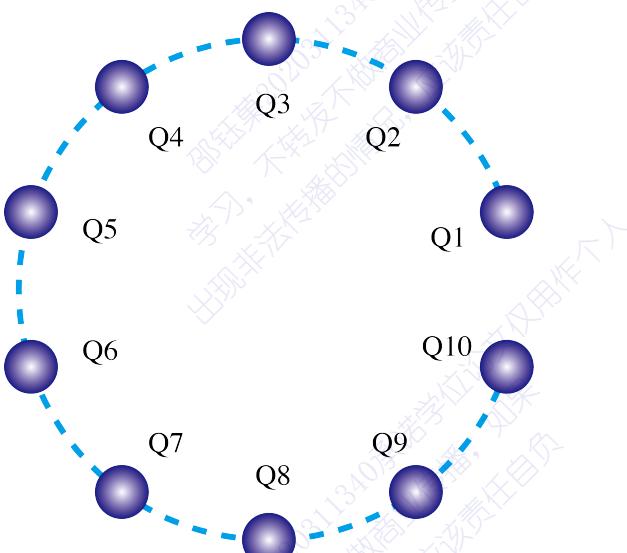


图 1.8 Quafu 平台 10 比特超导量子处理器 P-10 的拓扑结构

1.4 本章小结

本章首先介绍了量子力学的基本概念，给出了用来表示量子系统状态的纯态、混态、密度矩阵的概念，接着介绍了孤立系统的量子态在哈密顿量作用下的演化规律及演化算符的形式，并介绍了更一般的矩阵函数定义，随后介绍了量子测量

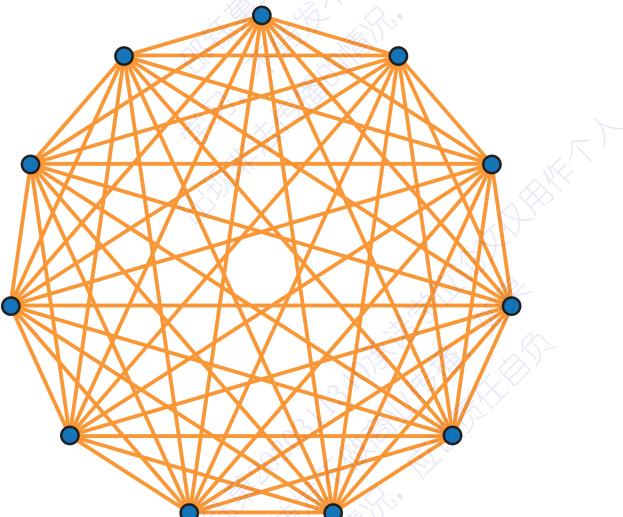


图 1.9 IonQ 平台 11 比特离子阱量子处理器的拓扑结构

的概念；之后我们对量子计算中量子比特、量子门、量子线路等常用概念进行了介绍，并介绍了非幺正量子线路；最后简要介绍了几种量子计算的物理实现方法及前沿进展，并简单介绍了量子计算云平台 Quafu 和 IonQ，便于后续内容中对算法演示实验的理解。

在后面的章节中，我们将介绍利用非幺正量子线路设计概率性虚时演化算法和量子稀疏成像算法、并求解量子多体系统和图像处理相关问题所取得的成果。第二章介绍了一些基础的量子算法、近年来量子算法的重要研究成果以及工作中涉及到的部分量子算法。第三章介绍了基于非幺正量子线路的概率性虚时演化算法，以及其在求解量子多体系统的基态方面的应用。第四章介绍了概率性虚时演化算法和变分量子算法在非厄米量子体系的动力学演化上的应用。第五章介绍了基于迭代收缩阈值方法的量子稀疏成像算法，并将其应用于图像处理问题。最后一章对本论文所做的工作进行总结，并对未来可能的发展方向做出展望。

表 1.1 Quafu 云平台中超导量子芯片 P-10 的性能参数

qubit	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}
$\omega_j^s/2\pi$ (GHz) ^a	5.536	5.069	5.660	4.742	5.528	4.929	5.451	4.920	5.540	4.960
$\omega_j^{10}/2\pi$ (GHz) ^b	5.456	4.424	5.606	4.327	5.473	4.412	5.392	4.319	5.490	4.442
$\omega_j^r/2\pi$ (GHz) ^c	5.088	4.702	5.606	4.466	5.300	4.804	5.177	4.697	5.474	4.819
$\eta_j/2\pi$ (GHz) ^d	0.250	0.207	0.251	0.206	0.251	0.203	0.252	0.204	0.246	0.208
$g_{j,j+1}/2\pi$ (MHz) ^e	12.07	11.58	10.92	10.84	11.56	10.00	11.74	11.70	11.69	-
$T_{1,j}$ (μs) ^f	20.0	52.5	15.9	16.3	36.9	44.4	30.8	77.7	22.8	25.0
$T_{2,j}^*$ (μs) ^g	8.60	1.48	9.11	2.10	12.8	2.73	15.7	1.88	4.49	2.05
$F_{0,j}$ (%) ^h	98.90	98.32	98.67	95.30	97.00	95.47	97.00	96.37	98.33	97.13
$F_{1,j}$ (%) ⁱ	92.90	92.30	92.97	91.53	86.17	87.93	93.40	93.37	94.63	92.07
$F_{j,j+1}$ (%) ^j	94.2	97.8	96.6	97.3	96.8	97.0	94.5	93.2	96.0	-

^a 比特 Q_j 的最大频率。^b 比特 Q_j 的中心频率。^c 比特 Q_j 读出时的谐振频率。^d 比特 Q_j 的非谐程度。^e 近邻比特 Q_j 与 Q_{j+1} 的耦合强度。^f 比特 Q_j 的弛豫时间 (relaxation time)。^g 比特 Q_j 的相干时间 (coherence time)。^h 比特 Q_j 的 $|0\rangle$ 态读出保真度。ⁱ 比特 Q_j 的 $|1\rangle$ 态读出保真度。^j 近邻比特 Q_j 与 Q_{j+1} 的 CZ 门保真度。

第2章 量子计算与常用量子算法介绍

在量子计算理论研究方面，人们开发出的量子算法已经广泛覆盖了各个应用领域，其中，部分量子算法能够解决基础且重要的数学或逻辑问题，并启发了一系列相关量子算法的研究，从而对整个量子计算领域的发展起到了至关重要的推动作用。

本章节中，我们将介绍一些基本的、对量子计算领域有重要意义的常用量子算法，以及一些与本论文工作相关的量子算法。

2.1 算法的复杂度

在介绍量子算法之前，我们首先需要介绍算法的复杂度（complexity）这一概念。在经典计算中人们就已经认识到，衡量一个算法的好坏，最重要的指标是算法求解一个问题的过程中所消耗的计算资源的量随着所求解问题的规模而增长的趋势。这里有两个关键词需要加以解释：计算资源与问题规模。第一，计算资源主要分为时间和空间两方面，其中时间资源是指算法需要的逻辑门的个数、门的深度、查询次数等，而空间资源是指算法需要用到的物理资源，一般是指比特。根据算法消耗的计算资源的类型，将复杂度分为时间复杂度和空间复杂度两个方面。第二，问题规模是指所求解的问题中输入数据的量，例如求解线性方程组问题中矩阵的维度、质数分解问题中整数的位数等。简单来说，算法的所消耗的时间和空间资源量是问题规模的一个单值函数，而算法的复杂度就是指这个单值函数的近似函数形式。常见的复杂度有常数、对数、对数多项式、线性、多项式、指数等。当问题规模较小时，我们一般不关心算法的复杂度，因为其消耗的时间和空间资源都可以忽略。只有当问题规模较大时，算法的复杂度才会十分重要。

对于量子算法而言，算法的复杂度与经典算法类似。其中空间复杂度是指算法需要用到的总的量子比特数。时间复杂度通常有三种计算方式：门复杂度（gate complexity），线路深度（circuit depth）和查询复杂度（query complexity）。这三种计算方式有相通之处，有时可以互相等效。第一，门复杂度就是指量子门的个数，值得一提的是，在实际的量子器件中构造线路时，一般只使用单比特门和两比特门，两种门需要的操作时间通常差距较大，因此有时会分开计算。第二，线路深度，有时也叫做线路层数，或门的层数，这一概念是基于量子线路中通常有作用在不同比特上的若干个量子门可以同时执行，这些可以同时执行的量子门就被称为线路的一层。第三，查询复杂度是基于“黑盒”（Oracle）这一概念定义的。假设

在求解某个问题时，需要求解一个子问题 A，某些情况下我们不知道或者不关心求解 A 的具体计算过程，只知道其功能，就把这部分计算过程整体打包为一个黑盒 O_A 。在求解问题的过程中，需要查询（调用、访问）黑盒 O_A 的次数，就叫做查询复杂度。这样做的好处是可以把很多能实现 O_A 的具体算法抽象成一个相同的算法，便于对算法进行设计和分析。

评估算法的复杂度需要借助渐进符号 \mathcal{O} 、 Ω 和 Θ ，分别表示渐近上界、渐近下界和渐近紧界。它们的数学含义分别为：对于两个非负函数 $f(n)$ 和 $g(n)$ ，若存在常数 c 和 n_0 ，使得 $n > n_0$ 时都满足 $f(n) \leq cg(n)$ ，则 $f(n) = \mathcal{O}(g(n))$ ，称 $g(n)$ 是 $f(n)$ 的渐进上界；若存在常数 c 和 n_0 ，使得 $n > n_0$ 时都满足 $f(n) \geq cg(n)$ ，则 $f(n) = \Omega(g(n))$ ，称 $g(n)$ 是 $f(n)$ 的渐进下界；若前面两种情况同时满足，即 $g(n)$ 同时是 $f(n)$ 的渐进上界和渐进下界，则 $f(n) = \Theta(g(n))$ ，称 $g(n)$ 是 $f(n)$ 的渐进紧界。一般情况下，我们更关心一个算法在最糟糕的情况下复杂的复杂度，即复杂度的渐进上界，因此通常用 \mathcal{O} 直接代表算法的复杂度。

2.2 几种基础的量子算法

2.2.1 量子搜索算法

量子搜索算法最早由 Lov Grover 于 1996 年提出^[8]，被称为 Grover 算法，可以在非结构化数据库中搜索满足特定条件的元素。该算法在密码学、优化问题、机器学习等领域有重要应用，是量子计算中最重要且著名的量子算法之一。

假设有一个规模为 N 的非结构化数据库，其中每个元素可以用一个整数 x 来代表， $x = 0, 1, \dots, N - 1$ ；同时，有一个函数 $f(x)$ 用来判断 x 是否为搜索目标，若 x 是搜索目标，则 $f(x) = 1$ ，否则 $f(x) = 0$ ；算法的目的是从整个数据库中找到满足 $f(x) = 1$ 的整数 x 。在该问题中，函数 $f(x)$ 是通过一个黑盒实现的，记为 O_f 。而我们所关心的复杂度主要是查询复杂度，即需要访问多少次 O_f 。经典算法解决该问题时，最坏的情况则需要遍历整个数据库，即访问 N 次 O_f ，平均则需要访问 $N/2$ 次，因此我们说，经典算法解决该问题的时间复杂度是 $\mathcal{O}(N)$ 。若能在量子计算机中访问 O_f ，则 Grover 算法仅需要 $\mathcal{O}(\sqrt{N})$ 次访问，就能以很高的概率找到搜索目标。相比较而言，Grover 算法实现了平方加速。下面我们将对 Grover 算法进行简要介绍。

在量子计算机中，首先我们使用 $n = \log N$ 个量子比特来表示数据库中的元素 x 。黑盒 O_f 是通过一个幺正操作 U_f 实现的，定义为： $U_f |x\rangle = (-1)^{f(x)} |x\rangle$ ，即如果 x 是搜索目标，则 $|x\rangle$ 的相位被取反，否则相位不变。此外，Grover 算法还需要用到一个相位门 $U_0 = 2|0\rangle\langle 0| - I$ ，即 $|0\rangle$ 态相位不变，其他态相位被取反。接着，

我们定义 Grover 算符: $\mathbf{G} = \mathbf{H}^{\otimes n} \mathbf{U}_0 \mathbf{H}^{\otimes n} \mathbf{U}_f$, 其中 $\mathbf{H}^{\otimes n}$ 表示在 n 个量子比特上分别执行 Hadamard 门。

Grover 算符是 Grover 算法的核心, 我们观察它作用在量子比特上的效果: 记数据库中所有元素的平均叠加态为 $|\psi\rangle = \sum_{x=0}^{N-1} |x\rangle / \sqrt{N}$, 记所有搜索目标的平均叠加态为 $|\beta\rangle = \sum_{f(x)=1} |x\rangle / \sqrt{M}$, 其中 M 为搜索目标的个数, 记所有非目标的平均叠加态为 $|\alpha\rangle = \sum_{f(x)=0} |x\rangle / \sqrt{N-M}$, 则有

$$|\psi\rangle = \cos \theta_0 |\alpha\rangle + \sin \theta_0 |\beta\rangle, \quad (2.1)$$

其中 $\theta_0 = \arcsin \sqrt{M/N}$ 。Grover 算法的关键步骤, 就是将 \mathbf{G} 算符连续作用在初始的平均叠加态 $|\psi\rangle$ 上, 作用一次被称为一次迭代。可以证明, 当进行了 k 次迭代后, 量子态会变为

$$\mathbf{G}^k |\psi\rangle = \cos((2k+1)\theta_0) |\alpha\rangle + \sin((2k+1)\theta_0) |\beta\rangle. \quad (2.2)$$

我们希望搜索目标态的振幅 $\sin((2k+1)\theta_0)$ 尽可能接近 1, 因此最佳迭代次数为

$$k_0 = \lceil \frac{\pi}{4\theta_0} - \frac{1}{2} \rceil, \quad (2.3)$$

这里 $\lceil \cdot \rceil$ 表示近似取整。通常情况下, 搜索目标在整个数据库中所占的比例很小, 即 $M/N \ll 1$, 此时 $\theta_0 \approx \sqrt{M/N}$, 最佳迭代次数近似为

$$k_0 \approx \frac{\pi}{4} \sqrt{\frac{N}{M}}. \quad (2.4)$$

也就是说, 当 N 个数据中有 M 个目标时, 利用 Grover 算法迭代 $\mathcal{O}(\sqrt{N/M})$ 次即可找到目标, 而经典算法平均需要访问 $\mathcal{O}(N/M)$ 次才能找到目标。因此, 相比于经典算法, 量子算法在该问题上实现了平方加速。目前 Grover 算法已经在量子计算物理平台上得到了实验验证^[41,123]。

Grover 算法并不是精确的搜索算法, 因为迭代 k_0 次后, 目标态的振幅 $\sin((2k_0+1)\theta_0)$ 只能尽可能接近 1, 而不一定精确等于 1。理论上, 当且仅当 $N/M = 4$ 时, 可以通过 1 次迭代精确得到目标态, 除此之外任何情况都无法精确求解。针对这一问题, 清华大学龙桂鲁提出了 Long 算法^[124], 它是在 Grover 算法的基础上进行了改进, 可以以 100% 的成功概率精确得到目标态。其具体方法为, 将构成 Grover 算符的黑盒 \mathbf{U}_f 与相位门 \mathbf{U}_0 替换为如下形式:

$$\mathbf{U}_f = e^{i\varphi_1 f(x)} |x\rangle, \quad \mathbf{U}_0 = (e^{i\varphi_2 - 1}) |0\rangle \langle 0| + \mathbf{I}, \quad (2.5)$$

其中 φ_1 与 φ_2 是可选择的相位。如果我们可以 $J+1$ 次迭代得到精确的目标

态，则 J 与 φ_1, φ_2 需要满足

$$J \geq \frac{\pi}{4\theta_0} - \frac{3}{2}, \quad (2.6)$$

$$\varphi_1 = \varphi_2 = 2 \arcsin \left(\frac{\sin \left(\frac{\pi}{4J+6} \right)}{\sin \theta_0} \right). \quad (2.7)$$

上面的2.7式被称为相位匹配条件^[125-127]。这样就可以通过 $J + 1$ 次迭代以 100% 的概率得到精确的目标态。

在搜索目标在数据库中所占的数量比例已知的情况下，Long 算法被证明是最优的精确搜索算法。以 Grover 算法与 Long 算法为基础的量子搜索算法，对于经典密码破译具有重要的意义^[128-129]，同时也为量子振幅放大^[130-131]、定点搜索算法^[132-133]等方法提供了理论基础，极大推动了量子计算的发展。

2.2.2 量子傅立叶变换与相位估计算法

傅立叶变换在信号处理、数据压缩、数据分析等众多领域具有重要的应用，是最重要的数学工具之一。傅立叶变换的实现效率决定了人类对信息的处理能力。在经典算法中，目前最高效的离散傅立叶变换算法是快速傅立叶变换（fast Fourier transformation, FFT），其对 N 维数据进行离散傅立叶变换需要 $\mathcal{O}(N \log N)$ 的时间。FFT 是目前最普遍使用的经典傅立叶变换算法。然而随着人类需要处理的信息量急速增长，FFT 在面对高维信息时也变得困难。而量子傅立叶变换（quantum Fourier transformation, QFT）^[134]则能够在 $\mathcal{O}(\log^2 N)$ 的时间内完成 N 维数据的离散傅立叶变换，因此相比于经典算法实现了指数加速。QFT 不仅提供了一种基于量子计算机的高效信号处理方法，同时在相位估计^[10,135]、质数分解^[9,136]等问题中也具有重要的应用。

对于一个 N 维向量 x_j ($j = 0, 1, \dots, N - 1$)，离散傅立叶变换能够给出：

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{i2\pi jk/N} x_j, \quad k = 0, 1, \dots, N - 1. \quad (2.8)$$

其物理含义是将向量 \mathbf{x} 所代表的时域信息转换为 \mathbf{y} 所代表的频域信息。在量子计算机中，我们可以将 \mathbf{x} 和 \mathbf{y} 都以振幅编码的方式储存在 $n = \log N$ 个量子比特的量子态中，即 $|\mathbf{x}\rangle = \sum_j x_j |j\rangle$, $|\mathbf{y}\rangle = \sum_k y_k |k\rangle$ (此处假定 \mathbf{x} 和 \mathbf{y} 都已归一化)，则量子计算机中的傅立叶变换需要构造幺正算符 U_{FT} ，使得 $U_{\text{FT}} |\mathbf{x}\rangle = |\mathbf{y}\rangle$ 。对于计算基

$|j\rangle$, 眄正算符 \mathbf{U}_{FT} 的作用为:

$$\mathbf{U}_{\text{FT}} |j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i2\pi j k / N} |k\rangle. \quad (2.9)$$

将等号左侧的计算基 $|j\rangle$ 写为二进制字符串的形式, 同时将等号右侧写成各个单比特态的直积形式, 则得到:

$$\begin{aligned} & \mathbf{U}_{\text{FT}} |j_1 j_2 \cdots j_n\rangle \\ &= \frac{1}{\sqrt{N}} (|0\rangle + e^{i2\pi 0.j_n} |1\rangle) (|0\rangle + e^{i2\pi 0.j_{n-1} j_n} |1\rangle) \cdots (|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n} |1\rangle), \end{aligned} \quad (2.10)$$

其中 $j_m \in \{0, 1\}$ ($m = 1, 2, \dots, n$), $0.j_m j_{m+1} \cdots j_n$ 表示二进制小数。从这个式子可以看出 \mathbf{U}_{FT} 的一种构造思路: 先将第 m 个比特从 $|j_m\rangle$ 变为 $(|0\rangle + e^{i2\pi 0.j_m j_{m+1} \cdots j_n} |1\rangle) / \sqrt{2}$, 再将所有比特倒序排列。其中, 对第 m 个比特的变换可以通过 Hadamard 门与一系列受控相位门实现, 对比特的倒序排列可以通过一系列 SWAP 门实现, 由此可以得到如图2.1所示的量子线路图, 其中 \mathbf{R}_k 是相位门, 形式为

$$\mathbf{R}_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi/2^k} \end{bmatrix}. \quad (2.11)$$

这样, 通过 n 个单比特门、 $n(n - 1)/2$ 个两比特门和 $n/2$ 个两比特 SWAP 门, 就可以实现 $N = 2^n$ 维数据的傅立叶变换。该算法的时间复杂度为 $\mathcal{O}(n^2)$, 相比于经典的 FFT 算法而言实现了指数加速。这里需要注意的是, 量子傅立叶变换给出的是振幅编码的量子态, 并不像经典数据那样可以直接得到向量的各个分量。如何高效地从量子态中获取所需要的经典信息, 也是量子计算中值得关注的重要问题。

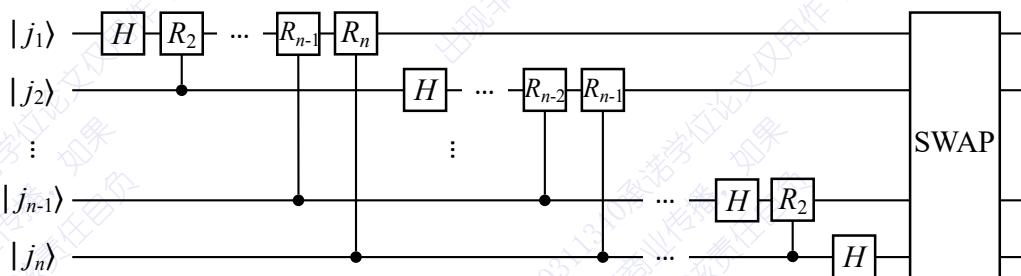


图 2.1 量子傅立叶变换算法的线路图

将 QFT 的过程逆转过来, 就可以实现逆傅立叶变换, 即根据已知的频域信息 \mathbf{y} 计算出时域信息 \mathbf{x} , 需要实现幺正算符 $\mathbf{U}_{\text{FT}}^{-1}$ 。根据量子线路的可逆性, 只需按线路图2.1的倒序进行操作, 并将其中的 \mathbf{R}_k 替换为 \mathbf{R}_k^\dagger , 就可以在 $\mathcal{O}(n^2)$ 的时间内实现逆傅立叶变换。

量子傅立叶变换及逆变换可以应用于相位估计、求阶问题、离散对数问题、大数分解问题等。其中, Shor 算法^[9,136]是著名的量子算法之一, 它将质数分解问题

转化为求阶问题，进而利用量子逆傅立叶变换求解。算法的具体过程与本论文研究内容无关，因此不再详细介绍。Shor 算法可以在 $\mathcal{O}(\text{poly}(n))$ 的时间内完成对 n 位整数的质数分解，相比于经典算法实现了指数加速，对目前基于 RSA 加密体系的保密通信构成了严重威胁，引起了量子算法和密码学领域的持续关注。

量子相位估计（quantum phase estimation, QPE）是 QFT 的一个重要应用。假设一个任意 N 维幺正算符 \mathbf{U} ，其本身是一个黑盒，给定 \mathbf{U} 的一个本征态 $|u\rangle$ ，相位估计就是希望求出 $|u\rangle$ 所对应的本征值 $e^{i2\pi\phi}$ 中的相位 ϕ 。由于 $0 \leq \phi < 1$ ，算法只需给出 ϕ 的小数点后的数字即可。

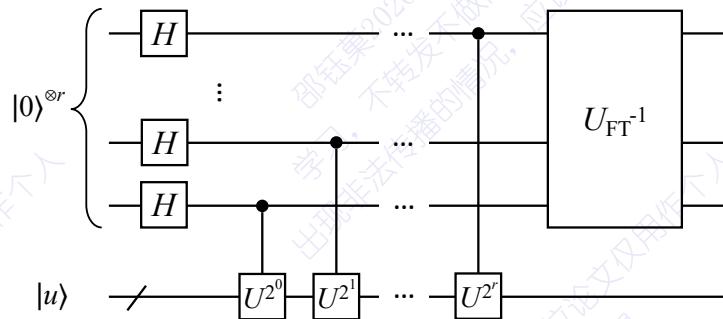


图 2.2 量子相位估计算法的线路图

量子相位估计的线路图如图2.2所示，其中额外引入 r 个量子比特用来储存相位估计的结果， r 代表相位估计的精度。经过量子相位估计的线路后，量子态会从初始的 $|0\rangle^{\otimes r}|u\rangle$ 演变为 $|\tilde{\phi}\rangle|u\rangle$ ，其中 $|\tilde{\phi}\rangle$ 中储存着 ϕ 的估计值，具体形式为

$$|\tilde{\phi}\rangle = \frac{1}{2^r} \sum_{j=0}^{2^r-1} \left[\sum_{k=0}^{2^r-1} e^{i2\pi k(\phi - j/2^r)} \right] |j\rangle. \quad (2.12)$$

当 ϕ 在二进制下是不超过 r 位的有限小数时，即 $2^r\phi$ 是二进制整数，则 $|\tilde{\phi}\rangle = |2^r\phi\rangle$ ，此时可以直接通过测量 $|\tilde{\phi}\rangle$ 得到 ϕ 的值。当 ϕ 在二进制下的小数位数超过 r 时，则 $|\tilde{\phi}\rangle$ 在最接近 $2^r\phi$ 的二进制整数对应的基矢上具有最大振幅，通过测量 $|\tilde{\phi}\rangle$ 可以以很高的概率得到 $2^r\phi$ 的近似值。理论分析表明，若希望以 $1 - \epsilon$ 的概率得到精度为 2^{-m} 的相位估计值，则引入的比特数 r 最少应为

$$r = m + \lceil \log \left(2 + \frac{1}{2\epsilon} \right) \rceil, \quad (2.13)$$

这里 $\lceil \cdot \rceil$ 表示上取整。在此情况下，量子相位估计需要 $\mathcal{O}(2^r)$ 个受控 \mathbf{U} 操作和 $\mathcal{O}(r^2)$ 个单比特和双比特门。

量子相位估计可以应用于本征值求解问题，假设幺正算符可以写成 $\mathbf{U} = e^{i\mathbf{A}}$ ，其中 \mathbf{A} 是厄米算符，则可以利用相位估计近似得到 \mathbf{A} 的本征值。此外，相位估计在线性方程组求解等问题中有重要应用，其中包括著名的 HHL 算法^[11]，以及基于相位估计近似实现任意矩阵函数的方法。关于 HHL 算法与任意矩阵函数将在2.5.1小

节中详细介绍。

2.2.3 哈密顿量模拟算法

在研究量子多体系统的动力学性质时，经常需要知道一个量子系统的状态在哈密顿量作用下的时间演化，即给定系统的初态 $|\psi_s(0)\rangle$ 和哈密顿量 \mathbf{H}_s ，求一段时间 t 后的演化末态 $|\psi_s(t)\rangle = e^{-i\mathbf{H}_s t} |\psi_s(0)\rangle$ 。假设已经找到一种方式，能够把要模拟的量子系统的状态映射到量子计算机中量子比特的状态上，即 $|\psi_s\rangle \rightarrow |\psi_q\rangle$ ，同时哈密顿量也可以映射过来 $\mathbf{H}_s \rightarrow \mathbf{H}_q$ ，那么只要我们能操控量子比特的态的演化 $|\psi_q(t)\rangle = e^{-i\mathbf{H}_q t} |\psi_q(0)\rangle$ ，就可以从量子比特的末态 $|\psi_q(t)\rangle$ 反推出要模拟的系统的末态 $|\psi_s(t)\rangle$ 。而对于任意的哈密顿量 \mathbf{H}_q ，其时间演化算符 $e^{-i\mathbf{H}_q t}$ 也是任意的，难以在量子线路中直接实现，这时就需要寻找一种方法，将时间演化算符拆分成常见的基本量子门，从而容易在量子线路中实现^[137]。

假设一个 n 量子比特的哈密顿量 \mathbf{H} 可以写为若干项之和： $\mathbf{H} = \sum_{k=1}^m \mathbf{H}_k$ 。如果其中任意两项都互易，即 $[\mathbf{H}_k, \mathbf{H}_{k'}] = 0$ ，则有如下式子：

$$e^{-i\mathbf{H}t} = e^{-i\mathbf{H}_1 t} e^{-i\mathbf{H}_2 t} \dots e^{-i\mathbf{H}_m t}. \quad (2.14)$$

此时就可以将哈密顿量演化算符拆分成其中各项演化算符的乘积。然而绝大多数情况下，哈密顿量中的各项并不互易，则上式不再成立。此时采用如下近似式：

$$e^{-i\mathbf{H}t} = (e^{-i\mathbf{H}_1 t/L} e^{-i\mathbf{H}_2 t/L} \dots e^{-i\mathbf{H}_m t/L})^L + \mathcal{O}(t/L). \quad (2.15)$$

这个式子叫做 Trotter 分解^[138-139]，它把总时间 t 切分成 L 个更短的时间间隔，每个时间间隔内的哈密顿量演化算符近似等于哈密顿量各项的演化算符的乘积。我们把每一项的短时间演化 $e^{-i\mathbf{H}_k t/L}$ 都称为一次 Trotter 迭代。Trotter 分解的误差项与总时间 t 呈正比，与时间切分的份数 L 呈反比。

一个 n 比特哈密顿量 \mathbf{H} 一定可以写成如下形式：

$$\mathbf{H} = \sum_{i_1=0}^3 \sum_{i_2=0}^3 \dots \sum_{i_n=0}^3 C_{i_1 i_2 \dots i_n} \sigma_{i_1} \otimes \sigma_{i_2} \otimes \dots \otimes \sigma_{i_n}, \quad (2.16)$$

其中 $C_{i_1 i_2 \dots i_n}$ 为实系数， σ_0 表示二维单位矩阵， $\sigma_{1,2,3}$ 分别表示泡利算符 $\sigma_{x,y,z}$ ，上式一共包含 4^n 项，每一项都是 n 个单位矩阵或泡利算符的直积，叫做泡利直积项或泡利项（有时会省略其中的直积符号 \otimes ）。 n 比特哈密顿量是 2^n 维厄米矩阵，具有 4^n 个独立自由度，而这 4^n 个泡利项刚好构成了 2^n 维厄米矩阵的一组独立基，因此上式可以看作是哈密顿量在这组基上的展开。根据 Trotter 分解，我们只需要在量子线路中实现各个泡利项的演化算符，就可以实现总哈密顿量的演化算符。

对于单比特的泡利算符 $\sigma_{x,y,z}$ ，其演化算符可直接通过单比特旋转门实现，例

如 $\mathbf{H}_k = c\sigma_x$, 则 $e^{-i\mathbf{H}_k t} = R_x(2ct)$ 。对于两比特泡利项, 如 $\mathbf{H}_k = c\sigma_z\sigma_z$, 其演化算符可以通过图2.3(a)中的量子线路实现, 仅需要2个CNOT门和一个旋转门 R_z 。而对于 $\mathbf{H}_k = c\sigma_x\sigma_y$, 则可以利用图2.3(b)中的量子线路实现其演化算符, 需要7个单比特门和2个CNOT门。两个线路图中旋转门 R_z 的角度都为 $\theta = 2ct$ 。

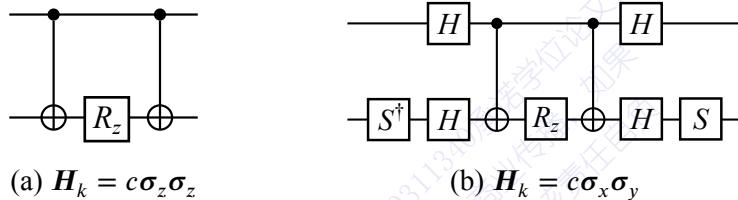


图 2.3 实现两比特泡利项演化算符的量子线路

依照这样的规律, 可以利用单比特门和CNOT门构造出任意多比特泡利项的演化算符。由此就可以实现任意哈密顿量的时间演化算符。

利用 Trotter 分解实现哈密顿量演化的时间复杂度为 $\mathcal{O}(mt/\delta)$, 其中 m 是哈密顿量中系数非零的泡利项的个数, t 是演化总时间, δ 是计算精度。值得注意的是, 如果 n 比特哈密顿量不具有任何特殊性质, 则 $m \sim 4^n$, 这意味着算法的复杂度是 n 的指数, 与经典算法复杂度相同, 此时量子算法不再具有优势。然而常见的量子多体系统, 例如化学分子和凝聚态体系, 其哈密顿量中主要包含单体能量和两体相互作用, 有时也考虑三体相互作用, 但更高阶的作用一般被忽略, 因此系数非零的项数 m 是粒子数 n 的多项式量级, 这时量子算法就具有优势。

随着量子算法理论的发展, 人们又针对具有稀疏性的哈密顿量提出了一系列更高效的模拟算法^[79,140-142]。这些算法能够利用哈密顿量的稀疏性, 通过能给出哈密顿量非零矩阵元的黑盒, 进一步提升哈密顿量模拟的效率, 达到正比于矩阵稀疏度或哈密顿量范数的时间复杂度。

2.3 参数化量子线路与变分量子算法

前面介绍的 Grover 算法、量子相位估计、哈密顿量模拟等算法, 如果想要解决有实际价值的问题, 其所需要的量子比特数量和量子线路深度都远远超过目前量子硬件的能力。而变分量子算法 (variational quantum algorithm, VQA)^[143]作为一类经典计算与量子计算混合的算法, 能有效利用当前中等规模含噪声的量子硬件, 在一定程度上发挥量子计算的优势。

变分量子算法的基本思想是: 利用参数化量子线路 (parametrized quantum circuit, PQC), 将量子比特的状态用一组可调的实参数 θ 表示, 即 $|\psi(\theta)\rangle$, 再根据待求解的问题设计经典参数优化器, 利用经典方法对参数 θ 进行优化, 从而使量子态逐渐演变为所求的结果。变分量子算法的基本框架如图2.4所示, 其中主要包括

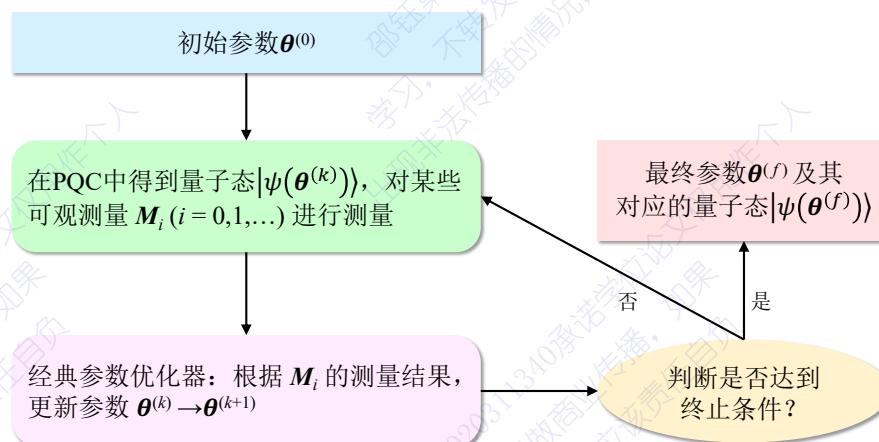


图 2.4 变分量子算法的基本框架

以下三个结构。

1. 参数化量子线路，又称作 *ansatz*，直译为“开端”、“拟设”。参数化量子线路中包含一系列由经典参数控制的量子门，而输入该线路的量子态是固定的，记为 $|\psi_0\rangle$ ，经过该线路后就变为参数化量子态。 $|\psi_0\rangle$ 通常为一些简单的态，例如 $|0\rangle^{\otimes n}$ 。我们用 $\mathbf{U}(\boldsymbol{\theta})$ 来表示整个参数化量子线路对应的幺正算符，则线路所输出的参数化量子态就可以表示为 $|\psi(\boldsymbol{\theta})\rangle = \mathbf{U}(\boldsymbol{\theta})|\psi_0\rangle$ 。

参数化量子线路通常包含大量结构相同的重复的量子门序列，每重复一次称为一层，如图2.5所示为参数化量子线路的一层的例子，其中共有 4 个 Rx 门和 4 个 Ry 门，每个门的旋转角参数都是独立的；将这样一层线路重复多次，各层线路中的参数都相互独立。线路层数越多，独立参数的个数就越多，能表达的量子态也越多，线路的表达能力就越强。通常而言，线路中独立参数的个数是量子比特数的多项式量级，这样既能保证线路有较强的表达能力，也能使算法复杂度处在较低的量级。

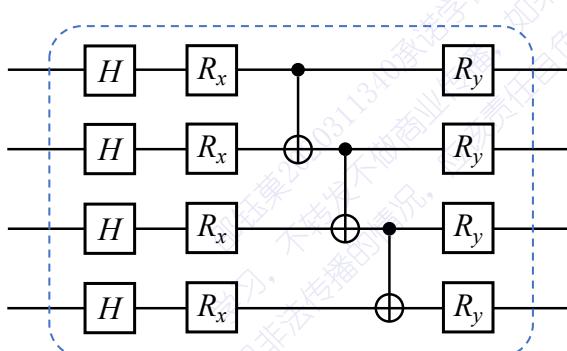


图 2.5 参数化量子线路的一层的例子

在实际问题中，参数化量子线路如何设计并没有固定的规律和原则，但通常有

一些经验，例如前面提到线路中通常包含大量结构相同的重复的量子门序列。此外，可以根据量子硬件所能高效实现的量子门来设计线路，也可以根据待求解问题的特点或目标量子态的部分已知性质来设计线路。

2. 损失函数 $C(\theta)$ ，用来衡量当前量子态与所求目标态之间的差距，通常由量子比特的某些可观测量的期望值计算得到。一般来说，损失函数值越小，表明当前量子态越接近目标态。例如，可以定义损失函数为哈密顿量的期望值： $C(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$ ，通过优化 θ 不断降低损失函数的值，当损失函数达到最小值时，量子态的能量平均值最低，意味着最接近哈密顿量的基态，这就是利用变分量子算法求解哈密顿量基态的过程。

损失函数决定了经典参数优化器的优化方向，其设计是否合理决定了变分量子算法的性能，如果损失函数的设计不合理，可能会导致参数优化过程收敛太慢，或者无法收敛。

3. 经典参数优化器，即一套经典算法，能够根据当前量子态的某些测量结果，对参数 θ 进行更新，以得到下一个更接近目标态的量子态。常见的经典参数优化方法包括梯度下降法、Nelder-Mead 方法、线性方程组求解等。经典参数优化器的性能有时也会影响变分量子算法的性能，不同类型的待求解问题所适用的优化方法也不尽相同，需要根据经验和尝试结果加以选择和调整。

2.3.1 变分量子本征求解器

变分量子本征求解器 (variational quantum eigensolver, VQE) 是由 Alberto Peruzzo 等人于 2014 年提出的算法^[144]，是最早的变分量子算法，它可以用来求解哈密顿量的基态本征值。其原理是：对于一个哈密顿量 H ，任意量子态 $|\psi\rangle$ 的能量平均值 $\langle \psi | H | \psi \rangle$ 都不可能低于基态能量 E_g ，因此若要求解基态能量，只需要在所有可能的 $|\psi\rangle$ 中寻找能量平均值最小值即可。

利用参数化量子线路，可以将量子态参数化： $|\psi\rangle = |\psi(\theta)\rangle$ ，这样就可以定义损失函数 $C(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle$ 。其中，哈密顿量可以按照2.16式写成泡利直积项之和：

$$H = \sum_{k=1}^m c_k H_k, \quad (2.17)$$

其中 H_k 是多比特泡利算符，其期望值 $\langle H_k \rangle$ 可以在量子线路中通过高效的方法测量得到。这样，损失函数值就可以通过如下方式测量：

$$C(\theta) = \langle \psi(\theta) | H | \psi(\theta) \rangle = \sum_{k=1}^m c_k \langle \psi(\theta) | H_k | \psi(\theta) \rangle. \quad (2.18)$$

将损失函数值输入到经典参数优化器中，通过一些经典的多元函数优化方法，例如梯度下降法、牛顿法、Nelder-Mead 方法等，就可以通过迭代的方式不断降低损失函数的值，直至收敛到局部最优解。这时我们就认为近似得到了哈密顿量的基态能量，对应的基态可以通过最后一步迭代的参数 θ 得到。通常而言，当哈密顿量中的泡利项个数较少、迭代的初始态更接近精确基态时，VQE 的收敛过程更快，得到的结果也更接近精确解。同时，算法的收敛过程与收敛结果的好坏也取决于 ansatz 的选取是否合适。VQE 被认为适用于求解化学分子、物理模型等体系的基态与基态能量，并被更广泛地应用于激发态求解^[145-146]、线性方程组求解^[147]、量子态分析^[148]等问题。针对 VQE 所做出的改进方案也已取得诸多进展^[149-151]。

对于固定类型的问题，VQE 的线路深度为 $\mathcal{O}(1)$ ，极大地降低了对量子比特相干时间的要求，因此适用于当今含噪声的量子硬件。而通常认为 VQE 存在以下问题：第一，参数化量子态无法表达希尔伯特空间中全部的态，因此 VQE 可能无法找到精确基态，只能在参数的表达范围内找到最接近基态的量子态；第二，经典参数优化可能给出的是局域最小值，而非全局最小值；第三，当比特数较多时，并没有理论保证 VQE 计算复杂度相对于经典算法的优越性^[152-153]。这些问题也是其它变分量子算法所面临的问题。但总体而言，VQE 仍然是能在当今量子硬件中发挥一定量子优势的算法。

2.3.2 量子近似优化算法

量子近似优化算法（quantum approximate optimization algorithm, QAOA）由 Edward Farhi 等人于 2014 年提出^[154]，它也是利用参数化量子线路来求解哈密顿量的基态，从基本原理和功能上与 VQE 基本相同，但在具体方法及应用场景上与 VQE 有所区别，且也引起了广泛关注与研究。

QAOA 可以理解为一种具有特殊的参数化量子线路结构的 VQE，其主要解决的是组合优化问题。QAOA 最开始被提出是为了解决最大割（max-cut）问题，该问题描述为：给定一个图 $G(V, E)$ ，包含一个顶点集合 V 和一个由其中的顶点构成的边的集合 E ，现要将图中的顶点划分为两组，求如何划分能使被切割的边的数量最大。图2.6给出了一些最大割问题中图的例子。最大割问题等价于求解如下哈密顿量的最大本征值：

$$\mathbf{H}_C = \sum_{\langle i,j \rangle \in E} \frac{1}{2} \left(1 - \sigma_z^i \sigma_z^j \right), \quad (2.19)$$

其中 $\langle i,j \rangle$ 是由点 i, j 构成的边。在这个哈密顿量中，每个比特表示图中的一个点，比特的 $|0\rangle$ 和 $|1\rangle$ 两种状态分别代表该点被划分在两个不同的组。

在 QAOA 方法中，定义初始哈密顿量 $\mathbf{H}_B = \sum_{j=1}^n \sigma_x^j$ ，是所有比特的 σ_x 算符

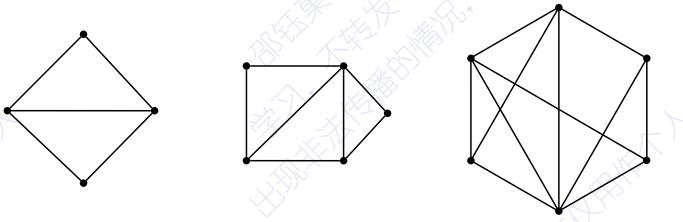


图 2.6 最大割问题的图的例子

之和，其本征值最高的本征态为 $|+\rangle^{\otimes n} = \sum_{k=0}^{2^n-1} |k\rangle / \sqrt{2^n}$ 。则 QAOA 的参数化量子线路由如下两种幺正算符产生：

$$\mathbf{U}_C(\gamma) = e^{-i\gamma H_C}, \quad \mathbf{U}_B(\beta) = e^{-i\beta H_B}. \quad (2.20)$$

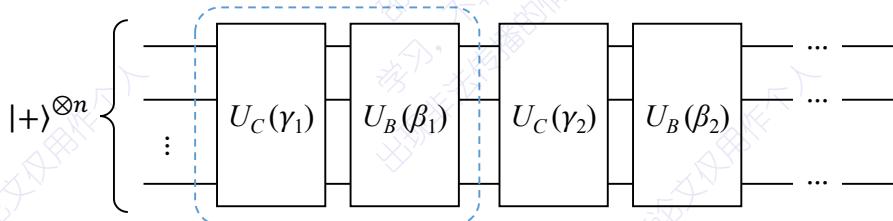


图 2.7 QAOA 的参数化量子线路结构

QAOA 的参数化量子线路结构如图2.7所示，其中每一层线路由一个 $\mathbf{U}_C(\gamma_i)$ 和一个 $\mathbf{U}_B(\beta_i)$ 构成，各层之间的 γ_i 和 β_i 均为独立参数。若线路共有 p 层，则 QAOA 的参数化量子态为：

$$|\gamma, \beta\rangle = \mathbf{U}_B(\beta_p)\mathbf{U}_C(\gamma_p) \cdots \mathbf{U}_B(\beta_2)\mathbf{U}_C(\gamma_2)\mathbf{U}_B(\beta_1)\mathbf{U}_C(\gamma_1)|+\rangle^{\otimes n}. \quad (2.21)$$

定义优化函数： $F(\gamma, \beta) = \langle \gamma, \beta | H_C | \gamma, \beta \rangle$ ，其中待优化的参数为 γ, β ，利用梯度方法等经典优化算法计算出优化函数的极大值，就可以得到目标量子态及其对应的参数，进而得到最大割问题的解。

QAOA 方法的实质是模拟了绝热演化的过程^[155]，当线路层数 p 趋于无穷时，线路的形式就趋于精确的绝热演化，因此足够多的层数就能保证量子态一定能演化至精确的目标态。QAOA 已引发了一系列相关的工作^[156-163]。

2.3.3 变分量子模拟

变分量子模拟 (variational quantum simulation, VQS)^[164] 是利用参数化量子线路求解哈密顿量模拟问题的算法。该算法的思想早在量子计算的概念诞生之前就已经存在^[165]，即把参数化量子态 $|\psi(\theta)\rangle$ 代入描述量子态随时间演化的方程，通过对方程做变分，把量子态随时间的演化等效为参数 θ 随时间的演化，并得到参数 θ 所满足的演化方程，再利用经典参数优化器进行求解即可。参数的演化方程

有几种不同的推导方法，分别基于不同的变分法则，推导结果也有所区别，但大同小异。这里我们只介绍 McLachlan 变分法：首先，将参数化量子态代入薛定谔方程得到

$$i \sum_k \frac{\partial |\psi\rangle}{\partial \theta_k} \dot{\theta}_k - \mathbf{H} |\psi\rangle = 0. \quad (2.22)$$

然后对该方程的模做变分，即

$$\delta \left\| i \sum_k \frac{\partial |\psi\rangle}{\partial \theta_k} \dot{\theta}_k - \mathbf{H} |\psi\rangle \right\| = 0, \quad (2.23)$$

这里 $\|\cdot\|$ 表示复模长。将这一变分方程展开并进一步处理，最终得到参数随时间 t 的演化方程：

$$\sum_k A_{jk} \dot{\theta}_k = B_j, \quad (2.24)$$

其中

$$A_{jk} = \operatorname{Re} \left(\frac{\partial \langle \psi |}{\partial \theta_j} \frac{\partial |\psi\rangle}{\partial \theta_k} \right), \quad B_j = \operatorname{Im} \left(\frac{\partial \langle \psi |}{\partial \theta_j} \mathbf{H} |\psi\rangle \right). \quad (2.25)$$

VQS 方法单次迭代的过程大概如下：根据 t 时刻的参数 $\theta(t)$ ，从参数化量子线路中得到此刻的量子态，并通过测量得到所有 A_{jk} 与 B_j 的值，在经典优化器中求解线性方程组 $\mathbf{A}\dot{\theta} = \mathbf{B}$ 得到 $\dot{\theta}$ ，则下一时刻的参数为 $\theta(t + \Delta t) = \theta(t) + \dot{\theta}\Delta t$ 。这其中的问题在于， A_{jk} 与 B_j 并不是量子力学中的可观测量，因此需要采用特殊方法进行测量。我们首先将参数化量子态写为如下形式：

$$|\psi(\theta)\rangle = \mathbf{U}_m(\theta_m) \cdots \mathbf{U}_2(\theta_2) \mathbf{U}_1(\theta_1) |\psi_0\rangle, \quad (2.26)$$

其中 $|\psi_0\rangle$ 是输入到参数化量子线路中的初始态， $\mathbf{U}_k(\theta_k)$ 是各个含参量子门，它们通常是单比特旋转门 $\mathbf{U}_k(\theta_k) = e^{-i\theta_k \sigma_{\alpha_k}/2}$ ，其中 $\alpha_k \in \{x, y, z\}$ ，于是有

$$\frac{\partial \mathbf{U}_k}{\partial \theta_k} = -\frac{i}{2} \sigma_{\alpha_k} \mathbf{U}_k, \quad \frac{\partial |\psi\rangle}{\partial \theta_k} = -\frac{i}{2} \mathbf{U}_m(\theta_m) \cdots \sigma_{\alpha_k} \mathbf{U}_k(\theta_k) \cdots \mathbf{U}_1(\theta_1) |\psi_0\rangle. \quad (2.27)$$

根据上式，就可以在量子线路中生成量子态 $\partial |\psi\rangle / \partial \theta_k$ （相差常系数）。这样就可以将 A_{jk} 的值转化为两个量子态的内积的实部。另一方面，将哈密顿量写成泡利项之和的形式： $\mathbf{H} = \sum_m c_m \mathbf{H}_m$ ，则

$$B_j = \sum_m c_m \operatorname{Im} \left(\frac{\partial \langle \psi |}{\partial \theta_j} \mathbf{H}_m |\psi\rangle \right). \quad (2.28)$$

由于 \mathbf{H}_m 为多比特泡利算符， $\mathbf{H}_m |\psi\rangle$ 也可以在量子线路中生成，因此 B_j 可以转化为两个量子态的内积的虚部，进而可以通过将其中一个态乘以相位因子 i ，将虚部转化为实部。因此我们需要在量子线路中实现对两个量子态内积实部的测量^[166]。

假设有两个量子态 $|\phi_1\rangle, |\phi_2\rangle$ 分别可以由两个幺正算符 $\mathbf{V}_1, \mathbf{V}_2$ 作用在同一个态上而产生，即 $|\phi_1\rangle = \mathbf{V}_1|\phi_0\rangle, |\phi_2\rangle = \mathbf{V}_2|\phi_0\rangle$ 。根据图2.8所示的量子线路，引入一个辅助比特，经过该线路后，所求的内积实部 $\text{Re}\langle\phi_1|\phi_2\rangle$ 可以通过测量辅助比特处于 $|0\rangle$ 态的概率 P_0 得出： $\text{Re}\langle\phi_1|\phi_2\rangle = 2P_0 - 1$ 。

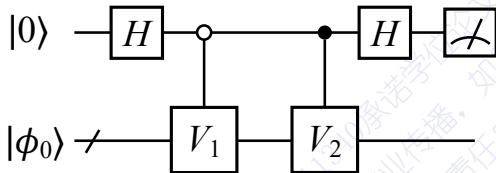


图 2.8 测量两个量子态的内积实部的量子线路

VQS 目前已被应用于更广泛的演化过程^[167-169]。而 VQS 面临的最大问题是误差的累积效应，相比其它变分量子算法，VQS 的误差累积效应对计算结果的影响更为显著，而针对此问题已有相关的工作^[170]。

2020 年，Cristina Cirstoiu 等人提出了另一种基于参数化量子线路的哈密顿量模拟算法，叫做变分快进（variational fast forwarding, VFF）算法^[171]。该方法定义参数化量子线路所对应的幺正算符为 $\mathbf{U}(\theta, \gamma) = \mathbf{W}(\theta)\mathbf{D}(\gamma)\mathbf{W}^\dagger(\theta)$ ，其中 $\mathbf{D}(\gamma)$ 是各个单比特 Rz 门的直积， γ 是这些门的旋转角参数。通过定义合适的损失函数，利用经典参数优化，使得 $\mathbf{U}(\theta, \gamma)$ 逐渐接近哈密顿量演化算符 $e^{-i\mathbf{H}\Delta t}$ ，其中 Δt 是一个较小的量。这样，如果要实现较长时间 $T = N\Delta t$ 的演化，只需要通过参数化量子线路 $\mathbf{U}^N(\theta, \gamma) = \mathbf{W}(\theta)\mathbf{D}(N\gamma)\mathbf{W}^\dagger(\theta)$ 即可。由于此方法中参数化量子线路的形式与哈密顿量对角化的数学形式相同，因此从参数 γ 中就可以近似得到哈密顿量的本征值，所以 VFF 方法也可以应用于本征值估计问题^[172]。

2.4 幺正算符线性组合及非幺正量子算法

前面介绍的量子算法都是幺正的算法，形如 $|\psi_f\rangle = \mathbf{U}_m \cdots \mathbf{U}_2 \mathbf{U}_1 |\psi_0\rangle$ ，其实现的是量子线路中各个幺正算符的乘积；但幺正算符的加减，或者说线性组合，却不能在量子线路中直接实现。而我们在1.2.2小节中介绍了非幺正量子线路，可以用来实现非幺正操作。非幺正量子线路的过程通常描述如下：假设工作比特的初始状态为 $|\psi_0^S\rangle$ （用上标 S 表示工作比特），辅助比特的初始状态通常为 $|0^A\rangle$ （用上标 A 表示辅助比特）。将整个量子线路中所有幺正操作的总效果记为 \mathbf{U} ，它是作用在所有比特上的算符，则经过 \mathbf{U} 后线路中的总量子态为 $\mathbf{U}|0^A\rangle|\psi_0^S\rangle$ 。接着对辅助比特进行测量，并希望得到的测量结果为 $|\phi^A\rangle$ ，通常 $|\phi^A\rangle$ 为计算基。若对辅助比特的测量成功得到了 $|\phi^A\rangle$ ，则线路中工作比特的状态变为 $\alpha\langle\phi^A|\mathbf{U}|0^A\rangle|\psi_0^S\rangle$ ，其中 α 为归一系数。可以看出，整个量子线路对工作比特的作用效果等价于算符

$\mathbf{N} = \alpha \langle \phi^A | \mathbf{U} | 0^A \rangle$ 。通常而言， \mathbf{N} 是非幺正的，我们可以通过对 \mathbf{U} 与 $|\phi^A\rangle$ 进行设计和选择，来精确地或近似地实现出任意想要的非幺正算符。

为了在量子计算机中更广泛地实现非幺正的量子算法，清华大学龙桂鲁于 2006 年提出了一种新的量子计算方法，称为对偶量子计算^[173-176]。不同于以往只能实现幺正算符乘积的量子计算，对偶量子计算可以利用非幺正量子线路实现幺正算符的线性组合，极大地扩展了量子算法的构造方式。

2.4.1 幺正算符线性组合方法

幺正算符线性组合 (linear combination of unitary, LCU) 可以理解为如下物理过程：一个波函数经过一个分波器，分成若干束子波，每一个子波各自经过一个不同的幺正算符，最后经过一个合波器重新合成一束波。这样就等效于实现了各个幺正算符的加法操作。

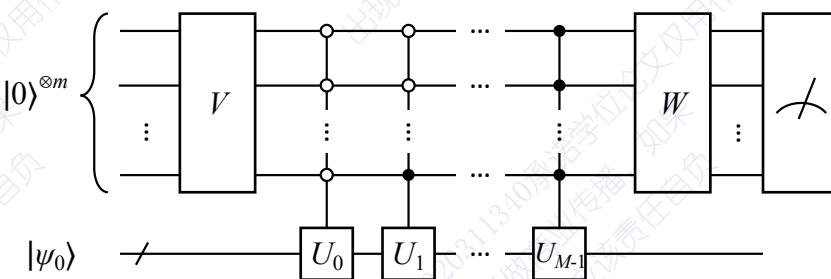


图 2.9 实现幺正算符线性组合的量子线路

在量子计算中，LCU 所对应的量子线路如图2.9所示。假设初始量子态为 $|\psi_0\rangle$ ，有 M 个幺正算符 U_0, U_1, \dots, U_{M-1} ，LCU 可以将它们的线性组合作用在初态上： $|\psi'\rangle = \alpha \sum_{j=0}^{M-1} \gamma_j U_j |\psi_0\rangle$ ，其中 γ_j 为组合系数， α 为量子态的归一系数。LCU 的量子线路使用了 $m = \log M$ 个辅助比特，先在辅助比特上作用 \mathbf{V} 算符，相当于分波器，把 $|\psi_0\rangle$ 分成 M 束子波，每个子波分别用辅助比特的 $|0\rangle, |1\rangle, \dots, |M-1\rangle$ 状态进行标记；接着作用一系列受控操作 $\sum_{j=0}^{M-1} |j\rangle\langle j| \otimes U_j$ ，相当于对于不同的子波进行不同的幺正操作；最后再作用 \mathbf{W} 算符，相当于合波器。经过 \mathbf{W} 后，线路中的总量子态为

$$\sum_{j=0}^{M-1} \mathbf{W} |j\rangle\langle j| \mathbf{V} |0\rangle^{\otimes m} \otimes U_j |\psi_0\rangle = \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} W_{kj} V_{j0} |k\rangle \otimes U_j |\psi_0\rangle. \quad (2.29)$$

这时对辅助比特进行测量，如果所有辅助比特都得到 0，则工作比特的量子态变为

$$\alpha_1 \sum_{j=0}^{M-1} W_{0j} V_{j0} U_j |\psi_0\rangle. \quad (2.30)$$

其中 α_1 为量子态的归一系数。只要 \mathbf{V} 和 \mathbf{W} 的矩阵形式满足 $W_{0j} V_{j0} = C \gamma_j$ ，

其中 C 为任意常系数，那么通过以上方式，就可以实现出幺正算符的线性组合 $\alpha \sum_{j=0}^{M-1} \gamma_j \mathbf{U}_j$ ，其中 $\alpha = C\alpha_1$ 。

已经证明，所有的线性有界算符都可以利用 LCU 方法实现^[177-178]，这使得 LCU 成为了一种能广泛容纳各种量子算法的理论。目前 LCU 已被应用于多项式函数优化^[179-180]、本征态求解^[181-185]、哈密顿量模拟^[142,186-187]、开放量子系统模拟^[182,188-189]、量子信号处理^[79,81-82,190-194]等问题。

2.4.2 全量子本征求解算法

全量子本征求解器（full quantum eigensolver, FQE）由魏世杰、李行、龙桂鲁于 2019 年提出^[181]，是 LCU 最为重要的应用之一，可以用来求解哈密顿量的基态和基态能量。与 VQE 类似，FQE 也把量子态的能量期望值作为损失函数： $f(|\mathbf{X}\rangle) = \langle \mathbf{X} | \mathbf{H} | \mathbf{X} \rangle$ ，其中 $|\mathbf{X}\rangle$ 是量子系统的量子态，并采用梯度下降法求损失函数的极小值，损失函数的梯度为

$$\nabla f(|\mathbf{X}\rangle) = 2\mathbf{H} |\mathbf{X}\rangle. \quad (2.31)$$

根据梯度下降法，每次迭代过程中量子态的变化为

$$|\mathbf{X}^{(k+1)}\rangle = |\mathbf{X}^{(k)}\rangle - 2\gamma_0 \mathbf{H} |\mathbf{X}^{(k)}\rangle = (\mathbf{I} - \gamma \mathbf{H}) |\mathbf{X}^{(k)}\rangle, \quad (2.32)$$

其中 γ_0 是梯度下降的步长， $\gamma = 2\gamma_0$ 。定义梯度下降算符： $\mathbf{H}^g = \mathbf{I} - \gamma \mathbf{H}$ ，并将其写成泡利直积项之和：

$$\mathbf{H}^g = \sum_{j=1}^M \beta_j \mathbf{H}_j^g, \quad (2.33)$$

其中 \mathbf{H}_j^g 是泡利直积算符，这样就可以利用 LCU 方法实现 \mathbf{H}^g 对量子态的作用，从而实现一步梯度下降过程。FQE 将算符 \mathbf{H}^g 不断作用在初始态上，经过足够多次迭代后，量子态将逐渐收敛至基态，能量期望值也将逐渐收敛至基态能量。

FQE 最早被应用于化学分子的基态求解问题，并在此后引发了许多相关的工作：闻经纬等人^[183]通过平移哈密顿量的能谱，将 FQE 扩展为可以求解分子激发态的算法；吕鹏等人^[184]将 FQE 应用于原子核壳模型的基态求解，并开发了完整的量子程序包；王博梽等人^[185]通过将 FQE 中的梯度下降算符进行幂次计算，减少了求解基态过程的迭代次数，提高了算法的总成功概率，并将改进后的算法应用于固体材料的能带计算。以上工作均在超导量子计算云平台中进行了实验验证。此外，Jin-Min Liang 等人^[195]通过将参数化量子线路引入辅助比特量子态制备过程，优化了单次迭代的成功概率，并在一定程度上降低了门复杂度。

2.5 线性系统的量子算法

针对线性系统（向量、矩阵）的运算在各学科中广泛存在。诸如主成分分析、奇异值分解、二次回归等线性系统求解在机器学习中有大量的应用。通常而言，实现线性系统运算的经典算法的时间复杂度都是系统维数的多项式量级，最好的情况也是线性量级。而大数据时代中，信息量的飞速增长使得经典算法逐渐变得困难。2008年，第一个求解线性方程组的量子算法由 Harrow、Hassidim 和 Lloyd 提出，被称为 HHL 算法^[11]，它可以在线性方程组维数的对数时间内得到线性方程组的解，从而实现了对经典算法的指数超越。HHL 算法的出现，直接推动了各类求解线性系统问题的量子算法的发展。

2.5.1 量子线性方程组求解算法与矩阵函数的实现

线性方程组的求解一直以来都是重要的数学问题。给定矩阵 \mathbf{A} 和向量 \mathbf{b} ，求解满足 $\mathbf{Ax} = \mathbf{b}$ 的向量 \mathbf{x} 。在经典算法中，求解线性方程组需要的时间通常是矩阵 \mathbf{A} 的维度的多项式量级。第一个求解线性方程组的量子算法是 HHL 算法，它采用振幅编码的方式将经典向量 \mathbf{b} 编码到量子态中：

$$\mathbf{b} = (b_0, b_1, \dots, b_{N-1})^T \rightarrow |\mathbf{b}\rangle = \frac{1}{\sqrt{|\mathbf{b}|^2}} \sum_{j=0}^{N-1} b_j |j\rangle, \quad (2.34)$$

其中 N 为向量 \mathbf{b} 的维数。这相当于用量子态各个计算基上的振幅来储存经典向量的各个分量，但由于量子态本身是归一的，这种方式会丢失经典向量模长的信息。这里我们不关心经典向量的模长。

HHL 算法假设矩阵 \mathbf{A} 为 $N \times N$ 维厄米矩阵，并希望给出线性方程组 $\mathbf{Ax} = \mathbf{b}$ 的解 $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ ，即需要计算出矩阵 \mathbf{A} 的逆 \mathbf{A}^{-1} 并将其作用在 \mathbf{b} 上。对应于量子态，则需要将算符 \mathbf{A}^{-1} 作用在量子态 $|\mathbf{b}\rangle$ 上。HHL 算法的量子线路如图2.10所示，其中在储存量子态 $|\mathbf{b}\rangle$ 的比特（下标 V）之外，额外引入了 r 个用来储存相位估计值的比特（下标 E）和 1 个辅助比特（下标 A）。线路主要分为四个部分，分别对应算法的四个主要步骤，下面来具体说明。

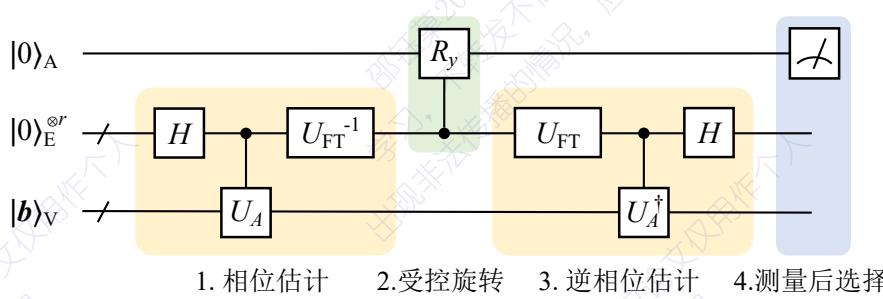


图 2.10 HHL 算法的量子线路图

1. 量子相位估计：假设矩阵 \mathbf{A} 的本征值和本征态分别为 λ_j 和 $|u_j\rangle$, $j = 0, 1, \dots, N - 1$ 。量子态 $|\mathbf{b}\rangle$ 可写为 \mathbf{A} 的本征态的叠加: $|\mathbf{b}\rangle = \sum_j \beta_j |u_j\rangle$ 。相位估计中的幺正矩阵为 $\mathbf{U}_A = e^{i\mathbf{At}}$, 其中 t 是人为选定的参数。则经过相位估计部分后, 线路中的总量子态为

$$|0\rangle \sum_{j=0}^{N-1} \beta_j |\tilde{\lambda}_j\rangle |u_j\rangle, \quad (2.35)$$

其中 $|\tilde{\lambda}_j\rangle$ 中储存了 \mathbf{U}_A 的相位估计值, 也就是 \mathbf{A} 的本征值估计值。

2. 受控旋转：受控旋转门的表达式为

$$\sum_{\tilde{\lambda}} |\tilde{\lambda}\rangle \langle \tilde{\lambda}| \otimes \begin{bmatrix} \sqrt{1 - \frac{C^2}{\tilde{\lambda}^2}} & -\frac{C}{\tilde{\lambda}} \\ \frac{C}{\tilde{\lambda}} & \sqrt{1 - \frac{C^2}{\tilde{\lambda}^2}} \end{bmatrix}. \quad (2.36)$$

其中 C 为可选择的常数。则经过受控旋转后, 线路中的总量子态为

$$\sum_{j=0}^{N-1} \beta_j \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right) |\tilde{\lambda}_j\rangle |u_j\rangle. \quad (2.37)$$

3. 逆相位估计：即相位估计的逆操作，使得线路中的总量子态变为

$$\sum_{j=0}^{N-1} \beta_j \left(\sqrt{1 - \frac{C^2}{\tilde{\lambda}_j^2}} |0\rangle + \frac{C}{\tilde{\lambda}_j} |1\rangle \right) |0\rangle^{\otimes r} |u_j\rangle. \quad (2.38)$$

4. 测量后选择：测量辅助比特，若得到结果为 1，则线路中其他比特的总状态变为

$$\alpha \sum_{j=0}^{N-1} \beta_j \frac{C}{\tilde{\lambda}_j} |0\rangle^{\otimes r} |u_j\rangle = \alpha C |0\rangle^{\otimes r} \sum_{j=0}^{N-1} \tilde{\lambda}_j^{-1} \beta_j |u_j\rangle = \alpha C |0\rangle^{\otimes r} \mathbf{A}^{-1} |\mathbf{b}\rangle. \quad (2.39)$$

经过以上步骤后就得到了量子态 $\mathbf{A}^{-1} |\mathbf{b}\rangle$, 其对应的就是线性方程组的解 \mathbf{x} 。通过对量子态进行测量, 即可恢复经典向量 \mathbf{x} 的信息。

HHL 算法假设了 \mathbf{A} 是厄米矩阵, 如果 \mathbf{A} 不是厄米矩阵 (包括 \mathbf{A} 不是方形矩阵的情形), 则可以定义:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^\dagger & \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{x} \end{bmatrix}, \quad \tilde{\mathbf{b}} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}, \quad (2.40)$$

那么原线性方程组就等价于 $\tilde{\mathbf{A}} \tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, 且 $\tilde{\mathbf{A}}$ 一定是厄米矩阵, 此时再按照 HHL 算法的步骤进行计算即可。

HHL 算法的复杂度主要来自于相位估计、逆相位估计和测量后选择的成功概率, 其复杂度表达式为 $\mathcal{O}(\log(N)s^2\kappa^2/\epsilon)$, 其中 N 为矩阵 \mathbf{A} 的维数、 s 是矩阵的稀疏度 (非零矩阵元个数在所有元素中的比例)、 κ 为矩阵的条件数 (最大本征值与

最小本征值之比)、 ϵ 为求解精度。当矩阵 \mathbf{A} 的稀疏度和条件数都较小、需要的精度不高时, HHL 算法相比于经典的线性方程组求解算法可以实现指数加速。HHL 算法为量子计算机求解线性系统提供了高效方法, 并直接推动了各类量子机器学习算法的发展, 如量子支持向量机^[196]、量子奇异值收缩^[197]、量子线性回归^[198]、量子推荐系统^[199]等。

将 HHL 算法的思路进行推广, 可以近似实现任意的矩阵函数。我们同样假设 \mathbf{A} 为厄米矩阵, 希望实现矩阵函数 $f(\mathbf{A})$ 。采用与 HHL 算法相同的量子线路图2.10, 并将其中的受控旋转操作替换为

$$\sum_{\tilde{\lambda}} |\tilde{\lambda}\rangle \langle \tilde{\lambda}| \otimes \begin{bmatrix} \sqrt{1 - C^2 f^2(\tilde{\lambda})} & -Cf(\tilde{\lambda}) \\ Cf(\tilde{\lambda}) & \sqrt{1 - C^2 f^2(\tilde{\lambda})} \end{bmatrix}, \quad (2.41)$$

并保持线路中其他操作不变, 则在测量后选择之前, 线路中的总量子态为

$$\begin{aligned} & \sum_{j=0}^{N-1} \beta_j \left(\sqrt{1 - C^2 f^2(\tilde{\lambda}_j)} |0\rangle + Cf(\tilde{\lambda}_j) |1\rangle \right) |0\rangle^{\otimes r} |u_j\rangle \\ & = |0\rangle |0\rangle^{\otimes r} \sqrt{1 - C^2 f^2(\mathbf{A})} |\mathbf{b}\rangle + |1\rangle |0\rangle^{\otimes r} Cf(\mathbf{A}) |\mathbf{b}\rangle. \end{aligned} \quad (2.42)$$

测量辅助比特得到 1 后, 就可以得到量子态为 $\alpha Cf(\mathbf{A}) |\mathbf{b}\rangle$, 其对应的经典向量就是将矩阵函数 $f(\mathbf{A})$ 作用在 \mathbf{b} 上得到的结果。这样的实现矩阵函数的方法叫做类 HHL 方法。

2.5.2 量子主成分分析与量子奇异值阈值算法

主成分分析 (principal components analysis, PCA) 是一种减少数据集的维数, 同时保留数据集明显特征的数据分析方法, 常用于图像处理、相关性分析、推荐系统等领域。一般来说, PCA 方法是对矩阵进行谱分解, 通过丢弃低于给定阈值的特征值和对应的特征向量, 从而得到矩阵的最优低秩近似。当面对大量的高维信息时, 经典的 PCA 方法将消耗大量的计算资源。

第一个针对量子系统实现 PCA 的量子算法由 Lloyd 等人于 2014 年提出, 被称为量子主成分分析 (quantum PCA, QPCA)^[200]。它可以针对未知的低秩量子态密度矩阵进行主成分分析, 以得到较大本征值及其对应的本征量子态。QPCA 的核心思想是, 对于一个未知密度矩阵 ρ , 通过量子相位估计来计算其本征值, 而量子相位估计需要我们能实现 ρ 的幺正演化算符 $e^{i\rho t}$ 。假设我们有两个维度相同的量

子系统 A 和 B, 密度矩阵分别为 ρ_A 和 σ_B , 那么有如下公式:

$$\begin{aligned} \text{tr}_A e^{-iS\Delta t} (\rho_A \otimes \sigma_B) e^{iS\Delta t} &= (\cos^2 \Delta t) \sigma + (\sin^2 \Delta t) \rho - i \cos \Delta t \sin \Delta t [\rho, \sigma] \\ &= \sigma - i \Delta t [\rho, \sigma] + \mathcal{O}(\Delta t^2) = e^{-i\rho\Delta t} \sigma e^{i\rho\Delta t} + \mathcal{O}(\Delta t^2), \end{aligned} \quad (2.43)$$

其中 tr_A 表示对子系统 A 求迹, S 是两个量子系统之间的交换算符, 满足 $S |i\rangle_A |j\rangle_A = |j\rangle_A |i\rangle_B$, 由于 S 是稀疏矩阵, 因此 $e^{-iS\Delta t}$ 可以高效实现^[140-142]。

根据这一公式, 只要有密度矩阵为 ρ 的量子系统, 就可以在其他量子系统上实现算符 $e^{i\rho\Delta t}$ 。

对于较长的演化时间 t , 则需要将 t 切分成更短的时间间隔 $\Delta t = t/n$, 并使用 n 份密度矩阵都为 ρ 的量子系统, 从而实现 $e^{i\rho t} = (e^{i\rho\Delta t})^n$ 。理论分析表明, 若希望相位估计的精度为 ϵ , 则需要 $t = \mathcal{O}(\epsilon^{-1})$, 同时需要密度矩阵 ρ 的份数为 $n = \mathcal{O}(\epsilon^{-3})$ 。

如果将算符 $e^{i\rho t}$ 作用于密度矩阵 ρ 本身, 则经过相位估计后, 量子线路中的量子态对应的密度矩阵为

$$\sum_j \lambda_j |\tilde{\lambda}_j\rangle \langle \tilde{\lambda}_j| \otimes |\phi_j\rangle \langle \phi_j|, \quad (2.44)$$

其中 λ_j 是 ρ 的本征值, $|\phi_j\rangle$ 是对应的本征态, $|\tilde{\lambda}_j\rangle$ 是储存 λ_j 二进制估计值的量子态。对这个量子态进行采样和测量, 就可以得到 ρ 的本征系统的特征。

矩阵的奇异值阈值计算在机器学习中有重要应用。量子奇异值阈值 (quantum singular value threshold, QSVT) 算法最早由 Bojia Duan 等人提出^[196-197]。对于给定的低秩矩阵 $A_0 \in \mathbb{R}^{p \times q}$, 假设其奇异值分解为 $A_0 = \sum_{k=1}^r \sigma_k u_k v_k^T$, QSVT 希望给出对 A_0 进行奇异值阈值操作后的结果:

$$S = D_\tau(A_0) = \sum_{k=1}^r s_\tau(\sigma_k) u_k v_k^T, \quad s_\tau(\sigma_k) = \max(\sigma_k - \tau, 0). \quad (2.45)$$

QSVT 算法首先需要构造量子态 $|\psi_{A_0}\rangle = \alpha_1 \sum_k \sigma_k |u_k\rangle |v_k\rangle$, 其中 α_1 是归一系数。这里由于 u_k 和 v_k 都是单位向量, 因此可以直接通过振幅编码方式储存在量子态中。接着, 定义算符 $A = A_0 A_0^\dagger$, 则有 $A = \sum_{k=1}^r \sigma_k^2 u_k u_k^T$ 。注意到, 如果对量子态 $|\psi_{A_0}\rangle$ 中的各个 $|v_k\rangle$ 对应的子系统求迹, 那么剩下部分的密度矩阵刚好正比于 A , 即 $\text{tr}_2(|\psi_{A_0}\rangle \langle \psi_{A_0}|) = \alpha_1^2 \sum_k \sigma_k^2 |u_k\rangle \langle u_k| = A / \text{tr} A$ 。这样就可以按照 QPCA 方法, 在量子线路中实现 A 的幺正演化 e^{iAt} 。

QSVT 算法的量子线路如图2.11所示。QSVT 是一种类 HHL 算法, 其本质也是利用相位估计得到 A 的本征值估计值, 再通过受控旋转实现 A 的矩阵函数。因此 QSVT 的线路结构与 HHL 相似, 包括相位估计、受控旋转、逆相位估计、测量

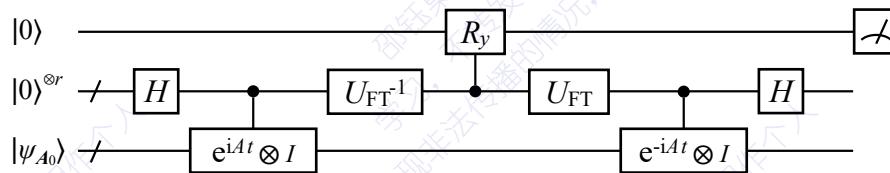


图 2.11 QSVD 算法的量子线路图

后选择这四个部分，其中受控旋转操作的具体形式为

$$\sum_{\tilde{\lambda}} |\tilde{\lambda}\rangle \langle \tilde{\lambda}| \otimes \begin{bmatrix} \cos \theta_{\tilde{\lambda}} & -\sin \theta_{\tilde{\lambda}} \\ \sin \theta_{\tilde{\lambda}} & \cos \theta_{\tilde{\lambda}} \end{bmatrix}, \quad \sin \theta_{\tilde{\lambda}} = \max \left(1 - \frac{\tau}{\sqrt{\tilde{\lambda}}}, 0 \right), \quad (2.46)$$

这样，当对辅助比特的测量得到结果为 1 时，就可以得到量子态 $\alpha_2 \sum_k \max(\sigma_k - \tau, 0) |\mathbf{u}_k\rangle |\mathbf{v}_k\rangle$ ，其中 α_2 是归一系数。该量子态所对应的矩阵即为奇异值阈值后的矩阵 \mathbf{S} ，由此我们就可以实现对矩阵 \mathbf{A}_0 的奇异值阈值操作。

2.6 本章小结

本章主要介绍了量子计算中常用的、较为重要的量子算法。首先，我们介绍了算法复杂度的概念，并介绍了量子计算早期的几个基础的量子算法——量子搜索算法、量子傅立叶变换、量子相位估计、基于 Trotter 分解的哈密顿量模拟算法，它们在量子计算中至今仍然发挥着重要作用；然后我们介绍了一类基于参数化量子线路的变分量子算法，其中包括变分量子本征求解器、量子近似优化算法、变分量子模拟等，在当今含噪声中等规模量子时代，这些算法能在一定程度上发挥量子硬件的优势；随后我们介绍了量子计算的另一种构造方式——幺正算符线性组合与非幺正量子算法，其重要的应用包括全量子本征求解算法等；最后我们介绍了线性系统求解相关的量子算法，包括 HHL、QPCA、QSVD 等算法。

本章所介绍的算法复杂度概念将为本论文所研究的算法提供重要的衡量依据；而本章所介绍的量子算法中，有些作为本论文所研究的算法的子程序出现，有些为本论文所研究的算法提供了研究方法，有些则因与本论文所研究的算法有共同应用而作为比较的对象。而与本文所研究的算法直接相关的内容，将在后面对应的章节中进行介绍。

第3章 基于非幺正量子线路的概率性虚时演化算法

虚时演化是量子模拟中的重要数学工具。本章节中，我们基于非幺正量子线路，构造了一种实现虚时演化的量子算法，并在演示实验与数值模拟中实现了算法在基态求解问题中的应用。

3.1 虚时演化及其相关算法

虚时演化 (imaginary-time evolution, ITE) 是量子模拟中的一个重要数学工具，它在基态求解^[80,201]、模拟有限温系统^[202-204]、开放量子系统模拟^[205]等问题中都有广泛应用。虚时演化是将薛定谔方程中的时间变成纯虚数： $t = -i\beta$ ，这里 β 为实数，称为虚时间。这样得到的薛定谔方程就称为虚时间薛定谔方程：

$$-\frac{\partial |\Phi_\beta\rangle}{\partial \beta} = \mathbf{H} |\Phi_\beta\rangle. \quad (3.1)$$

给定 $t = 0$ 时刻系统的初始量子态 $|\Phi_0\rangle$ ，虚时间薛定谔方程的解为

$$|\Phi_\beta\rangle = e^{-\beta\mathbf{H}} |\Phi_0\rangle. \quad (3.2)$$

该解从数学上符合演化方程，但从物理上并未考虑量子态的归一化。考虑归一化后，需要引入归一系数 $A = \langle\Phi_0|e^{-2\beta\mathbf{H}}|\Phi_0\rangle^{-1/2}$ ，则虚时演化量子态为

$$|\Phi_\beta\rangle = A e^{-\beta\mathbf{H}} |\Phi_0\rangle, \quad (3.3)$$

同时，虚时间薛定谔方程也应该修正为：

$$-\frac{\partial |\Phi_\beta\rangle}{\partial \beta} = \mathbf{H} |\Phi_\beta\rangle - E_\beta |\Phi_\beta\rangle, \quad (3.4)$$

其中 $E_\beta = \langle\Phi_\beta|\mathbf{H}|\Phi_\beta\rangle$ 。

在经典算法中，虚时演化的计算可以通过直接计算矩阵函数 $e^{-\beta\mathbf{H}}$ 并将其作用在初始态 $|\Phi_0\rangle$ 上而实现，也可以采用一些经典的量子多体系统计算方法，如量子蒙特卡洛^[206]、张量网络^[207-209]等。然而，量子体系的维数 N 与粒子数呈指数级关系，而经典算法计算矩阵指数通常需要 $\mathcal{O}(N^3)$ 的复杂度^[210]，随着所研究的粒子数的增长，经典计算所消耗的资源将迅速增加，使虚时演化的计算变得非常困难。量子计算是有效解决这一困难的途径之一。而虚时演化算符 $e^{-\beta\mathbf{H}}$ 是非幺正的，无法在一般的量子计算机中直接实现，或通过拆解成若干幺正算符的乘积实现，因此必须寻找其他方法。

目前，已有三类能够在量子计算机中实现虚时演化的算法。第一类是变分量子

模拟(VQS)^[167-169,211-212], 将VQS应用于虚时演化最早由Sam McArdle等人^[167]提出, 算法的具体过程与2.3.3小节中利用VQS实现哈密顿量的幺正演化类似: 首先将参数化量子态 $|\psi(\theta)\rangle$ 代入归一化的虚时演化方程, 得到

$$-\sum_k \frac{\partial |\Phi\rangle}{\partial \theta_k} \dot{\theta}_k = (\mathbf{H} - E) |\Phi\rangle, \quad (3.5)$$

其中 $\dot{\theta}_k = d\theta_k/d\beta$ 。对该方程的模长做变分, 并进一步处理, 最终得到参数随虚时间 β 的演化方程:

$$\sum_k A_{jk} \dot{\theta}_k = B_j, \quad (3.6)$$

其中

$$A_{jk} = \operatorname{Re} \left(\frac{\partial \langle \psi |}{\partial \theta_j} \frac{\partial | \psi \rangle}{\partial \theta_k} \right), \quad B_j = -\operatorname{Re} \left(\frac{\partial \langle \psi |}{\partial \theta_j} \mathbf{H} | \psi \rangle \right). \quad (3.7)$$

根据量子线路中的测量结果得到 A_{jk} 与 B_j 的值, 再通过经典优化器计算出下一步迭代参数, 不断重复这一过程直至达到所需要的演化时长。由于VQS并非基于梯度的优化方法, 因此不会收到局域极值的影响, 从这一方面来说VQS的收敛效果要优于VQE^[167]。

第二类是量子虚时演化(quantum imaginary-time evolution, QITE)算法, 这一方法最早由Mario Motta等人^[213]提出。假设哈密顿量为 $\mathbf{H} = \sum_m \mathbf{h}_m$, 其中每一项 \mathbf{h}_m 都作用在不超过 k 个相邻比特上。利用Trotter分解公式, 得到

$$e^{-\beta \mathbf{H}} = (e^{-\Delta \tau \mathbf{h}_1} e^{-\Delta \tau \mathbf{h}_2} \cdots)^{\beta/\Delta \tau} + \mathcal{O}(\Delta \tau). \quad (3.8)$$

在每一步Trotter迭代中, 需要实现如下演化: $|\psi'\rangle = e^{-\Delta \tau \mathbf{h}_m} |\psi\rangle$ 。QITE的基本思路是假设归一化末态 $|\bar{\psi}'\rangle = |\psi'\rangle / \| |\psi'\rangle \|$ 可以近似由幺正演化得到, 即 $|\bar{\psi}'\rangle = e^{-i\Delta \tau \mathbf{A}_m} |\psi\rangle + \mathcal{O}(\Delta \tau)$, 其中 \mathbf{A}_m 是厄米算符, 作用在与 \mathbf{h}_m 相同的 k 个比特上, 且可以写成泡利项之和:

$$\mathbf{A}_m = \sum_{i_1, \dots, i_k} a_{i_1, \dots, i_k} \sigma_{i_1} \cdots \sigma_{i_k} \equiv \sum_I a_I \sigma_I. \quad (3.9)$$

为了求出各系数 a_I , 将表达式 $\| |\psi'\rangle - e^{-i\Delta \tau \mathbf{A}_m} |\psi\rangle \|^2$ 对 $\Delta \tau$ 展开, 令其最低阶项系数为0, 由此得到线性方程组:

$$\sum_{I'} S_{II'} a_{I'} = b_I, \quad (3.10)$$

其中

$$S_{II'} = \langle \psi | \sigma_I \sigma_{I'} | \psi \rangle, \quad b_I = -\frac{i \langle \psi | \sigma_I \mathbf{h}_m | \psi \rangle}{\sqrt{1 - 2\Delta \tau \langle \psi | \mathbf{h}_m | \psi \rangle}}. \quad (3.11)$$

因此，对于每一步 Trotter 迭代，先测量出当前量子态的 $S_{II'}$ 与 b_I ，通过求解线性方程组得到 \mathbf{A}_m ，再对当前量子态进行幺正操作 $e^{-i\Delta\tau\mathbf{A}_m}$ 即可。其中需要求解的线性方程组的维数为 4^k ，因此算法要求 k 不能过大，即要求哈密顿量具有局域纠缠性。但当演化时间较长时，相互纠缠的比特数会增多，使得线性方程组的维数也迅速增大，因此算法在长演化时长下的复杂度也会迅速增加^[214]。总体而言，QITE 提供了一种新的非幺正量子算法的设计方式，引发了许多相关的工作^[205,214-220]。

第三类是基于非幺正量子线路的虚时演化算法，这一类算法中代表性工作包括文献^[221-226]等，它们都是通过对辅助比特的测量后选择来实现非幺正的虚时演化算符，但具体实现方法各不相同。例如，Tong Liu 等人提出的方法^[222]能够实现非幺正算符 Σ ，将幺正算符 $\mathbf{U} = \exp[-i\epsilon(\sigma_y \otimes \Sigma)]$ 作用在工作比特和 1 个辅助比特上，得到 $\mathbf{U}|0\rangle|\psi\rangle = |0\rangle\cos(\epsilon\Sigma)|\psi\rangle + |1\rangle\sin(\epsilon\Sigma)|\psi\rangle$ ，若测量辅助比特得到 $|1\rangle$ 且 ϵ 很小时，即可近似得到 $\epsilon\Sigma|\psi\rangle$ 。F. Turro 等人的方法^[223]通过如下幺正算符实现虚时演化：

$$\mathbf{U}(\tau) = \left(1 + e^{-2(H-E_T)\tau}\right)^{-1/2} \begin{bmatrix} e^{-(H-E_T)\tau} & 1 \\ 1 & -e^{-(H-E_T)\tau} \end{bmatrix}, \quad (3.12)$$

其中 E_T 为小于精确基态能量的值。引入初态为 $|0\rangle$ 的辅助比特，经过 $\mathbf{U}(\tau)$ 后测量辅助比特得到 $|0\rangle$ ，即可实现 $e^{-(H-E_T)\tau}$ 。Taichi Kosugi 等人的工作^[224]使用一个辅助比特，利用 LCU 方法在工作比特上实现哈密顿量的幺正演化算符 $e^{iH\Delta\tau}$ 与 $e^{-iH\Delta\tau}$ 的线性组合，从而以 $\mathcal{O}(\Delta\tau^2)$ 的误差近似实现虚时演化 $e^{-H\Delta\tau}$ 。此外，由魏世杰等人提出的 FQE 方法在数学形式上等价于虚时演化的一阶近似，因此也可以通过 FQE 近似实现虚时演化。

以上这些虚时演化算法在实际应用中各自会受到限制。例如，利用 VQS 实现虚时演化面临参数表达能力有限、计算复杂度缺乏理论保证等问题；QITE 对演化时长和局域纠缠性的要求限制了其复杂度的优越性；基于非幺正量子线路的虚时演化算法中，FQE 所需要的多比特受控量子门难以在现有的量子硬件上实现，而其他方法难以给出普适且具体的量子线路构造方式，难以针对一般形式的哈密顿量实现算法的具体应用。

本章节后面的内容将介绍我们所提出的一种全新的基于非幺正量子线路的虚时演化算法，该算法只需 1 个辅助比特，并且描述了针对任意由泡利项构成的哈密顿量给出利用基本量子门构造量子线路的系统性方法。我们还将展示将此算法应用到量子系统基态求解问题的一些数值模拟和演示实验结果。

3.2 适用于泡利形式哈密顿量的概率性虚时演化算法

基于非幺正量子线路，我们设计了一种能够模拟量子系统虚时演化的量子算法，叫做概率性虚时演化（probabilistic imaginary-time evolution, PITE）算法。PITE 算法的原始形式适用于任何由泡利项构成的哈密顿量，系统性地给出了利用基本量子门构造量子线路的具体方法，因此适合在量子硬件中进行实现。

假设一个 n 比特哈密顿量可以写成 m 个泡利项相加的形式，即 $\mathbf{H} = \sum_{k=1}^m c_k \mathbf{h}_k$ ，其中 c_k 是实系数， $\mathbf{h}_k = \otimes_{j=1}^n \sigma_{\alpha_j}^j$ ，这里 $\sigma_{\alpha_j}^j$ 表示作用在第 j 个比特上的泡利矩阵或单位矩阵， $\alpha_j \in \{0, x, y, z\}$ ，约定 $\sigma_0 = \mathbf{I}$ 。我们假设 m 是 $\text{poly}(n)$ 的量级，且哈密顿量中不包含常数项 $\mathbf{I}^{\otimes n}$ ，因为常数项仅导致哈密顿量的能谱平移。

我们希望在量子线路中实现虚时演化算符 $e^{-\beta \mathbf{H}}$ 。首先，我们对该算符进行 Trotter 分解：

$$e^{-\beta \mathbf{H}} = (e^{-c_1 \mathbf{h}_1 \Delta \tau} \cdots e^{-c_m \mathbf{h}_m \Delta \tau})^{\beta / \Delta \tau} + \mathcal{O}(\Delta \tau). \quad (3.13)$$

对于每一步 Trotter 迭代，我们希望得到 $|\Phi'\rangle = \tilde{T}_k |\Phi\rangle$ ，其中 $\tilde{T}_k = e^{-c_k \mathbf{h}_k \Delta \tau}$ 。由于 $c_k \mathbf{h}_k$ 仅有两个不同的本征值 $\pm |c_k|$ ，简并度各为 2^{n-1} ，因此一定存在一个幺正算符 U_k ，使得

$$U_k c_k \mathbf{h}_k U_k^\dagger = -|c_k| \sigma_z^{l_k}, \quad l_k \in \{1, \dots, n\}, \quad (3.14)$$

等号右侧为一个单比特算符。因此有

$$\tilde{T}_k = U_k^\dagger \exp(-|c_k| \sigma_z^{l_k} \Delta \tau) U_k. \quad (3.15)$$

我们利用如图3.1所示的量子线路图来实现 \tilde{T}_k 。其中 U_k 的构造最多需要 $\mathcal{O}(n)$ 个单比特门和 CNOT 门， $\exp(-|c_k| \sigma_z^{l_k} \Delta \tau)$ 的构造需要用到非幺正量子线路，仅需用到 1 个受控 R_y 操作和一次测量。下面讨论具体的实现细节。

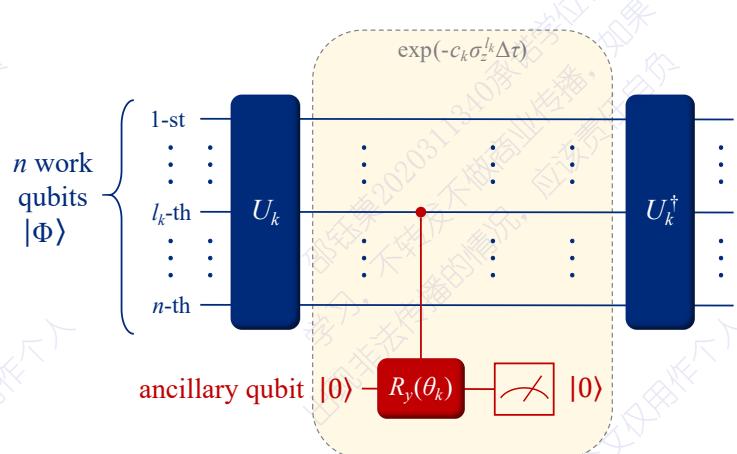


图 3.1 实现虚时演化算符 \tilde{T}_k 的量子线路

首先我们证明，满足上述条件的 \mathbf{U}_k 可以分解为三个幺正矩阵的乘积： $\mathbf{U}_k = \mathbf{V}_{k3}\mathbf{V}_{k2}\mathbf{V}_{k1}$ 。图3.2中给出了两个如何构造 \mathbf{U}_k 的例子。第一步，我们注意到

$$\mathbf{H}\sigma_x\mathbf{H} = \sigma_z, \quad \mathbf{H}\mathbf{S}^\dagger\sigma_y\mathbf{S}\mathbf{H} = \sigma_z, \quad (3.16)$$

其中 \mathbf{H} 和 \mathbf{S} 分别表示 Hadamard 门和 S 门，这样一来， \mathbf{V}_{k1} 可以通过将 \mathbf{H} 和 $\mathbf{H}\mathbf{S}^\dagger$ 分别作用在 σ_x 和 σ_y 对应的比特上来实现，并且有

$$\mathbf{V}_{k1}c_k \left(\sigma_{\alpha_1}^1 \otimes \cdots \otimes \sigma_{\alpha_n}^n \right) \mathbf{V}_{k1}^\dagger = c_k \sigma_{\gamma_1}^1 \otimes \cdots \otimes \sigma_{\gamma_n}^n, \quad (3.17)$$

其中，若 $\alpha_j = 0$ 则 $\gamma_j = 0$ ，否则 $\gamma_j = z$ 。

第二步，注意到

$$\mathbf{C}_{ij} \left(\sigma_z^i \otimes \sigma_z^j \right) \mathbf{C}_{ij}^\dagger = \mathbf{I}^i \otimes \sigma_z^j, \quad (3.18)$$

其中 \mathbf{C}_{ij} 表示第 i 个比特控制第 j 个比特的 CNOT 门。这样，构造 \mathbf{V}_{k2} 时首先选择一个任意的 l 满足 $\gamma_l = z$ ，然后对所有满足 $\gamma_j = z$ 且 $j \neq l$ 的 j 作用 \mathbf{C}_{jl} 操作，于是有

$$\mathbf{V}_{k2}\mathbf{V}_{k1}c_k \left(\sigma_{\alpha_1}^1 \otimes \cdots \otimes \sigma_{\alpha_n}^n \right) \mathbf{V}_{k1}^\dagger \mathbf{V}_{k2}^\dagger = \mathbf{I}^{\otimes l-1} \otimes c_k \sigma_z^l \otimes \mathbf{I}^{\otimes n-l}. \quad (3.19)$$

最后一步，由于

$$\sigma_x \sigma_z \sigma_x = -\sigma_z, \quad (3.20)$$

那么如果 $c_k > 0$ ，则 $\mathbf{V}_{k3} = \sigma_x^l$ ，否则 $\sigma_{k3} = \mathbf{I}$ 。到此就可以成功构造出 $\mathbf{U}_k = \mathbf{V}_{k3}\mathbf{V}_{k2}\mathbf{V}_{k1}$ ，且这一过程最多需要 $2n+1$ 个单比特门和 $n-1$ 个 CNOT 门，共计不超过 $3n$ 个基本量子门。

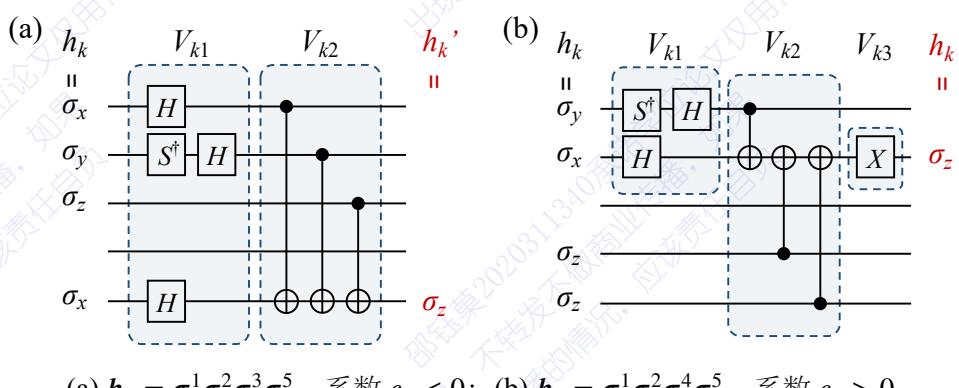


图 3.2 在 5 比特量子线路中利用单比特门和 CNOT 门构造 \mathbf{U}_k 的两个例子

经过 \mathbf{U}_k 的作用后，我们将量子态写为 $\mathbf{U}_k |\Phi\rangle = a_0 |\psi_0\rangle + a_1 |\psi_1\rangle$ ，其中 $|\psi_0\rangle$ 和 $|\psi_1\rangle$ 分别是 $\mathbf{U}_k |\Phi\rangle$ 在第 l_k 个比特的 $|0\rangle$ 和 $|1\rangle$ 两个子空间上的投影。可以证明 a_0 和 a_1 分别是 $|\Phi\rangle$ 在 $c_k \mathbf{h}_k$ 的基态子空间和激发态子空间的投影振幅。

为了在第 l_k 个比特上实现 $e^{-|c_k|\sigma_z\Delta\tau}$, 我们引入一个辅助比特 $|0\rangle$, 并在第 l_k 个比特和辅助比特上进行受控 \mathbf{R}_y 操作: $|0\rangle\langle 0| \otimes \mathbf{I} + |1\rangle\langle 1| \otimes \mathbf{R}_y(\theta_k)$, 其中

$$\theta_k = 2 \arccos(e^{-2|c_k|\Delta\tau}),$$

$$\mathbf{R}_y(\theta_k) = e^{-i\theta_k\sigma_y/2} = \begin{bmatrix} e^{-2|c_k|\Delta\tau} & -\sqrt{1-e^{-4|c_k|\Delta\tau}} \\ \sqrt{1-e^{-4|c_k|\Delta\tau}} & e^{-2|c_k|\Delta\tau} \end{bmatrix}, \quad (3.21)$$

此后, 总量子态变为

$$a_0 |\psi_0\rangle |0\rangle_{\text{anc}} + a_1 e^{-2|c_k|\Delta\tau} |\psi_1\rangle |0\rangle_{\text{anc}} + a_1 \sqrt{1-e^{-4|c_k|\Delta\tau}} |\psi_1\rangle |1\rangle_{\text{anc}}. \quad (3.22)$$

接着, 我们测量辅助比特, 如果测量得到 0, 则在剩余的比特中得到

$$\sqrt{\frac{1}{|a_0|^2 + |a_1|^2 e^{-4|c_k|\Delta\tau}}} (a_0 |\psi_0\rangle + a_1 e^{-2|c_k|\Delta\tau} |\psi_1\rangle), \quad (3.23)$$

在不考虑常系数的情况下, 这一结果与 $e^{-|c_k|\sigma_z\Delta\tau}$ 作用在第 l_k 个比特上的结果完全相同。

上述测量中, 得到 0 的概率为 $|a_0|^2 + |a_1|^2 e^{-4|c_k|\Delta\tau}$ 。如果测量得到 0, 则接下来只要进行 \mathbf{U}_k^\dagger 操作, 则输出态就等价于 $\tilde{\mathbf{T}}_k |\Phi\rangle$, 仅相差一个常系数。

综上而言, 忽略常系数, 非幺正算符 $\tilde{\mathbf{T}}_k$ 作用在量子态 $|\Phi\rangle$ 上的过程可以写成:

$$\tilde{\mathbf{T}}_k |\Phi\rangle = \mathbf{U}_k^\dagger (\langle 0|_{\text{anc}} \mathbf{C}_k \cdot \mathbf{U}_k |\Phi\rangle \otimes |0\rangle_{\text{anc}}), \quad (3.24)$$

其中 \mathbf{U}_k 可以将 \mathbf{h}_k 变换为仅作用在第 l_k 个比特上的泡利 z 算符, \mathbf{C}_k 表示作用在第 l_k 个比特和辅助比特上的受控 \mathbf{R}_y 操作, 旋转角为 $\theta_k = 2 \arccos -1 (e^{-2|c_k|\Delta\tau})$ 。

到此为止, 我们成功构造了 PITE 算法。接下来我们从理论上分析 PITE 算法的复杂度。对于该算法中成功执行的量子线路, 根据 Trotter 分解公式, 需要进行的 Trotter 迭代的次数为 mL , 其中 m 是哈密顿量中泡利项的数量, $L = \beta/\Delta\tau$ 是演化步数; 每一步 Trotter 迭代中, 需要的量子门个数为 $\mathcal{O}(n)$, 因此整个线路中的量子门个数为 $\mathcal{O}(nmL)$ 。线路的深度为 $\mathcal{O}(mL)$ 。而我们主要关心的复杂度则是成功得到所求末态 $|\Phi_\beta\rangle$ 的概率 P , 这一成功概率是源自于非幺正量子线路中的测量后选择。换言之, 平均需要将该算法重复运行 $1/P$ 次, 才能得到一次所求末态, 因此由成功概率而产生的时间复杂度为 $\mathcal{O}(1/P)$ 。

在实现非幺正算符 $\tilde{\mathbf{T}}_k$ 的量子线路中, 测量辅助比特得到 0 的概率为

$$P_k = |a_0|^2 + |a_1|^2 e^{-4|c_k|\Delta\tau} \geq e^{-4|c_k|\Delta\tau}, \quad (3.25)$$

其中 $|a_0|^2 + |a_1|^2 = 1$ 。由此, 我们可以给出总成功概率随虚时间 β 变化的严格下

界 (rigorous lower bound, RLB):

$$P_{\text{final}} \geq \left(\prod_{k=1}^m e^{-4|c_k|\Delta\tau} \right)^{\beta/\Delta\tau} = \exp \left(-4\beta \sum_k |c_k| \right) = P_{\text{RLB}}, \quad (3.26)$$

这一下界与虚时间 β 和各泡利项系数绝对值之和 $\sum_k |c_k|$ 呈指数衰减关系。需要强调的是, 当且仅当3.25式中的 $a_0 = 0$ 对整个虚时演化过程中所有泡利项都成立时, 成功概率才会达到严格下界, 这是最糟糕的情形, 在实例中不可能发生。在实际应用中, 成功概率往往远高于严格下界 (后面会给出相应的例子)。因此, 我们将通过另一种方式重新给出成功概率的下界, 称为近似下界 (approximate lower bound, ALB)。首先根据 $\langle \Phi | c_k h_k | \Phi \rangle = |c_k| (|a_1|^2 - |a_0|^2) = |c_k| (1 - 2|a_1|^2)$, 得到

$$\begin{aligned} P_k &\approx |a_0|^2 + |a_1|^2 (1 - 4|c_k|\Delta\tau) \\ &= 1 - 4|a_1|^2 |c_k| \Delta\tau \\ &= 1 - 2\Delta\tau (\langle \Phi | c_k h_k | \Phi \rangle + |c_k|) \\ &\approx \exp [-2\Delta\tau (\langle \Phi | c_k h_k | \Phi \rangle + |c_k|)], \end{aligned} \quad (3.27)$$

则在演化时长 $\Delta\tau$ 内的累积成功概率近似为

$$P(\Delta\tau) \approx \prod_{k=1}^m P_k = \exp \left[-2\Delta\tau \left(\langle \Phi | H | \Phi \rangle + \sum_k |c_k|^2 \right) \right]. \quad (3.28)$$

这里我们近似认为 $|\Phi\rangle$ 在 $\Delta\tau$ 的演化时长内保持不变。由此得到虚时间 β 的总累计成功概率:

$$P_{\text{final}} \approx \exp \left[-2 \int_0^\beta d\tau \left(\langle \Phi | H | \Phi \rangle + \sum_k |c_k|^2 \right) \right]. \quad (3.29)$$

为了计算这一积分, 我们需要计算能量平均值 $\langle \Phi | H | \Phi \rangle$ 随虚时间 τ 的函数关系。假设初始的归一化量子态 $|\Phi_0\rangle$ 是哈密顿量本征态的线性组合:

$$|\Phi_0\rangle = \sum_{i=0} \mu_i |E_i\rangle. \quad (3.30)$$

不妨认为本征值 E_i 按非降序排列: $E_0 \leq E_1 \leq E_2 \leq \dots$ 。则演化至虚时刻 τ 的归一化量子态为

$$|\Phi_\tau\rangle = \frac{\sum_{i=0} \mu_i e^{-E_i \tau} |E_i\rangle}{\sqrt{\sum_{i=0} |\mu_i|^2 e^{-2E_i \tau}}}. \quad (3.31)$$

虚时刻 τ 时的能量平均值为

$$\begin{aligned}\langle \Phi_\tau | H | \Phi_\tau \rangle &= \frac{\sum_{i=0} E_i |\mu_i e^{-E_i \tau}|^2}{\sum_{i=0} |\mu_i|^2 e^{-2E_i \tau}} = \frac{\sum_{i=0} s_i E_i e^{-2E_i \tau}}{\sum_{i=0} s_i e^{-2E_i \tau}} \\ &= \frac{s_0 E_0 + \sum_{i=1} s_i E_i e^{-2\Omega_i \tau}}{s_0 + \sum_{i=1} s_i e^{-2\Omega_i \tau}} = E_0 + \frac{\sum_{i=1} s_i \Omega_i e^{-2\Omega_i \tau}}{s_0 + \sum_{i=1} s_i e^{-2\Omega_i \tau}} \\ &\leq E_0 + \frac{(1 - s_0) \Omega_{\max} e^{-2\Omega_1 \tau}}{s_0},\end{aligned}\quad (3.32)$$

其中 $s_i = |\mu_i|^2$, $\Omega_i = E_i - E_0$, Ω_{\max} 是所有 Ω_i 的最大值; 特别而言, E_0 是精确基态能量, $s_0 = |\langle E_0 | \Phi_0 \rangle|^2$ 是初始态 $|\Phi_0\rangle$ 与精确基态 $|E_0\rangle$ 之间的保真度。

将 $\langle \Phi_\tau | H | \Phi_\tau \rangle$ 的表达式代入3.29式, 我们可以得到总成功概率的近似下界:

$$P_{\text{final}} \geq P_{\text{ALB}} = \exp \left[-2\beta\eta - \frac{(1 - s_0)\Omega_{\max}}{s_0\Omega_1} (1 - e^{-2\beta\Omega_1}) \right]. \quad (3.33)$$

其中 $\eta = E_0 + \sum_k |c_k|$, 由哈密顿量的能谱及其各个泡利项的系数所决定。

虚时演化最重要的应用之一是求解量子系统的基态和基态能量, 这一问题在量子计算中受到广泛关注^[81,144,154,167,181]。在求解基态这类问题中, 我们有时并不关心成功概率与虚时间 β 的关系, 而只关心成功概率与所求基态误差的关系。求解基态的误差定义为 $\epsilon = 1 - \mathcal{F}$, 其中 $\mathcal{F} = |\langle E_0 | \Phi \rangle|^2$ 是与精确基态的保真度。从3.31式中可得

$$\begin{aligned}\mathcal{F}(\beta) &= |\langle E_0 | \Phi_\beta \rangle|^2 = \left| \frac{\sum_{i=0} \langle E_0 | \mu_i e^{-\beta E_i} | E_i \rangle}{\sqrt{\sum_{i=0} |\mu_i|^2 e^{-2\beta E_i}}} \right|^2 \\ &= \frac{|\mu_0|^2 e^{-2\beta E_0}}{\sum_{i=0} |\mu_i|^2 e^{-2\beta E_i}} = \frac{s_0}{s_0 + \sum_{i=1} s_i e^{-2\beta\Omega_i}} \\ &\geq \frac{s_0}{s_0 + (1 - s_0)e^{-2\beta\Omega_1}}.\end{aligned}\quad (3.34)$$

这里用到了 $1 - s_0 = \sum_{i=1} s_i$, 且对于所有 i 都有 $\Omega_i \geq \Omega_1$, 由此得到

$$\beta\Omega_1 \leq \frac{1}{2} \ln \left(\frac{1 - s_0}{s_0} \frac{1 - \epsilon}{\epsilon} \right). \quad (3.35)$$

这里, 我们之所以关注 β 与 Ω_1 的乘积, 是因为我们可以选择哈密顿量的标度来改变 Ω_1 的值, 且 β 同样依赖于哈密顿量的标度。而无论哈密顿量的标度如何改变, $\beta\Omega_1$ 作为一个无量纲量, 并不随哈密顿量的标度而变化, 因此才是值得关注的量。上式表明, 当求解误差不超过 ϵ 时, 所需要的虚时演化时长 $\beta = \mathcal{O}(\Omega_1^{-1} \log \epsilon^{-1})$ 与 ϵ 的倒数呈对数关系, 且与哈密顿量能级差呈反比。这里我们并没有考虑 Trotter 分解所带来的误差, 若考虑这一部分, 则保真度还要降低 $\mathcal{O}(\Delta\tau)$ 的量级, 这会导致量子态不能以任意精度接近精确基态, 因此这一误差不能通过增加 β 而减小, 随

着 β 增大，量子态会收敛到 $|E_0\rangle + |\delta\rangle$ ，其中 $||\delta\rangle| \sim \mathcal{O}(\Delta\tau)$ 。然而，我们可以通过利用更高阶的 Trotter 分解将这部分误差降低至 $\mathcal{O}(\Delta\tau^2)$ 量级^[227]：

$$e^{-\beta H} = [(e^{-c_1 h_1 \Delta\tau/2} \dots e^{-c_m h_m \Delta\tau/2}) (e^{-c_m h_m \Delta\tau/2} \dots e^{-c_1 h_1 \Delta\tau/2})]^{\beta/\Delta\tau} + \mathcal{O}(\Delta\tau^2). \quad (3.36)$$

结合3.26、3.33和3.35式，我们可以得到求解基态误差不超过 ϵ 时，总成功概率的严格下界 (RLB) 和近似下界 (ALB)：

$$\begin{aligned} P_{\text{RLB}} &\geq \left(\frac{s_0}{1-s_0} \frac{\epsilon}{1-\epsilon} \right)^{\kappa_0} = \mathcal{O}(\epsilon^{\kappa_0}), \\ P_{\text{ALB}} &\geq \left(\frac{s_0}{1-s_0} \frac{\epsilon}{1-\epsilon} \right)^{\kappa_1} \exp \left[-\frac{\Omega_{\max}}{s_0 \Omega_1} \left(1 - \frac{s_0}{1-\epsilon} \right) \right] = \mathcal{O}(\epsilon^{\kappa_1}), \end{aligned} \quad (3.37)$$

其中 $\kappa_0 = 2 \sum_k |c_k| / \Omega_1$ 和 $\kappa_1 = (E_0 + \sum_k |c_k|) / \Omega_1$ 都由哈密顿量的能谱及其各泡利项的系数决定，但与哈密顿量的标度无关。

上述的严格下界和近似下界在采用高阶 Trotter 分解时都不会改变，因为 Trotter 分解公式保证哈密顿量中每一项的累积演化时长不变。

3.3 概率性虚时演化算法在求解基态问题中的应用

求解量子体系的基态是虚时演化最重要的应用之一。为了进一步探究 PITE 算法的性能，我们选取了三种不同的量子系统：氢分子 (H_2)、氢化锂分子 (LiH) 和一维横纵场量子伊辛链，并将 PITE 算法应用于这三种体系的基态求解问题中。其中，氢分子的计算是在 Quafu 量子云平台的 10 比特超导量子芯片和 IonQ 云平台的 11 比特囚禁离子阱量子芯片中运行的，而氢化锂分子与量子伊辛链的计算则是在量子模拟器中进行数值模拟，同时在数值模拟中我们也探究了量子噪声对算法的影响以及算法的成功概率。

3.3.1 量子计算云平台中的氢分子基态求解演示实验

为了能在量子计算机中计算氢分子或氢化锂分子的基态，首先我们要将这两种分子哈密顿量编码到量子比特中。这里我们选取分子轨道的 STO-3G 基^[228]将分子的原始哈密顿量写为二次量子化形式，再利用 Jordan-Wigner 变换将哈密顿量映射成泡利算符，得到的哈密顿量形如 $H(R) = \sum_k c_k(R) \sigma_{\alpha_1}^1 \cdots \sigma_{\alpha_n}^n$ ，其作用在 n 个量子比特上，且各项系数 c_k 随着分子中两个原子的距离 R (或称键长) 而变化。在本工作中，氢分子和氢化锂分子的哈密顿量被分别映射到 4 个和 6 个量子比特上，而氢分子哈密顿量又被进一步压缩编码并映射到 2 个量子比特上^[229]。最终得到的氢分子哈密顿量为

$$H_{H_2} = c_0(R) + c_1(R) \sigma_z^1 + c_1(R) \sigma_z^2 + c_2(R) \sigma_z^1 \sigma_z^2 + c_3(R) \sigma_x^1 \sigma_x^2, \quad (3.38)$$

其中包括常数项在内共有 5 个泡利项，各项系数 c_k 在不同的键长 R 下的取值见表3.1。而氢化锂分子的最终得到的哈密顿量中，包括常数项在内共有 62 个泡利项，各个泡利项以及最稳定键长 $R = 1.5\text{\AA}$ 下的各项系数在表3.2中给出。

表 3.1 不同键长 R 下，氢分子哈密顿量中各泡利项系数，数据来自文献^[229]

$R(\text{\AA})$	c_0	c_1	c_2	c_3
0.35	7.01273×10^{-1}	-7.47416×10^{-1}	1.31036×10^{-2}	1.62573×10^{-1}
0.45	2.67547×10^{-1}	-6.33890×10^{-1}	1.27192×10^{-2}	1.66621×10^{-1}
0.55	-1.83734×10^{-2}	-5.36489×10^{-1}	1.23003×10^{-2}	1.71244×10^{-1}
0.65	-2.13932×10^{-1}	-4.55433×10^{-1}	1.18019×10^{-2}	1.76318×10^{-1}
0.75	-3.49833×10^{-1}	-3.88748×10^{-1}	1.11772×10^{-2}	1.81771×10^{-1}
0.85	-4.45424×10^{-1}	-3.33747×10^{-1}	1.04061×10^{-2}	1.87562×10^{-1}
1.05	-5.62600×10^{-1}	-2.48783×10^{-1}	8.50998×10^{-3}	1.99984×10^{-1}
1.25	-6.23223×10^{-1}	-1.86173×10^{-1}	6.45563×10^{-3}	2.13102×10^{-1}
1.45	-6.52661×10^{-1}	-1.38977×10^{-1}	4.59760×10^{-3}	2.26294×10^{-1}

在量子云平台的演示实验中，我们共使用 3 个量子比特，其中 2 个比特作为工作比特来表示氢分子，1 个比特作为辅助比特。我们选取氢分子的 Hartree-Fock (HF) 态作为初始态，在量子比特中 HF 态表示为 $|\Phi_{\text{HF}}\rangle = |00\rangle$ 。下面介绍该实验中所运行的量子线路及相关细节。

两比特的氢分子哈密顿量（3.38式）包含 4 个非常数的泡利项，它们在 PITE 算法中分别对应如下四个非幺正的虚时演化算符：

$$\begin{aligned}\tilde{T}_1 &= \exp(-c_1 \Delta\tau \sigma_z^1), \\ \tilde{T}_2 &= \exp(-c_1 \Delta\tau \sigma_z^2), \\ \tilde{T}_3 &= \exp(-c_2 \Delta\tau \sigma_z^1 \sigma_z^2), \\ \tilde{T}_4 &= \exp(-c_3 \Delta\tau \sigma_x^1 \sigma_x^2).\end{aligned}\tag{3.39}$$

在 PITE 算法中，我们需要在工作比特上依次作用 $\tilde{T}_1, \tilde{T}_2, \tilde{T}_3, \tilde{T}_4$ 四个算符，并不断循环。在每一个 \tilde{T}_k 的作用后，按照 PITE 算法的流程，理论上应该对测量后选择得到的态继续进行下一步操作，然而在实验中，受限于量子硬件的性能，测量后选择会导致量子态产生严重的退相干，从而丢失其中的信息。因此我们采用等效替代方案，在每个 \tilde{T}_k 的作用后对工作比特的量子态进行态层析 (tomography)，态层析是指通过对量子态的多个可观测量进行重复测量，从而获得量子态矢量的各个分量。在每一步迭代开始时，先根据上一步的态层析结果重新制备上一步所得到的末态。每当 \tilde{T}_1 到 \tilde{T}_4 依次作用的一个循环结束后，就代表着一个 $\Delta\tau$ 时间间隔的演化结束，我们需要测量出量子态的能量期望值，以显示其收敛至基态能量的过程。

表3.2 最稳定键长 $R = 1.5\text{\AA}$ 下, 氢化锂分子哈密顿量中各泡利项系数, 以及在 GPITE 算法中泡利项的分组方式

k	c_k	\mathbf{h}_k	$\mathbf{H}_{k'}$	k	c_k	\mathbf{h}_k	$\mathbf{H}_{k'}$
1	-7.35094	$\mathbf{1}$	-	33	-1.49854×10^{-3}	$\sigma_x^2 \sigma_z^3 \sigma_x^4 \sigma_z^5$	\mathbf{H}_{13}
2	-1.58950×10^{-1}	σ_z^1	\mathbf{H}_1	34	-1.49854×10^{-2}	$\sigma_y^2 \sigma_z^3 \sigma_y^4 \sigma_z^5$	\mathbf{H}_{13}
3	-1.58950×10^{-1}	σ_z^2	\mathbf{H}_1	35	1.13678×10^{-2}	$\sigma_x^1 \sigma_z^2 \sigma_x^3 \sigma_z^6$	\mathbf{H}_{14}
4	7.82811×10^{-2}	$\sigma_z^1 \sigma_z^2$	\mathbf{H}_1	36	1.13678×10^{-2}	$\sigma_y^1 \sigma_z^2 \sigma_y^3 \sigma_z^6$	\mathbf{H}_{14}
5	-1.45795×10^{-1}	σ_z^3	\mathbf{H}_2	37	-1.17598×10^{-3}	$\sigma_z^1 \sigma_x^2 \sigma_z^3 \sigma_x^4$	\mathbf{H}_{15}
6	-1.45795×10^{-1}	σ_z^4	\mathbf{H}_2	38	-1.17598×10^{-3}	$\sigma_z^1 \sigma_y^2 \sigma_z^3 \sigma_y^4$	\mathbf{H}_{15}
7	8.51132×10^{-2}	$\sigma_z^3 \sigma_z^4$	\mathbf{H}_2	39	3.56300×10^{-3}	$\sigma_x^1 \sigma_z^2 \sigma_x^3 \sigma_z^5$	\mathbf{H}_{16}
8	2.96723×10^{-2}	σ_z^5	\mathbf{H}_3	40	3.56300×10^{-3}	$\sigma_y^1 \sigma_z^2 \sigma_y^3 \sigma_z^5$	\mathbf{H}_{16}
9	2.96723×10^{-2}	σ_z^6	\mathbf{H}_3	41	3.56300×10^{-3}	$\sigma_x^2 \sigma_z^3 \sigma_x^4 \sigma_z^6$	\mathbf{H}_{17}
10	1.24302×10^{-1}	$\sigma_z^5 \sigma_z^6$	\mathbf{H}_3	42	3.56300×10^{-3}	$\sigma_y^2 \sigma_z^3 \sigma_y^4 \sigma_z^6$	\mathbf{H}_{17}
11	5.36162×10^{-2}	$\sigma_z^1 \sigma_z^3$	\mathbf{H}_4	43	-1.03458×10^{-2}	$\sigma_x^1 \sigma_z^2 \sigma_y^3 \sigma_y^4$	\mathbf{H}_{18}
12	6.03396×10^{-2}	$\sigma_z^3 \sigma_z^5$	\mathbf{H}_4	44	-1.03458×10^{-2}	$\sigma_y^1 \sigma_y^2 \sigma_x^3 \sigma_x^4$	\mathbf{H}_{18}
13	6.28713×10^{-2}	$\sigma_z^1 \sigma_z^5$	\mathbf{H}_4	45	1.03458×10^{-2}	$\sigma_x^1 \sigma_y^2 \sigma_y^3 \sigma_x^4$	\mathbf{H}_{18}
14	5.64568×10^{-2}	$\sigma_z^1 \sigma_z^4$	\mathbf{H}_5	46	1.03458×10^{-2}	$\sigma_y^1 \sigma_x^2 \sigma_x^3 \sigma_y^4$	\mathbf{H}_{18}
15	6.03396×10^{-2}	$\sigma_z^4 \sigma_z^6$	\mathbf{H}_5	47	-2.84063×10^{-3}	$\sigma_x^3 \sigma_x^4 \sigma_y^5 \sigma_y^6$	\mathbf{H}_{19}
16	6.87743×10^{-2}	$\sigma_z^1 \sigma_z^6$	\mathbf{H}_5	48	-2.84063×10^{-3}	$\sigma_y^3 \sigma_y^4 \sigma_x^5 \sigma_x^6$	\mathbf{H}_{19}
17	5.36162×10^{-2}	$\sigma_z^2 \sigma_z^4$	\mathbf{H}_6	49	2.84063×10^{-3}	$\sigma_x^3 \sigma_y^4 \sigma_y^5 \sigma_x^6$	\mathbf{H}_{19}
18	6.87743×10^{-2}	$\sigma_z^2 \sigma_z^5$	\mathbf{H}_6	50	2.84063×10^{-3}	$\sigma_y^3 \sigma_x^4 \sigma_x^5 \sigma_y^6$	\mathbf{H}_{19}
19	7.06853×10^{-2}	$\sigma_z^4 \sigma_z^5$	\mathbf{H}_6	51	-5.90301×10^{-3}	$\sigma_x^1 \sigma_x^2 \sigma_y^5 \sigma_y^6$	\mathbf{H}_{20}
20	5.64568×10^{-2}	$\sigma_z^2 \sigma_z^3$	\mathbf{H}_7	52	-5.90301×10^{-3}	$\sigma_y^1 \sigma_y^2 \sigma_x^5 \sigma_x^6$	\mathbf{H}_{20}
21	6.28713×10^{-2}	$\sigma_z^2 \sigma_z^6$	\mathbf{H}_7	53	5.90301×10^{-3}	$\sigma_x^1 \sigma_y^2 \sigma_y^5 \sigma_x^6$	\mathbf{H}_{20}
22	7.06853×10^{-2}	$\sigma_z^3 \sigma_z^6$	\mathbf{H}_7	54	5.90301×10^{-3}	$\sigma_y^1 \sigma_x^2 \sigma_x^5 \sigma_y^6$	\mathbf{H}_{20}
23	-1.49854×10^{-3}	$\sigma_x^1 \sigma_x^3$	\mathbf{H}_8	55	-4.73898×10^{-3}	$\sigma_x^2 \sigma_x^3 \sigma_x^5 \sigma_x^6$	\mathbf{H}_{21}
24	-1.49854×10^{-3}	$\sigma_y^1 \sigma_y^3$	\mathbf{H}_8	56	-4.73898×10^{-3}	$\sigma_y^2 \sigma_y^3 \sigma_y^5 \sigma_y^6$	\mathbf{H}_{21}
25	1.13678×10^{-2}	$\sigma_x^2 \sigma_x^4$	\mathbf{H}_9	57	-4.73898×10^{-3}	$\sigma_x^2 \sigma_y^3 \sigma_y^5 \sigma_x^6$	\mathbf{H}_{21}
26	1.13678×10^{-2}	$\sigma_y^2 \sigma_y^4$	\mathbf{H}_9	58	-4.73898×10^{-3}	$\sigma_y^2 \sigma_x^3 \sigma_x^5 \sigma_y^6$	\mathbf{H}_{21}
27	1.04793×10^{-2}	$\sigma_x^1 \sigma_z^2 \sigma_x^3$	\mathbf{H}_{10}	59	-4.73898×10^{-3}	$\sigma_x^1 \sigma_z^2 \sigma_z^3 \sigma_x^4 \sigma_y^5 \sigma_y^6$	\mathbf{H}_{22}
28	1.04793×10^{-2}	$\sigma_y^1 \sigma_z^2 \sigma_y^3$	\mathbf{H}_{10}	60	-4.73898×10^{-3}	$\sigma_y^1 \sigma_z^2 \sigma_z^3 \sigma_y^4 \sigma_x^5 \sigma_x^6$	\mathbf{H}_{22}
29	1.04793×10^{-2}	$\sigma_x^2 \sigma_z^3 \sigma_x^4$	\mathbf{H}_{11}	61	4.73898×10^{-3}	$\sigma_x^1 \sigma_z^2 \sigma_z^3 \sigma_y^4 \sigma_y^5 \sigma_x^6$	\mathbf{H}_{22}
30	1.04793×10^{-2}	$\sigma_y^2 \sigma_z^3 \sigma_y^4$	\mathbf{H}_{11}	62	4.73898×10^{-3}	$\sigma_y^1 \sigma_z^2 \sigma_z^3 \sigma_x^4 \sigma_x^5 \sigma_y^6$	\mathbf{H}_{22}
31	-1.17598×10^{-3}	$\sigma_x^1 \sigma_z^2 \sigma_x^3 \sigma_z^4$	\mathbf{H}_{12}				
32	-1.17598×10^{-3}	$\sigma_y^1 \sigma_z^2 \sigma_y^3 \sigma_z^4$	\mathbf{H}_{12}				

我们利用图3.3所示的四个量子线路分别实现算符 \tilde{T}_k ($k = 1, 2, 3, 4$)，其中每个线路都包含如下四个部分。

(1) \mathbf{U} 与 \mathbf{U}^\dagger (图中蓝色部分)，对应 PITE 算法线路图3.1中的 \mathbf{U}_k 与 \mathbf{U}_k^\dagger 。

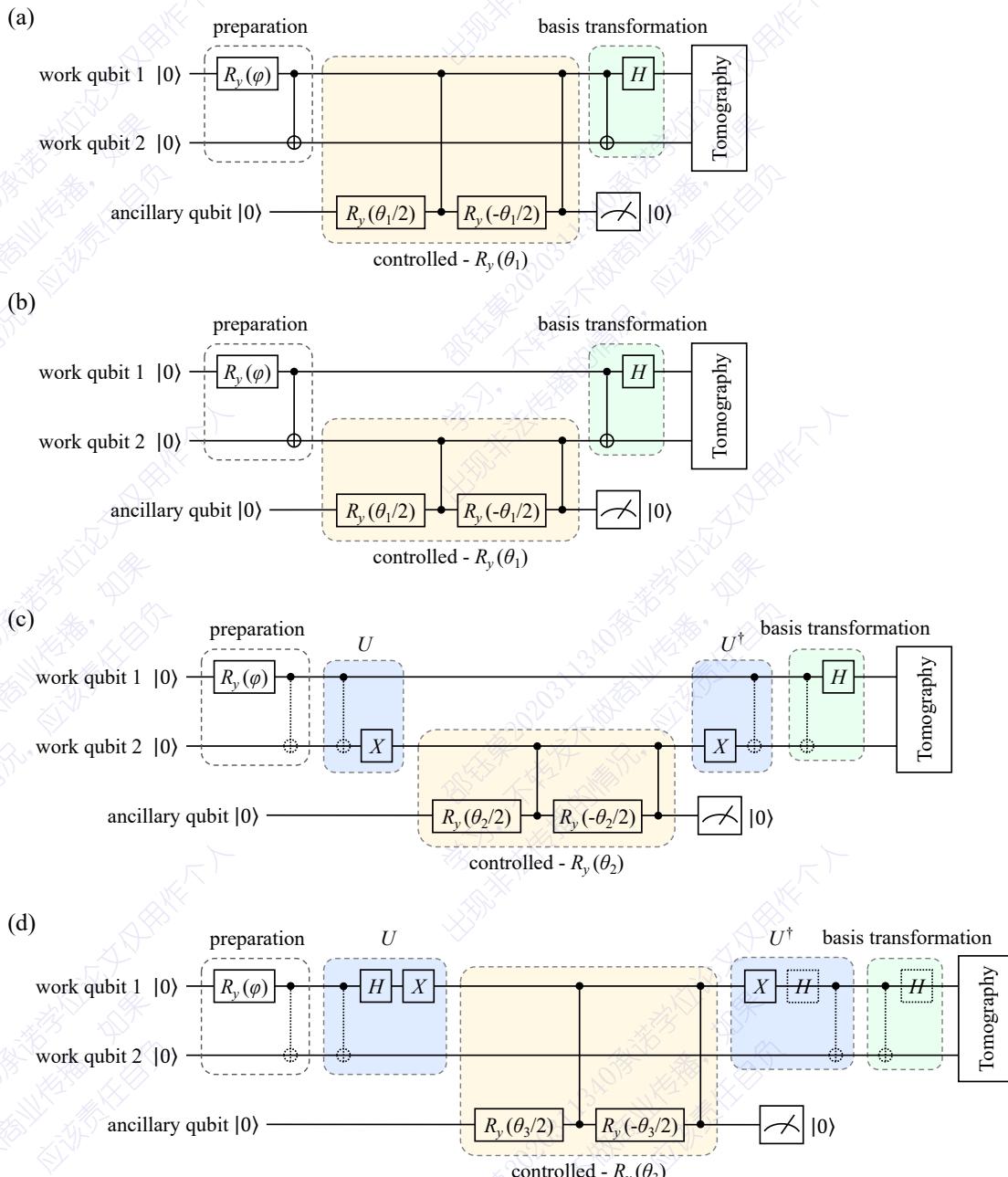
(2) 受控 \mathbf{R}_y 操作 (图中橙色部分)，对应 PITE 算法线路图3.1中的受控 \mathbf{R}_y 门。

(3) 基变换 (图中绿色部分)：理论分析表明，实验过程中的量子态可以写成 $\cos(\varphi/2)|00\rangle - \sin(\varphi/2)|11\rangle$ 的形式，其中 $0 < \varphi < \pi$ ，且 φ 大约在 $10^{-3} \sim 10^{-2}$ 量级，这一量级与量子比特的测量误差所在的量级相同。如果我们直接在量子比特的计算基上进行测量，那么测量得到 $|11\rangle$ 的概率就会严重受到测量误差的影响。因此，我们在测量量子比特之前先对其进行基变换，使量子态变为 $\frac{\cos(\varphi/2)-\sin(\varphi/2)}{\sqrt{2}}|00\rangle + \frac{\cos(\varphi/2)+\sin(\varphi/2)}{\sqrt{2}}|10\rangle$ 。这样就只需要测量第一个量子比特，且得到 0 和 1 的概率分别为 $p_0 = (\cos(\varphi/2) - \sin(\varphi/2))^2/2 = (1 - \sin \varphi)/2$, $p_1 = (1 + \sin \varphi)/2$ ，这里 p_0 和 p_1 大约为 $0.4 \sim 0.6$ ，远大于测量误差，因此测量误差产生的影响就可以忽略。

(4) 态层析与态制备 (图中白色部分)：在每一步 Trotter 迭代后，我们对已经进行了基变换的量子态进行态层析，即通过测量量子态的概率分布，获得态矢量。然后我们通过数值计算得到基变换之前的真实量子态，并将其作为下一个 Trotter 迭代的初始态。由于理论上量子态仅在 $|00\rangle$ 和 $|11\rangle$ 张成的实系数子空间中进行演化，因此制备该量子态仅需要一个单比特 Ry 门和一个 CNOT 门，其中 Ry 门的旋转角参数 φ 可以从态层析的结果中得到，具体而言，如上一段所述，角参数满足 $\sin \varphi = 2p_1 - 1 = 1 - 2p_0 = p_1 - p_0$ 。对每一个量子态的态层析都重复进行三次，每一次态层析在超导量子处理器中进行 2000 次测量，在离子阱量子处理器中进行 1000 次测量。每一次态层析后，计算角参数 φ ，取三次结果的平均参数作为下一步 Trotter 迭代的初始态制备参数。对于整个算法流程的第一个 Trotter 迭代，输入态为 $|\Phi_{HF}\rangle = |00\rangle$ ，即角参数 $\varphi = 0$ 。此外，在每一次 \tilde{T}_4 线路末尾的态层析之后，我们还要利用层析得到的真实量子态计算能量期望值 $\langle E \rangle = \langle \Phi | \mathbf{H}_{\mathrm{H}_2} | \Phi \rangle$ 。

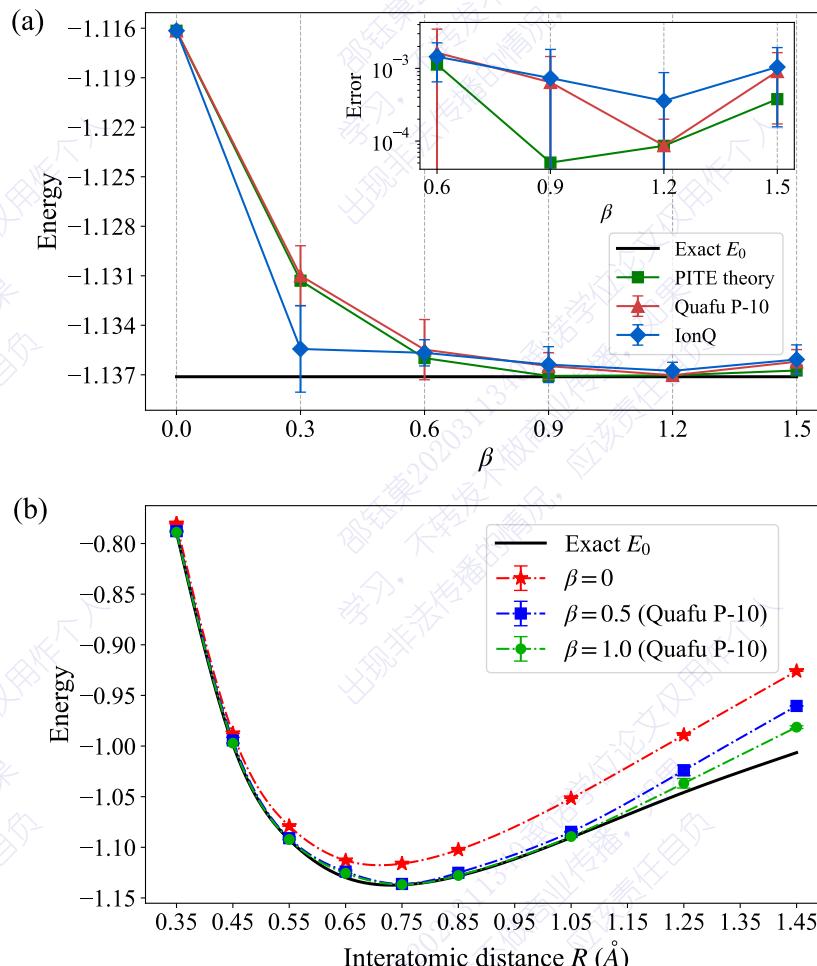
按照以上所叙述的演示实验方案，我们在两个量子计算云平台上进行利用 PITE 算法求解氢分子基态与基态能量的演示实验。

演示实验的第一部分是在固定氢分子键长 $R = 0.75\text{\AA}$ 下，根据 PITE 算法求解氢分子的基态能量，这部分实验在 Quafu 的超导量子芯片 P-10 和 IonQ 离子阱量子处理器中运行。图3.4(a) 为实验过程中能量期望值 $E = \langle \Phi | \mathbf{H}_{\mathrm{H}_2} | \Phi \rangle$ 随虚时间 β 的变化关系，其中绿色点为 PITE 算法的理论结果，红色和蓝色点分别为 Quafu 平台和 IonQ 平台的实验结果。两个实验平台给出的实验结果中，能量期望值都随着 β 增大而迅速收敛至精确基态能量，在 5 个演化步骤内就达到了 10^{-4} Hartree 能量的误差，满足化学精度的要求。



(a)-(d) 分别对应非幺正演化算符 \tilde{T}_k ($k = 1, 2, 3, 4$)。线路中带有虚线边框线的量子门会相互抵消，所以在实验中不会运行。

图 3.3 在量子计算云平台上，利用 PITE 算法计算氢分子基态与基态能量的演示实验的量子线路图



(a) 固定键长 $R = 0.75\text{\AA}$ 下，能量期望值随虚时间 β 的变化关系。(b) 不同的虚时间 β 对应的能量期望值随键长 R 的变化关系。两图中，黑色线表示精确对角化得到的精确基态能量。哈密顿量中的常数项在运行 PITE 算法时被忽略，但在计算能量平均值时被保留。

图 3.4 在 Quafu 和 IonQ 云平台上，利用 PITE 算法求解氢分子基态能量的演示实验结果

演示实验的第二部分是改变氢分子的键长 R ，在一系列不同的键长下利用 PITE 算法计算氢分子的基态能量，并绘制如图3.4(b) 所示的氢分子势能曲线，以得到氢分子最稳定的构型。这部分实验仅在 Quafu 平台上运行。对于每一个不同的 R ，初始态都选取为 HF 态 $|\Phi_{\text{HF}}\rangle$ ，然后分别测量量子态在虚时间 $\beta = 0, 0.5, 1$ 下的能量期望值，并与精确基态能量进行对比。实验结果表明，当 $\beta = 1$ 时实验结果已十分接近精确基态，在能量曲线的最低点处，我们得到了氢分子最稳定的键长，约为 0.75\AA 。

3.3.2 氢化锂分子与量子伊辛链基态求解的数值模拟

利用数值模拟，我们将 PITE 算法应用于氢化锂分子与量子伊辛链的基态能量计算中，并且研究了 PITE 算法的成功概率问题，以及量子噪声对算法的影响。

对于氢化锂分子，我们共使用了 7 个比特，其中包括 6 个用来表示氢化锂分子的工作比特和 1 个辅助比特。与氢分子类似，我们希望选取氢化锂分子的 Hartree-Fock 态作为算法的初始态，其在量子比特中表示为 $|\Phi_{HF}\rangle = |110000\rangle$ ，然而我们注意到 $|\Phi_{HF}\rangle$ 已经十分接近精确基态，因此在算法中收敛的步骤过少。为了更清楚地展示算法过程中量子态的收敛，我们实际上使用了 HF 态与某个激发态的叠加态作为算法的初始量子态： $|\Phi_0\rangle = \sqrt{0.96}|\Phi_{HF}\rangle + 0.2|000011\rangle$ 。

数值模拟的第一部分是固定氢化锂分子的键长为 $R = 2.0\text{\AA}$ ，计算量子态随虚时间 β 的演化过程。在图3.5(a)(b) 中，我们分别展示了氢化锂分子的能量期望值 $E(\beta)$ 和保真度 $\mathcal{F}(\beta)$ 随虚时间 β 的收敛过程，其中保真度 $\mathcal{F}(\beta) = |\langle E_0|\Phi_\beta\rangle|^2$ 定义为虚时演化量子态 $|\Phi_\beta\rangle$ 与精确基态 $|E_0\rangle$ 之间的内积模平方。除了运行理想的 PITE 算法之外，我们对量子线路中的噪声也进行了模拟。我们选取的噪声模型为量子信道，并在算法中每次测量辅助比特之前对量子比特施加噪声的影响。量子信道对量子态密度矩阵的影响为：

$$\mathcal{E}(\rho) = \sum_{v=1}^3 \hat{E}_v \rho \hat{E}_v^\dagger, \quad (3.40)$$

其中 \hat{E}_v 为 Kraus 算符：

$$\hat{E}_1 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1 - \epsilon_r - \epsilon_d} \end{bmatrix}, \quad \hat{E}_2 = \begin{bmatrix} 0 & \sqrt{\epsilon_d} \\ 0 & 0 \end{bmatrix}, \quad \hat{E}_3 = \begin{bmatrix} 0 & 0 \\ 0 & \sqrt{\epsilon_r} \end{bmatrix}, \quad (3.41)$$

这里 ϵ_r 和 ϵ_d 分别是量子比特的弛豫参数和退相位参数。针对量子噪声的数值模拟结果也在图3.5(a)(b) 中展示，其表明在噪声影响下，算法仍能使能量期望值以 10^{-3} Hartree 能量的误差收敛到精确基态能量。

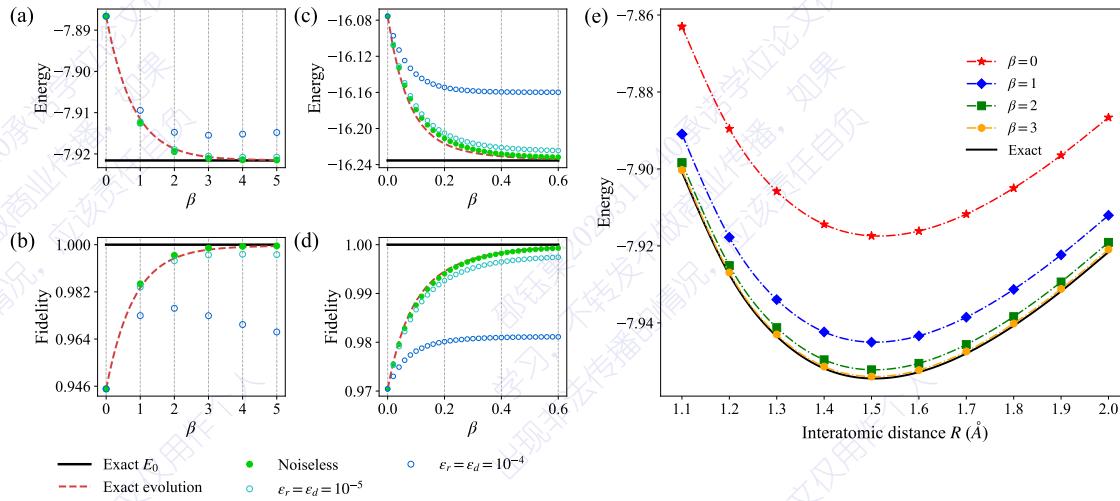
数值模拟的第二部分是在不同的氢化锂分子的键长 R 下计算基态能量，并绘制势能曲线。图3.5(e) 为各个不同的 β 下算法给出的势能曲线与精确的基态能量曲线的对比。其中能量曲线的最低点对应的就是氢化锂分子最稳定的键长，约为 1.5\AA 。

数值模拟的第三部分是对一维量子伊辛链的基态能量计算。这里我们选用了 10 格点周期性边界条件的量子伊辛链，哈密顿量为

$$\mathbf{H}_{\text{Ising}} = -J \sum_{j=1}^n \left(\sigma_z^j \sigma_z^{j+1} + g \sigma_x^j + h \sigma_z^j \right), \quad \sigma_z^{n+1} = \sigma_z^1, \quad (3.42)$$

其中 g 和 h 分别是施加在伊辛链上的横场和纵场的强度。在数值模拟中，哈密顿量中的各参数选为 $n = 10, J = 1, g = 1.2, h = 0.3$ 。算法的初始量子态选为 $|\Phi_0\rangle = \left(\cos \frac{\phi_0}{2} |0\rangle + \sin \frac{\phi_0}{2} |1\rangle \right)^{\otimes n}$ ，其中参数 ϕ_0 的选取使初始的能量期望值 $E(\beta = 0) = \langle \Phi_0 | \mathbf{H}_{\text{Ising}} | \Phi_0 \rangle$ 最小。数值模拟的结果如图3.5(c)(d) 所示，其中展示了 PITE

算法所给出的能量期望值 E 和保真度 F 随虚时间 β 的变化，此外量子噪声对算法的影响也在图中展示。结果表明，在无噪声的条件下，量子态的能量期望值可以以 10^{-3} Hartree 能量的误差收敛到精确基态能量，而在有噪声的情况下收敛误差为 10^{-2} Hartree。

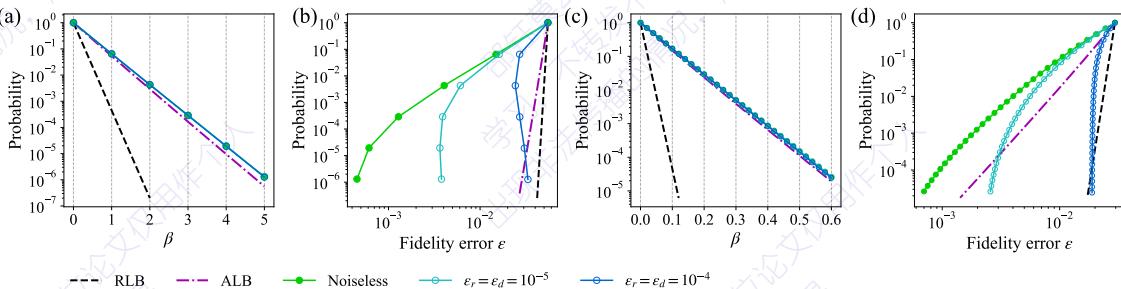


(a) 氢化锂分子的能量期望值随 β 变化。(b) 氢化锂分子的量子态保真度随 β 变化。(a) 与 (b) 固定了分子键长为 $R = 2.0\text{\AA}$ 。(c) 量子伊辛链的能量期望值随 β 变化。(d) 量子伊辛链的量子态保真度随 β 变化。(e) 不同虚时间 β 下，氢化锂分子的能量期望值随键长 R 变化。针对氢化锂分子的计算中，哈密顿量中的常数项在运行 PITE 算法时被忽略，但在计算能量平均值时被保留。针对量子伊辛链的计算中，参数为 $n = 10, J = 1, g = 1.2, h = 0.3$ 。(a)(c)(e) 图中的黑色线为精确对角化得到的精确基态能量。(a)(b)(c)(d) 图中的红色虚线为精确的虚时演化曲线。

图 3.5 在数值模拟中，利用 PITE 算法计算氢化锂分子与量子伊辛链的基态与基态能量（采用量子信道噪声模型）

在利用 PITE 求解氢化锂分子与量子伊辛链的基态能量的同时，我们也计算了 PITE 算法的成功概率，结果如图3.6所示。在针对这两种量子体系的计算中，累积成功概率随虚时间 β 的变化分别在图3.6(a) 和 (c) 中展示，都显示出了随着 β 增长而指数降低的趋势。可以看出，相比于公式3.26给出的成功概率的严格下界 (RLB，图中黑色实线)，公式3.33给出的近似下界 (ALB，图中紫色点划线) 更接近真实的成功概率曲线，这一情况在两种量子体系中都是如此。此外，图中的结果还表明，量子噪声对成功概率与虚时间 β 的关系几乎没有影响。而另一方面，图3.6(b) 和 (d) 则展示了累积成功概率与基态求解误差 ϵ 的关系，其中初始状态对应图中右上角的点，并在算法运行的过程中沿着曲线朝左下方演化。这里体现了量子噪声对算法的影响，即当存在量子噪声时，基态求解误差为 ϵ 时的算法成功概率可能会低于近似下界，这是由于相比于无噪声情况，噪声使得求解基态的误差变大，因此达到同一误差所需要的演化步骤更长，成功概率也会更低。

此外，在数值模拟中，我们还研究了另一种量子噪声的模型对算法的影响。这



(a) 求解氢化锂分子基态的成功概率随虚时间 β 的变化。(b) 求解氢化锂分子基态的成功概率与求解误差 ϵ 的关系。(c) 求解量子伊辛链基态的成功概率随虚时间 β 的变化。(d) 求解量子伊辛链基态的成功概率与求解误差 ϵ 的关系。在 (b) 和 (d) 中, 初始态位于曲线的右上角, 并朝左下方进行演化。在氢化锂分子的计算中, 严格下界 (RLB) 与近似下界 (ALB) 的计算都忽略了哈密顿量中的常数项。

图 3.6 在数值模拟中, 利用 PITE 算法求解氢化锂分子与量子伊辛链的基态的累积成功概率 (采用量子信道噪声模型)

一种噪声模型将所有的单比特门都当作 Bloch 球中绕某个轴的旋转操作, 噪声则体现在旋转角度的随机涨落, 且符合标准差为 e_1 的正态分布。对于两比特门, 将其拆分成单比特门和 CZ 门的组合, 并将受到噪声影响的 CZ 门当作受控相位门, 矩阵形式为 $\text{diag}(1, 1, 1, e^{i(\pi+\delta)})$, 其中 δ 符合标准差为 e_2 的正态分布。在该噪声模型下, 我们重新数值模拟了利用 PITE 算法计算氢化锂分子与量子伊辛链的基态能量, 模拟结果中能量与保真度随虚时间 β 的变化曲线以及算法的累积成功概率分别如图3.7和3.8所示。

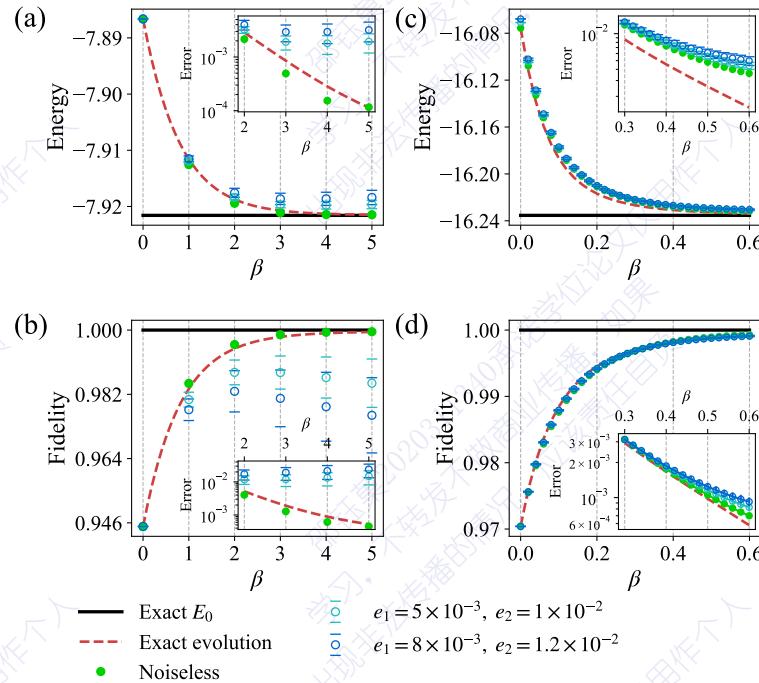
3.4 改进的概率性虚时演化算法及其应用

为了进一步提高 PITE 算法的成功概率, 降低算法的复杂度, 我们对 PITE 算法进行了改进, 改进后的 PITE 算法不仅在成功概率上有所提高, 而且适用于更广泛的哈密顿量形式, 不再仅局限于泡利形式的哈密顿量。我们将改进后的 PITE 算法称为 generalized PITE (GPITE) 算法。

GPITE 算法中, 哈密顿量同样可以写成若干项之和: $\mathbf{H} = \sum_{k=1}^m \mathbf{H}_k$, 这里 \mathbf{H}_k 不要求必须是泡利项。利用 Trotter 分解, 将虚时演化算符写为:

$$e^{-\beta \mathbf{H}} = (e^{-\mathbf{H}_1 \Delta \tau} \cdots e^{-\mathbf{H}_m \Delta \tau})^{\beta / \Delta \tau} + \mathcal{O}(\Delta \tau). \quad (3.43)$$

我们假设, 哈密顿量中每一项 \mathbf{H}_k 的特征值都可以通过经典计算高效得到, 且对应的本征态也可以在量子计算机上高效制备。这一假设大多情况下都可以满足, 例如当我们研究化学分子和物理系统时, 哈密顿量中的项通常是局域的, 因此维数较低。将 \mathbf{H}_k 的从小到大的第 i 个特征值记为 $\lambda_{k,i}$, 对应的本征态记为 $|\lambda_{k,i}\rangle$ 。定义 $\omega_{k,i} = \lambda_{k,i} - \lambda_{k,0}$, 其中 $\lambda_{k,0}$ 是 \mathbf{H}_k 的最小的本征值。与 PITE 算法的过程类似,



(a) 氢化锂分子的能量期望值随 β 变化。 (b) 氢化锂分子的量子态保真度随 β 变化。 (a) 与 (b) 固定了分子键长为 $R = 2.0\text{\AA}$ 。 (c) 量子伊辛链的能量期望值随 β 变化。 (d) 量子伊辛链的量子态保真度随 β 变化。 (e) 不同虚时间 β 下, 氢化锂分子的能量期望值随键长 R 变化。针对氢化锂分子的计算中, 哈密顿量中的常数项在运行 PITE 算法时被忽略, 但在计算能量平均值时被保留。针对量子伊辛链的计算中, 参数为 $n = 10, J = 1, g = 1.2, h = 0.3$ 。 (a)(c)(e) 图中的黑色线为精确对角化得到的精确基态能量。 (a)(b)(c)(d) 图中的红色虚线为精确的虚时演化曲线。

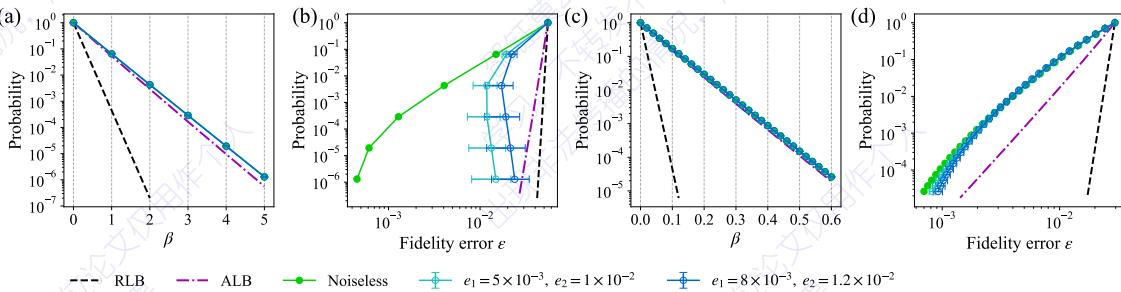
图 3.7 在数值模拟中, 利用 PITE 算法计算氢化锂分子与量子伊辛链的基态与基态能量(采用旋转角涨落噪声模型)

在量子线路中实现单步 Trotter 迭代的非幺正算符 $e^{-\mathbf{H}_k \Delta \tau}$ 的步骤如下:

1. 对工作比特施加 \mathbf{U}_k 操作, \mathbf{U}_k 可以将 \mathbf{H}_k 的本征态 $|\lambda_{k,i}\rangle$ 变换为计算基 $|x_{k,i}\rangle$ 。
2. 添加一个辅助比特, 其状态为 $|0\rangle$ 。
3. 对辅助比特施加由工作比特控制的旋转操作 $\sum_i |x_{k,i}\rangle \langle x_{k,i}| \otimes \mathbf{R}_y(\theta_{k,i})$, 其中旋转角为 $\theta_{k,i} = 2 \arccos(e^{-\omega_{k,i} \Delta \tau})$ 。
4. 对辅助比特进行测量; 如果得到 0, 则继续下一步操作; 否则算法重新开始。
5. 对工作比特施加 \mathbf{U}_k^\dagger 操作。结束。

这里我们强调, \mathbf{U}_k 和 \mathbf{U}_k^\dagger 在量子线路中可以高效构造, 因为我们假设了 \mathbf{H}_k 的本征态可以高效制备, 而制备本征态的过程实际上就是对计算基施加 \mathbf{U}_k^\dagger 操作。这一算法流程是对适用于泡利形式哈密顿量的原始 PITE 流程进行了推广, 泡利形式哈密顿量也属于 GPITE 适用范围中的某一种情况, 因此 GPITE 也可以称为推广的 PITE 算法。

现在我们关心 GPITE 算法的复杂度, 希望知道其如何优于原始的 PITE 算法。



(a) 求解氢化锂分子基态的成功概率随虚时间 β 的变化。(b) 求解氢化锂分子基态的成功概率与求解误差 ϵ 的关系。(c) 求解量子伊辛链基态的成功概率随虚时间 β 的变化。(d) 求解量子伊辛链基态的成功概率与求解误差 ϵ 的关系。在 (b) 和 (d) 中, 初始态位于曲线的右上角, 并朝左下方进行演化。在氢化锂分子的计算中, 严格下界 (RLB) 与近似下界 (ALB) 的计算都忽略了哈密顿量中的常数项。

图 3.8 在数值模拟中, 利用 PITE 算法求解氢化锂分子与量子伊辛链的基态的累积成功概率 (采用旋转角涨落噪声模型)

GPITE 算法复杂度的推导过程与 3.2 节中对原始 PITE 算法复杂度的推导类似, 这里我们只进行成功概率的近似下界 (ALB) 的推导。首先, 按照如前所述的步骤, 在量子线路中实现单步 Trotter 迭代 $e^{-H_k \Delta\tau}$ 的成功概率为

$$P_k(\Delta\tau) = |a_0|^2 + \sum_{i=1}^m |a_i|^2 e^{-2\omega_{k,i}\Delta\tau}, \quad (3.44)$$

其中 $a_i = \langle \lambda_{k,i} | \Phi \rangle$, $i \geq 0$, 且 $\omega_{k,i} = \lambda_{k,i} - \lambda_{k,0}$, $\lambda_{k,0}$ 是 H_k 最小的特征值。将上式作如下近似:

$$\begin{aligned} P_k(\Delta\tau) &\approx |a_0|^2 + \sum_{i=1}^m |a_i|^2 (1 - 2\omega_{k,i}\Delta\tau) \\ &= 1 - 2\Delta\tau \sum_{i=1}^m |a_i|^2 \omega_{k,i} \\ &= 1 - 2\Delta\tau (\langle \Phi_\tau | H_k | \Phi_\tau \rangle - \lambda_{k,0}) \\ &\approx \exp [-2\Delta\tau (\langle \Phi_\tau | H_k | \Phi_\tau \rangle - \lambda_{k,0})]. \end{aligned} \quad (3.45)$$

则在演化时长 $\Delta\tau$ 内, 累积成功概率近似为

$$P(\Delta\tau) \approx \prod_{k=1}^m P_k(\Delta\tau) = \exp \left[-2\Delta\tau \left(\langle \Phi_\tau | H | \Phi_\tau \rangle - \sum_k \lambda_{k,0} \right) \right]. \quad (3.46)$$

这里我们同样近似认为量子态 $|\Phi\rangle$ 在演化时长 $\Delta\tau$ 保持不变。由此可得, 总演化时长 β 所对应的总累积成功概率为

$$\begin{aligned} P_{\text{final}} &\approx \exp \left[-2 \int_0^\beta d\tau \left(\langle \Phi_\tau | H | \Phi_\tau \rangle - \sum_k \lambda_{k,0} \right) \right] \\ &\geq \exp \left[-2\beta\eta' - \frac{(1-s_0)\Omega_{\max}}{s_0\Omega_1} (1 - e^{-2\beta\Omega_1}) \right], \end{aligned} \quad (3.47)$$

其中 $\eta' = E_0 - \sum_k \lambda_{k,0}$, 由哈密顿量的能谱及其各个项的本征值决定 (式子中的积分利用3.32式来计算)。这一结果即为 GPITE 算法成功概率的近似下界。特别地, 当哈密顿量的每一项都是泡利项时, 即 $\mathbf{H}_k = c_k \mathbf{h}_k$, 则有 $\lambda_{k,0} = -|c_k|$, 这就使得3.47式转变为3.33, 即原始 PITE 的成功概率近似下界。

我们同样注意到, 当哈密顿量 \mathbf{H} 给定时, 成功概率下界表达式3.47中唯一可以变化的部分是 $\sum_k \lambda_{k,0}$, 即哈密顿量中各个项的最低本征值之和。这为我们提供了一个思路: 通过改变哈密顿量中各个项 \mathbf{H}_k 的划分, 增大 $\sum_k \lambda_{k,0}$ 的值, 我们就可以提高算法的成功概率。

除此之外, GPITE 的线路深度仍然与哈密顿量的项数呈线性关系, 所以我们也可以通过将哈密顿量划分成更少的项来降低线路深度。因此, 相比于原始的 PITE 算法, 改进后的 GPITE 算法既可以提高总成功概率, 也可以降低线路深度和门复杂度, 但是由于将 \mathbf{H}_k 对角化而产生对经典计算资源的需求增加。在降低量子计算资源与增加经典计算资源之间存在一个权衡, 而权衡的关键点就在于, 依据我们所求解的系统的特性, 寻找一个合适的方法, 将哈密顿量拆分成为若干合适的项之和。

进一步, 当我们利用 GPITE 算法求解量子体系的基态时, 我们实际关心的是算法成功概率与求解误差 $\epsilon = 1 - F$ 之间的关系, 而不是与虚时间 β 的关系。因此, 我们将3.35式与3.47式联立, 并得到累积成功概率与基态求解误差之间的关系:

$$P_{\text{final}} \geq \left(\frac{s_0}{1-s_0} \frac{\epsilon}{1-\epsilon} \right)^\kappa \exp \left[-\frac{\Omega_{\max}}{s_0 \Omega_1} \left(1 - \frac{s_0}{1-\epsilon} \right) \right] = \mathcal{O}(\epsilon^\kappa), \quad (3.48)$$

其中 $\kappa = (E_0 - \sum_k \lambda_{k,0}) / \Omega_1$, 该参数并不依赖于哈密顿量的标度。这表明如果我们希望以 ϵ 的误差得到基态, 成功概率是误差的多项式量级: $P = \mathcal{O}(\epsilon^\kappa)$, 我们需要将算法平均重复运行 $1/P$ 次才能够得到所希望的输出结果。

我们将 GPITE 算法应用于求解基态的问题中, 选取的体系仍然是氢化锂分子 (固定键长 $R = 2.0\text{\AA}$) 和一维量子伊辛链。与前面的计算方法不同, 我们不再将泡利项 $c_k \mathbf{h}_k$ 作为哈密顿量的各项 \mathbf{H}_k , 而是将这些泡利项重新排列, 并按照它们的性质分成若干组, 每组包含 2 到 4 个泡利项, 然后将每组中的泡利项合并成一个新的项。新项的个数比原始的泡利项的个数少, 且相比于求解总哈密顿量的本征系统, 求解每一个新项的本征系统所需要的经典计算资源都很少。这符合 GPITE 所适用的前提条件。

具体而言, 对于氢化锂分子在 $R = 2.0\text{\AA}$ 下的哈密顿量, 我们将不包括常数项在内的 61 个泡利项重新排列并分成 22 组, 各组求和后共得到了 22 个新项。分组能够保证每一个新项的最小本征值之和大于原始泡利项的最小本征值之和, 即 $\sum_{k'} \lambda_{k',0} > -\sum_k |c_k|$, 这样可以保证算法的成功概率高于原始 PITE 算法。分组的

方式以及得到的新项在表3.2中显示。以其中的某一项 \mathbf{H}_{18} 为例：

$$\mathbf{H}_{18} = -c (\sigma_x^1 \sigma_x^2 \sigma_y^3 \sigma_y^4 + \sigma_y^1 \sigma_y^2 \sigma_x^3 \sigma_x^4 - \sigma_x^1 \sigma_y^2 \sigma_y^3 \sigma_x^4 - \sigma_y^1 \sigma_x^2 \sigma_x^3 \sigma_y^4), \quad (3.49)$$

其中 $c = 1.03458 \times 10^{-2}$ 。 \mathbf{H}_{18} 只作用在前四个比特上，具有三个不同本征值： $\lambda_{\pm} = \pm 4c$ （简并度各为 2）和 $\lambda_0 = 0$ （简并度为 12），对应的本征态在四比特子系统中表示为

$$\begin{aligned} \lambda_{\pm} : & \frac{1}{\sqrt{2}}(|0110\rangle \pm |1001\rangle), \quad \frac{1}{\sqrt{2}}(|0011\rangle \mp |1100\rangle), \\ \lambda_0 : & |0000\rangle, \quad |0001\rangle, \quad |0010\rangle, \quad |0100\rangle, \quad |0101\rangle, \quad |0111\rangle, \\ & |1000\rangle, \quad |1010\rangle, \quad |1011\rangle, \quad |1101\rangle, \quad |1110\rangle, \quad |1111\rangle. \end{aligned} \quad (3.50)$$

按照 GPITE 算法，在四个比特上实现虚时演化算符 $e^{-\mathbf{H}_{18}\Delta\tau}$ 的量子线路如图3.9所示。其中， \mathbf{U}_{18} 算符可以将 λ_+ 的本征态变换为 $|1011\rangle$ 与 $|1111\rangle$ ，将 λ_- 的本征态变换为 $|0011\rangle$ 与 $|0111\rangle$ ，将 λ_0 的本征态变换为其余的计算基。受控 \mathbf{R}_y 操作中的角参数分别为 $\theta_1 = 2 \arccos(e^{-8c\Delta\tau})$ ， $\theta_0 = 2 \arccos(e^{-4c\Delta\tau})$ 。

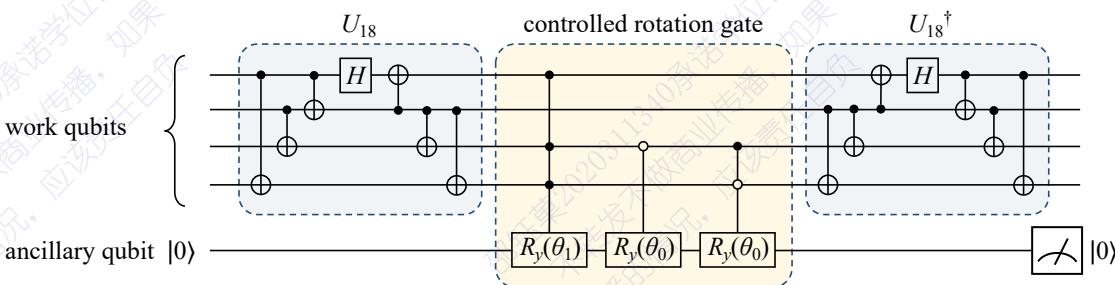


图 3.9 利用 GPITE 算法求解氢化锂分子的基态时，实现 \mathbf{H}_{18} 项的虚时演化的量子线路图

对于周期性量子伊辛链，我们将哈密顿量写为 $\mathbf{H} = \sum_{k=1}^n \mathbf{H}_k$ ，其中取 $J = 1$ 且

$$\mathbf{H}_k = -(\sigma_z^k \sigma_z^{k+1} + g \sigma_x^k + h \sigma_z^k). \quad (3.51)$$

按这样的方式，每一项 \mathbf{H}_k 都是一个局域项，仅作用在第 k 和第 $k+1$ 个比特上。我们在两比特计算基的表象下写出 \mathbf{H}_k 的矩阵形式：

$$\mathbf{H}_k = - \begin{pmatrix} 1+h & 0 & g & 0 \\ 0 & -1+h & 0 & g \\ g & 0 & -1-h & 0 \\ 0 & g & 0 & 1-h \end{pmatrix}. \quad (3.52)$$

该矩阵的本征值为 $\lambda_{0,3} = \mp \sqrt{g^2 + (h+1)^2}$ 与 $\lambda_{1,2} = \mp \sqrt{g^2 + (h-1)^2}$ （角标按本征值升序排列），同时可以得到对应的本征态 $|\lambda_{0,1,2,3}\rangle$ 。按照 GPITE 算法的过程，我

们利用如图3.10所示的量子线路来实现虚时演化算符 $e^{-H_k \Delta\tau}$, 其中量子门的角参数分别为

$$\begin{aligned}\varphi_1 &= \arccos \frac{1-h}{\sqrt{g^2 + (h-1)^2}}, \\ \varphi_2 &= \arccos \frac{-1-h}{\sqrt{g^2 + (h+1)^2}}, \\ \theta_i &= 2 \arccos(e^{-\Delta\tau(\lambda_i - \lambda_0)}), \quad i = 1, 2, 3.\end{aligned}\quad (3.53)$$

线路中的 U_k 可以将本征态 $|\lambda_{0,1,2,3}\rangle$ 分别变换为 $|10\rangle, |11\rangle, |01\rangle, |00\rangle$ 。

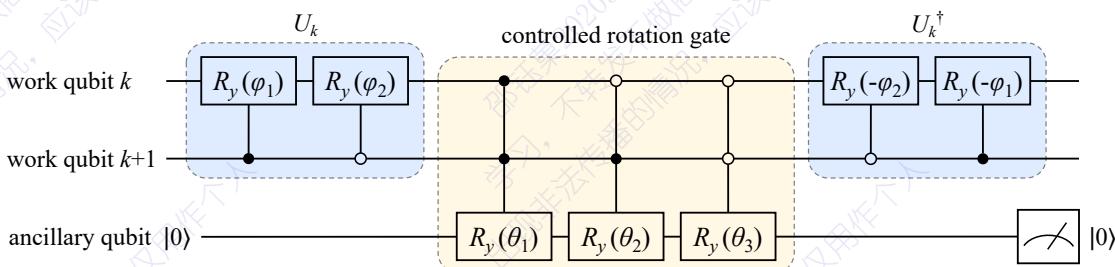


图 3.10 利用 GPITE 算法求解量子伊辛链的基态时, 实现 H_k 项的虚时演化的量子线路图

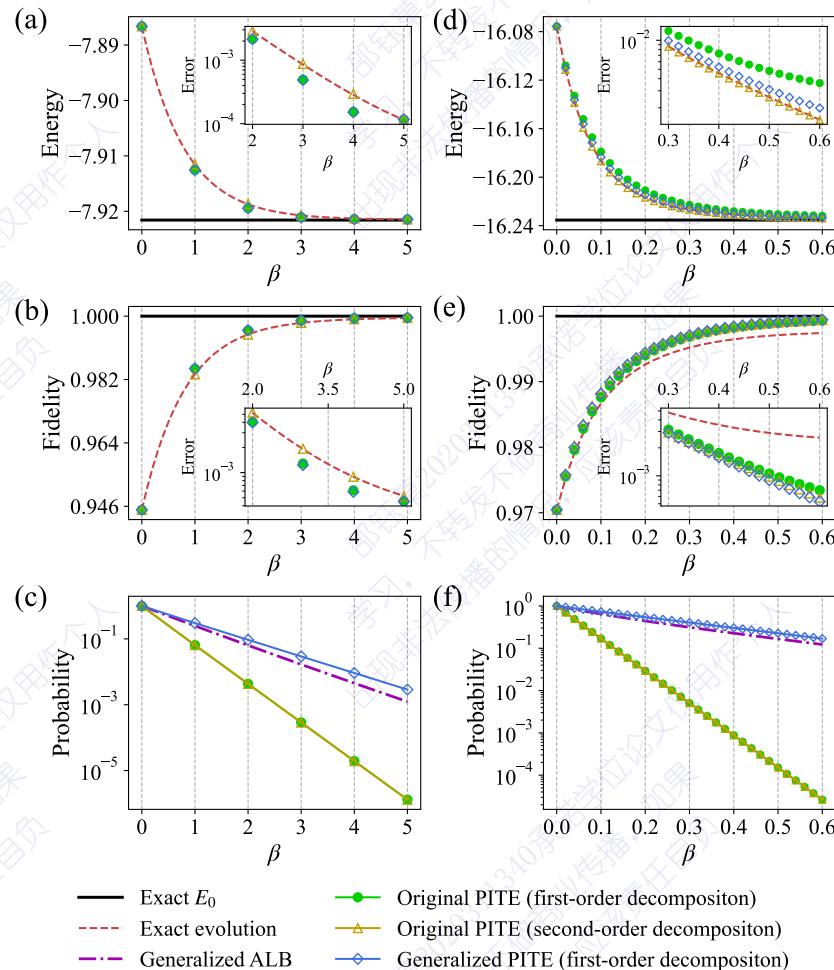
利用 GPITE 求解氢化锂分子和量子伊辛链基态能量的数值模拟结果如图3.11所示, 其中, (a)(b)(d)(e) 展示了能量期望值与量子态保真度的收敛情况, (c) 和 (f) 将模拟过程中的真实成功概率与公式3.47给出的成功概率下界进行了比较。结果表明, 从量子态能量与保真度的收敛情况来看, GPITE 与原始 PITE 算法的效果几乎没有差别。但从最终累积成功概率上看, GPITE 算法相比于原始 PITE 而言在数量级上提升了 $10^2\text{-}10^4$, 使得算法的实用性更强。

在针对 GPITE 算法的研究之外, 我们还模拟了原始 PITE 算法在更高阶 Trotter 展开 (3.36式) 下的性能。模拟结果在图3.11中显示, 可以看出高阶 Trotter 展开下的 PITE 算法给出的演化曲线更接近精确的虚时演化, 而在算法的成功概率方面, 高阶 Trotter 展开并没有明显的效果。

3.5 概率性虚时演化算法的复杂度讨论

PITE 与 GPITE 算法复杂度的最主要来源是由成功概率所产生的时间复杂度。由于 PITE 是 GPITE 的一种特殊情况, 这里我们只讨论 GPITE。

在利用 GPITE 算法求解量子体系的基态时, 算法以 ϵ 的误差得到基态, 成功概率为 $P_{\text{final}} = O(\epsilon^\kappa)$, 其中 $\kappa = (E_0 - \sum_k \lambda_{k,0})/\Omega_1$ 反比于基态与第一激发态之间的能隙。能隙减小, 成功概率也迅速减小。而一般来说, 大多数现有的量子算法



(a) 氢化锂分子的能量期望值随 β 变化。(b) 氢化锂分子的量子态保真度随 β 变化。(c) 求解氢化锂分子基态的成功概率随虚时间 β 的变化。(d) 量子伊辛链的能量期望值随 β 变化。(e) 量子伊辛链的量子态保真度随 β 变化。(f) 求解量子伊辛链基态的成功概率随虚时间 β 的变化。(a)(b)(d)(e) 图中的红色虚线为精确的虚时演化曲线。(c)(f) 图中的紫色点划线为公式3.47给出的 GPITE 成功概率近似下界。求解氢化锂分子的过程中，键长固定为 $R = 2.0\text{\AA}$ ，且哈密顿量中的常数项在运行算法及计算成功概率下界时被忽略，但在计算能量平均值时被保留。求解量子伊辛链时，选取的参数为 $n = 10, J = 1, g = 1.2, h = 0.3$ 。

图 3.11 在数值模拟中，利用 GPITE 算法和高阶 Trotter 展开的 PITE 算法求解氢化锂分子与量子伊辛链的基态与基态能量，以及算法的累积成功概率

都受到哈密顿量能隙的影响，例如：变分算法 VQE 利用梯度下降等经典优化方法寻找能量期望值的极小值，在这一过程中，能隙的大小会影响经典优化过程，可能使优化过程遇到“贫瘠高原”问题^[152]；量子相位估计、HHL 及相关算法中，能隙与矩阵的条件数有关，而条件数影响了算法所消耗的时间资源。对于经典算法而言，计算 N 维矩阵的指数 $e^{-\beta H}$ 需要消耗 $\mathcal{O}(N^3)$ 的计算资源^[210]。而如前所述，对于一个成功执行的量子线路，门复杂度为 $\mathcal{O}(\log(N)mL)$ 。因此，整个算法的空间复杂度为 $\mathcal{O}(\log N)$ ，时间复杂度为 $\mathcal{O}(\log(N)mL/P_{\text{final}})$ 。在 $P_{\text{final}} \geqslant 1/N^2$ 的情况下，GPITE 算法优于相应的经典算法。

PITE 与 GPITE 算法所实现的是量子系统的虚时间演化模拟，而不只是求解基态能量。算法的成功概率随虚时间 β 和哈密顿量的谱半径呈指数衰减，这是源自于非幺正量子线路天然的概率性。也正因此，累积成功概率的衰减在所有基于非幺正量子线路的迭代性算法^[181,186,223]中都是常见问题。为了提高算法的成功概率，我们努力进行改进，将 PITE 算法推广为 GITE 算法。除此之外，另一种可能提高成功概率的做法是将概率性算法和其他方法结合起来，例如变分量子算法^[230]，这是未来的工作中需要研究的课题。

3.6 本章小结

在本章中，我们提出了一种基于非幺正量子线路的概率性虚时演化算法（PITE）。针对任意比特数量、任意可高效表达成泡利算符形式的哈密顿量，该算法都可以给出量子线路的具体构造方式。PITE 算法可应用于求解量子体系的基态和基态能量，当求解基态的保真度误差为 ϵ 时，算法的累积成功概率为 $P_{\text{final}} = O(\epsilon^\kappa)$ ，其中 κ 由哈密顿量的能谱决定。对于一个成功执行的量子线路，其量子门复杂度为 $\mathcal{O}(\log(N)mL)$ ，其中 n 为量子比特数， m 为哈密顿量中泡利项的个数， L 为演化的步数。

此外，我们通过超导和离子阱量子计算云平台，以求解氢分子的基态为例，对 PITE 算法的原理和过程进行了演示实验。在数值模拟中，我们利用 PITE 算法求解氢化锂分子和量子伊辛链的基态，并探究了模拟过程中算法的累积成功概率，以及算法在量子噪声情况下的表现。最后，我们还将 PITE 算法推广到哈密顿量不是由泡利项组成的情况，即 GPITE 算法，并通过数值模拟证明了其在成功概率上的显著提高。

有了在量子计算机上高效模拟虚时演化的能力，人们可以在吉布斯态制备或研究有限温度量子系统的性质等方面进行探索。此外，该算法还可以应用于非厄米物理体系的量子模拟。非厄米哈密顿量可以分解为厄米部分和反厄米部分，其中反厄米部分的动力学演化可以等效于虚时演化的过程。在下一章中我们将详细介绍如何利用 PITE 算法模拟非厄米体系的动力学演化过程。

第4章 非厄米体系动力学演化的量子模拟

非厄米体系的动力学演化过程体现了非厄米体系的一些特殊性质，是我们所感兴趣的问题。本章节中，我们将概率性虚时演化算法应用于非厄米体系动力学演化模拟问题中。同时我们也探究了变分量子算法在非厄米体系动力学演化模拟中的应用。

4.1 非厄米体系及其动力学演化

在前面的内容中，我们始终假定所有的可观测量都是厄米算符，这是量子力学的公设之一。哈密顿量的厄米性保证了能谱是实数，且保证了时间演化算符的幺正性，从而使得量子态密度 $\langle \psi | \psi \rangle$ 守恒。然而随着理论的深入发展，人们发现，哈密顿量的厄米性实际上是孤立量子系统的要求。许多情况下，我们所研究的量子系统与周围的环境存在相互作用，或者我们只关心一个庞大系统中的有限子系统，这时我们感兴趣的部分就是一个开放的量子系统，其能谱不一定是实数，量子态的密度也不再守恒^[231-232]。

描述一个开放量子系统通常采用密度矩阵 ρ 。一种常用的描述开放量子系统动力学演化的方式是量子信道：

$$\rho(t) = \sum_k E_k \rho(0) E_k^\dagger, \quad (4.1)$$

其中 E_k 称为 Kraus 算符，其满足归一性条件 $\sum_k E_k^\dagger E_k = I$ 。另一种常用的描述开放量子系统动力学演化的方式是密度矩阵的演化方程，被称为 Lindblad 主方程^[233]：

$$\frac{\partial \rho}{\partial t} = -i [H, \rho] + \sum_i \gamma_i \left(L_i \rho L_i^\dagger - \frac{1}{2} \{ L_i^\dagger L_i, \rho \} \right), \quad (4.2)$$

其中 H 是系统自身的厄米哈密顿量， $\{ \cdot, \cdot \}$ 是反对易括号， L_i 是 Lindblad 算符，表示系统与环境的相互作用， γ_i 是对应的衰减率。定义等效哈密顿量：

$$H_{\text{eff}} = H - \frac{i}{2} \gamma_i L_i^\dagger L_i, \quad (4.3)$$

则主方程可重写为 $\dot{\rho} = -i (H_{\text{eff}} \rho - \rho H_{\text{eff}}^\dagger) + \sum_i \gamma_i L_i \rho L_i^\dagger$ ，其中的 $\sum_i \gamma_i L_i \rho L_i^\dagger$ 项对应于量子跳跃（quantum jump）。在一些特定条件下（例如系统与环境不产生纠缠、对环境的微观信息不关心，等等），量子跳跃项可以忽略，那么主方程就可以看作具有非厄米的等效哈密顿量 H_{eff} 的孤立系统的演化，这时我们所研究的量子体系就可以称为非厄米量子体系。非厄米体系的量子态仍可看作纯态，因

此可以直接用态矢量 $|\psi\rangle$ 来描述，其时间演化遵循非厄米哈密顿量的薛定谔方程： $i\partial|\psi\rangle/\partial t = \mathbf{H}_{\text{eff}}|\psi\rangle$ 。

常见的非厄米体系包括但不限于具有增益和损耗的光学系统^[234-239]、具有耗散的开放系统^[240-244]、具有有限寿命准粒子的电子系统^[245-248]、具有拓扑性质的晶格体系^[249-261]等。非厄米体系通常具有一些厄米系统所不具有的特性，如趋肤效应^[256,260]、边界态^[249-250]、拓扑相变^[251,255]、异常点（exceptional point, EP）^[247,262-263]等，从而引发了广泛的关注与研究兴趣。

2022 年，Wen-Tan Xue 等人的工作^[264]揭示了非厄米体系中的一种新奇的现象，被称为非厄米边缘突起（non-Hermitian edge burst, NHEB）。边缘突起现象所出现的体系是一个如图4.1(a) 所示的格点体系，其中每个元胞有 A、B 两个格点，所有的 A 和 B 分别构成两条子链。该体系所等效的非厄米哈密顿量为

$$\mathbf{H} = \sum_x \left[\left(a_x^\dagger b_x + \frac{t_2}{2} a_x^\dagger b_{x+1} + \frac{t_2}{2} a_{x+1}^\dagger b_x + \frac{it_2}{2} a_{x+1}^\dagger a_x - \frac{it_2}{2} b_{x+1}^\dagger b_x + \text{h.c.} \right) - i\gamma b_x^\dagger b_x \right], \quad (4.4)$$

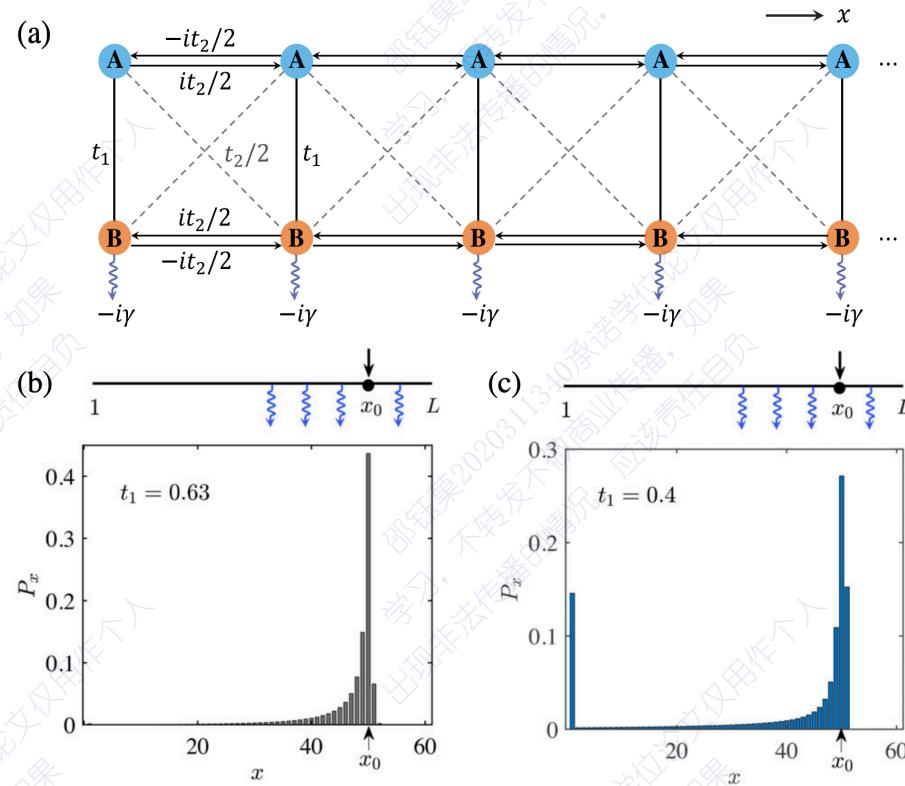
其中 h.c. 代表其前面所有项的共轭转置之和， t_1, t_2, γ 是描述体系相互作用强度的参数， a_x^\dagger, b_x^\dagger 分别为 x 位置 A 和 B 格点处的粒子产生算符， a_x, b_x 为对应的粒子湮灭算符。该体系中导致非厄米效应的项为 $-i\gamma b_x^\dagger b_x$ ，其物理含义为每个 B 格点处存在强度为 γ 的粒子逃逸。若初始时刻将一个粒子放在 x_0 位置的 A 格点上，随着哈密顿量的演化，粒子的波函数会在各个格点间转移并产生概率分布，但每个 B 格点处波函数都受到 γ 强度的衰减。经过足够长时间后，格点内粒子的总波函数将衰减为 0，这意味着粒子一定从某个 B 格点离开了体系，逃逸到环境中。通过计算，可以得到粒子从各个 x 位置的 B 格点逃逸的概率 P_x ，结果如图4.1(b) 和 (c) 所示。注意到，当参数 t_1 的选取满足某个范围时，粒子的逃逸概率在格点体系的边缘将产生突起，而其它情形则不会产生突起。研究人员将这一现象称为非厄米边缘突起，并进一步探究了其产生的条件及背后的物理效应。NHEB 引发了一系列相关的研究^[265-266]，并且已在实验中得到了验证^[267]。本文中，我们不关心 NHEB 产生的原因和相关物理图像，只关心如何利用量子算法实现 NHEB 的计算。

根据薛定谔方程，可以得到粒子数密度随时间变化的方程：

$$\frac{d}{dt} \langle \psi(t) | \psi(t) \rangle = i \langle \psi(t) | (\mathbf{H}^\dagger - \mathbf{H}) | \psi(t) \rangle = - \sum_x 2\gamma |\psi_x^B(t)|^2, \quad (4.5)$$

其中 $\psi_x^B(t)$ 是 t 时刻 x 位置的 B 格点处的粒子波函数。由此可以得到粒子从 x 位置的 B 格点逃逸的概率：

$$P_x = 2\gamma \int_0^\infty dt |\psi_x^B(t)|^2. \quad (4.6)$$



(a) 非厄米格点体系的模型，每个元胞用空间位置 x 标记，包含 A、B 两个格点。(b)(c) 粒子从各个 x 位置的 B 格点逃逸的概率 P_x 。其中格点构成的链的总长度为 $L = 60$ ，粒子的初始位置为 $x_0 = 50$ ，哈密顿量中的参数为 $t_2 = 0.5$ 和 $\gamma = 0.8$ ，对于 (b) 图取 $t_1 = 0.63$ ，不存在边缘突起现象，对于 (c) 图取 $t_1 = 0.4$ ，存在边缘突起现象。

图 4.1 非厄米格点体系中存在的边缘突起现象，图片来自参考文献^[264]

由于概率守恒，一定有 $\sum_x P_x = 1$ 。这一式子表明，逃逸概率 P_x 的计算依赖于对体系的动力学演化的模拟。当格点数量较大时，经典算法将消耗过多的计算资源，我们希望能利用量子算法对体系的动力学演化进行更高效的模拟。而非厄米哈密顿量 \mathbf{H} 所对应的时间演化算符 $e^{-i\mathbf{H}\tau}$ 是一个非幺正算符，并不能在量子线路中直接实现，因此需要寻找其它等效的方法。

目前，人们已经在利用量子计算机模拟非厄米体系或开放量子体系的动力学性质方面进行了一些研究^[51,182,186,188-189]。例如，D. W. Berry 等人^[186]于 2015 年提出将演化算符进行 Taylor 展开，并将带截断的 Taylor 序列写成幺正算符线性组合形式，进而通过 LCU 方法实现，这种方法既可以模拟幺正演化，也可以模拟非幺正演化。魏世杰等人^[188]于 2016 年提出可以利用 LCU 方法实现 Kraus 算符，从而模拟开放量子系统的量子信道；在此方法的基础上，A. W. Schlimgen 等人^[189]于 2021 年进一步将任意 Kraus 算符分解成厄米部分与反厄米部分，并近似写成不超过 4 个指数形式的幺正算符之和，从而在 LCU 方法中利用 2 个辅助比特进行模拟。Jin-Min Liang 等人^[182]于 2022 年提出将密度矩阵编码到量子比特的纯态中，通过

FQE 和梯度下降方法得到开放量子体系的非平衡稳态。此外在实验方面, Yang Wu 等人^[51]于 2019 年在金刚石-氮空位中心体系中通过在两个量子比特上施加幺正操作, 等效实现了单比特非厄米哈密顿量的时间演化。

以上这些方法在任意非厄米体系动力学模拟中应用时都会受到限制, 主要原因是难以给出普适、具体且高效的量子线路构造方式。本章后面的内容中, 我们将介绍如何将 PITE 算法应用于任意非厄米体系的动力学演化模拟中, 并给出适用于任意哈密顿量的线路构造方式。此外我们还研究了变分量子算法在非厄米体系动力学演化模拟中的应用。

4.2 基于概率性虚时演化的非厄米体系动力学模拟

考虑一个最简单的具有 NHEB 现象的格点模型, 其具有两个 A 格点和两个 B 格点, 即链长 $L = 2$, 如图4.2所示。在初始时刻, 我们将粒子放置在右上位置 ($x = 2$) 的 A 格点, 经过足够长时间后, 粒子从左下 ($x = 1$) 或右下 ($x = 2$) 的 B 格点逃逸。如果粒子从左下逃逸的概率大于从右下处逃逸的概率, 则说明出现了边缘突起现象, 否则没有出现边缘突起。

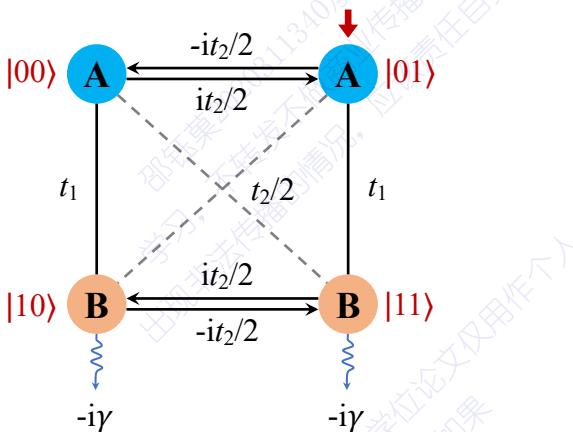


图 4.2 链长 $L = 2$ 的非厄米格点模型

若要利用量子算法模拟这样一个体系的演化过程, 需要将格点体系的哈密顿量编码到量子比特上。我们使用两个量子比特用来表示粒子在格点体系中的位置, 其中第一个比特的 $|0\rangle$ 和 $|1\rangle$ 态分别表示粒子处在 A 链和 B 链中, 第二个比特的 $|0\rangle$ 和 $|1\rangle$ 态分别表示粒子处在 $x = 1, 2$ 位置。两比特的四个计算基分别对应粒子处在四个不同的格点上, 如图4.2中所标注。将格点体系的哈密顿量对应到量子比特上, 得到

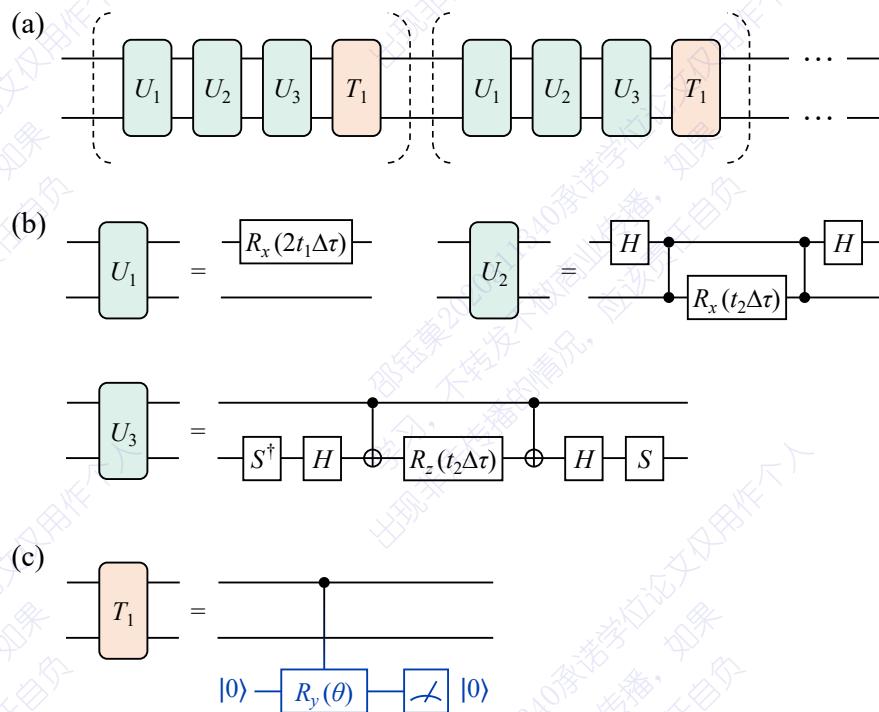
$$\mathbf{H} = t_1 \boldsymbol{\sigma}_x^1 + \frac{t_2}{2} \boldsymbol{\sigma}_x^1 \boldsymbol{\sigma}_x^2 + \frac{t_2}{2} \boldsymbol{\sigma}_z^1 \boldsymbol{\sigma}_y^2 - i\gamma \frac{\mathbf{I} - \boldsymbol{\sigma}_z^1}{2}. \quad (4.7)$$

若想在量子线路中实现该哈密顿量的演化算符 e^{-iHT} , 我们利用 Trotter 分解:

$$e^{-iHT} = e^{-\gamma T/2} \left(e^{-it_1 \sigma_x^1 \Delta\tau} e^{-it_2 \sigma_x^1 \sigma_x^2 \Delta\tau/2} e^{-it_2 \sigma_z^1 \sigma_y^2 \Delta\tau/2} e^{i\gamma \sigma_z^1 \Delta\tau/2} \right)^{T/\Delta\tau} + \mathcal{O}(\Delta\tau), \quad (4.8)$$

其中 $e^{-\gamma T/2}$ 是常数因子, 在模拟量子态的演化时可以忽略, 因此每一个 $\Delta\tau$ 间隔内包含 4 个 Trotter 迭代的步骤。

模拟非厄米格点体系动力学演化的量子线路的结构如图4.3(a)所示, 每个 $\Delta\tau$ 内的 4 个 Trotter 迭代对应的算符分别定义为 $U_1 = e^{-it_1 \sigma_x^1 \Delta\tau}$, $U_2 = e^{-it_2 \sigma_x^1 \sigma_x^2 \Delta\tau/2}$, $U_3 = e^{-it_2 \sigma_z^1 \sigma_y^2 \Delta\tau/2}$, $T_1 = e^{i\gamma \sigma_z^1 \Delta\tau/2}$, 其中 $U_{1,2,3}$ 是幺正的演化, 可具体通过图4.3(b)中的线路实现; 而 T_1 则是非幺正的演化, 其数学形式刚好等价于一个哈密顿量为 $-\gamma \sigma_z^1$ 的系统的虚时演化, 因此我们可以利用 PITE 算法来实现, 所需要的线路如图4.3(c)所示, 其中旋转门的参数为 $\theta = 2 \arccos(e^{-\gamma \Delta\tau})$ 。由此我们就可以对非厄米格点体系中的粒子波函数随时间的变化进行模拟。



(a) 根据 Trotter 展开得到的线路的总结构。(b)Trotter 迭代中的幺正算符的构造, 可以通过一般的哈密顿量演化算法实现。(c)Trotter 迭代中的非幺正算符的构造, 由于其数学形式与虚时演化相同, 可以通过 PITE 算法实现。

图 4.3 利用 Trotter 分解与 PITE 算法模拟 $L = 2$ 的非厄米格点体系所需要的量子线路

事实上, 任意非厄米体系的动力学演化都可以借助 PITE 算法进行模拟。首先, 一个任意的哈密顿算符 H 都可以写成一个厄米矩阵与一个反厄米矩阵之和:

$$H_S = \frac{H + H^\dagger}{2}, \quad H_A = \frac{H - H^\dagger}{2i}, \quad H = H_S + iH_A, \quad (4.9)$$

其中 H_S 和 H_A 都是厄米的, 则 iH_A 是反厄米的。根据 Trotter 展开, 我们就可以

将 \mathbf{H} 的时间演化算符写为

$$e^{-i\mathbf{H}T} = (e^{-i\mathbf{H}_S\Delta\tau} e^{\mathbf{H}_A\Delta\tau})^{T/\Delta\tau} + \mathcal{O}(\Delta\tau), \quad (4.10)$$

其中, 厄米部分的演化 $e^{-i\mathbf{H}_S\Delta\tau}$ 是幺正的, 可以通过一般的哈密顿量模拟算法来实现, 而反厄米部分的演化 $e^{\mathbf{H}_A\Delta\tau}$ 则等价于一个哈密顿量为 $-\mathbf{H}_A$ 的体系的虚时演化, 就可以利用 PITE 算法来实现。这样就可以通过厄米部分的幺正演化与反厄米部分的非幺正演化不断交替来实现 Trotter 迭代, 从而实现非厄米哈密顿量的动力学演化。

在模拟过程中需要注意的问题是, 由于演化算符的非幺正性, 实际格点体系中的波函数的模长时刻在变化, 其物理含义即为格点体系中粒子数密度由于逃逸而不断衰减; 但利用量子算法对波函数的演化进行模拟时, 量子比特的状态是归一的。因此我们在根据量子比特的状态还原真实的粒子波函数时, 需要乘以模长系数。考虑真实体系的波函数在 $|\psi\rangle$ 在 $\Delta\tau$ 时间内的演化: $|\psi'\rangle = e^{-i\mathbf{H}\Delta\tau} |\psi\rangle$, 当 $\Delta\tau$ 很小时, 波函数的模长变化为

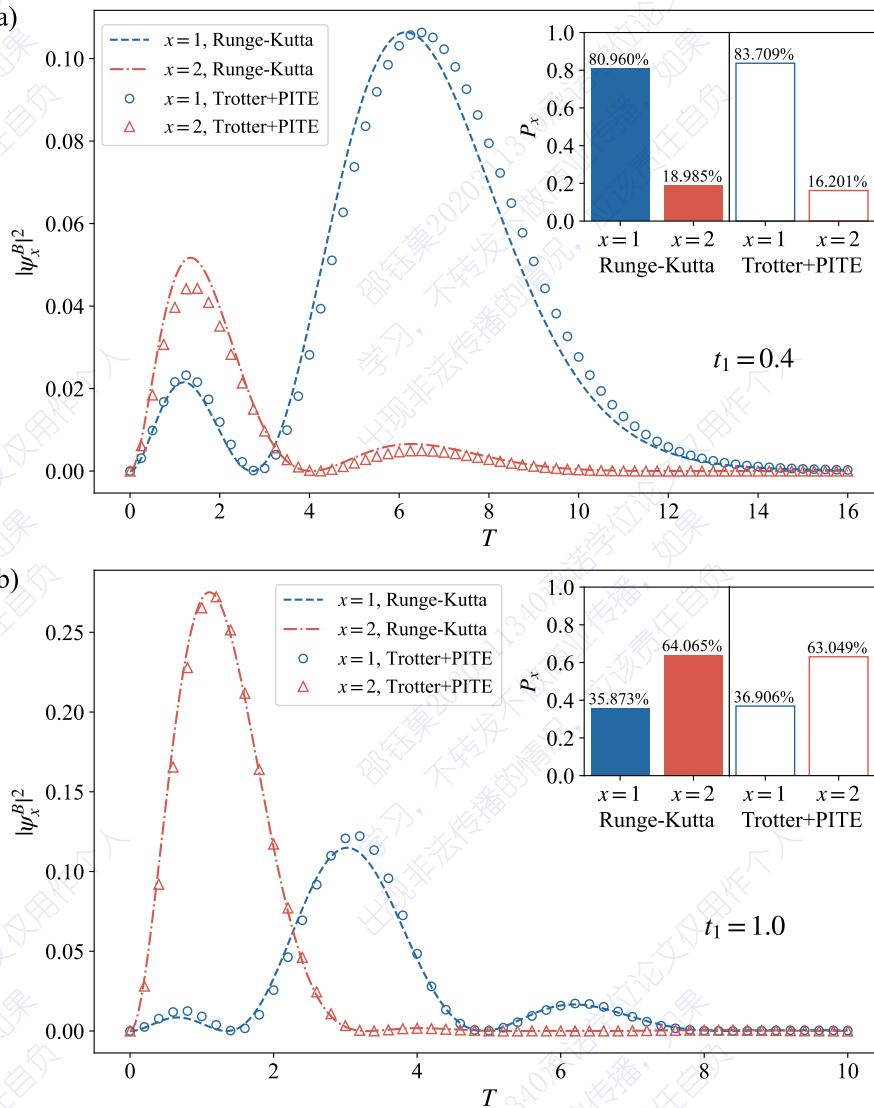
$$\begin{aligned} \langle\psi'|\psi'\rangle &= \langle\psi|e^{i\mathbf{H}^\dagger\Delta\tau} e^{-i\mathbf{H}\Delta\tau} |\psi\rangle = \langle\psi| (1 + i\mathbf{H}^\dagger\Delta\tau) (1 - i\mathbf{H}\Delta\tau) |\psi\rangle + \mathcal{O}(\Delta\tau^2) \\ &= \langle\psi|\psi\rangle + \Delta\tau \langle\psi| (i\mathbf{H}^\dagger - i\mathbf{H}) |\psi\rangle \\ &= \langle\psi|\psi\rangle + \gamma\Delta\tau \langle\psi| (\sigma_z^1 - \mathbf{I}) |\psi\rangle \\ &= (1 - \gamma\Delta\tau + \gamma\Delta\tau \langle\sigma_z^1\rangle) \langle\psi|\psi\rangle, \end{aligned} \quad (4.11)$$

其中 $\langle\sigma_z^1\rangle = \langle\psi|\sigma_z^1|\psi\rangle / \langle\psi|\psi\rangle$ 是 σ_z^1 的观测期望值, 也可以写为 $\langle\sigma_z^1\rangle = \langle\bar{\psi}|\sigma_z^1|\bar{\psi}\rangle$, 其中 $|\bar{\psi}\rangle$ 是归一化后的 $|\psi\rangle$ 态, 是 $|\psi\rangle$ 所对应的量子比特的状态。这意味着, 每一个 $\Delta\tau$ 的时间间隔后, 波函数的模长就会变为原先的 $1 - \gamma\Delta\tau + \gamma\Delta\tau \langle\sigma_z^1\rangle$ 倍, 那么在量子算法过程中, 只需要在 τ 时刻测量出当前 σ_z^1 的期望值, 再根据当前的波函数模长, 就可以得到 $\tau + \Delta\tau$ 时刻的波函数模长。

算法希望得到的是波函数 ψ_x^B 在演化过程中各个时刻的值, 并最终通过计算积分得到粒子逃逸概率 P_x 。我们总结以上分析和讨论, 给出算法的具体步骤:

1. 设定演化总时长 T 和每一步的时间间隔 $\Delta\tau$;
2. $\tau = 0$ 时刻, 制备量子比特的初始状态 $|\bar{\psi}(0)\rangle$, 设定初始的总波函数模长 $\alpha(0) = 1$;
3. 对当前量子比特的状态进行测量, 得到期望值 $\langle\sigma_z^1\rangle$, 并更新下一时刻波函数的模长 $\alpha(\tau + \Delta\tau) = (1 - \gamma\Delta\tau + \gamma\Delta\tau \langle\sigma_z^1\rangle) \alpha(\tau)$;
4. 按照顺序在量子线路中执行 $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \mathbf{T}_1$ 操作, 得到下一时刻的归一化波函数 $|\bar{\psi}(\tau + \Delta\tau)\rangle$;

5. 此时 $\tau = \tau + \Delta\tau$, 测量当前 $|\bar{\psi}\rangle$ 在 $|10\rangle$ 和 $|11\rangle$ 上的概率, 乘以当前总波函数模平方 α^2 , 得到当前各个 B 格点上的波函数模平方 $|\psi_x^B|^2$;
6. 重复第 3-5 步, 直至 $\tau = T$, 则粒子逃逸概率可以近似计算为 $P_x = \gamma\Delta\tau \sum_{\tau} (|\psi_x^B(\tau)|^2 + |\psi_x^B(\tau + \Delta\tau)|^2)$ 。



(a) $t_1 = 0.4$, 存在边缘突起现象。(b) $t_1 = 1.0$, 不存在边缘突起现象。两组模拟中的共同参数为 $t_2 = 0.5, \gamma = 0.8$ 。图中的散点是利用量子算法 PITE 得到的结果, 实线是利用经典算法龙格-库塔方法得到的结果。

图 4.4 利用 Trotter 分解和 PITE 算法模拟 $L = 2$ 的非厄米格点体系演化及边缘突起现象

按照以上过程, 我们对 $L = 2$ 的非厄米格点中的波函数的时间演化进行模拟, 得到的结果如图4.4所示, 其中利用量子算法得到的结果与利用经典龙格-库塔法得到的数值结果进行了对比, 两者显示出了很好的一致性。两组模拟中分别取 $t_1 = 0.4$ 和 $t_1 = 1.0$, 其他参数固定为 $t_2 = 0.5, \gamma = 0.8$ 。可以看到当 $t_1 = 0.4$ 时,

子的逃逸概率 $P_{x=1}$ 远大于 $P_{x=2}$, 即此时体系中存在边缘突起现象; 而 $t_1 = 1.0$ 时则没有边缘突起现象。量子算法得到的结果很好地符合了预期效果。

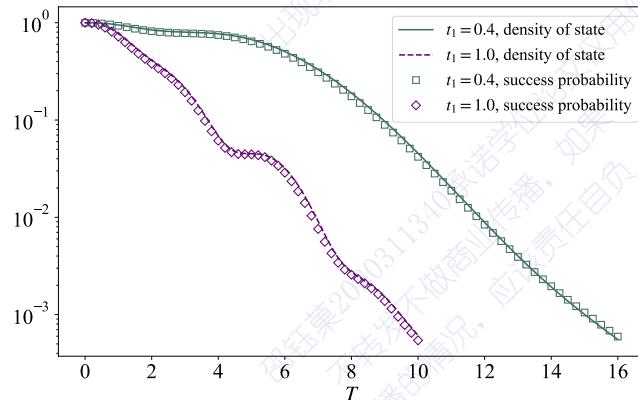


图 4.5 利用 Trotter 分解和 PITE 算法模拟 $L = 2$ 的非厄米格点体系演化的累积成功概率与格点体系中真实波函数的密度变化

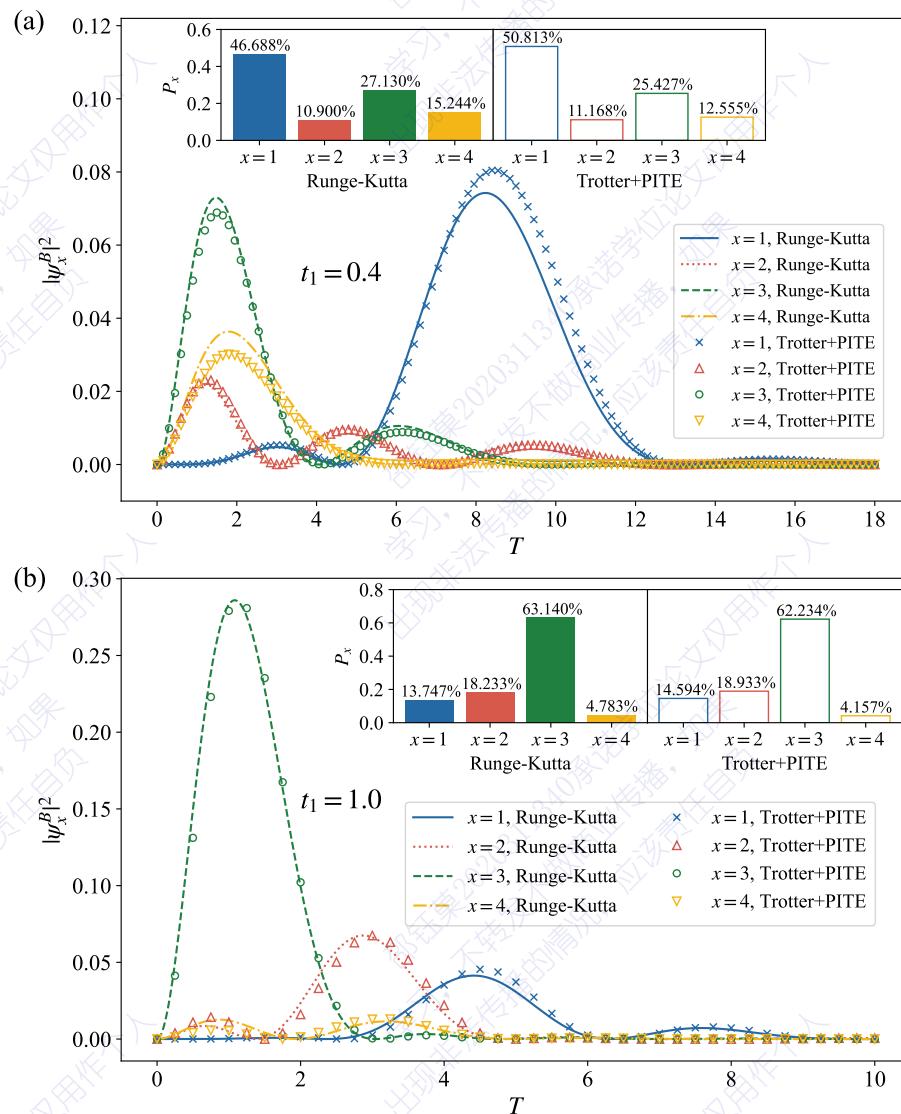
在量子算法运行的过程中, 我们同时计算了由测量后选择而引起的累积成功概率。在图4.5中, 我们将量子算法的累积成功概率与格点体系中真实波函数的密度进行了对比, 并展示了二者的一致性。这里我们指出, 这种一致性并非在所有非厄米体系的模拟中都存在, 因为算法的成功概率与所模拟体系的真实粒子数密度并没有必然联系。而之所以在我们所研究的非厄米格点体系中会出现这样的现象, 是由于在每一个 $\Delta\tau$ 内, 导致体系粒子数衰减的演化项 $e^{-\gamma\Delta\tau/2}e^{i\sigma_z^1\Delta\tau/2}$ 刚好与 PITE 算法所等效实现的非幺正算符完全相同, 不相差系数因子, 因此每个 $\Delta\tau$ 内体系内实际波函数模长的衰减比例就等于 PITE 算法的成功概率。只要非厄米体系符合这一条件, 就可以观察到算法成功概率与体系真实的波函数密度一致的现象。

进一步, 我们利用 PITE 算法对 $L = 4$ 的非厄米格点体系进行了模拟, 其中粒子初始时刻放置于 $x = 3$ 处的 A 格点。将 $L = 4$ 的格点编码到量子比特中, 需要 3 个量子比特, 其中第一个比特用来表示粒子处于 A 链或 B 链, 第二和第三个比特的四种状态 $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ 分别表示粒子处于 $x = 1, 2, 3, 4$ 位置。将格点体系的哈密顿量对应到量子比特中, 得到

$$\begin{aligned} \mathbf{H} = & t_1 \sigma_x^1 + \frac{t_2}{2} \sigma_x^1 \sigma_x^3 + \frac{t_2}{2} \sigma_z^1 \sigma_y^3 + \frac{t_2}{4} \sigma_x^1 \sigma_x^2 \sigma_x^3 + \frac{t_2}{4} \sigma_x^1 \sigma_y^2 \sigma_y^3 \\ & - \frac{t_2}{4} \sigma_z^1 \sigma_x^2 \sigma_y^3 + \frac{t_2}{4} \sigma_z^1 \sigma_y^2 \sigma_x^3 - i\gamma \frac{\mathbf{I} - \sigma_z^1}{2}. \end{aligned} \quad (4.12)$$

利用 Trotter 分解, 将 PITE 算法应用于模拟演化过程, 其中量子比特的初始状态为 $|010\rangle$ 。模拟所得到 B 格点上的波函数随时间的演化曲线如图4.6所示。量子算法与经典龙格-库塔方法得到的结果呈现出一致性, 且在 $t_1 = 0.4$ 时可以观察到边缘突起, 在 $t_1 = 1.0$ 时则没有出现边缘突起。图4.7则展示了量子算法的累积成功概率

与格点体系中真实波函数的密度，二者之间依然存在一致性。



(a) $t_1 = 0.4$, 存在边缘突起现象。(b) $t_1 = 1.0$, 不存在边缘突起现象。两组模拟中的共同参数为 $t_2 = 0.5, \gamma = 0.8$ 。图中的散点是利用量子算法 PITE 得到的结果, 实线是利用经典算法龙格-库塔方法得到的结果。

图 4.6 利用 Trotter 分解和 PITE 算法模拟 $L = 4$ 的非厄米格点体系演化及边缘突起现象

4.3 基于变分量子算法的非厄米体系动力学模拟

另一种能够模拟非厄米体系动力学演化的算法就是变分量子算法。在2.3.3小节和3.1节中, 我们分别介绍了利用变分量子算法模拟一般厄米哈密顿量的实时间演化和虚时间演化, 但由于归一化因子所导致的演化方程有所不同, 这两种情形所对应的参数演化方程并不能直接应用于非厄米体系的演化中。因此我们在这里首先基于变分量子模拟的思路, 对非厄米哈密顿量演化所对应的参数演化方程进

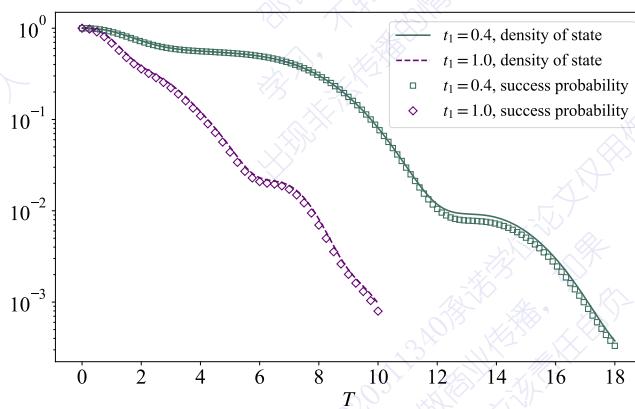


图 4.7 利用 Trotter 分解和 PITE 算法模拟 $L = 4$ 的非厄米格点体系演化的累积成功概率与格点体系中真实波函数的密度变化

行推导。

假设非厄米哈密顿量可以写成厄米部分与反厄米部分之和: $\mathbf{H} = \mathbf{H}_S + i\mathbf{H}_A$ 。非厄米体系演化过程中的归一化量子态为

$$|\bar{\psi}(t)\rangle = A(t)^{-1/2} e^{-i\mathbf{H}t} |\bar{\psi}(0)\rangle, \quad A(t) = \langle \bar{\psi}(0) | e^{i\mathbf{H}^\dagger t} e^{-i\mathbf{H}t} | \bar{\psi}(0) \rangle, \quad (4.13)$$

因此有

$$\begin{aligned} \frac{\partial |\bar{\psi}(t)\rangle}{\partial t} &= -\frac{1}{2} \frac{\partial A}{\partial t} A(t)^{-3/2} e^{-i\mathbf{H}t} |\bar{\psi}(0)\rangle - i\mathbf{H} A(t)^{-1/2} e^{-i\mathbf{H}t} |\bar{\psi}(0)\rangle \\ &= \left(-i\mathbf{H} - \frac{1}{2} \frac{\partial A}{\partial t} \right) |\bar{\psi}(t)\rangle, \end{aligned} \quad (4.14)$$

其中

$$\begin{aligned} \frac{\partial A}{\partial t} &= \langle \bar{\psi}(0) | e^{i\mathbf{H}^\dagger t} (i\mathbf{H}^\dagger - i\mathbf{H}) e^{-i\mathbf{H}t} | \bar{\psi}(0) \rangle \\ &= 2 \langle \bar{\psi}(0) | e^{i\mathbf{H}^\dagger t} \mathbf{H}_A e^{-i\mathbf{H}t} | \bar{\psi}(0) \rangle \\ &= 2A \langle \mathbf{H}_A \rangle, \end{aligned} \quad (4.15)$$

这里 $\langle \mathbf{H}_A \rangle = \langle \bar{\psi}(t) | \mathbf{H}_A | \bar{\psi}(t) \rangle$ 表示当前时刻 \mathbf{H}_A 的观测期望值。由此可以得到非厄米体系中归一化量子态所满足的演化方程:

$$\frac{\partial |\bar{\psi}(t)\rangle}{\partial t} = (-i\mathbf{H} - \langle \mathbf{H}_A \rangle) |\bar{\psi}(t)\rangle. \quad (4.16)$$

将参数化量子态 $|\bar{\psi}(\theta)\rangle$ 代入上述演化方程, 得到

$$\sum_k \frac{\partial |\bar{\psi}\rangle}{\partial \theta_k} \dot{\theta}_k = (-i\mathbf{H} - \langle \mathbf{H}_A \rangle) |\bar{\psi}\rangle. \quad (4.17)$$

对该方程的模长做变分, 并进一步处理, 最终得到参数所满足的演化方程:

$$\sum_k A_{jk} \dot{\theta}_k = B_j, \quad (4.18)$$

其中

$$A_{jk} = \operatorname{Re} \left(\frac{\partial \langle \bar{\psi} |}{\partial \theta_j} \frac{\partial | \bar{\psi} \rangle}{\partial \theta_k} \right), \quad B_j = \operatorname{Im} \left(\frac{\partial \langle \bar{\psi} |}{\partial \theta_j} \mathbf{H} | \bar{\psi} \rangle \right). \quad (4.19)$$

以上变分法则存在一个问题，即它对量子态的全局相位有冗余的要求。简单来说，假设有一个这样的随时间变化的量子态： $|\bar{\psi}'(t)\rangle = e^{i\phi(t)} |\bar{\psi}(t)\rangle = e^{i\phi(t)} A(t)^{-1/2} e^{-i\mathbf{H}t} |\bar{\psi}(0)\rangle$ ，其中 $\phi(t)$ 是任意实数，由于全局相位因子可忽略，这个态也可以描述哈密顿量 \mathbf{H} 的时间演化，但它不满足演化方程。也就是说，由于对全局相位的限制，演化方程得到的解只是所有可以描述演化的态的必要不充分条件。为了解除演化方程对全局相位的限制，人们通常采用两种方法。第一种方法是利用密度矩阵演化方程^[169-170]：非厄米体系中的归一化密度矩阵 $\rho = |\bar{\psi}\rangle \langle \bar{\psi}|$ 满足演化方程 $\dot{\rho} + i(\mathbf{H}\rho - \rho\mathbf{H}^\dagger) + 2\langle \mathbf{H}_A \rangle \rho = 0$ ，该方程不受全局相位的影响，可以将参数化密度矩阵 $\rho(\theta)$ 代入上述方程，整理得到参数的演化方程。第二种方法是将全局相位 ϕ 作为待优化参数的一个分量，即参数化量子态为 $|\bar{\psi}(\phi, \theta)\rangle$ ，再按照原来的方法得到参数演化方程后，消去其中的 $\dot{\phi}$ 即可得到其他参数的方程。这两种方法得到的结果虽有差异，但都可以消除全局相位的影响。

作为另一种可以消除全局相位影响的方法，我们提出了利用演化态与参数态的内积推导参数演化方程的方法。考虑 $\Delta\tau$ 的时间内，参数化量子态由 $|\bar{\psi}(\theta)\rangle$ 变为 $|\bar{\psi}(\theta + \Delta\theta)\rangle$ ，而非厄米体系中的归一化量子态演化为

$$|\bar{\psi}'\rangle = \frac{e^{-i\mathbf{H}\Delta\tau} |\bar{\psi}\rangle}{\sqrt{\langle \bar{\psi} | e^{i\mathbf{H}^\dagger\Delta\tau} e^{-i\mathbf{H}\Delta\tau} | \bar{\psi} \rangle}}. \quad (4.20)$$

在不考虑全局相位的情况下，上面两个演化末态应该相等。我们令二者的内积之模等于 1，即令下式等于 0：

$$\epsilon = 1 - |\langle \bar{\psi}(\theta + \Delta\theta) | \bar{\psi}' \rangle|^2 = 1 - \frac{|\langle \bar{\psi}(\theta + \Delta\theta) | e^{-i\mathbf{H}\Delta\tau} | \bar{\psi} \rangle|^2}{\langle \bar{\psi} | e^{i\mathbf{H}^\dagger\Delta\tau} e^{-i\mathbf{H}\Delta\tau} | \bar{\psi} \rangle}. \quad (4.21)$$

这一表达式中不包含全局相位，因此可以消除全局相位的影响。将上式对小量 $\Delta\tau$ 和 $\Delta\theta$ 做展开，并利用参数化量子态始终归一这一条件，展开后保留到二阶小量，整理后便得到参数的演化方程：

$$\sum_k A_{jk} \dot{\theta}_k = B_j, \quad (4.22)$$

其中

$$\begin{aligned} A_{jk} &= \operatorname{Re} \left(\frac{\partial \langle \bar{\psi} |}{\partial \theta_j} \frac{\partial | \bar{\psi} \rangle}{\partial \theta_k} - \frac{\partial \langle \bar{\psi} |}{\partial \theta_j} | \bar{\psi} \rangle \langle \bar{\psi} | \frac{\partial | \bar{\psi} \rangle}{\partial \theta_k} \right), \\ B_j &= \operatorname{Im} \left(\frac{\partial \langle \bar{\psi} |}{\partial \theta_j} \mathbf{H} | \bar{\psi} \rangle + \langle \mathbf{H} \rangle \langle \bar{\psi} | \frac{\partial | \bar{\psi} \rangle}{\partial \theta_j} \right). \end{aligned} \quad (4.23)$$

我们将基于这一变分法则的变分量子算法应用于非厄米格点体系的动力学演化模拟中。同上一节一样，考虑到量子比特的状态始终是归一的，而实际体系中的粒子波函数是不归一的，因此需要在模拟演化的过程中时刻计算二者相差的系数。这里我们给出利用变分量子算法模拟非厄米体系动力学演化的具体步骤：

1. 设定演化总时长 T 和每一步的时间间隔 $\Delta\tau$ ，选取合适的参数化量子线路；
2. $\tau = 0$ 时刻，设定初始参数 $\theta(0)$ ，设定初始的总波函数模长 $\alpha(0) = 1$ ；
3. 将当前参数 $\theta(\tau)$ 输入到参数化量子线路中，对量子比特的状态进行测量，得到期望值 $\langle \sigma_z^1 \rangle$ ，并更新下一时刻波函数的模长 $\alpha(\tau + \Delta\tau) = (1 - \gamma\Delta\tau + \gamma\Delta\tau \langle \sigma_z^1 \rangle) \alpha(\tau)$ ；
4. 将当前参数 $\theta(\tau)$ 输入到参数化量子线路中，对量子比特的状态进行测量，得到各个 A_{jk} 和 B_j 的值（ A_{jk} 和 B_j 并不是量子力学中的可观测量，不能直接测量，但可以转化为两个量子态的内积的实部从而进行测量）；
5. 将 A_{jk} 和 B_j 输入到经典参数优化器中，求解线性方程组 $\mathbf{A}\dot{\theta} = \mathbf{B}$ ，并更新下一时刻的参数 $\theta(\tau + \Delta\tau) = \theta(\tau) + \dot{\theta}\Delta\tau$ ；
6. 此时 $\tau = \tau + \Delta\tau$ ，将当前参数 θ 输入到参数化量子线路中，测量量子态在 $|10\rangle$ 和 $|11\rangle$ 上的概率，并乘以当前总波函数模平方 α^2 ，得到当前各个 B 格点上的波函数模平方 $|\psi_x^B|^2$ ；
7. 重复第 3-6 步，直至 $\tau = T$ ，则粒子逃逸概率可以近似计算为 $P_x = \gamma\Delta\tau \sum_{\tau} (|\psi_x^B(\tau)|^2 + |\psi_x^B(\tau + \Delta\tau)|^2)$ 。

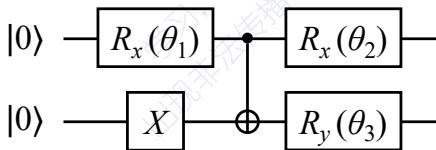
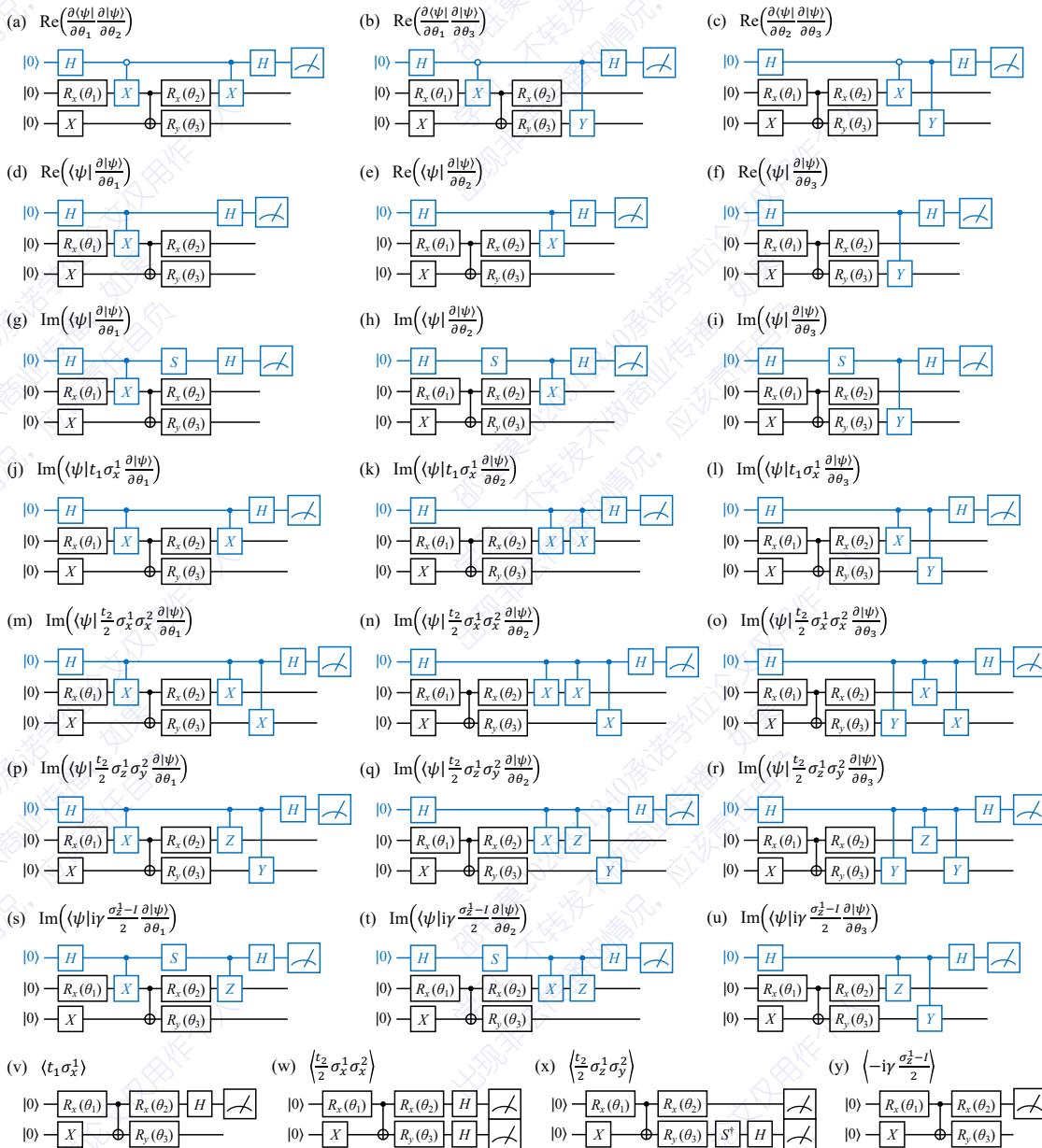


图 4.8 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系所使用的参数化量子线路

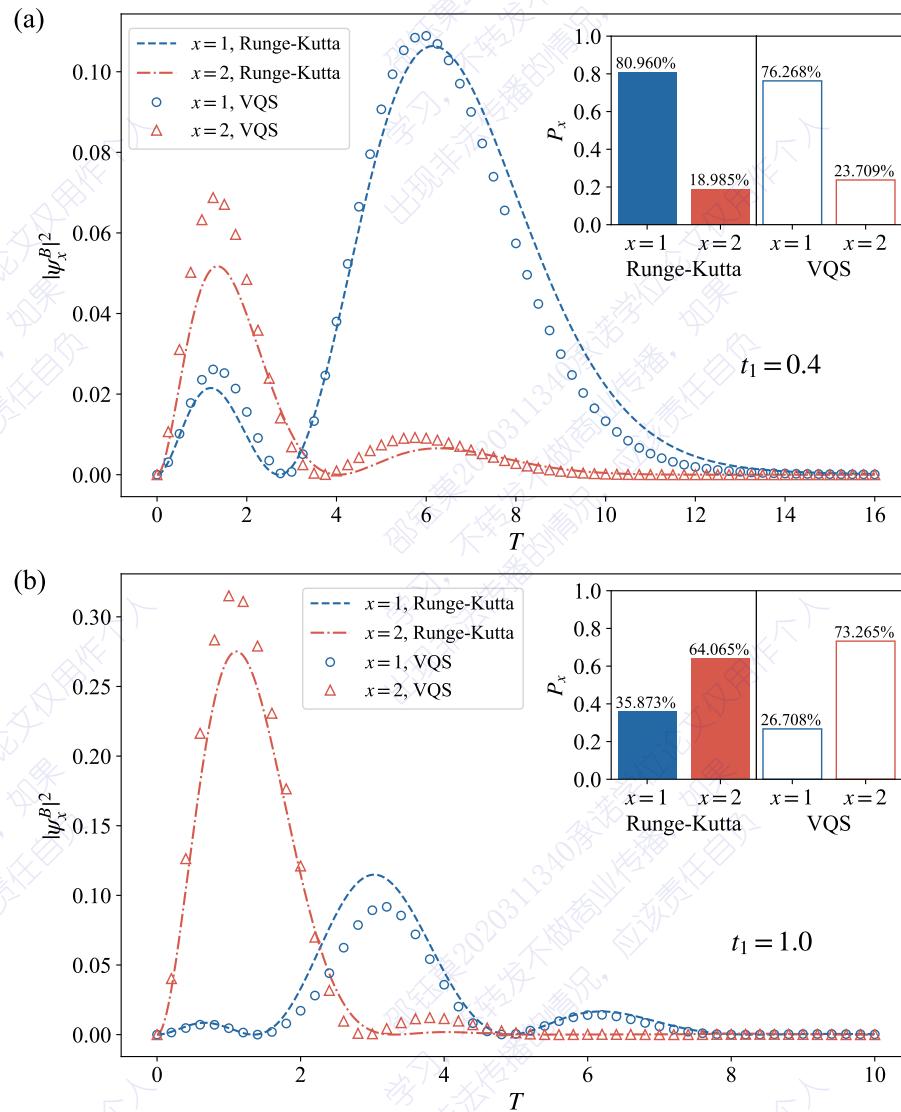
我们采用如图4.8所示的两比特参数化量子线路，线路中包含 $\theta_{1,2,3}$ 三个独立参数，当所有参数均为 0 时，参数化量子线路的输出态为 $|01\rangle$ ，即体系的演化初态。因此初始时刻的参数均选取为 0。在变分量子算法执行的过程中，需要测量 A_{jk} 和 B_j 的值，所需要的全部量子线路如图4.9所示。按照以上描述的过程，我们对 $L = 2$ 的非厄米格点中的波函数的时间演化进行模拟，得到的结果如图4.4所示，其中利用变分量子算法得到的结果与利用经典龙格-库塔法得到的数值结果进行了对比，两者显示出了较好的一致性。两组模拟中分别取 $t_1 = 0.4$ 和 $t_1 = 1.0$ ，其他参数固定为 $t_2 = 0.5, \gamma = 0.8$ 。当 $t_1 = 0.4$ 时，粒子的逃逸概率 $P_{x=1}$ 远大于 $P_{x=2}$ ，即此时体系中存在边缘突起现象；而 $t_1 = 1.0$ 时则没有边缘突起现象，很好地符合了预期效果。



每个线路中，蓝色的线路或量子门是与辅助比特相关的部分，黑色的部分只与工作比特相关。各个线路对应的所标注的待测量值都是通过线路结尾测量量子比特得到 0 和 1 的概率进行计算得到的。其中，(j)-(y) 对应的测量值仅对应哈密顿量中的一个泡利项。

图 4.9 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系时，测量 A_{jk} 和 B_j 所需要的量子线路

进一步，我们对比了利用 PITE 算法和变分量子算法两种不同的方法得到的模拟演化结果。我们以经典龙格-库塔方法得到的结果为基准，定义量子算法模拟演化的误差： $\epsilon = 1 - |\langle \bar{\psi}_{\text{QA}}(\tau) | \bar{\psi}_{\text{RK}}(\tau) \rangle|^2$ ，其中 $|\bar{\psi}_{\text{QA}}(\tau)\rangle$ 是利用 PITE 算法或变分量子算法得到的归一的演化态， $|\bar{\psi}_{\text{RK}}(\tau)\rangle$ 是利用经典龙格-库塔方法得到的归一的演化态。图 4.11 中展示了两种不同的量子算法模拟 $L = 2$ 的非厄米格点体系得到的演



(a) $t_1 = 0.4$, 存在边缘突起现象。(b) $t_1 = 1.0$, 不存在边缘突起现象。两组模拟中的共同参数为 $t_2 = 0.5, \gamma = 0.8$ 。图中的散点是利用变分量子算法得到的结果, 实线是利用经典算法龙格-库塔方法得到的结果。

图 4.10 利用变分量子算法模拟 $L = 2$ 的非厄米格点体系演化及边缘突起现象

化态的误差, 其中在无论是否存在边缘突起的情形下, 两种量子算法得到的演化态的误差都不超过 0.3%, 说明了两种算法在模拟非厄米体系动力学演化问题中的有效性。

4.4 本章小结

本章我们探究了两种不同的量子算法在非厄米体系动力学演化模拟中的应用。第一种方法中, 我们利用 Trotter 分解可以将非厄米体系的演化分解为幺正演化和虚时演化, 并将上一章所描述的 PITE 算法应用于其中的虚时演化部分, 从而进行

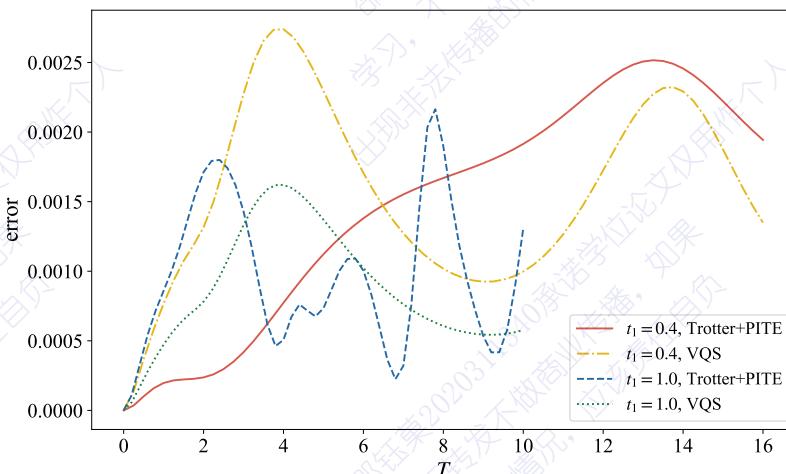


图 4.11 利用 PITE 算法与利用变分量子算法模拟 $L = 2$ 的非厄米格点体系得到演化态的误差

模拟。第二种方法中，我们通过非厄米体系归一化量子态演化方程，给出了一种不受全局相位影响的变分量子算法的参数演化方程。两种方法都能高效对非厄米格点体系中出现的边缘突起效应进行模拟，并得到符合预期的结果，由此证明了 PITE 算法与变分量子算法在非厄米物理体系的研究中所具有的重要应用价值。

第5章 量子稀疏成像算法

在第三、四章中，我们介绍了非幺正量子算法在量子多体系统模拟问题中的应用。本章节中，我们将研究非幺正量子算法在另一类实用问题中的应用，即稀疏成像问题。稀疏成像是近年来快速发展的一种成像技术，在医学、遥感、探测等领域具有广泛应用。基于经典算法中能实现稀疏成像的迭代收缩阈值算法，我们提出了一种实现稀疏成像的非幺正量子算法，并在演示实验和数值模拟中实现了算法在图像处理问题中的应用。

5.1 稀疏成像问题及相关算法

稀疏成像的基本原理是利用投影数据进行重建^[268]。在稀疏成像中，一般采用X射线、光学、超声、电子显微镜等成像方式，将样品在不同角度下的投影数据收集起来，然后通过重建算法将这些数据还原成三维空间内的物体，有时也只收集并重建物体的二维信息。稀疏成像的稀疏性体现在成像的大部分区域是不包含信息的背景，因此可以通过一些手段降低所需要处理的数据的规模，从而实现高效的成像。常用的稀疏成像算法包括两类：一类是基于压缩感知的稀疏重建算法，其基本思想是通过收集物体的投影数据，利用合适稀疏基进行数据压缩，降低信号的维数，并利用压缩后的数据进行重建。常用的压缩基有小波基、离散余弦基等。另一类是基于模型的稀疏成像算法，其基本思想是通过建立物体的数学模型，对物体进行建模和优化。常用的优化方法有最小二乘法、非负矩阵分解等。

大规模数据的稀疏成像主要面临如下方面的困难：(1) 计算量大。大规模数据稀疏成像往往需要采用一些复杂的算法和优化方法，如基于压缩感知的算法、迭代重构算法、正则化技术等，来处理大规模数据并获得高质量的成像结果。此外，高维度和稀疏性数据的特点也会导致计算量增加，例如基于压缩感知的算法需要进行大量的矩阵计算和向量计算。这些计算方法需要消耗的资源巨大，需要高性能计算平台和大规模分布式计算系统的支持。(2) 噪声和误差。数据所受到噪声和误差会对成像结果产生不良影响，因此需要采用一些优化方法，如正则化技术、先验知识等，来减小噪声和误差对成像结果的影响。(3) 数据获取困难。对于大规模数据，需要大量的时间和资源来获取完整的数据集。因此需要寻找合适的数据采集和处理方法，以提高数据获取的效率和精度。(4) 模型选择和参数调节。稀疏成像的模型选择和参数调节非常重要，直接影响成像的效果和精度。随着数据规模的增大，优化和调节会更加复杂，模型选择和参数调节的难度也会增大。面

对以上这些困难和挑战，需要采用先进的计算技术和优化方法来提高计算效率和精度，以实现高效、准确的稀疏成像。

稀疏成像过程可以用以下公式描述：

$$\mathbf{b} = \mathbf{Ax} + \mathbf{n}, \quad (5.1)$$

其中 \mathbf{x} 是表示原始图像的向量， \mathbf{A} 是成像变换矩阵或采样矩阵， \mathbf{n} 是成像过程中的噪声和误差， \mathbf{b} 是变换或采样后得到的数据。在稀疏成像问题中， \mathbf{A} 和 \mathbf{b} 是已知的， \mathbf{n} 一般是小量，而成像的过程就是通过 \mathbf{A} 和 \mathbf{b} 恢复出 \mathbf{x} ，可以利用最小二乘法来求解：

$$\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2, \quad (5.2)$$

其中 $\|\cdot\|_2$ 代表矩阵或向量的 2 范数（后面出现的所有形如 $\|\cdot\|_q$ 的符号都表示矩阵或向量的 q 范数）。若采样次数大于 \mathbf{x} 的维度，即过采样情形，则通过求解该方程一般可以很好地恢复出 \mathbf{x} 。若采样次数小于 \mathbf{x} 的维度，即欠采样情形，则方程的解不唯一，单凭此方程无法得到真实的 \mathbf{x} ；此时若已知 \mathbf{x} 是稀疏的，且成像矩阵 \mathbf{A} 满足约束等距性（restricted isometry property, RIP）条件，则可以转化为求解以下优化问题来很好地恢复出 \mathbf{x} ：

$$\min_{\mathbf{x}} \|\mathbf{x}\|_q, \quad \text{s.t. } \mathbf{b} = \mathbf{Ax}. \quad (5.3)$$

在不同的应用场景中， q 的选择不同，算法的效果与适用性也不同。通常可以选择 $q = 0$ ，此时向量的 0 范数等于非零元素的个数。然而 $q = 0$ 时该问题无法在经典计算中高效求解，但在一定条件下，可以将其转化为 $q = 1$ 的优化问题，此时向量的 1 范数等于所有元素复模长之和。通过拉格朗日乘子法，可以将 $q = 1$ 的优化问题转化为最小完全收缩和选择算符（least absolute shrinkage and selection operator, LASSO）问题的等价形式^[269-270]，即

$$\min_{\mathbf{x}} (\|\mathbf{b} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_1), \quad (5.4)$$

其中拉格朗日乘子 λ 也称为正则化参数。

在经典算法中，解决 LASSO 问题最常用的算法叫做迭代收缩阈值算法（iterative shrinkage thresholding algorithm, ISTA）^[271]。该方法将待优化函数中的两部分分别定义为 $F(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|_2^2$ 和 $G(\mathbf{x}) = \lambda \|\mathbf{x}\|_1$ ，并进行如下步骤：(1) 选定初始向量 $\mathbf{x}^{(0)}$ ；(2) 对 $F(\mathbf{x})$ 进行梯度下降，得到 $\mathbf{x}_g^{(k)} = \mathbf{x}^{(k)} - \xi F'(\mathbf{x}^{(k)}) = \mathbf{x}^{(k)} - \xi \mathbf{A}^\dagger (\mathbf{Ax}^{(k)} - \mathbf{b})$ ，其中 ξ 为梯度下降速率；(3) 对 $\mathbf{x}_g^{(k)}$ 进行阈值收缩操作得到 $\mathbf{x}^{(k+1)}$ ，阈值收缩操作

是对 $\mathbf{x}_g^{(k)}$ 中的每个分量进行如下变换：

$$x_i \rightarrow \begin{cases} x_i - \xi\lambda, & x_i > \xi\lambda, \\ 0, & |x_i| \leq \xi\lambda, \\ x_i + \xi\lambda, & x_i < -\xi\lambda. \end{cases} \quad (5.5)$$

(4) 重复第 2-3 步，直到 \mathbf{x} 达到收敛条件。

对于稀疏成像问题中凸优化迭代需要的海量数据处理问题，量子计算是一种有效的解决方案。然而目前利用量子算法解决稀疏成像问题仍是量子计算领域中的缺口。在公开资料中，空军工程大学等单位研究人员提出量子增强重构算法 (quantum-enhanced reconstruction algorithm, QRA)^[272]，可以实现图像稀疏重构，该方法在5.3式所示的优化问题中取 $q = 2$ ，并将其转化为如下无约束线性方程组：

$$(\eta \mathbf{A}^\dagger \mathbf{A} - \lambda_0 \mathbf{I}) \mathbf{x} = \mathbf{A}^\dagger \mathbf{b}, \quad (5.6)$$

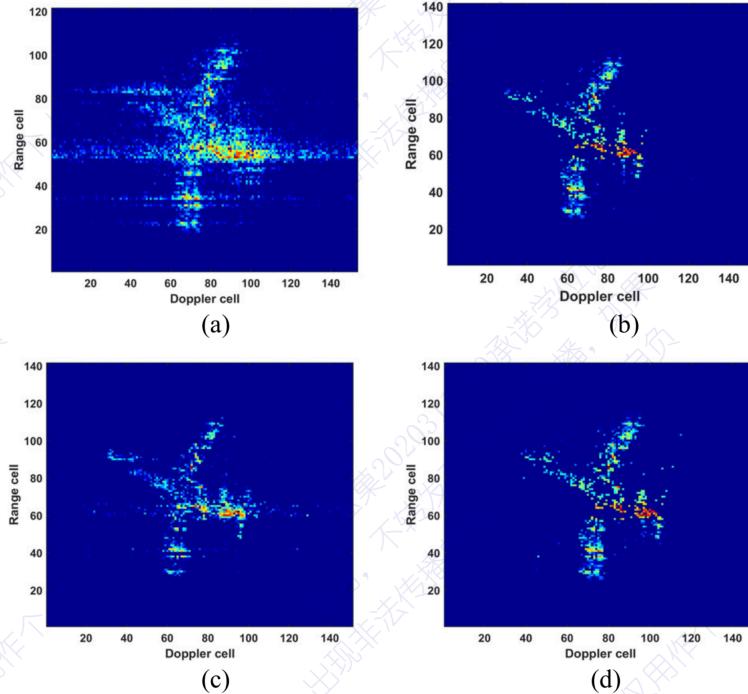
其中 η 是标度因子， λ_0 是正则化系数。令 $\Xi = \eta \mathbf{A}^\dagger \mathbf{A} - \lambda_0 \mathbf{I}$, $\gamma = \mathbf{A}^\dagger \mathbf{b}$ ，则上式重写为 $\Xi \mathbf{x} = \gamma$ ，即标准的线性方程组，进一步便可利用 HHL 算法对其进行求解。研究人员利用 QRA 对 Yak-42 飞机的采样图像进行稀疏重构，并将成像结果与一些常用经典算法的结果进行对比，其中包括距离-多普勒 (range-Doppler, RD) 算法、正交匹配追踪 (orthogonal matching pursuit, OMP) 算法和快速迭代收缩阈值算法 (fast ISTA, FISTA)，结果如图5.1所示。

QRA 能够较好地实现稀疏成像，但该方法所解决的是 $q = 2$ 时的优化问题，优化结果虽然对噪声有一定的抵抗，但其稀疏性只体现在稀疏采样过程，算法给出的解通常不具有稀疏性。若希望解稀疏，需要取 $q = 0$ 或 $q = 1$ ^[273]，而对于这类问题目前还没有对应的量子算法。本章后面的内容将介绍我们如何构造能够解决 $q = 1$ 时的优化问题的量子算法，并展示将其应用到稀疏成像问题的一些数值模拟和演示实验结果。

5.2 基于阈值迭代收缩算法的量子稀疏成像算法

基于经典算法中的迭代收缩阈值方法，我们设计了一种实现稀疏成像的量子算法，叫做量子迭代收缩阈值算法 (quantum ISTA, QISTA)，可以用来求解形如5.4式的凸优化问题。在 ISTA 的算法流程中有两个关键的迭代步骤，一个是梯度下降，另一个是阈值收缩。我们需要在 QISTA 中实现这两个关键步骤。

量子算法首先需要解决的问题是如何将经典向量储存在量子比特的状态中。一种最常用的储存经典向量的量子态编码方式叫做振幅编码，在2.5.1小节中介绍的 HHL 算法采用的就是这种方法。振幅编码以如下方式将经典向量 \mathbf{x} 编码在量子



(a) 距离-多普勒算法; (b) 正交匹配追踪算法; (c) 快速迭代收缩阈值算法; (d) 量子增强重构算法。

图 5.1 利用各类算法对 Yak-42 飞机进行稀疏成像的结果, 图片来自参考文献^[272]

态 $|\mathbf{x}\rangle$ 中:

$$\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T \rightarrow |\mathbf{x}\rangle = \frac{1}{\sqrt{|\mathbf{x}|^2}} \sum_{j=0}^{N-1} x_j |j\rangle. \quad (5.7)$$

振幅编码的缺点是, 量子态天然的归一性质使得经典向量的模长信息丢失, 只能保留各分量之间的相对比值。例如, 有两个经典向量 $\mathbf{x}_1 = (1, 2)$ 和 $\mathbf{x}_2 = (2, 4)$, 将它们进行振幅编码得到的是相同的量子态 $|\mathbf{x}_1\rangle = |\mathbf{x}_2\rangle = (|0\rangle + 2|1\rangle)/\sqrt{5}$, 这时只凭借量子态中的信息无法区分这两个经典向量。为了解决这一问题, 我们使用带修饰的振幅编码方式^[179]:

$$\mathbf{x} = (x_1, x_2, \dots, x_N)^T \rightarrow |\mathbf{X}\rangle = \cos \gamma \left(|0\rangle + \sum_{j=1}^N x_j |j\rangle \right). \quad (5.8)$$

其中 $\cos \gamma = 1/\sqrt{1 + |\mathbf{x}|^2}$, $0 \leq \gamma < \pi/2$ 。这种方式相当于定义新向量 $\mathbf{X} = (1, \mathbf{x})^T$, 再将 \mathbf{X} 用原来的振幅编码方式储存在量子态中。这样只需测量出量子态在 $|0\rangle$ 上的振幅, 即 $\cos \gamma$, 就可以恢复出原向量的模长 $|\mathbf{x}| = \tan \gamma$ 。

在带修饰的振幅编码下, 我们将利用量子算法实现 ISTA 中的梯度下降和阈值收缩两个关键步骤。首先, 对于梯度下降操作, 我们的第一种思路是采用 Pan Gao 等人于 2021 年提出的量子梯度算法^[179-180]。需要进行梯度操作的函数为 $F(\mathbf{x}) =$

$\|\mathbf{b} - \mathbf{Ax}\|_2^2$, 用 $\mathbf{X} = (1, \mathbf{x})^\top$ 表示则有 $F(\mathbf{x}) = \mathbf{X}^\dagger \mathbf{D} \mathbf{X}$, 其中

$$\mathbf{D} = \begin{bmatrix} \mathbf{b}^\dagger \mathbf{b} & -\mathbf{b}^\dagger \mathbf{A} \\ -\mathbf{A}^\dagger \mathbf{b} & \mathbf{A}^\dagger \mathbf{A} \end{bmatrix} \quad (5.9)$$

是一个厄米矩阵。由此得

$$\nabla_{\mathbf{x}} F = (\nabla_{\mathbf{x}} \mathbf{X}) (\nabla_{\mathbf{X}} F) = \mathbf{K} \mathbf{D} \mathbf{X}, \quad (5.10)$$

其中 $\mathbf{K} = \text{diag}(0, 1, 1, \dots) = \mathbf{I} - |0\rangle\langle 0|$ 为分量置零算符。因此, 梯度下降操作需要实现的变换为:

$$\mathbf{X} \rightarrow \mathbf{X}_g = \mathbf{X} - \xi \mathbf{K} \mathbf{D} \mathbf{X} = (\mathbf{I} - \xi \mathbf{K} \mathbf{D}) \mathbf{X}. \quad (5.11)$$

将以上变换对应到量子态的变换中, 由于量子态归一系数 $\cos \gamma$ 并不影响优化函数的极值点和梯度方向, 因此可以直接把经典向量映射到量子态上, 即 $|\mathbf{X}_g\rangle = (\mathbf{I} - \xi \mathbf{K} \mathbf{D}) |\mathbf{X}\rangle$ 。这一操作需要我们在量子线路中实现非幺正算符 $\mathbf{I} - \xi \mathbf{K} \mathbf{D}$, 其中算符 \mathbf{D} 可以通过相位估计方法实现, 算符 \mathbf{K} 可以通过对辅助比特进行受控翻转再进行测量后选择来实现, 而 \mathbf{I} 与 $\mathbf{K} \mathbf{D}$ 的线性组合则可以利用 LCU 来实现。

图5.2所示为实现量子梯度下降的量子线路图。线路中共包含 5 部分量子比特: 下标 U 表示实现 LCU 所需要的辅助比特, 下标 K 表示实现算符 \mathbf{K} 所需要的辅助比特, 下标 D 表示实现算符 \mathbf{D} 所需要的辅助比特, 下标 E 表示用来储存相位估计结果所需要的比特, 下标 V 表示储存向量 \mathbf{X} 的工作比特。图中橙色部分等效实现的是算符 \mathbf{D} , 其原理是2.5.1小节中所介绍的, 利用量子相位估计(或类 HHL 方法)实现矩阵函数, 这里的函数就是算符 \mathbf{D} 的线性函数 $f(\mathbf{D}) = \mathbf{D}$ 。图中的绿色部分等效实现的是算符 \mathbf{K} , 其具体形式为, 当下标为 V 的比特全部为 0 时, 对比特 K 进行翻转操作, 否则比特 K 保持不变, 这样当测量辅助比特 K 得到 0 时, 就等效于对下标为 V 的比特实现了 $\mathbf{K} = \mathbf{I} - |0\rangle\langle 0|$ 的作用效果。

对整个线路的过程进行分析: 量子比特的初始状态为 $|0\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E |\mathbf{X}\rangle_V$, 经过 $\mathbf{R}_y(\varphi)$ 操作后, 得到

$$\left(\cos \frac{\varphi}{2} |0\rangle + \sin \frac{\varphi}{2} |1\rangle \right)_U |0\rangle_K |0\rangle_D |0\rangle_E |\mathbf{X}\rangle_V. \quad (5.12)$$

经过线路中的橙色部分, 得到

$$\begin{aligned} & \cos \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E |\mathbf{X}\rangle_V + C \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E \mathbf{D} |\mathbf{X}\rangle_V \\ & + \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |1\rangle_D |\perp\rangle_E_V, \end{aligned} \quad (5.13)$$

其中我们用 $| \perp \rangle$ 来表示我们不关心的态。然后经过线路中的绿色部分，得到

$$\begin{aligned} & \cos \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E |\mathbf{X}\rangle_V + C \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E \mathbf{KD} |\mathbf{X}\rangle_V \\ & + \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |1\rangle_D |\perp\rangle_{EV} + \sin \frac{\varphi}{2} |1\rangle_U |1\rangle_K |\perp\rangle_{DEV}. \end{aligned} \quad (5.14)$$

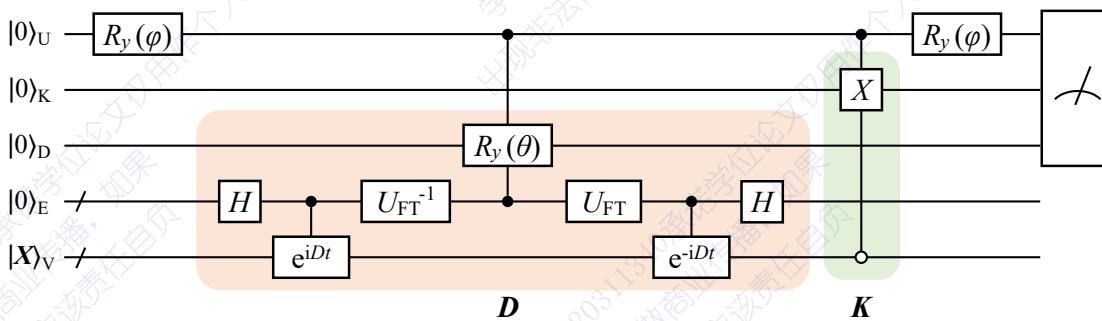
最后再经过 $\mathbf{R}_y(\varphi)$ 操作后，得到

$$\begin{aligned} & \cos^2 \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E |\mathbf{X}\rangle_V - C \sin^2 \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |0\rangle_E \mathbf{KD} |\mathbf{X}\rangle_V \\ & + |1\rangle_U |\perp\rangle_{KDEV}. \end{aligned} \quad (5.15)$$

在线路的末端，对辅助比特 U,K,D 进行测量，若测量均得到 0，即可得到量子态

$$\alpha \left(|\mathbf{X}\rangle_V - C \tan^2 \frac{\varphi}{2} \mathbf{KD} |\mathbf{X}\rangle_V \right). \quad (5.16)$$

通过设定 C 和 φ 使得 $C \tan^2(\varphi/2) = \xi$ ，即可在量子线路中实现梯度下降操作。

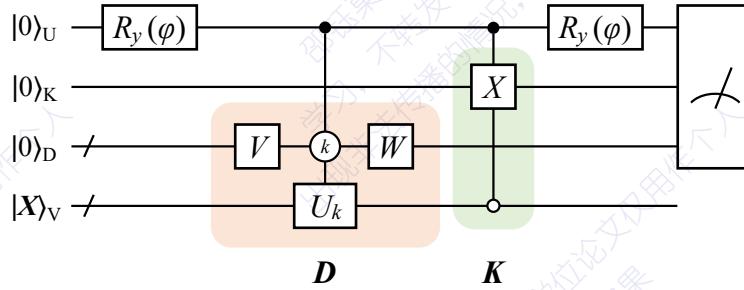


橙色部分等效为梯度算符 \mathbf{D} ，绿色部分等效为分量置零算符 \mathbf{K} 。整个线路利用了类 HHL 方法和 LCU 方法来实现。

图 5.2 基于相位估计方法实现量子梯度下降的线路图

以上所描述的量子梯度算法在被提出时，用来解决的是任意多项式函数的优化问题。对于大于二阶的多项式，等效梯度算符 \mathbf{D} 一定依赖于当前的量子态 $|\mathbf{X}\rangle$ 。如何构造一个依赖于量子态自身的算符并作用在自身上？这就需要采用 QPCA 方法^[200]。而 QPCA 所实现的是密度矩阵的演化算符，具有指数的形式，因此利用 QPCA 可以方便地构造出算符 $e^{-i\mathbf{D}t}$ ，进而可以通过相位估计实现算符 \mathbf{D} 。这也正是为什么上述量子梯度算法采用相位估计来实现梯度算符的原因。然而，在稀疏成像问题中，梯度下降所优化的目标函数 $F(\mathbf{x})$ 是向量 \mathbf{x} 的二阶多项式，对应到带修饰的振幅编码量子态 $|\mathbf{X}\rangle$ 上时，等效梯度算符 \mathbf{D} 是固定的，不依赖于量子态 $|\mathbf{X}\rangle$ ，因此在稀疏成像问题中构造算符 \mathbf{D} 并不需要复杂的 QPCA 方法，只需在算法开始时计算出 \mathbf{D} 并在量子线路中给出固定的构造方式，就可以在之后的算法过程中将 \mathbf{D} 当作黑盒来使用。

考虑到以上原因，同时也考虑到相位估计所需要的线路深度较大，不利于在量子计算机上高效地实现，这里我们提出了另外一种实现梯度下降的方法：利用



橙色部分等效为梯度算符 \mathbf{D} ，绿色部分等效为分量置零算符 \mathbf{K} 。

图 5.3 基于 LCU 方法实现量子梯度下降的线路图

LCU 方式构造梯度算符 \mathbf{D} 。LCU 作为一种常用的构造任意算符的通用方法，可以帮助我们在算法开始之前给出 \mathbf{D} 的固定构造方式。一般地，假设 \mathbf{D} 可以写成若干个较简单的幺正算符的线性组合： $\mathbf{D} = \sum_k \gamma_k \mathbf{U}_k$ ，则可以利用图5.3所示的量子线路来构造梯度算符 \mathbf{D} 并实现梯度下降，其中算符 \mathbf{V} 和 \mathbf{W} 分别对应 LCU 中的分波和合波操作。整个线路仅包含四部分量子比特。分析该线路的过程：量子比特的初始状态为 $|0\rangle_U |0\rangle_K |0\rangle_D |X\rangle_V$ 。经过 $\mathbf{R}_y(\varphi)$ 操作后，得到

$$\left(\cos \frac{\varphi}{2} |0\rangle + \sin \frac{\varphi}{2} |1\rangle \right)_U |0\rangle_K |0\rangle_D |X\rangle_V. \quad (5.17)$$

经过线路中的橙色部分，若 \mathbf{V} 和 \mathbf{W} 的矩阵形式满足 $\mathbf{W}_{0j} \mathbf{V}_{j0} = C \gamma_j$ ，则得到

$$\begin{aligned} & \cos \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |X\rangle_V + \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} \mathbf{W}_{kj} \mathbf{V}_{j0} |k\rangle_D \mathbf{U}_j |X\rangle_V \\ &= \cos \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |X\rangle_V + C \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |0\rangle_D \mathbf{D} |X\rangle_V + |1\rangle_U |0\rangle_K |\perp\rangle_D, \end{aligned} \quad (5.18)$$

这里同样用 $|\perp\rangle$ 表示不关心的态。然后经过线路中的绿色部分，得到

$$\cos \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |X\rangle_V + C \sin \frac{\varphi}{2} |1\rangle_U |0\rangle_K |0\rangle_D \mathbf{K} \mathbf{D} |X\rangle_V + |1\rangle_U |\perp\rangle_D. \quad (5.19)$$

最后再经过 $\mathbf{R}_y(\varphi)$ 操作后，得到

$$\cos^2 \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D |X\rangle_V - C \sin^2 \frac{\varphi}{2} |0\rangle_U |0\rangle_K |0\rangle_D \mathbf{K} \mathbf{D} |X\rangle_V + |1\rangle_U |\perp\rangle_D. \quad (5.20)$$

在线路的末端，对辅助比特 U,K,D 进行测量，若测量均得到 0，即可得到量子态

$$\alpha \left(|X\rangle_V - C \tan^2 \frac{\varphi}{2} \mathbf{K} \mathbf{D} |X\rangle_V \right). \quad (5.21)$$

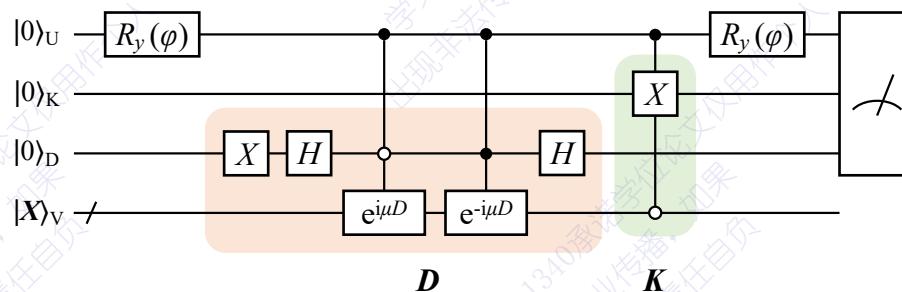
同样通过设定 C 和 φ 使得 $C \tan^2(\varphi/2) = \xi$ ，即可在量子线路中实现梯度下降操作。

以上两种实现量子梯度下降的方法分别利用相位估计方法和 LCU 方法实现梯

度算符 \mathbf{D} , 它们所实现的功能是相同的, 因此理论上可以随意选择其中一种。但考虑在实际应用中的实现效率和适用情况, 通常而言 \mathbf{D} 较为稀疏, 因此其演化算符 $e^{-i\mathbf{D}t}$ 很容易构造^[140-141], 具体而言, 若 \mathbf{D} 的维度为 N 、稀疏度为 s , 则构造 $e^{-i\mathbf{D}t}$ 的时间复杂度为 $\mathcal{O}(\log(N)s^2t)$ 。这样一看似利用相位估计方法更加合适, 但实际上, 相位估计所需要的线路深度更大, 且需要引入更多量子比特来储存相位估计值, 从而不利于实现。如果采用 LCU 方法, 则需要将 \mathbf{D} 写成较简单的幺正算符线性组合形式, 最常用且通用的方式是将 \mathbf{D} 写成泡利项之和, 但如果 \mathbf{D} 不具备特别的性质, 它所包含的系数非零的泡利项的个数可能是 4^n 量级, 其中 n 是储存经典向量的量子比特个数, 这样 LCU 方法将不具备与经典算法相比较的优势。因此这里我们给出了一种能够利用优势并避免劣势的方法: 根据 \mathbf{D} 的厄米性, 有如下公式^[189]:

$$\mathbf{D} = \lim_{\mu \rightarrow 0} \frac{e^{i\mu\mathbf{D}} - e^{-i\mu\mathbf{D}}}{2i\mu}, \quad (5.22)$$

当 μ 为接近 0 的实数时, 可以将 \mathbf{D} 近似写为两个幺正算符 $e^{i\mu\mathbf{D}}$ 和 $e^{-i\mu\mathbf{D}}$ 的线性组合, 其中 μ 称为梯度算符演化参数, 进而就可以利用 LCU 方法实现 \mathbf{D} 。这样一来, 既能利用 \mathbf{D} 的演化算符容易实现的这一特点, 也能避免相位估计带来过大的线路深度, 同时将实现 \mathbf{D} 所需的额外的比特数量减少到 1 个 (用于实现 LCU)。利用这样的方式, 我们重新给出基于 LCU 方法实现梯度下降的量子线路, 如图5.4所示。后文中在利用 QISTA 解决实际问题时, 如不加说明, 则都是使用图5.4所示的线路实现梯度下降过程。



橙色部分等效为梯度算符 \mathbf{D} , 绿色部分等效为分量置零算符 \mathbf{K} 。

图 5.4 基于 LCU 方法, 利用梯度算符的幺正演化实现量子梯度下降的线路图

经过梯度下降之后, 接下来我们需要在量子线路中实现阈值收缩操作。定义阈值收缩函数为 $s_\tau(x) = \text{sign}(x) \max(|x| - \tau, 0)$, 其中 τ 为阈值, sign 为符号函数 (若 x 为复数, 则 sign 为复相位因子)。对向量 \mathbf{x} 进行阈值收缩操作, 需要对 \mathbf{x} 的所有分量 x_i 都进行 s_τ 操作。在量子计算中, 阈值收缩可以利用 QSVT^[197] 来实现, 但是 QSVT 并不能直接用于带修饰振幅编码的量子态。在 QSVT 算法的基础上, 我

们进行了改进，并给出了针对带修饰振幅编码的量子态实现阈值收缩操作的量子算法和相应的量子线路。

对于量子态 $|X\rangle = \cos \gamma (|0\rangle + \sum_{i=1}^N x_i |i\rangle)$ ，我们首先要制备出具有如下密度矩阵的量子态：

$$\rho_{XX^\dagger} = \cos^2 \gamma \left(|0\rangle\langle 0| + \sum_{i=1}^N |x_i|^2 |i\rangle\langle i| \right). \quad (5.23)$$

制备该密度矩阵的量子线路如图5.5所示，其中需要额外引入相同数量的量子比特，并在各个对应的比特之间分别进行 CNOT 操作，这样得到的总量子态为 $\cos \gamma (|0\rangle|0\rangle + \sum_{i=1}^N x_i |i\rangle|i\rangle)$ 。此时如果只关注原有的这部分量子比特，即对新引入的量子比特子系统求迹，就可以得到如上式所示的密度矩阵。

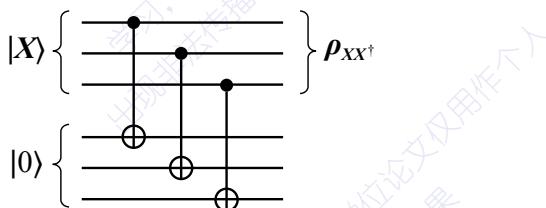


图 5.5 根据量子态 $|X\rangle$ 制备密度矩阵 ρ_{XX^\dagger} 的量子线路图（以三比特为例）

根据 QPCA 方法，有了具有密度矩阵 ρ_{XX^\dagger} 的量子态，我们就可以实现密度矩阵的幺正演化 $e^{i\rho_{XX^\dagger}t}$ ，进而可以通过相位估计算法，对密度矩阵的本征值，即各个 $|x_i|^2$ 的值进行估计，并利用类 HHL 方法实现密度矩阵的阈值收缩函数。需要注意的是，量子态 $|X\rangle$ 的 $|0\rangle$ 分量在经典向量中对应的是固定值 1，该分量不应该进行阈值收缩，因此在实现矩阵函数时不能对所有本征值进行阈值收缩，需要根据分量角标进行有条件的操作，因此我们需要在类 HHL 方法的基础上进一步改进。

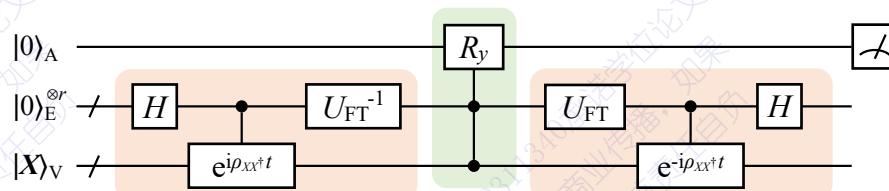


图 5.6 实现阈值收缩操作的量子线路图

利用量子算法对带修饰振幅编码的量子态实现阈值收缩的所需要的线路如图5.6所示，其中下标为 A 的是辅助比特，下标为 E 的是储存相位估计值的比特，下标为 V 的是储存向量 x 的工作比特。该线路图包含四部分：量子相位估计（左侧橙色部分）、受控旋转操作（中间绿色部分）、逆相位估计（右侧橙色部分）、测量后选择。其中与类 HHL 方法不同的是，受控旋转操作不仅要受控于相位估计比特（下标 E），还要受控于储存向量 X 的比特（下标 V），此外受控操作的具体形

式也与类 HHL 方法有所不同。下面对线路的四个过程进行具体分析。

1. 量子相位估计：经过相位估计操作后，线路中的总量子态为

$$\cos \gamma |0\rangle_A \left(|\tilde{\sigma}_0\rangle_E |0\rangle_V + \sum_{i=1}^N x_i |\tilde{\sigma}_i\rangle_E |i\rangle_V \right), \quad (5.24)$$

其中 $\tilde{\sigma}_0$ 是 $\cos^2 \gamma$ 的估计值， $\tilde{\sigma}_i$ 是 $\cos^2 \gamma |x_i|^2$ 的估计值。我们可以通过测量得到 $\cos^2 \gamma$ 的估计值，该值在下一步操作中会用到。

2. 受控旋转操作：受控操作所实现的具体变换形式为

$$|0\rangle_A |\tilde{\sigma}\rangle_E |i\rangle_V \rightarrow \begin{cases} |1\rangle_A |\tilde{\sigma}\rangle_E |0\rangle_V, & i = 0, \\ (\cos \theta_{\tilde{\sigma}} |0\rangle + \sin \theta_{\tilde{\sigma}} |1\rangle)_A |\tilde{\sigma}\rangle_E |i\rangle_V, & i > 0, \end{cases} \quad (5.25)$$

其中 $\sin \theta_{\tilde{\sigma}} = s_{\tau}, (\sqrt{\tilde{\sigma}})/\sqrt{\tilde{\sigma}}$, $\tau' = \tau \cos \gamma$ 。该受控旋转可以通过一系列 CNOT 门、Toffoli 门和单比特旋转门实现。则经过受控旋转后，线路中的量子态为

$$\begin{aligned} \cos \gamma & \left(|1\rangle_A |\tilde{\sigma}_0\rangle_E |0\rangle_V + \sum_{i=1}^N s_{\tau}(x_i) |1\rangle_A |\tilde{\sigma}_i\rangle_E |i\rangle_V \right. \\ & \left. + \sum_{i=1}^N \sqrt{1 - s_{\tau}^2(x_i)} |0\rangle_A |\tilde{\sigma}_i\rangle_E |i\rangle_V \right). \end{aligned} \quad (5.26)$$

3. 逆相位估计：经过逆相位估计后，下表为 E 的量子比特将全部变为 $|0\rangle$ 态，忽略这部分比特后，线路中的量子态变为

$$\cos \gamma \left(|1\rangle_A |0\rangle_V + \sum_{i=1}^N s_{\tau}(x_i) |1\rangle_A |i\rangle_V + \sum_{i=1}^N \sqrt{1 - s_{\tau}^2(x_i)} |0\rangle_A |i\rangle_V \right). \quad (5.27)$$

4. 测量辅助比特 A，若得到结果为 1，则工作比特的态为

$$\cos \gamma' \left(|0\rangle_V + \sum_{i=1}^N s_{\tau}(x_i) |i\rangle_V \right), \quad (5.28)$$

其中 $\cos \gamma'$ 为新的归一系数，对应阈值收缩后的经典向量的模长。至此我们就完成了对带修饰振幅编码的量子态进行阈值收缩的操作。

到此为止，我们就利用量子算法实现了 QISTA 中的两个关键步骤：梯度下降和阈值收缩。完整的 QISTA 的算法框架为：选定初始向量并将其编码到量子态中，通过不断地迭代梯度下降和阈值收缩这两个步骤，使量子态逐渐收敛，最终量子态中所编码的经典向量就对应稀疏成像的结果。如果稀疏成像的效果很好，最终得到的量子态应该是稀疏的，因此很容易通过态层析等测量方法恢复出其中的经典向量。

接下来，我们从理论上对 QISTA 的复杂度进行分析。其中，空间复杂度主要

是算法所需要的量子比特数，时间复杂度主要有两方面，一是算法需要的量子门的数量，二是算法的成功概率（称为测量复杂度）。我们针对量子梯度下降和量子阈值收缩这两个步骤分别进行讨论。

首先我们讨论梯度下降步骤。若我们采用相位估计方法实现梯度算符 \mathbf{D} ，则单步梯度下降过程的计算复杂度分析如下：

(1) 空间复杂度：储存 N 维经典向量 \mathbf{X} 所需要的量子比特数量为 $\log N$ ；当要求所实现的梯度下降的精度为 ϵ 时，用于存储相位估计值的比特数量为 $\mathcal{O}(\log(1/\epsilon))$ ；此外还需要 1 个辅助比特来实现梯度算符 \mathbf{D} 、1 个辅助比特来实现分量置零算符 \mathbf{K} 以及 1 个辅助比特用来实现 \mathbf{I} 与 \mathbf{KD} 的线性组合。共计空间复杂度 $\mathcal{O}(\log(N/\epsilon))$ 。

(2) 门复杂度：利用相位估计实现 \mathbf{D} 的过程与 HHL 算法类似，参考 HHL 算法的复杂度分析^[11]，当梯度算符 \mathbf{D} 具有稀疏度 s 和条件数 κ ，且要求精度为 ϵ 时，则实现 \mathbf{D} 的门复杂度为 $\mathcal{O}(\log(N)s^2\kappa^2/\epsilon)$ 。此外，实现算符 \mathbf{K} 所需的多比特受控翻转操作的复杂度为 $\mathcal{O}(\log(N))$ 。共计门复杂度 $\mathcal{O}(\log(N)s^2\kappa^2/\epsilon)$ 。

(3) 测量复杂度：由于梯度下降过程利用了非幺正量子线路，测量后选择过程的成功概率总小于 1。然而如文献^[179]中所述，可以通过设置迭代更新步长将成功概率限制于不小于 $3/16$ 的量级。因此单次迭代中，为了成功得到梯度下降后的量子态，所需测量复杂度可控制在 $16/3$ 这一常数量级。

若我们将梯度算符 \mathbf{D} 写成 $e^{i\mu\mathbf{D}}$ 与 $e^{-i\mu\mathbf{D}}$ 的线性组合，并采用 LCU 方法实现梯度算符 \mathbf{D} ，则单步梯度下降过程的计算复杂度分析如下：

(1) 空间复杂度：储存经典向量 \mathbf{X} 需要的量子比特数量为 $\log N$ ；此外还需要 1 个辅助比特来实现梯度算符 \mathbf{D} 、1 个辅助比特来实现分量置零算符 \mathbf{K} 以及 1 个辅助比特用来实现 \mathbf{I} 与 \mathbf{KD} 的线性组合。共计空间复杂度 $\mathcal{O}(\log N)$ 。

(2) 门复杂度：实现幺正算符 $e^{\pm i\mu\mathbf{D}}$ 的复杂度为 $\mathcal{O}(\log(N)s^2\mu)$ 。实现算符 \mathbf{K} 所需的多比特受控翻转操作的复杂度为 $\mathcal{O}(\log(N))$ 。共计门复杂度 $\mathcal{O}(\log(N)s^2\mu)$ 。

(3) 测量复杂度：由于 LCU 实际上实现的是算符 $\mu\mathbf{D}$ ，因此单次迭代的成功概率大约为 $\mathcal{O}(\mu)$ ，即成功得到一次测量结果需要 $\mathcal{O}(1/\mu)$ 的时间。结合前面所分析的门复杂度，单次迭代的时间复杂度为 $\mathcal{O}(\log(N)s^2)$ 。

接着我们讨论阈值收缩步骤。对于单步阈值迭代过程的计算复杂度分析如下：

(1) 空间复杂度：储存经典向量 \mathbf{X} 需要的量子比特数量为 $\log N$ ；当要求精度为 ϵ 时，利用 QPCA 方法实现幺正算符 $e^{i\rho_{XX^\dagger t}}$ 所需要的辅助比特数量为 $\mathcal{O}(\log(N)t^2/\epsilon)$ ；用于存储相位估计值的比特数量为 $\mathcal{O}(\log(1/\epsilon))$ ；此外还需要 1 个辅助比特来实现测量后选择。共计空间复杂度 $\mathcal{O}(\log(N)t^2/\epsilon)$ 。

(2) 门复杂度：当相位估计的精度为 ϵ 时，受控旋转操作引起的误差为

$\mathcal{O}(\epsilon\tau/|x_i|)$, 又由于阈值收缩的限制, 有 $|x_i| > \tau$, 因此总误差为 $\mathcal{O}(\epsilon)$, 与相位估计的误差为同一量级。此时相位估计所需要的幺正算符 $e^{i\rho_{XX^\dagger}t}$ 应满足 $t = \mathcal{O}(1/\epsilon)$, 同时又由于实现 $e^{i\rho_{XX^\dagger}t}$ 的门复杂度为 $\mathcal{O}(\log(N)t^2/\epsilon)$, 因此共计门复杂度 $\mathcal{O}(\log(N)/\epsilon^3)$ 。

(3) 测量复杂度: 实际应用中, 阈值收缩会使模较小的元素减小, 而对模较大的元素影响很小。该步骤的成功概率主要由梯度下降速率 ξ 和正则化参数 λ 决定, 在成像结果较好时, 成功概率一般维持在较高水平。后面我们在数值模拟中得到的结果也验证了这一点。

最后, 经过数次迭代, 量子态逐渐收敛并得到稀疏解, 然后对量子态进行测量以得到其中的经典信息。考虑到稀疏成像问题的解一定是稀疏的, 当量子态稀疏度为 s 时, 仅有 Ns 个基矢上存在概率分布, 因此当恢复经典向量的精度为 ϵ 时, 需要对量子态进行测量的次数为 $\mathcal{O}(Ns/\epsilon^2)$ 。

整体而言, QISTA 的空间复杂度为 $\mathcal{O}(\log(N)/\epsilon^3)$; 单次迭代的门复杂度为 $\mathcal{O}(\log(N)s^2\kappa^2/\epsilon + \log(N)/\epsilon^3)$ 或 $\mathcal{O}(\log(N)s^2 + \log(N)/\epsilon^3)$ 。由于稀疏成像需要对噪声进行抑制处理, 图像矩阵中的小量会在阈值收缩过程中被丢弃, 因此量子态的实现精度 ϵ 不需要很高。实现稀疏成像的经典算法的空间复杂度和单次迭代的门复杂度都是 $\mathcal{O}(N^2)$, 因此相比于经典算法, QISTA 在稀疏成像问题上实现了指数级超越。

5.3 量子稀疏成像算法在成像问题中的应用

为了进一步验证 QISTA 的性能, 我们利用数值模拟将算法应用于稀疏图像恢复的问题中, 并且在 Quafu 量子计算云平台上对算法进行了原理演示性的实验。

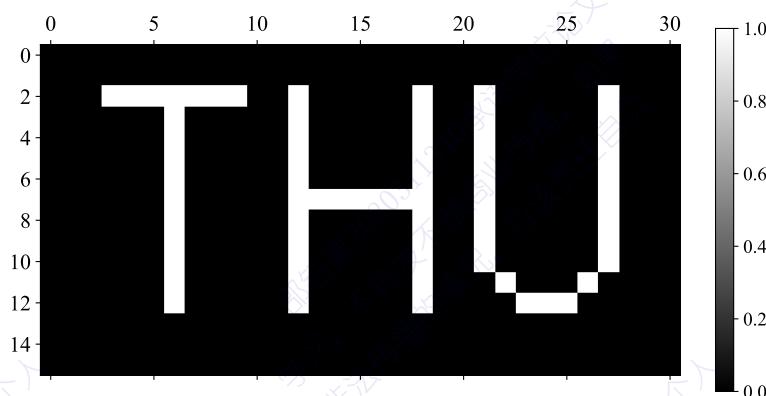


图 5.7 “THU”字样采样前的原始图像

数值模拟的第一个例子是利用 QISTA 对带有“THU”字样的图片进行稀疏成像。“THU”字样采样前的原始图像如图5.7所示, 它对应于一个 16×31 的矩阵, 其

中黑色像素对应的矩阵元为 0，白色像素对应的矩阵元为 1。我们将其看作 16 个 31 维的行向量，分别记为 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{16}$ 。首先，我们在计算机中模拟对该原始图像进行采样的过程，所使用的采样矩阵 \mathbf{A} 为傅立叶变换矩阵：

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{N-1} & \cdots & \omega^{(N-1)^2} \end{bmatrix}, \quad (5.29)$$

其中 $\omega = e^{i2\pi/N}$, $N = 31$ 。模拟采样过程按照 $\mathbf{b}_i = \mathbf{A}\mathbf{x}_i + \mathbf{n}_i$ 进行 ($i = 1, 2, \dots, 16$)，其中 \mathbf{n}_i 为随机噪声向量，每个像素点上的噪声符合以复平面原点为中心的二维正态分布。在根据 \mathbf{A} 和 \mathbf{b}_i 恢复 \mathbf{x}_i 时，一个最直接的方法是将 \mathbf{A} 的逆直接作用在 \mathbf{b}_i 上，得到的结果为 $\mathbf{x}'_i = \mathbf{A}^{-1}\mathbf{b}_i = \mathbf{x}_i + \mathbf{A}^{-1}\mathbf{n}_i$ ，相当于在原始图像上直接引入随机噪声。利用这种方式，我们可以对图像粗糙地进行恢复，其结果如图 5.8 所示。随着噪声方差的增大，图中的“THU”字样逐渐不可辨认，图像也逐渐失去稀疏性，从而会消耗更多的计算资源。

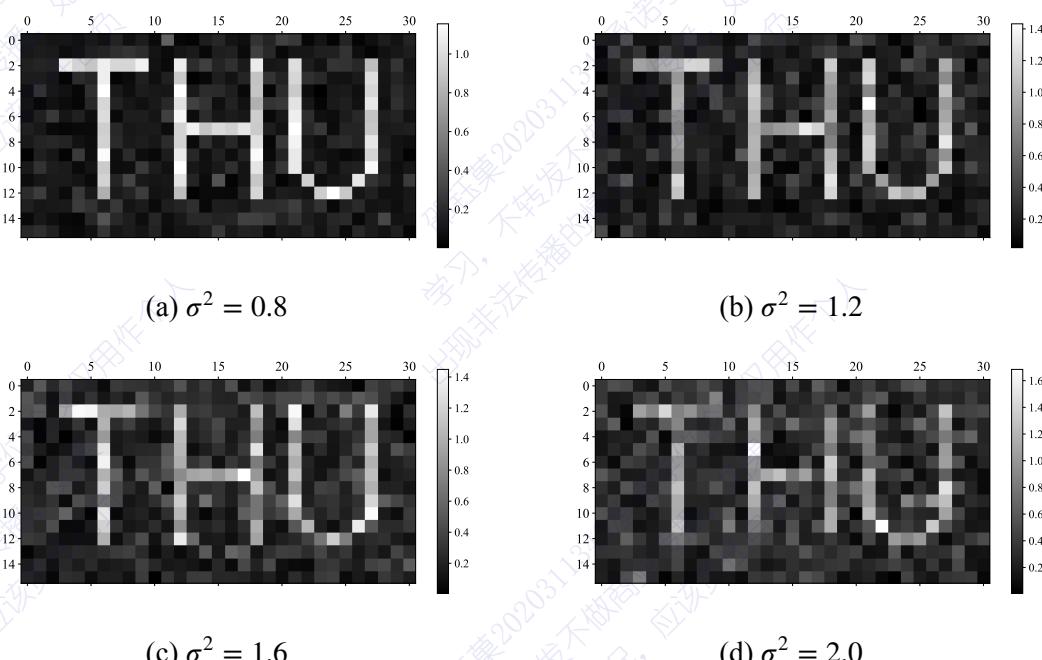
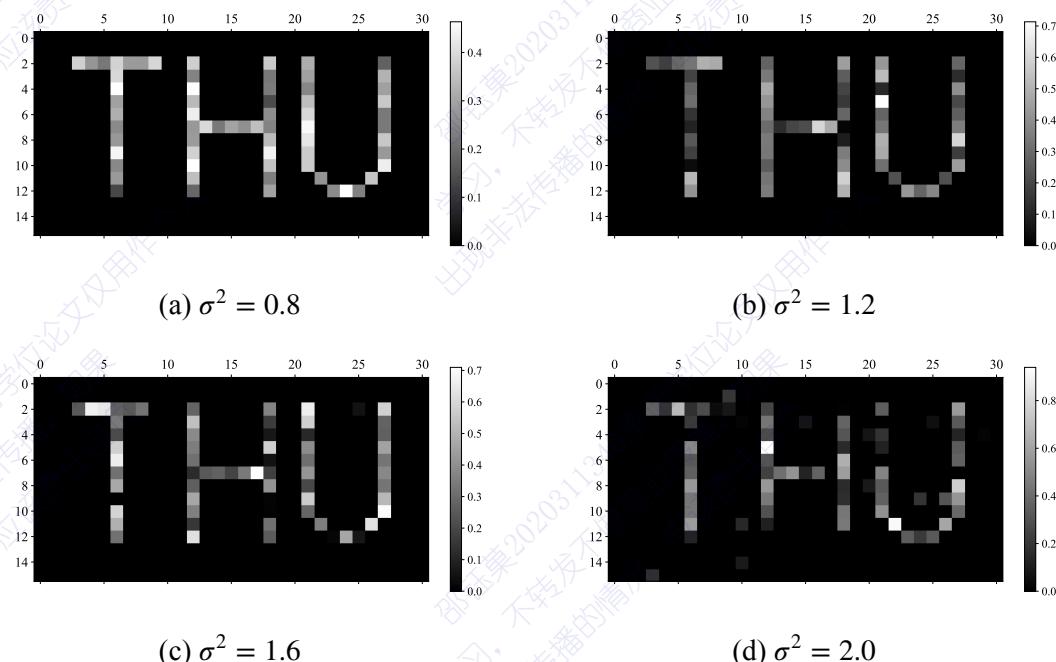


图 5.8 不同方差的噪声下，利用采样矩阵的逆直接恢复“THU”字样的图像

为了降低噪声的影响，使成像效果更好，我们需要考虑图像的稀疏性，通过稀疏成像的方法恢复出原始图像。这里我们将 QISTA 应用于该问题中。由于图像的行向量的维数为 $N = 31$ ，将行向量按照带修饰的振幅编码方式储存到量子态中时，需要至少 32 维的量子态。因此我们使用 5 个量子比特用来储存图像的每个行

向量。对于图像的每一行，我们需要单独进行运算。

在利用 QISTA 对以上图像进行稀疏成像的过程中，所有行向量的初始值都设为 $\mathbf{x}_i^{(0)} = (1, 0, 0, \dots)$ ，则初始量子态为 $|\mathbf{X}_i^{(0)}\rangle = (|00000\rangle + |00001\rangle)/\sqrt{2}$ 。选取正则化参数为 $\lambda = 20$ ，设定梯度算符的演化参数为 $\mu = 10^{-3}$ ，选取梯度下降速率为 $\xi = 5 \times 10^{-3}$ ，设定迭代次数为 15。按照 QISTA 的步骤，对量子算法所执行的过程进行数值模拟，最终得到的稀疏成像结果如图5.9所示。随着噪声方法的增大，“THU”字样逐渐产生部分失真，但能够清晰辨认，且图像能够保持较好的稀疏性。



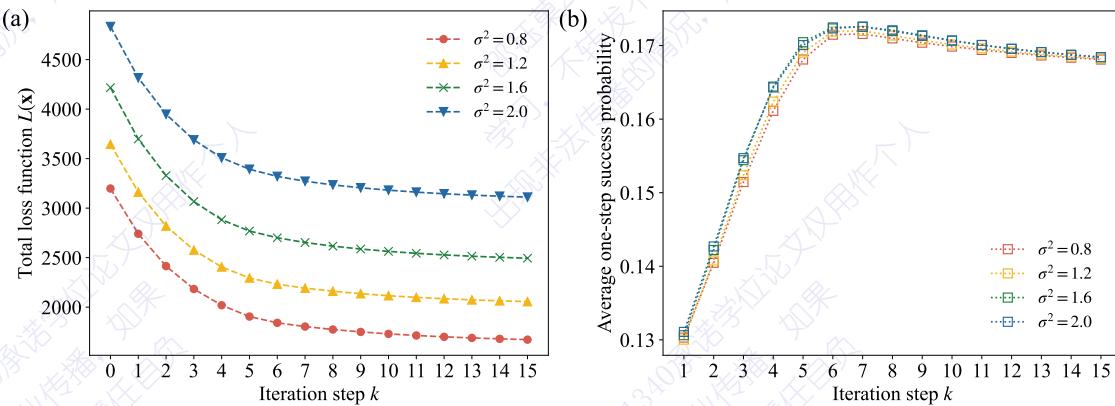
选取正则化参数为 $\lambda = 20$ ，梯度算符演化参数为 $\mu = 10^{-3}$ ，梯度下降速率为 $\xi = 5 \times 10^{-3}$ ，迭代次数为 15。

图 5.9 不同方差的噪声下，利用 QISTA 对“THU”字样进行稀疏成像的结果

在求解以上问题的迭代过程中，我们选取了两组数据进行了计算与分析，分别是图像的损失函数值和算法的单步成功概率。损失函数定义为

$$L(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_1, \quad (5.30)$$

也就是 LASSO 问题中需要寻找最小值点的目标函数。损失函数值的大小表明了成像效果的好坏，通常而言，损失函数值越低，成像结果与原始图像越接近，且成像结果的稀疏性更好。在算法迭代的过程中，我们计算每一步迭代结束后图像的各个行向量对应的损失函数值，再将各个行向量的损失函数值相加得到整个图像的总损失函数。图5.10(a)展示了 QISTA 迭代过程中图像的总损失函数值的变化，可以看出损失函数随迭代过程逐渐下降并收敛，且噪声方差越大，收敛的最终结果越高，这表明了噪声对成像效果的影响，即成像效果随着噪声方差的增大而下降。



选取正则化参数为 $\lambda = 20$, 梯度算符演化参数为 $\mu = 10^{-3}$, 梯度下降速率为 $\xi = 5 \times 10^{-3}$, 迭代次数为 15。(a) QISTA 迭代过程中损失函数值的变化。损失函数定义为 $L(\mathbf{x}) = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1$, 图像的总损失函数等于图像中各个行向量的损失函数之和。(b) QISTA 迭代过程中平均单步成功概率的变化。单步成功概率等于单次迭代中梯度下降的成功概率与阈值收缩的成功概率的乘积, 平均单步成功概率等于图像中各个行向量的单步成功概率的平均值。

图 5.10 利用 QISTA 对“THU”字样进行稀疏成像迭代过程中的损失函数值与单步成功概率

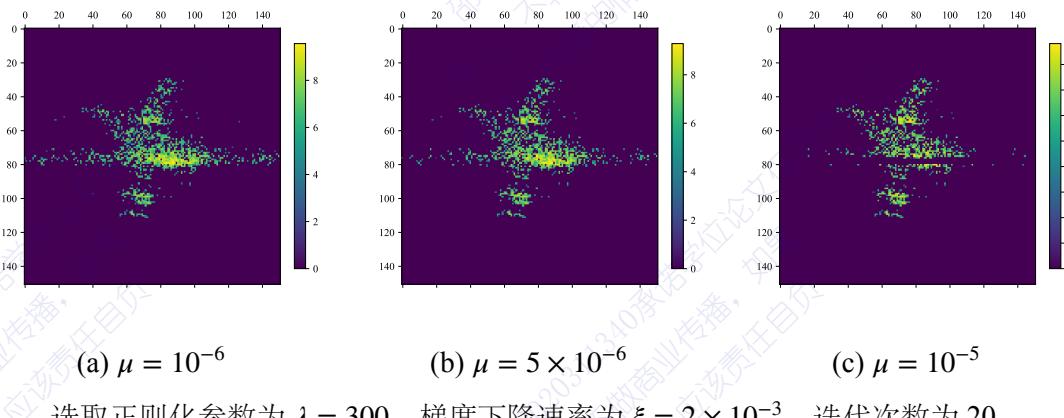
这一结论从成像结果的图像上也可以直观地感受到。

另一方面, QISTA 中梯度下降和阈值收缩都是通过测量后选择实现的, 我们也在模拟过程中探究了算法的成功概率。对于算法的单次迭代, 我们将梯度下降的成功概率和阈值收缩的成功概率相乘, 得到算法在单次迭代中的累积成功概率, 即单步成功概率。进一步, 我们将图像中各个行向量的单步成功概率进行平均, 得到整个图像的平均单步成功概率。图5.10(b)展示了 QISTA 迭代过程中图像的平均单步成功概率的变化, 其中可看出噪声方差对成功概率的影响很小。整体而言, 算法的单步成功概率可以基本稳定在常数量级。

数值模拟的第二个例子是文献^[272]中所使用的 Yak-42 飞机图像。在这个例子中, 我们并不知道采样前的原始图像, 只知道采样矩阵 \mathbf{A} 和采样后的结果 \mathbf{b} , 这一情况更接近稀疏成像算法的真实应用场景。Yak-42 飞机图像的采样结果为 151×151 的矩阵, 同上一个例子一样, 我们将其当作 151 个 151 维的行向量, 记为 \mathbf{b}_i ($i = 1, 2, \dots, 151$), 同时我们将待恢复的原始图像记为 \mathbf{x}_i ($i = 1, 2, \dots, 151$)。对于每一个行向量, 我们单独进行运算, 将其按照带修饰的振幅编码方式储存到量子态中时, 需要至少 152 维的量子态, 因此我们使用 8 个量子比特用来储存图像的每个行向量。

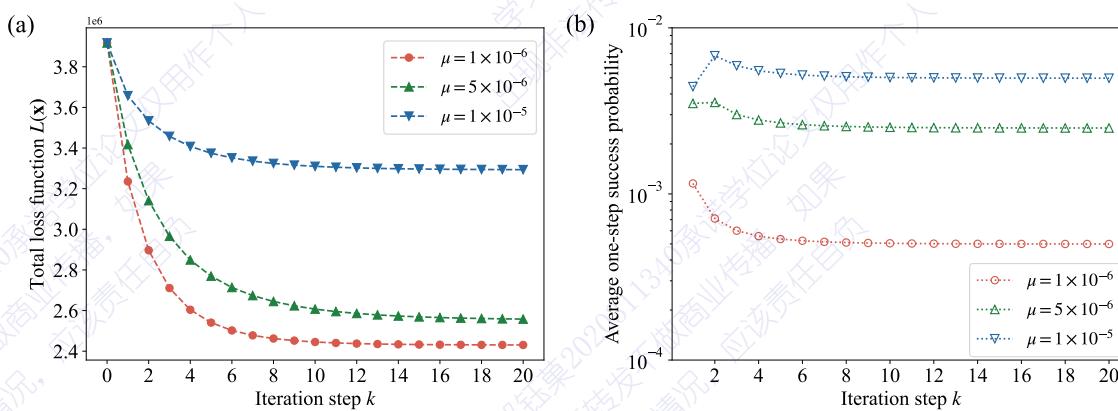
在利用 QISTA 对 Yak-42 飞机进行稀疏成像的过程中, 所有行向量的初始值都设为 $\mathbf{x}_i^{(0)} = (1, 0, 0, \dots)$, 则初始量子态为 $|\mathbf{X}_i^{(0)}\rangle = (|00000000\rangle + |00000001\rangle)/\sqrt{2}$ 。正则化参数与梯度下降速率选取为 $\lambda = 300, \xi = 2 \times 10^{-3}$, 并设定迭代次数为 20,

梯度算符的演化参数分别取 $\mu = 10^{-6}$, $\mu = 5 \times 10^{-6}$, $\mu = 10^{-5}$ 三个不同的值。按照 QISTA 的步骤, 对量子算法所执行的过程进行数值模拟, 最终得到的稀疏成像结果如图5.11所示。注意到, 随着 μ 的增大, 图像中央部分有若干行像素的亮度出现了缺失, 这是由于 μ 的增大使得利用 LCU 方法实现梯度算符 \mathbf{D} 的误差变大而导致的。此外, 我们也对 QISTA 迭代过程中图像的损失函数值和算法的单步成功概率进行了计算和分析, 结果如图5.12(a)和(b)所示, 其中, 随着 μ 的增大, 损失函数的收敛值逐渐升高, 说明成像效果变差, 这同样是由于实现梯度算符 \mathbf{D} 的误差变大而导致; 然而算法的单步成功概率会随着 μ 而升高, 使得算法的测量复杂度降低。



选取正则化参数为 $\lambda = 300$, 梯度下降速率为 $\xi = 2 \times 10^{-3}$, 迭代次数为 20。

图 5.11 利用 QISTA 对 Yak-42 飞机进行稀疏成像的结果



选取正则化参数为 $\lambda = 300$, 梯度下降速率为 $\xi = 2 \times 10^{-3}$, 迭代次数为 20。(a) QISTA 迭代过程中损失函数值的变化, 图像的总损失函数等于图像中各个行向量的损失函数之和。(b) QISTA 迭代过程中平均单步成功概率的变化, 单步成功概率等于单次迭代中梯度下降的成功概率与阈值收缩的成功概率的乘积, 平均单步成功概率等于图像中各个行向量的单步成功概率的平均值。

图 5.12 利用 QISTA 对 Yak-42 飞机进行稀疏成像迭代过程中的损失函数值与单步成功概率

接下来，我们希望能在真实的量子计算硬件中实现 QISTA 在稀疏成像问题中的应用。然而，QISTA 中所用到的子程序包括相位估计、LCU、QPCA、QSVD 等，这些方法普遍具有较深的量子线路层数，并且需要控制比特数较多的受控量子门，比特之间纠缠程度高，这些因素使得 QISTA 对量子硬件性能的要求远超现有的量子计算硬件的能力，即使在处理维度很小的数据时也是如此。为了能避免这些困难，将 QISTA 在量子云平台上实现，我们保留 QISTA 的核心原理，对 QISTA 的大部分流程进行了等效简化，使其能够在现有的量子计算硬件中发挥性能。

在实验中，我们选取了一个 3 维向量作为原始图像 \mathbf{x} ，并且选取了一个 3×2 的矩阵作为稀疏采样矩阵 \mathbf{A} ，则采样后的图像 $\mathbf{b} = \mathbf{Ax}$ 是一个 2 维向量。矩阵 \mathbf{A} 、向量 \mathbf{x} 和 \mathbf{b} 的具体值如下：

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} 0.08508329 & -0.15324513 & 1.06886209 \\ 0.40171687 & 0.38096134 & -0.83001098 \end{bmatrix}, \\ \mathbf{x} &= \begin{bmatrix} -2.93696394 & -1.27694084 & -0.8 \end{bmatrix}, \\ \mathbf{b} &= \begin{bmatrix} -0.90929126 & -1.00228429 \end{bmatrix}.\end{aligned}\quad (5.31)$$

在量子计算硬件中利用 QISTA 实现该稀疏成像问题时，我们使用两个量子比特来储存 3 维经典向量 \mathbf{x} 。演示实验方案如下：

1. 计算梯度算符 \mathbf{D} ，并利用基本量子门对 \mathbf{D} 进行简单构造，如图 5.13(a) 所示；
2. 在量子线路中，制备量子态 $|\mathbf{X}\rangle$ ，将算符 \mathbf{D} 作用在上面，得到梯度量子态 $|\mathbf{g}\rangle = \mathbf{D} |\mathbf{X}\rangle$ ，对 $|\mathbf{g}\rangle$ 进行量子态层析，将 $|\mathbf{g}\rangle$ 的信息储存在经典计算机中；
3. 在经典计算机中，对 $|\mathbf{g}\rangle$ 的首分量置零，相当于进行了分量置零算符 \mathbf{K} 的操作，得到 $|\mathbf{g}'\rangle = \mathbf{K} |\mathbf{g}\rangle$ ；
4. 在经典计算机中模拟变分量子态学习算法，得到可以表示 $|\mathbf{g}'\rangle$ 的经典参数 θ_g （后面将介绍变分量子态学习算法的基本原理）；
5. 在量子线路中，利用 LCU 方法，实现量子态 $|\mathbf{X}\rangle$ 与 $|\mathbf{g}'\rangle$ 的线性组合，从而实现梯度下降，得到量子态 $|\mathbf{X}_g\rangle$ ；并对 $|\mathbf{X}_g\rangle$ 进行量子态层析，将其信息储存在经典计算机中；
6. 在经典计算机中模拟变分量子态学习算法，得到可以表示 $|\mathbf{X}_g\rangle$ 的经典参数 θ_d ；
7. 在经典计算机中，计算对 $|\mathbf{X}_g\rangle$ 的阈值收缩操作所对应的变化量子态 $|\mathbf{s}\rangle$ ，并模拟变分量子态学习算法，得到可以表示 $|\mathbf{s}\rangle$ 的经典参数 θ_s ；
8. 在量子线路中，利用 LCU 方法，实现量子态 $|\mathbf{X}_g\rangle$ 与 $|\mathbf{s}\rangle$ 的线性组合，从而实现阈值收缩，得到本次迭代结束后的量子态 $|\mathbf{X}'\rangle$ ；并对 $|\mathbf{X}'\rangle$ 进行量子态层析，将其信息储存在经典计算机中；

9. 如果迭代还未终止，则在经典计算机中模拟变分量子态学习算法，得到可以表示 $|X'\rangle$ 的经典参数 θ_x ，用以下一步迭代初始时对 $|X'\rangle$ 进行重新制备；
10. 重复第 2-9 步，直至迭代终止。

该过程中需要使用变分量子态学习（variational quantum state learning, VQSL）算法 [230,274-275]，它是利用参数化量子线路重建出一个给定的未知量子态或目标量子态 $|\psi_t\rangle$ 。该方法通过定义损失函数 $C(\theta) = 1 - \langle\psi(\theta)|\psi_t\rangle\langle\psi_t|\psi(\theta)\rangle$ 并利用梯度下降等优化方法得到最优参数，则最优参数所对应的参数化量子态可以认为最接近于目标态。在本实验所使用的 VQSL 算法中，参数化量子线路的构造如图5.13(b) 所示。实验方案中除了初始的 $|X\rangle$ 之外，其余的量子态全部利用图5.13(b) 所示的参数化量子线路及 VQSL 算法给出的最优参数进行制备。

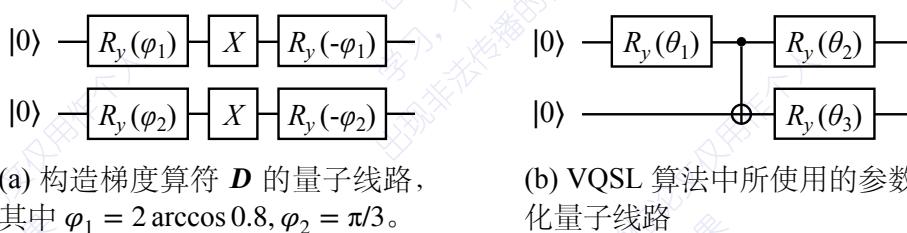


图 5.13 在 QISTA 演示实验中所使用的量子线路

在演示实验中，正则化参数选取 $\lambda = 10^{-3}$ ，设定迭代步数为 2，其中第一步迭代中的梯度下降速率为 $\xi_1 = 15$ ，第二步迭代中的梯度下降速率为 $\xi_2 = 0.7$ 。按照如上所描述的实验方案，我们在 Quafu 量子计算云平台上使用 P-10 量子芯片上进行演示实验。经过三次重复实验后得到的平均结果为： $\mathbf{x} = (1.46349835, 1.34006879, 1.16845978)$ 。从数值上看，该结果与原始图像之间差异较大，但我们在成像问题中主要关心像素点之间的相对亮度。我们将演示实验所得到的结果进行了可视化，并与原始图像进行了对比，如图5.14所示，其中，演示实验结果的可视化效果与原始图像非常接近。我们对图像的相对亮度进行处理，将两幅图像对应的向量 \mathbf{x} 中的元素分别取绝对值，进行归一化后，计算两个向量的内积，可以得到演示实验的结果与原始图像之间的相似度约为 91.35%，说明了 QISTA 在演示实验中可以得到较好的成像效果。

5.4 本章小结

本章中，我们基于经典算法中的迭代收缩阈值算法，设计了一种实现稀疏成像的量子算法，即量子迭代收缩阈值算法（QISTA）。该算法通过带修饰的振幅编码方式将经典向量储存在量子态中，并根据相位估计、LCU、QPCA、QSVD 等方法，构造非幺正量子线路，在量子线路中等效实现梯度下降与阈值收缩操作，从



图 5.14 QISTA 演示实验得到的成像结果与原始图像对比

而实现稀疏成像。从空间复杂度和时间复杂度上进行分析，QISTA 相比于已有的经典算法都能够实现指数级超越。

此外，在数值模拟中，我们利用 QISTA 对受噪声影响的“THU”字样图像与 Yak-42 飞机图像进行了稀疏成像，并探究了模拟过程中的损失函数与算法成功概率的变化。我们还在 Quafu 量子计算云平台上，以一个 3 维向量的恢复问题为例，对 QISTA 的原理和过程进行了演示实验。

量子计算硬件中的演示实验已对 QISTA 做了等效的简化处理，效果仍然不尽如人意，这是由于现有的量子计算硬件性能的限制所导致的。尽管如此，理论分析和数值模拟结果都说明了 QISTA 在处理稀疏成像问题上的可行性与高效性，展现了其在稀疏成像问题中具有极大的应用潜力。

第6章 总结与展望

量子计算自被提出四十余年来，理论体系逐渐丰富。量子计算被认为具有超越经典计算的巨大优越性，尽管没有严格的理论证明这一点，但目前已发展出的量子算法在计算复杂度上都优于解决同问题的经典算法。在量子计算的理论体系不断发展的同时，人们在实验上也取得了巨大进展，包括2019年以来人们在超导、光学等量子计算物理体系中相继实现量子霸权。量子计算开始朝向实用化迈进，人们也越来越期待量子计算为现代信息技术带来革命性突破。因此，发展更高效的量子算法、从理论基础上充分发挥量子优越性变得越来越紧迫。

本文中，我们对基于非幺正量子线路的量子算法与量子模拟开展了研究。第一章中我们首先回顾了量子力学的基础知识，简要介绍了量子计算的理论概念与实现方法。第二章中，我们介绍了几类重要且基础的量子算法，并介绍了复杂度的概念，其中的哈密顿量演化、变分量子算法、相位估计、LCU、线性系统求解等内容为后文所介绍的研究工作提供了重要的理论基础。

在第三章中，我们构造了一种基于非幺正量子线路的概率性虚时演化算法(PITE)，通过理论分析与推导给出了算法的复杂度。通过在超导和离子阱量子计算云平台上的演示实验以及在经典计算机上的数值模拟，我们验证并研究了该算法在求解量子多体系统基态问题上的应用。通过对该算法进行理论推广，我们使算法能适用于具有非常广泛的哈密顿量形式的量子系统，并在数量级上降低了算法的复杂度，使该算法在实际应用中具有更高的效率。

在第四章中，我们将虚时演化算法与变分量子算法应用于非厄米量子体系的动力学演化问题中。我们强调了虚时演化算法在任意非厄米体系的量子模拟中的普适性。同时，我们根据非厄米体系中归一化纯态的演化方程，通过量子态内积进行推导给出了不受全局相位影响的变分算法参数演化方程。通过数值模拟，我们利用两种量子算法来计算非厄米体系中的边缘突起效应，验证了算法在实际应用中的可行性。

在第五章中，我们构造了一种基于迭代收缩阈值方法的量子稀疏成像算法，通过理论分析与推导给出了算法的复杂度。我们在量子计算云平台上，对等效简化的算法进行了原理性演示，验证了算法在原理上的可行性。我们通过在经典计算机上的数值模拟，验证并研究了该算法在稀疏图像恢复问题上的应用。

本文的研究内容作为对量子算法理论体系的补充，能够在未来量子计算机解决实际问题时发挥一定程度上的作用。然而，本论文的研究内容中在某些方面仍

需要进一步的研究。例如，我们在设计量子算法的过程中，通常关心如何从初始量子态出发、最终得到末量子态，而如何根据所输入的经典数据高效地制备初始量子态、以及如何从末量子态中高效地获得经典的输出结果，在本文中没有进行深入探讨。另外，基于非幺正量子线路的量子算法一定面临着成功概率的问题，而本论文所研究的 PITE 算法与量子稀疏成像算法都是迭代性非幺正算法，算法的累积成功概率总是随着迭代次数呈指数增长。尽管本文所研究的 GPITE 算法可以在一定程度上缓解这一问题，但我们仍希望寻找一种合适的机制能够从更本质的原理上提高测量后选择的成功概率，甚至可以将概率性算法变为确定性算法。此外，本文所研究的算法已在小规模问题中进行了演示和验证，但这些算法在从理论模型走向实际应用时仍会遇到许多问题，如解决大规模问题时的误差影响与资源限制、不同类型的量子硬件对算法效果的影响等。这些都是值得探讨与关注的问题，在未来的进一步研究中将针对这些问题对算法进行改进和优化。

参考文献

- [1] Benioff P. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines[J]. *Journal of Statistical Physics*, 1980, 22: 563-591.
- [2] Benioff P. Quantum mechanical models of turing machines that dissipate no energy[J]. *Physical Review Letters*, 1982, 48(23): 1581.
- [3] Benioff P. Quantum mechanical hamiltonian models of turing machines[J]. *Journal of Statistical Physics*, 1982, 29: 515-546.
- [4] Manin Y I. Computable and uncomputable[J]. Sovetskoye Radio, Moscow, 1980, 128: 15.
- [5] Feynman R P. Simulating physics with computers[J]. *International Journal of Theoretical Physics*, 1982, 21(6/7).
- [6] Deutsch D, Jozsa R. Rapid solution of problems by quantum computation[J]. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 1992, 439(1907): 553-558.
- [7] Simon D R. On the power of quantum computation[J]. *SIAM Journal on Computing*, 1997, 26 (5): 1474-1483.
- [8] Grover L K. A fast quantum mechanical algorithm for database search[C]//*Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*. 1996: 212-219.
- [9] Shor P W. Algorithms for quantum computation: discrete logarithms and factoring[C]//*Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994: 124-134.
- [10] Kitaev A Y. Quantum measurements and the abelian stabilizer problem[A]. 1995.
- [11] Harrow A W, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations[J]. *Physical Review Letters*, 2009, 103(15): 150502.
- [12] Hallgren S. Fast quantum algorithms for computing the unit group and class group of a number field[C]//*Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*. 2005: 468-474.
- [13] Hallgren S. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem[J]. *Journal of the ACM (JACM)*, 2007, 54(1): 1-19.
- [14] Van Dam W, Shparlinski I E. Classical and quantum algorithms for exponential congruences [C]//*Theory of Quantum Computation, Communication, and Cryptography: Third Workshop, TQC 2008 Tokyo, Japan, January 30-February 1, 2008. Revised Selected Papers 3*. Springer, 2008: 1-10.
- [15] Chen J, Childs A M, Hung S H. Quantum algorithm for multivariate polynomial interpolation [J]. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 2018, 474(2209): 20170480.

参考文献

- [16] Ambainis A. Quantum walk algorithm for element distinctness[J]. SIAM Journal on Computing, 2007, 37(1): 210-239.
- [17] Wang Z G, Wei S J, Long G L. A quantum circuit design of aes requiring fewer quantum qubits and gate operations[J]. Frontiers of Physics, 2022, 17(4): 41501.
- [18] Abrams D S, Lloyd S. Simulation of many-body fermi systems on a universal quantum computer [J]. Physical Review Letters, 1997, 79(13): 2586.
- [19] Aspuru-Guzik A, Dutoi A D, Love P J, et al. Simulated quantum computation of molecular energies[J]. Science, 2005, 309(5741): 1704-1707.
- [20] Babbush R, Love P J, Aspuru-Guzik A. Adiabatic quantum simulation of quantum chemistry [J]. Scientific Reports, 2014, 4(1): 6603.
- [21] Gu S, Somma R D, Şahinoğlu B. Fast-forwarding quantum evolution[J]. Quantum, 2021, 5: 577.
- [22] Holmes Z, Muraleedharan G, Somma R D, et al. Quantum algorithms from fluctuation theorems: Thermal-state preparation[J]. Quantum, 2022, 6: 825.
- [23] Biamonte J, Wittek P, Pancotti N, et al. Quantum machine learning[J]. Nature, 2017, 549(7671): 195-202.
- [24] Schuld M, Sinayskiy I, Petruccione F. An introduction to quantum machine learning[J]. Contemporary Physics, 2015, 56(2): 172-185.
- [25] Moll N, Barkoutsos P, Bishop L S, et al. Quantum optimization using variational algorithms on near-term quantum devices[J]. Quantum Science and Technology, 2018, 3(3): 030503.
- [26] Li Y, Tian M, Liu G, et al. Quantum optimization and quantum learning: A survey[J]. Ieee Access, 2020, 8: 23568-23593.
- [27] Shor P W. Scheme for reducing decoherence in quantum computer memory[J]. Physical Review A, 1995, 52(4): R2493.
- [28] Knill E, Laflamme R, Viola L. Theory of quantum error correction for general noise[J]. Physical Review Letters, 2000, 84(11): 2525.
- [29] Bacon D. Operator quantum error-correcting subsystems for self-correcting quantum memories [J]. Physical Review A, 2006, 73(1): 012340.
- [30] Fowler A G, Mariantoni M, Martinis J M, et al. Surface codes: Towards practical large-scale quantum computation[J]. Physical Review A, 2012, 86(3): 032324.
- [31] Knapp C, Zaletel M, Liu D E, et al. The nature and correction of diabatic errors in anyon braiding[J]. Physical Review X, 2016, 6(4): 041003.
- [32] Bernstein E, Vazirani U. Quantum complexity theory[J]. SIAM Journal on Computing, 1997, 26(5): 1411-1473.
- [33] Cleve R. An introduction to quantum complexity theory[J]. Collected Papers on Quantum Computation and Quantum Information Theory, 2000: 103-127.
- [34] Brandão F G S L, Chemissany W, Hunter-Jones N, et al. Models of quantum complexity growth [J]. PRX Quantum, 2021, 2(3): 030316.
- [35] Clarke J, Wilhelm F K. Superconducting quantum bits[J]. Nature, 2008, 453(7198): 1031-1042.

参考文献

- [36] Cirac J I, Zoller P. Quantum computations with cold trapped ions[J]. Physical Review Letters, 1995, 74(20): 4091.
- [37] Loss D, DiVincenzo D P. Quantum computation with quantum dots[J]. Physical Review A, 1998, 57(1): 120.
- [38] O'Brien J L, Pryde G J, White A G, et al. Demonstration of an all-optical quantum controlled-not gate[J]. Nature, 2003, 426(6964): 264-267.
- [39] Nizovtsev A P, Kilin S Y, Jelezko F, et al. A quantum computer based on nv centers in diamond: optically detected nutations of single electron and nuclear spins[J]. Optics and Spectroscopy, 2005, 99: 233-244.
- [40] Cory D G, Price M D, Havel T F. Nuclear magnetic resonance spectroscopy: An experimentally accessible paradigm for quantum computing[J]. Physica D: Nonlinear Phenomena, 1998, 120 (1-2): 82-101.
- [41] Jones J A, Mosca M, Hansen R H. Implementation of a quantum search algorithm on a quantum computer[J]. Nature, 1998, 393(6683): 344-346.
- [42] Monz T, Nigg D, Martinez E A, et al. Realization of a scalable shor algorithm[J]. Science, 2016, 351(6277): 1068-1070.
- [43] Lee Y, Joo J, Lee S. Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience[J]. Scientific Reports, 2019, 9(1): 4778.
- [44] Zhang M, Dong L, Zeng Y, et al. Improved circuit implementation of the hhl algorithm and its simulations on qiskit[J]. Scientific Reports, 2022, 12(1): 13287.
- [45] Arute F, Arya K, Babbush R, et al. Quantum supremacy using a programmable superconducting processor[J]. Nature, 2019, 574(7779): 505-510.
- [46] Gong M, Wang S, Zha C, et al. Quantum walks on a programmable two-dimensional 62-qubit superconducting processor[J]. Science, 2021, 372(6545): 948-952.
- [47] Wu Y, Bao W S, Cao S, et al. Strong quantum computational advantage using a superconducting quantum processor[J]. Physical Review Letters, 2021, 127(18): 180501.
- [48] Zhu Q, Cao S, Chen F, et al. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling[J]. Science Bulletin, 2022, 67(3): 240-245.
- [49] Zhong H S, Wang H, Deng Y H, et al. Quantum computational advantage using photons[J]. Science, 2020, 370(6523): 1460-1463.
- [50] Zhong H S, Deng Y H, Qin J, et al. Phase-programmable gaussian boson sampling using stimulated squeezed light[J]. Physical Review Letters, 2021, 127(18): 180502.
- [51] Wu Y, Liu W, Geng J, et al. Observation of parity-time symmetry breaking in a single-spin system[J]. Science, 2019, 364(6443): 878-880.
- [52] Xu Y, Chu J, Yuan J, et al. High-fidelity, high-scalability two-qubit gate scheme for superconducting qubits[J]. Physical Review Letters, 2020, 125(24): 240503.
- [53] Neill C, McCourt T, Mi X, et al. Accurately computing the electronic properties of a quantum ring[J]. Nature, 2021, 594(7864): 508-512.
- [54] Wang C, Li X, Xu H, et al. Towards practical quantum computers: Transmon qubit with a lifetime approaching 0.5 milliseconds[J]. npj Quantum Information, 2022, 8(1): 3.

- [55] Shi S, Xu B, Zhang K, et al. High-fidelity photonic quantum logic gate based on near-optimal rydberg single-photon source[J]. *Nature Communications*, 2022, 13(1): 4454.
- [56] Li Z, Liu P, Zhao P, et al. Error per single-qubit gate below 10⁻⁴ in a superconducting qubit[J]. *npj Quantum Information*, 2023, 9(1): 111.
- [57] Shi Y H, Liu Y, Zhang Y R, et al. Quantum simulation of topological zero modes on a 41-qubit superconducting processor[J]. *Physical Review Letters*, 2023, 131(8): 080401.
- [58] Somoroff A, Ficheux Q, Mencia R A, et al. Millisecond coherence in a superconducting qubit [J]. *Physical Review Letters*, 2023, 130(26): 267001.
- [59] Xu S, Sun Z Z, Wang K, et al. Digital simulation of projective non-abelian anyons with 68 superconducting qubits[J]. *Chinese Physics Letters*, 2023, 40(6): 060301.
- [60] Julsgaard B, Sherson J, Cirac J I, et al. Experimental demonstration of quantum memory for light[J]. *Nature*, 2004, 432(7016): 482-486.
- [61] Lvovsky A I, Sanders B C, Tittel W. Optical quantum memory[J]. *Nature Photonics*, 2009, 3 (12): 706-714.
- [62] Hedges M P, Longdell J J, Li Y, et al. Efficient quantum memory for light[J]. *Nature*, 2010, 465(7301): 1052-1056.
- [63] Wallucks A, Marinković I, Hensen B, et al. A quantum memory at telecom wavelengths[J]. *Nature Physics*, 2020, 16(7): 772-777.
- [64] Dridi G, Liu K, Guérin S. Optimal robust quantum control by inverse geometric optimization [J]. *Physical Review Letters*, 2020, 125(25): 250403.
- [65] Debroux R, Michaels C P, Purser C M, et al. Quantum control of the tin-vacancy spin qubit in diamond[J]. *Physical Review X*, 2021, 11(4): 041041.
- [66] Wang Y, Haze M, Bui H T, et al. Universal quantum control of an atomic spin qubit on a surface [J]. *npj Quantum Information*, 2023, 9(1): 48.
- [67] Degen C L, Reinhard F, Cappellaro P. Quantum sensing[J]. *Reviews of Modern Physics*, 2017, 89(3): 035002.
- [68] Ho K O, Wong K C, Leung M Y, et al. Recent developments of quantum sensing under pressurized environment using the nitrogen vacancy (nv) center in diamond[J]. *Journal of Applied Physics*, 2021, 129(24).
- [69] Marciniak C D, Feldker T, Pogorelov I, et al. Optimal metrology with programmable quantum sensors[J]. *Nature*, 2022, 603(7902): 604-609.
- [70] Preskill J. Quantum computing in the nisq era and beyond[J]. *Quantum*, 2018, 2: 79.
- [71] Bharti K, Cervera-Lierta A, Kyaw T H, et al. Noisy intermediate-scale quantum algorithms[J]. *Reviews of Modern Physics*, 2022, 94(1): 015004.
- [72] Callison A, Chancellor N. Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond[J]. *Physical Review A*, 2022, 106(1): 010101.
- [73] 张永德. 量子力学[M]. 科学出版社, 2002.
- [74] Griffiths D J, Schroeter D F. Introduction to quantum mechanics[M]. Cambridge university press, 2018.

参考文献

- [75] Sakurai J J, Napolitano J. Modern quantum mechanics[M]. Cambridge University Press, 2020.
- [76] Schrödinger E. An undulatory theory of the mechanics of atoms and molecules[J]. Physical Review, 1926, 28(6): 1049.
- [77] Frommer A, Simoncini V. Matrix functions[M]//Model Order Reduction: Theory, Research Aspects and Applications. Springer, 2008: 275-303.
- [78] Nielsen M A, Chuang I L. Quantum computation and quantum information: volume 2[M]. Cambridge university press Cambridge, 2001.
- [79] Low G H, Chuang I L. Optimal hamiltonian simulation by quantum signal processing[J]. Physical Review Letters, 2017, 118(1): 010501.
- [80] Lehtovaara L, Toivanen J, Eloranta J. Solution of time-independent schrödinger equation by the imaginary time propagation method[J]. Journal of Computational Physics, 2007, 221: 1.
- [81] Dong Y, L L, Tong Y. Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices[J]. PRX Quantum, 2022, 3(4): 040305.
- [82] Martyn J M, Rossi Z M, Tan A K, et al. Grand unification of quantum algorithms[J]. PRX Quantum, 2021, 2(4): 040203.
- [83] Terashima H, Ueda M. Nonunitary quantum circuit[J]. International Journal of Quantum Information, 2005, 3(04): 633-647.
- [84] DiVincenzo D P. The physical implementation of quantum computation[J]. Fortschritte der Physik: Progress of Physics, 2000, 48(9-11): 771-783.
- [85] Makhlin Y, Schön G, Shnirman A. Josephson-junction qubits and the readout process by single-electron transistors[A]. 1998.
- [86] Blais A, Huang R S, Wallraff A, et al. Cavity quantum electrodynamics for superconducting electrical circuits: An architecture for quantum computation[J]. Physical Review A, 2004, 69 (6): 062320.
- [87] You J Q, Nori F. Superconducting circuits and quantum information[J]. Physics Today, 2005, 58(11): 42-47.
- [88] Wendin G, Shumeiko V S. Quantum bits with josephson junctions[J]. Low Temperature Physics, 2007, 33(9): 724-744.
- [89] DiCarlo L, Chow J M, Gambetta J M, et al. Demonstration of two-qubit algorithms with a superconducting quantum processor[J]. Nature, 2009, 460(7252): 240-244.
- [90] Geerlings K, Leghtas Z, Pop I M, et al. Demonstrating a driven reset protocol for a superconducting qubit[J]. Physical Review Letters, 2013, 110(12): 120501.
- [91] Barends R, Kelly J, Megrant A, et al. Coherent josephson qubit suitable for scalable quantum integrated circuits[J]. Physical Review Letters, 2013, 111(8): 080502.
- [92] Kockum A F, Nori F. Quantum bits with josephson junctions[M]//Fundamentals and Frontiers of the Josephson Effect. Springer, 2019: 703-741.
- [93] Sørensen A, Mølmer K. Quantum computation with ions in thermal motion[J]. Physical Review Letters, 1999, 82(9): 1971.

参考文献

- [94] Milburn G J, Schneider S, James D F V. Ion trap quantum computing with warm ions[J]. *Fortschritte der Physik: Progress of Physics*, 2000, 48(9-11): 801-810.
- [95] Kielpinski D, Monroe C, Wineland D J. Architecture for a large-scale ion-trap quantum computer[J]. *Nature*, 2002, 417(6890): 709-711.
- [96] Bruzewicz C D, Chiaverini J, McConnell R, et al. Trapped-ion quantum computing: Progress and challenges[J]. *Applied Physics Reviews*, 2019, 6(2).
- [97] Burkard G, Loss D, DiVincenzo D P. Coupled quantum dots as quantum gates[J]. *Physical Review B*, 1999, 59(3): 2070.
- [98] Lovett B W, Reina J H, Nazir A, et al. Optical schemes for quantum computation in quantum dot molecules[J]. *Physical Review B*, 2003, 68(20): 205319.
- [99] O'brien J L. Optical quantum computing[J]. *Science*, 2007, 318(5856): 1567-1570.
- [100] Kok P, Munro W J, Nemoto K, et al. Linear optical quantum computing with photonic qubits [J]. *Reviews of Modern Physics*, 2007, 79(1): 135.
- [101] Childress L, Hanson R. Diamond nv centers for quantum computing and quantum networks[J]. *MRS Bulletin*, 2013, 38(2): 134-138.
- [102] Zhang J, Hegde S S, Suter D. Efficient implementation of a quantum algorithm in a single nitrogen-vacancy center of diamond[J]. *Physical Review Letters*, 2020, 125(3): 030501.
- [103] Jones J A, Mosca M. Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer[J]. *The Journal of Chemical Physics*, 1998, 109(5): 1648-1653.
- [104] Wen J, Kong X, Wei S, et al. Experimental realization of quantum algorithms for a linear system inspired by adiabatic quantum computing[J]. *Physical Review A*, 2019, 99(1): 012320.
- [105] Yang F, Zhao D, Wei C, et al. A parallel quantum eigensolver for quantum machine learning [J]. *New Journal of Physics*, 2023.
- [106] Freedman M, Kitaev A, Larsen M, et al. Topological quantum computation[J]. *Bulletin of the American Mathematical Society*, 2003, 40(1): 31-38.
- [107] Nayak C, Simon S H, Stern A, et al. Non-abelian anyons and topological quantum computation [J]. *Reviews of Modern Physics*, 2008, 80(3): 1083.
- [108] Stern A, Lindner N H. Topological quantum computation—from basic concepts to first experiments[J]. *Science*, 2013, 339(6124): 1179-1184.
- [109] 章立源, 张金龙, 崔广霖. 超导物理学[M]. 电子工业出版社, 1995.
- [110] Martinis J M, Devoret M H, Clarke J. Energy-level quantization in the zero-voltage state of a current-biased josephson junction[J]. *Physical Review Letters*, 1985, 55(15): 1543.
- [111] Bravyi S, Cross A W, Gambetta J M, et al. High-threshold and low-overhead fault-tolerant quantum memory[A]. 2023.
- [112] Aksенов M A, Zalivako I V, Semerikov I A, et al. Realizing quantum gates with optically addressable yb+ 171 ion qudits[J]. *Physical Review A*, 2023, 107(5): 052612.
- [113] Katz O, Cetina M, Monroe C. N-body interactions between trapped ion qubits via spin-dependent squeezing[J]. *Physical Review Letters*, 2022, 129(6): 063603.

- [114] Moses S A, Baldwin C H, Allman M S, et al. A race-track trapped-ion quantum processor[J]. Physical Review X, 2023, 13(4): 041052.
- [115] [EB/OL]. <http://quafu.baqis.ac.cn/>.
- [116] [EB/OL]. <http://github.com/ScQ-Cloud/pyquafu>.
- [117] [EB/OL]. <http://scq-cloud.github.io/>.
- [118] [EB/OL]. <https://www.ibm.com/quantum>.
- [119] [EB/OL]. <https://github.com/Qiskit>.
- [120] [EB/OL]. <https://ionq.com/quantum-cloud>.
- [121] [EB/OL]. <https://aws.amazon.com/cn/braket/>.
- [122] Wright K, Beck K M, Debnath S, et al. Benchmarking an 11-qubit quantum computer[J]. Nature Communications, 2019, 10(1): 5464.
- [123] Figgatt C, Maslov D, Landsman K A, et al. Complete 3-qubit grover search on a programmable quantum computer[J]. Nature Communications, 2017, 8(1): 1918.
- [124] Long G L. Grover algorithm with zero theoretical failure rate[J]. Physical Review A, 2001, 64 (2): 022307.
- [125] Long G L, Li Y S, Zhang W L, et al. Phase matching in quantum searching[J]. Physics Letters A, 1999, 262(1): 27-34.
- [126] Long G, WeiLin Z, Li Y, et al. Arbitrary phase rotation of the marked state cannot be used for grover's quantum search algorithm[J]. Communications in Theoretical Physics, 1999, 32(3): 335.
- [127] Li D F, Li X X, Huang H T. Phase condition for the grover algorithm[J]. Theoretical and Mathematical Physics, 2005, 144: 1279-1287.
- [128] Brassard G, Høyer P, Tapp A. Quantum cryptanalysis of hash and claw-free functions[C]// LATIN'98: Theoretical Informatics: Third Latin American Symposium Campinas, Brazil, April 20–24, 1998 Proceedings 3. Springer, 1998: 163-169.
- [129] Grassl M, Langenberg B, Roetteler M, et al. Applying grover's algorithm to aes: quantum resource estimates[C]//International Workshop on Post-Quantum Cryptography. Springer, 2016: 29-43.
- [130] Høyer P. Arbitrary phases in quantum amplitude amplification[J]. Physical Review A, 2000, 62(5): 052304.
- [131] Brassard G, Hoyer P, Mosca M, et al. Quantum amplitude amplification and estimation[J]. Contemporary Mathematics, 2002, 305: 53-74.
- [132] Grover L K. Fixed-point quantum search[J]. Physical Review Letters, 2005, 95(15): 150501.
- [133] Yoder T J, Low G H, Chuang I L. Fixed-point quantum search with an optimal number of queries[J]. Physical Review Letters, 2014, 113(21): 210501.
- [134] Coppersmith D. An approximate fourier transform useful in quantum factoring[A]. 2002.
- [135] Abrams D S, Lloyd S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors[J]. Physical Review Letters, 1999, 83(24): 5162.

参考文献

- [136] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer[J]. SIAM Review, 1999, 41(2): 303-332.
- [137] Lloyd S. Universal quantum simulators[J]. Science, 1996, 273(5278): 1073-1078.
- [138] Trotter H F. On the product of semi-groups of operators[J]. Proceedings of the American Mathematical Society, 1959, 10(4): 545-551.
- [139] Chernoff P R. Note on product formulas for operator semigroups[J]. Journal of Functional Analysis, 1968, 2(2): 238-242.
- [140] Berry D W, Ahokas G, Cleve R, et al. Efficient quantum algorithms for simulating sparse hamiltonians[J]. Communications in Mathematical Physics, 2007, 270: 359-371.
- [141] Childs A M. On the relationship between continuous-and discrete-time quantum walk[J]. Communications in Mathematical Physics, 2010, 294: 581-603.
- [142] Wiebe N, Childs A. Hamiltonian simulation using linear combinations of unitary operations [C]//APS March Meeting Abstracts: volume 2012. 2012: T30-003.
- [143] Cerezo M, Arrasmith A, Babbush R, et al. Variational quantum algorithms[J]. Nature Reviews Physics, 2021, 3(9): 625-644.
- [144] Peruzzo A, McClean J, Shadbolt P, et al. A variational eigenvalue solver on a photonic quantum processor[J]. Nature Communications, 2014, 5(1): 4213.
- [145] Higgott O, Wang D, Brierley S. Variational quantum computation of excited states[J]. Quantum, 2019, 3: 156.
- [146] Xie Q X, Liu S, Zhao Y. Orthogonal state reduction variational eigensolver for the excited-state calculations on quantum computers[J]. Journal of Chemical Theory and Computation, 2022, 18(6): 3737-3746.
- [147] Bravo-Prieto C, LaRose R, Cerezo M, et al. Variational quantum linear solver[J]. Quantum, 2023, 7: 1188.
- [148] Cerezo M, Sharma K, Arrasmith A, et al. Variational quantum state eigensolver[J]. npj Quantum Information, 2022, 8(1): 113.
- [149] Kandala A, Mezzacapo A, Temme K, et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets[J]. Nature, 2017, 549: 242-246.
- [150] Wang D, Higgott O, Brierley S. Accelerated variational quantum eigensolver[J]. Physical Review Letters, 2019, 122: 140504.
- [151] Amaro D, Modica C, Rosenkranz M, et al. Filtering variational quantum algorithms for combinatorial optimization[J]. Quantum Science and Technology, 2022, 7: 015021.
- [152] McClean J R, Boixo S, Smelyanskiy V N, et al. Barren plateaus in quantum neural network training landscapes[J]. Nature Communications, 2018, 9: 4812.
- [153] Bittel L, Kliesch M. Training variational quantum algorithms is np-hard[J]. Physical Review Letters, 2021, 127: 120502.
- [154] Farhi E, Goldstone J. A quantum approximate optimization algorithm[A]. 2014.
- [155] Wurtz J, Love P J. Counterdiabaticity and the quantum approximate optimization algorithm[J]. Quantum, 2022, 6: 635.

- [156] Farhi E, Harrow A W. Quantum supremacy through the quantum approximate optimization algorithm[A]. 2016.
- [157] Guerreschi G G, Matsuura A Y. Qaoa for max-cut requires hundreds of qubits for quantum speed-up[J]. *Scientific Reports*, 2019, 9(1): 6903.
- [158] Anschuetz E, Olson J, Aspuru-Guzik A, et al. Variational quantum factoring[C]//Quantum Technology and Optimization Problems: First International Workshop, QTOP 2019, Munich, Germany, March 18, 2019, Proceedings 1. Springer, 2019: 74-85.
- [159] Zhou L, Wang S T, Choi S, et al. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices[J]. *Physical Review X*, 2020, 10(2): 021067.
- [160] Harrigan M P, Sung K J, Neeley M, et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor[J]. *Nature Physics*, 2021, 17(3): 332-336.
- [161] Wurtz J, Love P. Maxcut quantum approximate optimization algorithm performance guarantees for $p > 1$ [J]. *Physical Review A*, 2021, 103(4): 042612.
- [162] Majumdar R, Madan D, Bhoumik D, et al. Optimizing ansatz design in qaoa for max-cut[A]. 2021.
- [163] Zhou Z, Du Y, Tian X, et al. Qaoa-in-qaoa: solving large-scale maxcut problems on small quantum machines[J]. *Physical Review Applied*, 2023, 19(2): 024027.
- [164] Li Y, Benjamin S C. Efficient variational quantum simulator incorporating active error minimization[J]. *Physical Review X*, 2017, 7(2): 021050.
- [165] McLachlan A D. A variational solution of the time-dependent schrodinger equation[J]. *Molecular Physics*, 1964, 8(1): 39-44.
- [166] Mitarai K, Fujii K. Methodology for replacing indirect measurements with direct measurements [J]. *Physical Review Research*, 2019, 1(1): 013006.
- [167] McArdle S, Jones T, Endo S, et al. Variational ansatz-based quantum simulation of imaginary time evolution[J]. *npj Quantum Information*, 2019, 5: 75.
- [168] Yuan X, Endo S, Zhao Q, et al. Theory of variational quantum simulation[J]. *Quantum*, 2019, 3: 191.
- [169] Endo S, Sun J, Li Y, et al. Variational quantum simulation of general processes[J]. *Physical Review Letters*, 2020, 125(1): 010501.
- [170] Yao Y X, Gomes N, Zhang F, et al. Adaptive variational quantum dynamics simulations[J]. *PRX Quantum*, 2021, 2(3): 030307.
- [171] Cirstoiu C, Holmes Z, Iosue J, et al. Variational fast forwarding for quantum simulation beyond the coherence time[J]. *npj Quantum Information*, 2020, 6(1): 82.
- [172] Filip M A, Ramo D M, Fitzpatrick N. Variational phase estimation with variational fast forwarding[J]. *Quantum*, 2024, 8: 1278.
- [173] Long G L. General quantum interference principle and duality computer[J]. *Communications in Theoretical Physics*, 2006, 45(5): 825.

- [174] Long G, Liu Y. Duality quantum computing[J]. Frontiers of Computer Science in China, 2008, 2: 167-178.
- [175] Long G, Liu Y. Duality computing in quantum computers[J]. Communications in Theoretical Physics, 2008, 50(6): 1303.
- [176] Long G L. Duality quantum computing and duality quantum information processing[J]. International Journal of Theoretical Physics, 2011, 50: 1305-1318.
- [177] Gudder S. Mathematical theory of duality quantum computers[J]. Quantum Information Processing, 2007, 6(1): 37-48.
- [178] Zou X, Qiu D, Wu L, et al. On mathematical theory of the duality computers[J]. Quantum Information Processing, 2009, 8: 37-50.
- [179] Gao P, Li K, Wei S, et al. Quantum gradient algorithm for general polynomials[J]. Physical Review A, 2021, 103(4): 042403.
- [180] Gao P, Li K, Wei S, et al. Quantum second-order optimization algorithm for general polynomials [J]. Science China Physics, Mechanics & Astronomy, 2021, 64(10): 100311.
- [181] Wei S, Li H, Long G. A full quantum eigensolver for quantum chemistry simulations[J]. Research, 2020.
- [182] Liang J M, Wei S J, Fei S M. Quantum gradient descent algorithms for nonequilibrium steady states and linear algebraic systems[J]. Science China Physics, Mechanics & Astronomy, 2022, 65(5): 250313.
- [183] Wen J, Wang Z, Chen C, et al. A full circuit-based quantum algorithm for excited-states in quantum chemistry[J]. Quantum, 2024, 8: 1219.
- [184] Lv P, Wei S, Xie H N, et al. Qcsh: A full quantum computer nuclear shell-model package[J]. Science China Physics, Mechanics & Astronomy, 2023, 66(4): 240311.
- [185] Wang B, Wen J, Wu J, et al. A powered full quantum eigensolver for energy band structures [A]. 2023.
- [186] Berry D W, Childs A M, Cleve R, et al. Simulating hamiltonian dynamics with a truncated taylor series[J]. Physical Review Letters, 2015, 114(9): 090502.
- [187] Wei S J, Long G L. Duality quantum computer and the efficient quantum simulations[J]. Quantum Information Processing, 2016, 15: 1189-1212.
- [188] Wei S J, Ruan D, Long G L. Duality quantum algorithm efficiently simulates open quantum systems[J]. Scientific Reports, 2016, 6(1): 30727.
- [189] Schlimgen A W, Head-Marsden K, Sager L M, et al. Quantum simulation of open quantum systems using a unitary decomposition of operators[J]. Physical Review Letters, 2021, 127 (27): 270503.
- [190] Gilyén A. Quantum singular value transformation & its algorithmic applications[D]. University of Amsterdam, 2019.
- [191] Gilyén A, Su Y, Low G H, et al. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics[C]//Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. 2019: 193-204.

参考文献

- [192] Dong Y, Meng X, Whaley K B, et al. Efficient phase-factor evaluation in quantum signal processing[J]. Physical Review A, 2021, 103(4): 042419.
- [193] Rossi Z M, Chuang I L. Multivariable quantum signal processing (m-qsp): prophecies of the two-headed oracle[J]. Quantum, 2022, 6: 811.
- [194] Motlagh D, Wiebe N. Generalized quantum signal processing[A]. 2023.
- [195] Liang J M, Lv Q Q, Shen S Q, et al. Improved iterative quantum algorithm for ground-state preparation[J]. Advanced Quantum Technologies, 2022, 5(12): 2200090.
- [196] Duan B, Yuan J, Liu Y, et al. Quantum algorithm for support matrix machines[J]. Physical Review A, 2017, 96(3): 032301.
- [197] Duan B, Yuan J, Liu Y, et al. Efficient quantum circuit for singular-value thresholding[J]. Physical Review A, 2018, 98(1): 012308.
- [198] Wiebe N, Braun D, Lloyd S. Quantum algorithm for data fitting[J]. Physical Review Letters, 2012, 109(5): 050505.
- [199] Kerenidis I, Prakash A. Quantum recommendation systems[A]. 2016.
- [200] Lloyd S, Mohseni M, Rebentrost P. Quantum principal component analysis[J]. Nature Physics, 2014, 10(9): 631-633.
- [201] Kraus C V, Cirac J I. Generalized hartree-fock theory for interacting fermions in lattices: numerical methods[J]. New Journal of Physics, 2010, 12: 113004.
- [202] Verstraete F, Garcia-Ripoll J J, Cirac J I. Matrix product density operators: Simulation of finite-temperature and dissipative systems[J]. Physical Review Letters, 2004, 93: 207204.
- [203] White S R. Minimally entangled typical quantum states at finite temperature[J]. Physical Review Letters, 2009, 102: 190601.
- [204] Haah J, Kothari R, Tang E. Optimal learning of quantum hamiltonians from high-temperature gibbs states[C]//2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS). IEEE, 2022: 135-146.
- [205] Kamakari H, Sun S N, Motta M, et al. Digital quantum simulation of open quantum systems using quantum imaginary-time evolution[J]. PRX Quantum, 2022, 3(1): 010320.
- [206] McClean J R, Aspuru-Guzik A. Clock quantum monte carlo technique: An imaginary-time method for real-time quantum dynamics[J]. Physical Review A, 2015, 91(1): 012311.
- [207] Czarnik P, Cincio L, Dziarmaga J. Projected entangled pair states at finite temperature: Imaginary time evolution with ancillas[J]. Physical Review B, 2012, 86: 245101.
- [208] Stoudenmire E M, White S R. Minimally entangled typical thermal state algorithms[J]. New Journal of Physics, 2010, 12(5): 055026.
- [209] Phien H N, McCulloch I P, Vidal G. Fast convergence of imaginary time evolution tensor network algorithms by recycling the environment[J]. Physical Review B, 2015, 91(11): 115137.
- [210] Moler C, Van Loan C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later[J]. SIAM Review, 2003, 45(1): 3-49.
- [211] Jones T, Endo S, McArdle S, et al. Variational quantum algorithms for discovering hamiltonian spectra[J]. Physical Review A, 2019, 99(6): 062304.

- [212] Gomes N, Mukherjee A, Zhang F, et al. Adaptive variational quantum imaginary time evolution approach for ground state preparation[J]. Advanced Quantum Technologies, 2021, 4(12): 2100114.
- [213] Motta M, Sun C, Tan A T K, et al. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution[J]. Nature Physics, 2020, 16(2): 205-210.
- [214] Huang Y, Shao Y, Ren W, et al. Efficient quantum imaginary time evolution by drifting real-time evolution: an approach with low gate and measurement complexity[J]. Journal of Chemical Theory and Computation, 2023, 19(13): 3868-3876.
- [215] Yeter-Aydeniz K, Pooser R C, Siopsis G. Practical quantum computation of chemical and nuclear energy levels using quantum imaginary time evolution and lanczos algorithms[J]. npj Quantum Information, 2020, 6(1): 63.
- [216] Sun S N, Motta M, Tazhigulov R N, et al. Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution[J]. PRX Quantum, 2021, 2: 010317.
- [217] Nishi H, Kosugi T, Matsushita Y. Implementation of quantum imaginary-time evolution method on nisq devices by introducing nonlocal approximation[J]. npj Quantum Information, 2021, 7 (1): 85.
- [218] Jouzdani P, Johnson C W, Mucciolo E R, et al. Alternative approach to quantum imaginary time evolution[J]. Physical Review A, 2022, 106(6): 062435.
- [219] Cao C, An Z, Hou S Y, et al. Quantum imaginary time evolution steered by reinforcement learning[J]. Communications Physics, 2022, 5(1): 57.
- [220] Yeter-Aydeniz K, Moschandreas E, Siopsis G. Quantum imaginary-time evolution algorithm for quantum field theories with continuous variables[J]. Physical Review A, 2022, 105(1): 012412.
- [221] Lin S H, Dilip R, Green A G, et al. Real- and imaginary-time evolution with compressed quantum circuits[J]. PRX Quantum, 2021, 2: 010342.
- [222] Liu T, Liu J G, Fan H. Probabilistic nonunitary gate in imaginary time evolution[J]. Quantum Information Processing, 2021, 20: 204.
- [223] Turro F, Roggero A, Amitrano V, et al. Imaginary-time propagation on a quantum chip[J]. Phys. Rev. A, 2022, 105: 022440.
- [224] Kosugi T, Nishiya Y, Nishi H, et al. Imaginary-time evolution using forward and backward real-time evolution with a single ancilla: First-quantized eigensolver algorithm for quantum chemistry[J]. Physical Review Research, 2022, 4(3): 033121.
- [225] Leadbeater C, Fitzpatrick N, Ramo D M, et al. Non-unitary trotter circuits for imaginary time evolution[A]. 2023.
- [226] Silva T L, Taddei M M, Carrazza S, et al. Fragmented imaginary-time evolution for early-stage quantum signal processors[J]. Scientific Reports, 2023, 13(1): 18258.
- [227] Sornborger A T, Stewart E D. Higher-order methods for simulations on quantum computers[J]. Physical Review A, 1999, 60: 1956.

参考文献

- [228] Hehre W J, Ditchfield R, Pople J A. Self-consistent molecular orbital methods. xii. further extensions of gaussian-type basis sets for use in molecular orbital studies of organic molecules [J]. *Journal of Chemical Physics*, 1972, 56: 5.
- [229] Colless J I, Ramasesh V V, Dahlen D, et al. Computation of molecular spectra on a quantum processor with an error-resilient algorithm[J]. *Physical Review X*, 2018, 8(1): 011021.
- [230] Benedetti M, Fiorentini M, Lubasch M. Hardware-efficient variational quantum algorithms for time evolution[J]. *Physical Review Research*, 2021, 3(3): 033083.
- [231] Bender C M. Making sense of non-hermitian hamiltonians[J]. *Reports on Progress in Physics*, 2007, 70(6): 947.
- [232] Ashida Y, Gong Z, Ueda M. Non-hermitian physics[J]. *Advances in Physics*, 2020, 69(3): 249-435.
- [233] Lindblad G. On the generators of quantum dynamical semigroups[J]. *Communications in Mathematical Physics*, 1976, 48: 119-130.
- [234] Klaiman S, Günther U, Moiseyev N. Visualization of branch points in \mathcal{PT} -symmetric waveguides[J]. *Physical Review Letters*, 2008, 101(8): 080402.
- [235] Makris K G, El-Ganainy R, Christodoulides D N, et al. Beam dynamics in \mathcal{PT} symmetric optical lattices[J]. *Physical Review Letters*, 2008, 100(10): 103904.
- [236] Guo A, Salamo G J, Duchesne D, et al. Observation of \mathcal{PT} -symmetry breaking in complex optical potentials[J]. *Physical Review Letters*, 2009, 103(9): 093902.
- [237] Longhi S. Bloch oscillations in complex crystals with \mathcal{PT} symmetry[J]. *Physical Review Letters*, 2009, 103(12): 123601.
- [238] Chong Y, Ge L, Stone A D. \mathcal{PT} -symmetry breaking and laser-absorber modes in optical scattering systems[J]. *Physical Review Letters*, 2011, 106(9): 093902.
- [239] Feng L, Wong Z J, Ma R M, et al. Single-mode laser by parity-time symmetry breaking[J]. *Science*, 2014, 346(6212): 972-975.
- [240] Carmichael H J. Quantum trajectory theory for cascaded open systems[J]. *Physical Review Letters*, 1993, 70(15): 2273.
- [241] Choi Y, Kang S, Lim S, et al. Quasieigenstate coalescence in an atom-cavity quantum composite [J]. *Physical Review Letters*, 2010, 104(15): 153601.
- [242] Lin Z, Ramezani H, Eichelkraut T, et al. Unidirectional invisibility induced by p t-symmetric periodic structures[J]. *Physical Review Letters*, 2011, 106(21): 213901.
- [243] Lee T E, Chan C K. Heralded magnetism in non-hermitian atomic systems[J]. *Physical Review X*, 2014, 4(4): 041001.
- [244] Li J, Harter A K, Liu J, et al. Observation of parity-time symmetry breaking transitions in a dissipative floquet system of ultracold atoms[J]. *Nature Communications*, 2019, 10(1): 855.
- [245] Zyuzin A A, Zyuzin A Y. Flat band in disorder-driven non-hermitian weyl semimetals[J]. *Physical Review B*, 2018, 97(4): 041203.
- [246] Shen H, Fu L. Quantum oscillation from in-gap states and a non-hermitian landau level problem [J]. *Physical Review Letters*, 2018, 121(2): 026403.

参考文献

- [247] Zhou H, Peng C, Yoon Y, et al. Observation of bulk fermi arc and polarization half charge from paired exceptional points[J]. *Science*, 2018, 359(6379): 1009-1012.
- [248] Papaj M, Isobe H, Fu L. Nodal arc of disordered dirac fermions and non-hermitian band theory [J]. *Physical Review B*, 2019, 99(20): 201107.
- [249] Lee T E. Anomalous edge state in a non-hermitian lattice[J]. *Physical Review Letters*, 2016, 116(13): 133903.
- [250] Leykam D, Bliokh K Y, Huang C, et al. Edge modes, degeneracies, and topological numbers in non-hermitian systems[J]. *Physical Review Letters*, 2017, 118(4): 040401.
- [251] Shen H, Zhen B, Fu L. Topological band theory for non-hermitian hamiltonians[J]. *Physical Review Letters*, 2018, 120(14): 146402.
- [252] Yao S, Song F, Wang Z. Non-hermitian chern bands[J]. *Physical Review Letters*, 2018, 121 (13): 136802.
- [253] Lee C H, Li L, Gong J. Hybrid higher-order skin-topological modes in nonreciprocal systems [J]. *Physical Review Letters*, 2019, 123(1): 016805.
- [254] Liu T, Zhang Y R, Ai Q, et al. Second-order topological phases in non-hermitian systems[J]. *Physical Review Letters*, 2019, 122(7): 076801.
- [255] Longhi S. Topological phase transition in non-hermitian quasicrystals[J]. *Physical Review Letters*, 2019, 122(23): 237601.
- [256] Zhang K, Yang Z, Fang C. Correspondence between winding numbers and skin modes in non-hermitian systems[J]. *Physical Review Letters*, 2020, 125(12): 126402.
- [257] Xiao L, Deng T, Wang K, et al. Non-hermitian bulk–boundary correspondence in quantum dynamics[J]. *Nature Physics*, 2020, 16(7): 761-766.
- [258] Hu H, Zhao E. Knots and non-hermitian bloch bands[J]. *Physical Review Letters*, 2021, 126 (1): 010401.
- [259] Kawabata K, Shiozaki K, Ryu S. Topological field theory of non-hermitian systems[J]. *Physical Review Letters*, 2021, 126(21): 216405.
- [260] Sun X Q, Zhu P, Hughes T L. Geometric response and disclination-induced skin effects in non-hermitian systems[J]. *Physical Review Letters*, 2021, 127(6): 066401.
- [261] Wu D, Xie J, Zhou Y, et al. Connections between the open-boundary spectrum and the generalized brillouin zone in non-hermitian systems[J]. *Physical Review B*, 2022, 105(4): 045422.
- [262] Miri M A, Alu A. Exceptional points in optics and photonics[J]. *Science*, 2019, 363(6422): eaar7709.
- [263] Long D, Mao X, Qin G Q, et al. Dynamical encircling of the exceptional point in a largely detuned multimode optomechanical system[J]. *Physical Review A*, 2022, 106(5): 053515.
- [264] Xue W T, Hu Y M, Song F, et al. Non-hermitian edge burst[J]. *Physical Review Letters*, 2022, 128(12): 120401.
- [265] Yuce C, Ramezani H. Non-hermitian edge burst without skin localization[J]. *Physical Review B*, 2023, 107(14): L140302.

参考文献

- [266] Wen P, Pi J, Long G L. Investigation of a non-hermitian edge burst with time-dependent perturbation theory[J]. Physical Review A, 2024, 109(2): 022236.
- [267] Xiao L, Xue W T, Song F, et al. Observation of non-hermitian edge burst in quantum dynamics [A]. 2023.
- [268] 张冰尘, 洪文, 吴一戎. 稀疏微波成像导论[M]. 科学出版社, 2019.
- [269] Tibshirani R. Regression shrinkage and selection via the lasso[J]. Journal of the Royal Statistical Society Series B: Statistical Methodology, 1996, 58(1): 267-288.
- [270] Vorlíčková J. Least absolute shrinkage and selection operator method[M]. Univerzita Karlova, Fakulta sociálních věd, 2017.
- [271] Daubechies I, Defrise M, De Mol C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint[J]. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, 2004, 57(11): 1413-1457.
- [272] Liu X, Dong C, Luo Y, et al. Sparse reconstruction for radar imaging based on quantum algorithms[J]. IEEE Geoscience and Remote Sensing Letters, 2021, 19: 1-5.
- [273] Donoho D L, Tsaig Y. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse[J]. IEEE Transactions on Information theory, 2008, 54(11): 4789-4812.
- [274] Liu Y, Wang D, Xue S, et al. Variational quantum circuits for quantum state tomography[J]. Physical Review A, 2020, 101(5): 052316.
- [275] Xiao J, Wen J, Wei S, et al. Reconstructing unknown quantum states using variational layerwise method[J]. Frontiers of Physics, 2022, 17(5): 51501.

致 谢

从本科入学至今，我已在清华园中度过了九年。我非常庆幸自己能在青春时代享受这样一段愉快而充实的求学生活。短暂而又漫长的九年的时光给我留下了许多回忆，也为我的人生留下了不可磨灭的烙印。在这段经历中，我也有幸拥有了许多人的陪伴，他们不仅为我提供精神上的慰藉，也为我在生活中的成长和学业上的进步提供了莫大的帮助。在今后的人生中，他们值得我永远铭记。

最为感谢导师龙桂鲁教授在学习和生活中对我的指导和关心。龙老师在学术上务实严谨、不断进取，在生活中真诚谦和、朴素大方，对待学生既严格要求、也宽容相待，这些品格对我影响甚远，使我感到温馨的同时，也鼓励着我不断完善自我、努力前行。

魏世杰师兄帮助我确定了研究方向，并在此后的科研过程中细致入微地指导我，与我一同讨论研究内容与研究方法，在我怠惰时督促我努力工作，为我的学业和生活提供了巨大的帮助。对此我表示最真诚的感谢。

还要特别感谢北京量子信息科学研究院的范桁老师和王正安等人，他们为我的科研课题提供了重要的研究资源和帮助。

向我的女友马天贶致以最真挚的情意，她既是爱人也是朋友。在生活中她体贴入微的照顾和陪伴让我倍感温暖，在我感到焦虑和迷茫时她为我提供可靠的情绪支持和安慰，同时她也督促我改正缺点、不断成长，让我能成为更好的自己。一同致意的还包括马七酱与谢墨菲，他们用可爱的身体与灵魂为疲倦的人类带来抚慰和温暖。

高攀、王泽国、杨帆、皮竞辉师兄以及王博桢师姐扎实的专业基础让我获益匪浅，与他们交流讨论对我的科研内容和思路起到了重要帮助；毛璇师兄优秀的科研成果为我树立了榜样；张江、李行、曾进峰、邢同昊、张飞昊、周增荣、张浩然、潘栋师兄以及王敏师姐在我参与科研项目的过程中提供了宝贵的帮助；胡晓晓、李小刚、赵大法、吴家为、肖俊祥、陈鑫宇师兄，同级的温棚宇和李相洁，以及龙丹、雷雨霆、孙泽洲、刘雨辰等师弟师妹，为我的学习和生活增长了见识，增添了乐趣。感谢龙组大家庭的每一个人，恕不能在此一一列举。

感谢赵凌飞、李子硕、乔宗远等一众挚友，即使有些朋友不常见面，也会在心中相互支持鼓励，相聚之时的欢乐则更加难以忘怀。

感谢伯格曼、塔科夫斯基、费里尼、伍迪·艾伦、基耶斯洛夫斯基、安哲罗普洛斯、特吕弗、戈达尔、朱塞佩·罗曼·波兰斯基、希区柯克、卓别林、库布里克、

致 谢

斯科塞斯、昆汀、诺兰、维伦纽瓦、韦斯·安德森、宫崎骏、奉俊昊、姜文、王家卫、北野武、黑泽明……他们为人类电影艺术贡献了无价的珍宝。感谢阿加莎·克里斯蒂、岛田庄司、江户川乱步等本格推理作家和阿西莫夫、阿瑟·克拉克、刘慈欣等科幻作家。他们让我的精神生活免遭枯竭。他们和他们的作品将永远被世人铭记。

最后，也感谢我的亲人在远方默默的支持和关心。

愿所有人安好，愿大家都有精彩的人生。

声 明

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：_____ 日 期：_____

个人简历、在学期间完成的相关学术成果

个人简历

1997 年 08 月 22 日出生于河北省邯郸市。

2015 年 8 月考入清华大学物理系数理基础科学专业，2019 年 7 月本科毕业并获得理学学士学位。

2019 年 8 月免试进入清华大学物理系攻读物理学博士至今。

在学期间完成的相关学术成果

学术论文：

- [1] **Hao-Nan Xie**, Shi-Jie Wei, Fan Yang, Zheng-An Wang, Chi-Tong Chen, Heng Fan, Gui-Lu Long. Probabilistic imaginary-time evolution algorithm based on non-unitary quantum circuit[J]. Physical Review A, 2024, 109(5): 052414. (SCI 收录)
- [2] Peng Lv, Shijie Wei, **Hao-Nan Xie**, Guilu Long. QCSH: A full quantum computer nuclear shell-model package[J]. Science China Physics, Mechanics & Astronomy, 2023, 66(4): 240311. (SCI 收录)
- [3] Bozhi Wang, Jingwei Wen, Jiawei Wu, **Haonan Xie**, Fan Yang, Shijie Wei, Gui-lu Long. A powered full quantum eigensolver for energy band structures. (Physical Review B 已接收)

专利：

- [4] 魏世杰, 龙桂鲁, 谢浩楠. 一种计算蛋白质分子构型的方法、装置: 中国, CN116189788A[P]. 2022-12-30.

指导教师评语

谢浩楠同学独立完成了论文的研究与撰写，围绕基于非酉量子线路的量子算法展开研究，构造了量子体系虚时演化与稀疏成像问题的量子算法，并通过演示实验和数值模拟进行了验证。该论文取得了以下创造性成果：

1. 利用非酉量子线路构造了一种模拟量子体系虚时演化的量子算法，提供了一种普适且具体的量子线路构造方法，给出了算法的误差与复杂度分析，将这一算法应用于量子体系基态求解问题中，并在量子计算云平台演示实验与经典数值模拟中进行了验证。这一虚时演化算法相比较于经典方法，在较大的量子体系中具有指数加速的效果，相比较于已有的量子算法，则具有更好的普适性以及适合在实验中实现的特点。
2. 进一步研究了基于非酉量子线路的虚时演化算法在非厄米边缘突起效应计算中的应用，论证了本文所构造的虚时演化算法在非厄米体系模拟问题中的普适性。同时，还研究了变分量子算法在计算非厄米边缘突起效应中的应用，推导出了一种适用于非厄米体系的不受全局相位影响的变分方法。通过数值模拟，验证了两种量子算法在非厄米体系模拟中的适用性。
3. 利用非酉量子线路，提出了一种基于迭代收缩阈值方法的量子稀疏成像算法，给出了算法的误差与复杂度分析，并进行了相关数值模拟和实验演示。这一量子稀疏成像算法相比较于经典方法具有指数加速的效果，相比较于已有的量子算法，可以在成像结果上更好地体现稀疏性，从而达到更高的成像质量。

论文发展了非酉量子线路方法，将其应用于量子多体系统的计算和稀疏成像，取得显著效果，具有很强的创新性、科研创造性，具有重要的理论意义，并在未来量子计算的实际应用中具有较大的应用价值。

答辩委员会决议书

谢浩楠同学的博士学位论文研究了基于非酉线路的量子多体系统模拟和稀疏成像量子算法，论文选题具有重要的科学意义和应用价值。论文所取得的主要创新性成果如下：

- (1) 提出基于非酉量子线路的虚时演化量子算法，给出了一种适合实验实现的构建量子线路的系统性方法，对算法的复杂度进行了理论分析，在量子计算云平台验证了算法的有效性，提高了算法的实用性。
- (2) 将概率性虚时演化算法和变分量子算法应用于非厄米边缘突起效应的模拟，给出了任意非厄米体系动力学演化的模拟方法，并通过数值模拟验证了两种量子算法的有效性与准确性。
- (3) 提出基于非酉量子线路的量子迭代收缩阈值算法，对算法的复杂度进行了理论分析；通过量子计算云平台验证了算法在求解稀疏成像问题中的有效性。

该博士学位论文写作规范，表述准确，逻辑严谨，文献综述全面，研究结果可靠，是一篇优秀的博士论文。论文表明作者具有扎实的理论基础和系统的专业知识，具有很强的独立从事科研工作的能力。谢浩楠同学在答辩过程中表述清晰，回答问题准确。答辩委员会经无记名投票表决，一致同意通过谢浩楠同学的学位论文答辩，并建议授予其理学博士学位。