全面量子模拟编译器的验证与评估

BhuwanJoshi ✉爱荷华州
立大学，爱荷华州，美国
Zoukarneini Tagba ✉爱
荷华州立大学，爱荷华州，美国
ZhiXiang Chong ✉爱荷
华州立大学，爱荷华州，美国
LiyiLi ✉爱荷华州立大学，爱
荷华州，美国

arXiv:2611.28388v1 [quant-ph] 20 Nov 2026

摘要

哈密顿量模拟是量子计算的一项核心应用，但现有的大多数编译器都无法提供机器检查保证，证明其生成的电路能够近似预期演化。之前的 QBlue 编译器验证了一个仅包含基于 Trotter 的算法的哈密顿量模拟编译器。该编译器接收以泡利字符串表示的哈密顿量、模拟时间和误差界，并生成一个电路或模拟调度；在 Rocq 证明助手中，已证明该电路或调度能够在给定误差界内近似理想时间演化，同时还证明了一个端到端误差界，将各个阶段的误差进行合成；编译器通过资源估算，利用该误差界选择满足用户要求的成本最低的算法。本文通过以下两个步骤进一步推进验证工作。首先，我们将 QBlue 扩展为 QBlue$_{\text{BE}}$，验证另一种重要的哈密顿量模拟编译方法：基于块编码的（基于 BE 的）算法。其次，我们针对当前最先进的基于 BE 的编译器 PennyLane，对 279 个哈密顿量模拟程序进行了实证评估。对于许多程序而言，与 PennyLane 相比，我们的编译器实现了 94% 的门数减少。此外，我们还发现了 PennyLane 中的问题和漏洞。我们还比较了基于 Trotter 的算法与基于 BE 的算法，为长期以来关于哪类算法性能更优的争论提供了实践层面的比较。我们的结果出人意料地表明，基于 Trotter 的算法可能具有实践优势，尽管 BE 算法在理论上更具优势。

2012年 ACM 学科分类 计算理论 →量子计算理论；软件及其工程 → 形式化软件验证；软件及其工程 → 经验性软件验证

关键词和短语 量子计算、哈密顿量模拟、编译器验证

数字对象标识符 10.4230/LIPIcs...

1 引言

量子模拟是量子计算的一项旗舰应用，有望以超越经典能力的精度对物理系统 [95, 49, 33]，化学反应 [51, 20, 36, 62]，和材料 [35, 37, 68]进行建模 [38]。对于 Quantum 模拟，物理系统由哈密顿量 H 描述，而模拟其演化时间 r 意味着要将酉算子 exp($-ir\hat{}H$) 实现为设备可运行的量子电路或模拟调度。实际使用的算法 [18, 27, 17, 60, 12, 59, 61, 27] 只能近似这个酉算子，因此正确性需要定量衡量：当编译后的程序所表示的酉算子与 exp($-ir\hat{}H$) 之间的距离在规定范围内时，该程序才是正确的。答案的价值取决于误差有多大，以及编译器是否能获知误差。目前的哈密顿量模拟编译器 [52, 69, 18, 53]会积极进行优化，但没有一个能提供机器检查保证，证明其输出接近预期演化。错误编译的模拟会悄无声息地出错：量子电路照常运行并返回



© Bhuwan Joshi、Zoukarneini Tagba、Zhi Xiang Chong 和 Liyi Li；依据知识共享署名 4.0 国际许可协议授权 莱布尼茨国际信息学论文集 达格施图尔城堡 – 莱布尼茨信息学中心，德国达格施图尔出版社

XX：2 验证与评估全面的量子模拟编译器

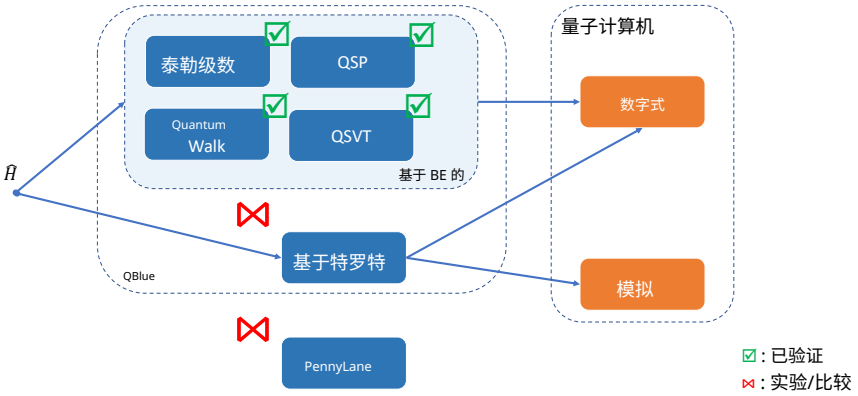


图 1 QBlue 项目流程；顶部是 QBlue$_{BE}$；我们将其与 PennyLane 进行比较，并比较 QBlue 中基于 BE 的算法和基于特罗特的算法。

数值，而这些数值描述的却是另一个物理系统。选择算法是第二个难题。人们提出了许多算法，各自声称性能更优；而成本最低的选择取决于哈密顿量、误差容限和所用平台。

此前，QBlue [54] 开发了一款已验证的量子模拟编译器，用于基于 Trotter-Suzuki 分解的基于特罗特的哈密顿量模拟编译。QBlue 的输入包括由泡利字符串构成的哈密顿量表达式、模拟时间 r 和误差界 ϵ。这种输入格式具有通用性：作用于 n 个量子比特的泡利字符串构成了作用于 n 个量子比特的所有算符的基，因此每个量子比特哈密顿量都可以用这种形式表示。编译器实现了四种不同的基于特罗特的算法，即一阶和二阶 Trotter 化 [83]，QDrift [17], 以及 MarQSim [18], ，它们将 exp($-ir\hat{}H$) 转换为由单个泡利串指数函数组成的序列，并针对三种后端之一合成该序列：标准数字式门序列、IBM 模拟超导硬件 [69], ，以及一种离子阱处理器，其全局驱动可直接实现多量子比特相互作用 [50, 74]。该编译器的主要成果是给出正确性保证，并通过 Rocq 证明助手 [14]进行机械化验证。该保证指出，对于厄米泡利字符串哈密顿量 $\hat{}H$、时间 r 和误差界 ϵ，编译后的程序 U 满足 $\|[\![U]\!](\psi) - [\![\exp(-ir\ \hat{}H)]\!](\psi)\| < \epsilon$ ，对任意归一化态 ψ均成立。为满足 ϵ，编译器根据 H的 $\hat{}$ 结构进行符号计算，求得目标后端上每种候选算法的误差上界，并设置算法参数（例如 Trotter 步数），以确保误差低于 ϵ。在满足 ϵ的算法中，编译器会选择估算成本最低的一种。

QBlue 完全基于特罗特，忽略了其他哈密顿量模拟算法，即基于 BE 的算法。基于 BE 的算法不执行 Trotter-Suzuki 分解，而是认识到 exp($-ir\hat{}H$) 的实现可以通过 Taylor 和 Jacobi-Anger 等展开公式展开，从而得到一个泡利字符串线性和的表达式；借助块编码（BE）技术，该表达式可以在量子计算机上实现——该技术主要将线性和操作与量子受控门连接起来。

本文扩展了 QBlue，使其支持基于 BE 的哈密顿量模拟算法，并开展实证研究，对比了不同的编译算法，如图 3 所示。我们的首要贡献是识别出基于 BE 的算法的编译流水线、QBlue$_{BE}$，并在 Rocq 中建立正确性保证，对所有这些算法进行验证。在验证过程中，我们发现了现有平台 PennyLane [11] 中的问题和漏洞；为此，我们在大型基准测试上进行评估，并将 QBlue$_{BE}$ 编译与

PennyLane 进行对比，以识别这些问题和漏洞。据我们所知，这是首个针对哈密顿量模拟的经验证编译器，能够在多种基于 BE 的算法中携带机械化误差界。

另一项贡献是对比基于 Trotter 和基于 BE 的编译算法进行的实证研究。基于 BE 的算法预计具有比基于 Trotter 的算法更低的理论复杂度 [27], ，但其实际优势尚不明确。在本文中，我们使用涵盖真实世界量子化学和物理学程序的综合基准测试来编译这些算法。我们考察不同算法所生成的编译后的电路中的量子比特数和量子门数，以了解这两类算法各自的优势。我们发现了令人意外的结果。尽管基于 BE 的算法在更长的模拟时间和更严格的误差界下能够在量子门节省方面提供理论优势，但对于具有合理模拟时间和误差界的真实世界案例，基于 Trotter 的算法在量子比特数和量子门数方面具有实际优势。

1.0.0.1 贡献总结

我们对基于 BE 的哈密顿量模拟编译器进行端到端验证，并在 Rocq 中使用经验证的定理机械化流水线中的每个步骤（图 3）。我们还对基于 BE 的哈密顿量模拟编译器开展了实证研究。在该研究中，我们在 279 个哈密顿量模拟程序上评估 QBlue$_{BE}$ ，这些程序涵盖从自旋模型到分子哈密顿量的不同类型，并将其与未经验证的最先进编译器 PennyLane [11],进行比较；此外，我们还考察哈密顿量模拟编译器的若干方面，包括比较基于 Trotter 和基于 BE 的算法，以及比较基于 BE 的算法组内的不同算法。一个关键区别在于，基于 Trotter 的算法适用于模拟量子计算机，而基于 BE 的算法只能在数字量子计算机上实现。我们比较了模拟量子计算机上的基于 Trotter 的算法与数字量子计算机上的基于 BE 的算法的效率，并发现前者效率更高。本文的贡献总结如下。

- 一条已验证的编译流水线，涵盖约束、动态阶段，以及面向基于 BE 的算法并提供经认证误差界的编译器；一个经机械化验证的端到端编译定理。我们为 QBlue$_{BE}$ 哈密顿量模拟编译器的每个步骤开发了一个库，作为基于 BE 的哈密顿量模拟的标准；全面评估基于 BE 的算法的哈密顿量模拟编译，并将 QBlue$_{BE}$ 编译器与工业界编译器 PennyLane 进行比较。我们发现了 PennyLane 中的问题，并表明我们的编译器比 PennyLane 节省更多量子门，即在许多情况下节省了 90% 以上的量子门；一项针对基于 BE 的算法和基于 Trotter 的算法的实证研究，其中包括对数字式和模拟平台的比较，以及对不同基于 BE 的算法的考察；一套扎实的基准测试，涵盖 279 个小型、中型和大型真实程序，这些基准测试可作为哈密顿量模拟的标准集合。

1.0.0.2 论文路线图

本文其余部分安排如下。第2节以伊辛模型为例介绍 QBlue$_{BE}$ ，并在此过程中介绍相关背景。第3节介绍编译流水线并证明其正确性。第4节和第5节评估 QBlue$_{BE}$，第6节讨论相关工作，第7节总结全文。

$$\begin{array}{llllll} \text{Nat. Num} & m,n,j,k,g \in \mathbb{N} & \text{Reals} & \theta,r \in \mathbb{R} & \text{Complex} & z \in \mathbb{C} \\ \text{Variable} & x,y,f & \text{Bit/Bool} & b ::= 0 \mid 1 & \text{Basis Vector} & \eta \in |b\rangle \end{array}$$

$$n\text{-Site Basis-ket State} \quad w ::= \bigotimes_{k=0}^{n-1} \eta_k \qquad \text{Quantum State} \quad \psi ::= \sum_j z_j w_j| \quad o$$

图2 粒子的 QBlue 状态结构。我们将 $z \cdot \bigotimes_{k=0}^{0} \eta_k \equiv z\,\eta_0$ 和 $z_1\eta_1 \otimes z_2\eta_2 \equiv (z_1 * z_2)\eta_1\eta_2$ 简记为。

2 背景与概述

本节以一个小型哈密顿量为例，展示它经过 QBlue$_{\text{BE}}$ 的处理过程，从物理描述到量子电路，并介绍每个步骤所需的背景知识。图 3 展示了各个阶段。

2.1 量子态背景

我们首先介绍量子模拟中使用的量子态的背景信息。

在量子计算中，系统的状态由希尔伯特空间中的一个单位向量描述。量子态由一个或多个量子比特（位点，即物理位置）组成，每个量子比特都是一个二维希尔伯特空间 H门。通过哈密顿量对粒子系统建模时，量子态中的单个量子比特通常称为位点，而量子计算态通常称为量子比特。两者指代相似的概念，但前者描述粒子系统中的单位向量，后者描述量子计算。本文泛指“位点”，并用“量子比特”指代变换后的基于泡利算符的哈密顿量和量子幺正操作。

单个量子比特可表示为一个二维向量（$\begin{smallmatrix} z_1 \\ z_2 \end{smallmatrix}$），其中 z_1 和 z_2是复振幅，且满足 $|z_1|^2 + |z_2|^2 = 1$。使用狄拉克符号，可将其写作$z_1|0\rangle + z_2|1\rangle$，其中 $|0\rangle$ 和 $|1\rangle$ 是计算基态。当 z_1 和 z_2 均不为零时，该量子比特处于 $|0\rangle$和 $|1\rangle$的叠加态 中。多量子比特数据通过张量积构造，例如 $|0\rangle \otimes |1\rangle = |01\rangle$。然而，并非所有多量子比特态都能写成张量积；有些态是纠缠态，例如贝尔对 $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$。

图2定义了本文使用的量子态语法，并使用计算基向量和基矢对这类系统进行形式化描述。单个位点的基向量写作量子态 $\eta_k = |b\rangle$，其中 $k \in [0,1]$ 用于标识一个局部基向量。由 n 个位点组成的基矢态则表示为 $w = \bigotimes_{k=0}^{n-1} \eta_k$。例如，两个位点分别处于基向量 $|0\rangle$和 $|1\rangle$ 时，组合成联合态 $|0\rangle \otimes |1\rangle$，可简写为 $|0\rangle|1\rangle$。

单站点基向量可乘以复振幅 z，形成基矢$z|k\rangle$。对于一个 n 站点系统，我们将基矢写作 $w = z \cdot \bigotimes_{k=0}^{n-1} \eta_k$，并逐分量关联振幅。例如，状态 $z_1\eta_1 \otimes z_2\eta_2$ 可简写为 $(z_1 * z_2)\eta_1\eta_2$。一般而言，量子叠加态的规范写法为 $\psi = \sum_j z_j w_j$。我们还引入零向量 o 来表示空态，它在第二量子化的语义中发挥作用。作为一种便捷的记法，我们用 $|j\rangle_n$ 表示一个 n 量子比特系统，它等于 $|j[0]\rangle \otimes |j[1]\rangle \otimes \ldots \otimes |j[n-1]\rangle$。

2.2 示例：双格点哈伯德模型

哈伯德模型 [45] 描述晶格中相邻位点之间跃迁并相互排斥的电子。我们采用其最小实例 [64]：两个位点，标记为 0 和 1，容纳无自旋费米子，因此每个位点要么为空，要么容纳一个粒子。系统的状态是 C

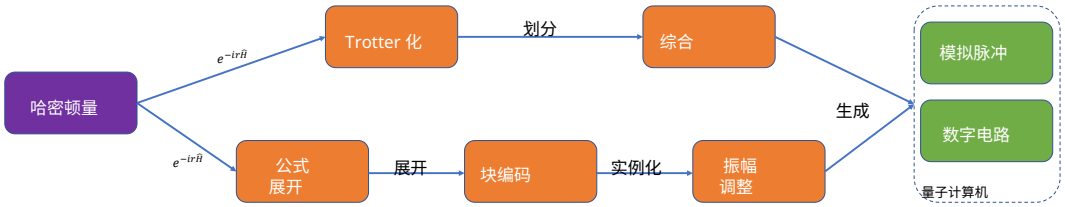


图 3 哈密顿量模拟编译流程。上方流程基于特罗特，下方流程基于 BE。

系统的状态是 $\mathbb{C}^2 \otimes \mathbb{C}^2$ 中的一个单位向量，其基态为 $|m\rangle\,|n\rangle$，其中 $m, n \in \{0,1\}$ 记录位点 0 和 1 的占据数。一般状态是叠加态 $\sum_{m,n} z_{mn}\,|m\rangle\,|n\rangle$，其中 $\sum_{m,n} |z_{mn}|^2 = 1$。这正是两个量子比特的状态空间，因此此类模型天然适合作为量子计算机的目标。

为描述该模型，人们使用哈密顿量（附录 A 中的语法）来描述物理约束。在该模型中，我们对不同约束求和，哈伯德模型包含两个主要组成部分：一个描述位点 1 和位点 0 之间的电子跃迁，另一个描述两个位点之间的排斥作用。以泡利串形式写出的哈密顿量如下所示，其中 $-\frac{1}{2}\,X(0)\circ X(1)\; -\frac{1}{2}\,Y(0)\circ Y(1)$ 项是运动项，而 $\frac{1}{2}\,Z(0)\circ Z(1)$ 项是排斥项。

$$\hat{H} = -\tfrac{1}{2}\,X(0)\circ X(1) - \tfrac{1}{2}\,Y(0)\circ Y(1) + \tfrac{1}{2}\,Z(0)\circ Z(1). \tag{2.3}$$

与上文的哈伯德模型示例一样，哈密顿量是一个厄米矩阵，$\hat{H} = \hat{H}^\dagger$，其特征值就是系统的能级。模拟系统演化时间 r，意味着应用酉算子 $\exp(-ir\hat{H})$；它之所以是酉算子，正是因为 $\hat{H}$ 是厄米的。这就是 QBlue 要编译的酉算子。

上面的示例使用了泡利矩阵运算。量子计算机通过泡利矩阵 ($\mathcal{P}$) 对量子比特进行操作

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

作用于 n 个量子比特的泡利串 ($\hat{P}$) 是 n 个泡利矩阵的张量积。对于一个含 n 个位点的系统，$\hat{}$ 泡利串的形式为 $P_j = \mathcal{P}_n \otimes \cdots \otimes \mathcal{P}_1$,，其中每个 $\mathcal{P}_k$ 都作用于一个量子比特的希尔伯特空间 $\mathcal{H}_2$。哈密顿量程序可以改写为泡利串的加权和：$\hat{H} = \sum_j r_j \hat{P}_j$，其中 $r_j \in \mathbb{R}^2$。

我们用 $Z(j)$ 表示作用于量子比特 j 上、并在其他位置作用单位算符的 Z，因此 $Z(0)\circ Z(1)$ 是泡利串 $Z \otimes Z$。n 个量子比特上的 4^n 个泡利字符串构成所有 $2^n \times 2^n$ 矩阵的一个基，因此，n 个量子比特上的每个哈密顿量都可以表示为泡利字符串的加权和。

2.3 基于特罗特的哈密顿量模拟编译与 QBlue

基于 Trotter 的哈密顿量模拟编译基于 Trotter–Suzuki 分解。如图 3 顶部的流程所示，基于 Trotter 的编译器分为两个步骤：Trotter 化和电路综合。

要编译 $\exp(\hat{H}_1 + \hat{H}_2)$，我们将指数和转换为两个指数函数的乘积 $\exp(\hat{H}_1)\circ \exp(\hat{H}_2)$。然而，只有当各项对易时，和的指数才等于各指数函数的乘积。Trotter 化通过一系列单个泡利串的指数函数来近似 $\exp(-ir\hat{H})$。一阶乘积公式 [58, 83]

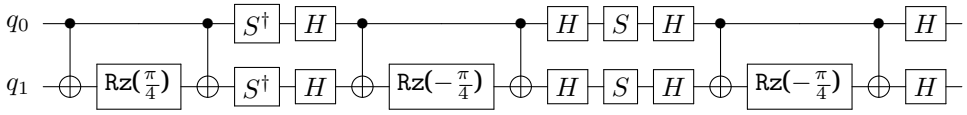


图4 双位点模型中 exp($-i\frac{\pi}{4}\hat{}H$) 的数字电路。从左到右，三个模块 s
在公式（2.4）中分别实现绕 $Z(0)\circ Z(1)$、 $Y(0)\circ Y(1)$ 和 $X(0)\circ X(1)$ 的旋转。

将时间分成 m步，

$$\exp\big(-ir(A+B)\big) \;\approx\; \Big(\exp\big(-i\tfrac{r}{m}A\big)\circ\exp\big(-i\tfrac{r}{m}B\big)\Big)^{m},$$

误差的算子范数至多为 $\frac{r^2}{2m}\|[A,B]\|$ [28],，因此误差会随着 m 增大而减小，并且当 A 和 B 对易时误差为零。二阶 Trotter 分解 [83] 对称地执行这些步骤，将误差降低至 $O(\frac{r^3}{m^2})$。QDrift [17] 按照各项的权重比例随机抽样，因此其成本取决于哈密顿量的总权重，而非项数；MarQSim [18] 则结合了确定性和随机排序。

对于 $r=\frac{\pi}{4}$ 这个例子，一步一阶 Trotter 化给出

$$\exp\big(-\tfrac{i\pi}{4}\hat{H}\big) \;\approx\; \exp\big(\tfrac{i\pi}{8}\,X(0)\circ X(1)\big)\circ\exp\big(\tfrac{i\pi}{8}\,Y(0)\circ Y(1)\big)\circ\exp\big(-\tfrac{i\pi}{8}\,Z(0)\circ Z(1)\big). \quad (2.4)$$

公式 (2.3) 中的三个泡利串两两对易，因为每一对在两个量子比特上都不相同，因此在这个例子中，近似结果就是精确等式。一般而言，这些项并不对易。接着，QBlue 针对四种算法中的每一种，根据其参数计算误差上界，并设置参数（步数 m，或随机化算法的抽样次数），使该上界低于用户指定的 ϵ，然后选择估算成本最低的算法（附录 B）。对于这个例子，它选择了一阶 Trotter 化。

Trotter 化后，QBlue 对量子电路进行综合，以生成数字电路或模拟脉冲。公式（2.4）的每个因子都是围绕泡利串 $\hat{}P$ 的旋转 exp($-i\theta\hat{}P$)。在数字后端上，围绕 $Z(0)\circ Z(1)$ 的旋转是一种标准模式，

$$\exp\big(-i\theta\,Z(0)\circ Z(1)\big) = \mathtt{CNOT}_{0,1}\circ\big(I\otimes\mathtt{Rz}(2\theta)\big)\circ\mathtt{CNOT}_{0,1}, \qquad \mathtt{Rz}(\varphi)=\exp(-i\tfrac{\varphi}{2}Z),$$

而围绕其他泡利字符串的旋转，则可通过对每个量子比特进行基变换化归为该模式： H将 X 映射为 Z， $S^\dagger$ 随后接 H 将 Y 映射为 Z。图4展示了所得的量子电路，该电路能够精确实现公式（2.4）。面向模拟量子计算机的编译会将每个旋转实现为对硬件原生相互作用的定时施加，而不是门序列；因禁离子后端使用能够同时耦合多个离子的全局驱动 [50, 74]。

2.4 基于 BE 的算法

本文主要关注 QBlue$_{\text{BE}}$，它基于基于 BE 的算法。这是将 exp($-ir\ \hat{}H$) 编译为量子电路的另一种方式，其中 $\hat{}H$ 中的子项均为酉算子，可以直接实现为量子电路，即表示为泡利串 $\hat{}H=\sum_j r_j\,\hat{}P_j$，其中 $\hat{}P_j$ 为酉算子。利用块编码技术，我们可以将泡利字符串的线性和转换为一系列控制操作，其中每个控制操作都可以直接实现为量子电路。

最简单的基于 BE 的算法是泰勒展开，我们通过截断泰勒级数展开 $\exp(-ir\hat{}H)$，$\sum_j \frac{-ir\hat{}k}{H^k}!$，从而在归一化后得到求和操作 $\sum_k \theta_k \hat{} P_k$，其中包含一些泡利字符串 $\hat{}\ P_k$ 和 $\sum_k |\theta_k| = 1$。这种操作可以通过酉算子的线性组合（LCU）技术实现为量子电路，具体电路为 $\texttt{PRE}; \texttt{SELECT}; \texttt{PRE}^\dagger$，其中 SELECT 电路如图 6 所示。 PRE 制备叠加态，而 SELECT 是一个多路复用器电路，其中包含 $\hat{}$ 一系列受控操作，每个操作都将一个泡利串 P_k 实现为量子电路；也就是说，受控操作可以视为量子条件操作 $\hat{}\texttt{if}\,(x = k)\,\{P_k\}$。详情见第3节。

2.5 保证

$\hat{}$ 对于厄米泡利字符串哈密顿量 H、时间 r 和误差界 ϵ，定理 1 指出，编译后的程序 U 满足 $\|[\![U]\!](\psi) - [\![\exp(-ir\hat{}H)]\!](\psi)\| < \epsilon$，对任意归一化态 ψ均成立。该证明通过四种基于 BE 的算法确保编译器正确性，每种算法都有各自的流水线。QBlueBE 界定了其选择算法及其参数时的范围。

3 QBlueCompilation过程

本节介绍 QBlueBE 编译步骤，如图 3 底部路径所示，并说明 Rocq 如何组织和验证正确性证明。

3.1 通用过程

编译器将哈密顿量模拟表达式 $\exp(-ir\ \hat{}H)$ 转换为量子电路，其中 $\hat{}H$ 是 QBlueBE 哈密顿量程序，r 是时间参数；该量子电路以用户定义的误差界 ϵ 近似实现 $\exp(-ir\ \hat{}H)$，并可在量子硬件上执行。我们将编译建模为如下形式的判断：$(\hat{}H,\ r) \gg U$。QBlueBE 编译器支持多种编译路径，帮助用户从四种基于 BE 的算法中选择最佳方案。

四种不同的编译算法具有相同的通用流程，如图 3 底部路径所示。该流程包含以下三个编译步骤。

1. **展开公式**：主要步骤是将哈密顿量模拟算子展开$\exp(-ir\ \hat{}H)$ 为泡利字符串的线性和。这些算法使用两种矩阵展开公式：泰勒展开和 Jacobi–Anger 展开。2. **块编码**：基于 BE 的算法的一个关键组成部分是块编码，它通过可在量子计算机上执行的酉算子实现非酉矩阵。主要有两种块编码方法：线性组合酉算子（LCU）和量子游走。给定的非酉操作可以改写为线性和操作，而这些方法通过多路复用器实现求和。我们将在下一节讨论这些方法。
3. **振幅调整**：振幅调整步骤通过提升正确进行哈密顿量模拟的振幅概率来改进结果，包括无相位振幅放大（OAA）和量子信号处理。振幅调整操作以通用框架的形式实现，将哈密顿量模拟块编码作为输入，并生成重复模式以提升振幅。

现在，我们给出将哈密顿量模拟$\hat{}$约束表达式 H 编译为量子电路的正确性定理。

▶**定理 1**（编译正确性）。 给定$\hat{}H$、一个时间周期 r以及期望误差 ϵ，我们将其模拟 exp(-$ir\hat{}H$)编译为量子电路，形式为（$\hat{}H$， r） $\gg U$。对于任意状态 ψ，令 $\psi_1 = [\![U]\!](\psi)$；量子电路与模拟之间的距离受 ϵ限制，即$\|\psi_1 - [\![\exp(-ir\ \hat{}H)]\!](\psi)\| < \epsilon$。

证明。 我们在 Rocq 中为四种算法的四个不同版本完成了完全机械化证明。◀

定理指出，编译后的量子电路 U 与语义定义 exp(-$ir\ \hat{}H$)之间的距离小于用户定义的上界 ϵ，我们将在下一节将其与其他因素联系起来。我们的编译器经过端到端验证，所有定理均已在 Rocq 中完全机械化。

3.2 单独的基于 BE 的算法

本节讨论基于上述一般 1
上述框架。

3.2.1 基于线性组合酉算子的泰勒展开

如第 2.1 节所述，执行哈密顿量模拟的一种简单方法是使用泰勒展开。我们在上述一般情形下，按照三个步骤讨论该算法。第 2.4 节解释了**展开公式**步骤，该步骤将 exp(-$ir\ \hat{}H$) 展开为截断泰勒级数。以下介绍另外两个步骤。

3.2.1.1 块编码

在块编码步骤中，泰勒展开利用线性组合酉算子算法，通过组合多个小型幺正算子，在量子计算机上实现非酉操作。给定一个 2^n 维哈密顿量 $\hat{}H = \sum_j^m r_j\ \hat{}P_j$，并将其作用于 n 量子比特数组 x，我们分配一个 $\lceil \log m \rceil$ 辅助量子比特数组 y；线性组合酉算子算法由三个依次作用于 x 和 y 的部分组成，如下所示： PRE； SELECT； PRE†。其中， PRE 准备叠加态 $\sum_j \sqrt{\frac{|r_j|}{\lambda}}\ |j\rangle$，其对应 $\lambda = \sum_j |r_j|$； SELECT 在 $y = j$ 时将 $\hat{}P_j$ 作用于 x （当 $\overline{r_j} < 0$ 时则翻转符号）。下面讨论这两个部分。

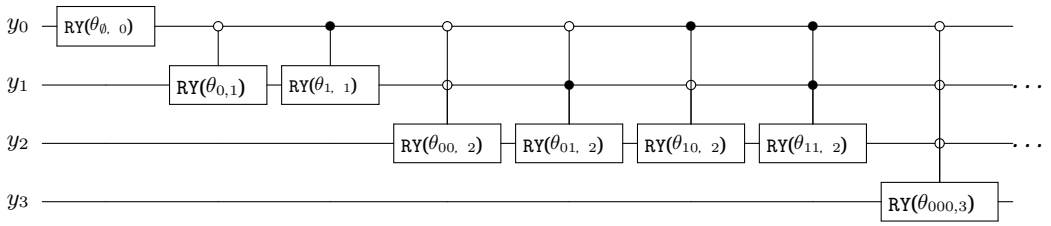


图5 基于二叉树分解和 RY 量子门的 PRE 量子电路。

图5展示了通过 Grover-Rudolph算法制备 $\lceil \log m \rceil$ 个辅助量子比特叠加态的 PRE 量子电路。该算法将态制备问题转化为二叉树分解问题。在电路生成中，对于一个 $\lceil \log m \rceil$ 量子比特态，第 k 个量子比特（$k \in [0, \lceil \log\ m \rceil)$ 有 2^k 个多路复用器，其控制门为 RY 门；其中，第 k 个量子比特有带 k 个节点的多路复用器，因此第一个量子比特只有一个 RY 门

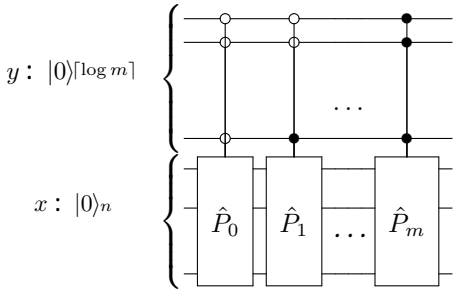


图 6 基于多路复用器的 SELECT 量子电路。

（$k = 0$）。为更好地描述该量子电路，我们假设长度为 k 的二进制位串 $b \in \{0,\ 1\}^k$，其中 $k \in [0, \lceil \log\ m \rceil)$，即第 k 个量子比特需要 2^k 个比特串，每个比特串包含 k 位。我们记录第 k 个量子比特的 RY（$\theta_{b,\ k}$）门中的角度值 $\theta_{b,\ k}$，如下所示（假设 $t = \lceil \log m \rceil$）。

$$\theta_{b,k} \triangleq 2 \arctan 2\left(\sqrt{\sum_{j=b\cdot 2^{t-k}}^{(b+1)2^{t-k}-1} |r_j|},\ \sqrt{\sum_{j=b\cdot 2^{t-k}}^{b\cdot 2^{t-k}+2^{t-k-1}-1} |r_j|}\right)$$

例如，$\theta_{\emptyset,\ 0}$ 等于 2arctan $2\left(\sqrt{\sum_{j=2^{t-1}}^{2^t-1} |r_j|},\ \sqrt{\sum_{j=0}^{2^{t-1}-1} |r_j|}\right)$。给定初始态 $|0\rangle^{\lceil \log m \rceil}$，整个量子电路制备出 $\sum_j \sqrt{\frac{|r_j|}{\lambda}} |j\rangle$；每个均匀受控的 RY都被编译为 RY和 CNOT 量子门。

如图 6 所示，SELECT 操作是一个多路复用器，它将 j的泡利串应用编码为受控操作，受控节点位于 j上。该编码将索引 j 写入为一个 $\lceil \log m \rceil$ 位字符串，并处理每一位：1 编码为标准受控节点（黑色节点），0 编码为带有受控节点的 X 门（白色节点）。图 6 展示了受控节点编码的示例，其中白色和黑色节点分别表示对 0 位和 1 位的条件。

3.2.1.2 振幅调整

在振幅调整步骤中，泰勒展开使用 OAA 算法来放大目标态的振幅。给定上述应用于两个寄存器 x 和 y的 LCU 量子电路，我们定义一个对全 0 比特串执行反射的反射操作，如下所示。

$$R \triangleq \mathtt{X}(u); \mathtt{H}(u); \mathtt{if}\ (y = 0)\ \{\mathtt{X}(u)\}; \mathtt{H}(u); \mathtt{X}(u)$$

这里，if $(y = 0)$ {X(u)} 是一个量子条件操作，通过多路复用器实现，用于查询量子数组 y 是否全为 0（$y = 0$）。

一步 OAA 算法实现为 OAA $\triangleq$ LCU；R；LCU†；R；LCU，其中 R 已在上文定义，LCU 已在第 3.2.1.1 节中定义。一般而言，将振幅 s 提升需要进行 $O(\frac{1}{s})$ 轮；我们使用一次额外的 RY旋转，将每个分段的振幅补齐至恰好 $\frac{1}{2}$，因此单轮即可精确达到目标。OAA 框架是一个顶层框架，用于封装上文开发的 LCU 操作，即 OAA(LCU)。

3.2.1.3 效率

最终，我们验证了泰勒展开算法的定理 1，并确立了误差界、时间周期和步数之间的以下关系。对于 $M \geq 1$ 个模拟分段，形式化验证的误差预算引理表明，局部误差界 $\delta \leq \frac{\epsilon}{M}$ 蕴含 $M\delta \leq \epsilon$。这为组合泰勒模拟分段提供了所需的误差预算。

3.2.2 使用量子游走进行量子比特化

量子比特化利用块编码构造量子游走算子，其幂作用于 $\frac{H}{\lambda}$ 时会应用切比雪夫多项式；将这些幂与适当的权重结合，可得到 $\exp(-ir\hat{H})$。该算法遵循图 3 所示的流程。

3.2.2.1 展开公式

采用量子游走方法的量子比特化使用雅可比–安格尔展开近似 $H=\sum_j r_j P_j$ 的时间演化，如下所示。

$$\exp(-ir\hat{H}) \approx J_0(-\lambda r)+2\sum_{k=1}^{K} i^k J_k(-\lambda r)\, T_k\left(\frac{\hat{H}}{\lambda}\right) \tag{1}$$

这里，$\lambda=\sum_j |r_j|$。我们在 K 处截断展开，$J_k(r)$ 是贝塞尔函数，其定义如下。

$$J_k(r) \triangleq \frac{1}{\pi}\int_0^{\pi} \cos(k\theta - r\sin(\theta))d\theta$$

注意，函数参数 k 和 r 是数字；因此，计算成本极低。$T_k(\hat{H})$ 项使用截断的第一类切比雪夫多项式近似 $\cos(\hat{H})$，其递推公式如下所列。

$$T_0(\hat{H}) \triangleq I \qquad T_1(\hat{H}) \triangleq \hat{H} \qquad T_{n+1}(\hat{H}) \triangleq 2\hat{H}\circ T_n(\hat{H}) - T_{n-1}(\hat{H})$$

在公式 (1) 中，我们需要通过执行 $\frac{H}{\lambda}$ 来调整 $\hat{H}$ 的振幅，因为切比雪夫多项式仅在 $[-1,1]$ 上有界。展开收敛很快。每个 T_k 的绝对值都不超过 1，因此将求和截断在 K 处，误差至多为 $\sum_{k>K} 2|J_k(\tau)|$；一旦 k 超过 τ，$J_k(\tau)$ 就会以快于指数的速度衰减。实际中，我们将时间 r 分成 $M=\lceil \lambda r \rceil$ 段，因此每段的长度为 $\tau \le 1$，并选取尾项小于 $\frac{\epsilon}{4M}$ 的最小 K。这样通常每段只剩下少数几项。

3.2.2.2 块编码

我们使用与第 3.2.1.1 节相同的 PRE 和 SELECT 构造量子游走，并将它们作用于 x 和 y 寄存器：

$$\mathrm{WALK} \triangleq \mathrm{SELECT};\ \mathrm{PRE}^{\dagger};\ R;\ \mathrm{PRE}$$

这里，R 是前文定义的关于 $|0\rangle$ 在 y 上的反射，因此 $\mathrm{PRE}^{\dagger};\ R;\ \mathrm{PRE}$ 是关于 $|G\rangle=\mathrm{PRE}|0\rangle$ 的反射，即振幅放大的扩散操作 [16]。它的幂直接给出切比雪夫项：$\langle G|\,\mathrm{WALK}^k\,|G\rangle = (-1)^k T_k(\frac{H}{\lambda})$，因此公式 (1) 中的第 k 项就是 WALK 的一次幂。这里，WALK^k 表示重复 WALK k 次。

3.2.2.3 振幅调整

我们将 WALK^0、……、WALK^K 与另外一层 LCU 层结合。我们制备一个外部寄存器，其振幅为 $\sqrt{\frac{|\beta_k|}{B}}$，其中 $\beta_0=J_0(\tau)$、$\beta_k=2J_k(\tau)$ 和 $B=\sum_k|\beta_k|$；相位 i^k 以及

β_k 的符号应用于该寄存器，其比特控制 $\texttt{WALK}^{2^j}$。填充旋转将成功振幅精确设置为 $1\ 2$，因此，对整个片段而非单个 WALK 应用一次 OAA 轮次，即可使该片段具有确定性。

3.2.2.4 效率

我们针对量子游走算法的量子比特化验证定理 1，并确立误差界、时间周期与步数之间的以下关系。对于模拟时间 r，将其划分为 M 个时间段，每段模拟时长为 $\frac{r}{M}$。在满足分段级正确性和非扩张性假设的条件下，我们在 Rocq 中证明，累积模拟误差不超过 $M\delta$，其中 δ 是单个编译分段的误差界。因此，选择 $\delta \leq \frac{\epsilon}{M}$ 可保证总体误差至多为 ϵ。

—

3.2.3 使用量子信号处理进行量子比特化

量子信号处理建立在量子比特化的基础上：它不再通过线性组合酉算子组合 WALK 的幂，而是直接对 WALK 应用多项式，因此不需要 OAA。我们使用广义量子信号处理 [65], ，它将受控游走步骤与一个额外量子比特 u 上的旋转交错执行：

$$\texttt{QSP}(\Phi) \triangleq \texttt{PRE};\ R_u(\phi_0);\ \big(\texttt{C}_u\texttt{WALK};\ R_u(\phi_1)\big);\ \ldots;\ \big(\texttt{C}_u\texttt{WALK};\ R_u(\phi_{2K})\big);\ (\texttt{WALK}^\dagger)^K;\ \texttt{PRE}^\dagger$$

此处，当 $u = 0$ 时， $\texttt{C}_u\texttt{WALK}$ 应用 WALK ，且每个 R_u（ϕ_j）都是作用于 $\hat{}u$ 的单量子比特旋转。在 H 的每个本征向量上， WALK 都表现为相位 z，因此受 $2K$ 控制的步骤会应用一个 $2K$ 次多项式 P（z），而 K 逆游走会将其平移为 $z^{-K}P$（z），这与从 $-K$ 到 K 的雅可比–安格尔级数相符。通过逐次剥离一个旋转，可根据 P 的系数求出这些角度：给定满足单位圆上 $|P|^2 + |Q|^2 = 1$ 的互补多项式 Q，选择每一层，使得撤销该层后，$(P,\ Q)$ 的次数降低一 [65]。

算法 1 量子信号处理相位因子优化求解器（QSPBFGS）

要求：目标实多项式 f，次数为 d，在 $[-1,1]$ 上满足 $|f| < 1$，容差为 ϵ。确保：对称相位向量 $\Phi = (\phi_0, \ldots, \phi_d)$，满足 $\phi_k = \phi_{d-k}$。1：令 $\tilde{d} = \lceil \frac{d+1}{2} \rceil$，并选取节点 $x_j = \cos\left(\frac{(2j-1)\pi}{4\tilde{d}}\right), j = 1, \ldots, \tilde{d}$（$T_{2\tilde{d}}$ 的正根）。2：定义 ~~$L(\hat{\Phi})$~~ $= \frac{1}{\tilde{d}} \sum_j \left|\mathrm{Re}\ 0|U_\Phi(x_j)|0 - f(x_j)\right|^2$，其中 $\hat{}\Phi$ 包含 $\tilde{d}$ 个自由相位。3：设定 $\hat{}\Phi^{(0)} = (\frac{\pi}{4}, 0, \ldots, 0)$，初始化逆 Hessian 近似 B_0，并设定 $t = 0$。4：当 $L(\hat{}\Phi^{(t)}) > \epsilon$ 且 L-BFGS 仍有进展时，执行以下操作：5：计算 $\nabla L(\hat{}\Phi^{(t)})$ 并使用 L-BFGS 步更新 $\hat{}\Phi^{(t+1)}$； $t \leftarrow t + 1$。6：循环结束。7：使用牛顿迭代步细化 $\hat{}\Phi^{(t)}$，使 $\mathrm{Re}\ 0|U_\Phi(x_j)|0 = f(x_j)$，直至残差低于 ϵ。8：返回由 $\hat{}\Phi^{(t)}$ 构造的完整对称 Φ。

对于高次多项式而言，求出相位角 Φ 是最困难的部分，因此我们采用 Dong 等人提出的基于优化的求解器 [32], ，其总结见算法 1。该求解器并不从多项式根中逐一计算相位（在高阶情况下这种方法会变得数值不稳定），而是将相位视为未知量并直接进行拟合：在第 1 步，它在切比雪夫节点处计算 QSP 序列，并使用 L-BFGS 最小化其与目标多项式之间的平方差。由于相位具有对称性，只有其中一半是自由变量；从平凡相位（$\frac{\pi}{4}$，0，……，0）开始，即使对于数百阶的多项式，优化也能可靠收敛；在我们的实现中，经过几次牛顿法

迭代步即可将结果优化至所需容差。我们使用该求解器计算 QSVT 相位（第3节.2.4）；而我们的 QSP 路径则通过逐层剥离来计算其角度 [65]。

3.2.3.1 效率与误差界

为建立截断阶数、模拟时间与近似误差之间的关系，令 r 表示总模拟时间，$\lambda = \sum_j |r_j|$ 表示线性组合酉算子的归一化因子，ϵ 表示目标误差界。量子信号处理编译将演化划分为

$$M = \max\{1, \lceil \frac{\lambda r}{10} \rceil\}$$

个片段，设 $r \geq 0$，每个片段的归一化模拟时间为 $\tau = \frac{\lambda r}{M}$。令 d 表示雅可比–安格截断阶数。相应的广义量子信号处理多项式的次数为 $2d$。

1. **谱分解：** 对于归一化哈密顿量 $x = \frac{H}{\lambda}$，其谱位于 $[-1, 1]$中，时间演化函数可利用雅可比–安格尔展开表示为：

$$\exp(-i\tau\, x) = J_0(\tau)T_0(x) + 2\sum_{k=1}^{\infty}(-i)^k J_k(\tau)T_k(x),$$

其中，T_k 是第一类切比雪夫多项式，J_k 是第一类贝塞尔函数。

2. **级数截断：** 将展开截断至 d 阶会引入多项式 1
近似误差的上界为

$$\delta_{\text{poly}}(d, \tau) = 2\sum_{k=d+1}^{\infty} |J_k(\tau)|.$$

由于对于 $x \in [-1, 1]$，$|T_k(x)| \leq 1$，因此该贝塞尔尾项为截断误差提供了统一上界：

$$\sup_{x\in[-1,1]} |\exp(-i\tau\, x) - P_d(x)| \leq \delta_{\text{poly}}(d, \tau),$$

其中

$$P_d(x) = J_0(\tau)T_0(x) + 2\sum_{k=1}^{d}(-i)^k J_k(\tau)T_k(x).$$

3. **截断阶数与电路步骤：** 编译器使用局部容差 d选择截断阶数

$$\epsilon_0 = \frac{\epsilon}{4M}.$$

预期的截断条件为

$$\delta_{\text{poly}}(d, \tau) \leq \epsilon_0.$$

广义QSP构造生成一个次数为$2d$ 的多项式，并在每个分段中实现 $2d$次受控量子游走操作和 d 次逆游走操作。因此，电路构造将量子游走操作的数量与所选截断阶数直接关联起来。

为了将这些近似界与机械化误差分析联系起来，令 δ_{phase}表示与相位或角度合成相关的误差预算，包括互补多项式构造；令 δ_{impl} 表示剩余的实现误差预算，包括多项式缩放和电路实现误差。所有误差预算均按分段定义。

经机械化证明的 Rocq 引理 `component_budget` 确立了：对于 $M \geq 1$ 和非负误差预算，

$$\delta_{\text{poly}}(d, \tau) + \delta_{\text{phase}} + \delta_{\text{impl}} \leq \frac{\epsilon}{M}$$

意味着

$$M\left(\delta_{\text{poly}}(d, \tau) + \delta_{\text{phase}} + \delta_{\text{impl}}\right) \leq \epsilon.$$

因此，该机械化结果确立了局部误差预算、模拟分段数量与全局容差之间的算术关系。当各个组件对完整的每段语义误差给出界限，且满足分段组合假设时，累积的模拟误差便有界于 ϵ。相应的局部近似和量子电路级正确性义务必须另行确立。

3.2.4 Quantum 奇异值变换

QSVT 也会对 $\frac{H}{\lambda}$ 应用多项式，但它将相位施加在块编码本身上，而非受控游走上。展开公式采用带有贝塞尔系数的 Jacobi–Anger 展开。块编码步骤使用第 3.2.1.1 节中的 LCU 块 $U = \text{PRE}; \text{SELECT}; \text{PRE}^{\dagger}$。关键区别在于振幅调整步骤。

3.2.4.1 振幅调整

QSVT 交替使用 U 和 $U^{\dagger}$，并施加投影算子相位 $\exp(-i\phi_j(2\Pi - I))$，其中 $\Pi = |0\rangle\langle 0|_y$；每个相位都是作用于一个辅助量子比特的多重受控标志，后接一个 `RZ`。采用合适的角度 $\phi_1, \ldots, \phi_d$ 后，结果的左上角块是关于 $\frac{H}{\lambda}$ 的 d 次多项式。单个序列只能生成具有固定奇偶性的多项式，因此我们构造两个序列：一个偶次多项式对应雅可比–安格尔级数的余弦部分，一个奇次多项式对应其正弦部分。额外的一个量子比特取出各自的实部（分别使用 ϕ 和 $-\phi$ 运行序列并取平均），第二个量子比特将两者合并，得到 $\frac{1}{2}(P_{\cos} + iP_{\sin}) \approx \frac{1}{2}\exp(\frac{-i\tau}{\lambda}H)$。由于振幅已经是 $\frac{1}{2}$，进行一轮 OAA 即可使该片段成为确定性的。角度由算法 1 给出。

3.2.4.2 效率

在上述设置下，我们建立步骤与误差界之间的以下关系。对于模拟时间 r，QSVT 编译会构造 $M = \lceil \frac{\lambda r}{50} \rceil$ 个片段，每个片段的目标演化时长为 $\frac{r}{M}$，归一化片段时长为 $\tau = \frac{\lambda r}{M}$。编译器将多项式截断容差和多项式缩放参数 η 均设为 $\frac{\epsilon}{8M}$。在片段级正确性和非扩张性假设下，我们在 Rocq 中证明，累积模拟误差不超过 $M\delta$，其中 δ 表示经 OAA 处理后一个完整 QSVT 片段的误差。因此，要求 $\delta \leq \frac{\epsilon}{M}$ 即可保证整体模拟误差至多为 ϵ。局部误差界由这些参数推出。截断和 $(1 - \eta)$ 缩放各自引入的误差均不超过 $\frac{\epsilon}{8M}$，因此编码算子与 $\exp(\frac{-i\tau}{\lambda}H)$ 的差距不超过 $\gamma = \frac{\epsilon}{4M}$；一次 OAA 迭代可将误差控制在 $\gamma + 2\gamma^2 \leq \frac{\epsilon}{2M}$ 以内，满足 $\delta \leq \frac{\epsilon}{M}$

并留有两倍的余量。相位通过数值计算得出，并与其目标多项式进行核验；在我们的基准测试中，最大残差为 8.5×10^{-13}。

4 评估

本节对 QBlueBE 进行实验评估。我们首先定义用于比较的量子程序基准测试套件和最先进的基线框架。随后，我们展示结果，并围绕以下研究问题进行组织：

- RQ1：适用性：QBlueBE 能否在允许的误差界内，将基于 BE 的量子模拟程序编译为量子电路？RQ2：有效性：与其他框架相比，QBlueBE 能否为广泛的程序生成有效的量子
- 电路？RQ3：正确性与可靠性：与其他框架相比，QBlueBE 能否生成正确的电路并获得可靠的结果？
-

4.1 实验设置

名称	#	平均量子比特数	量子比特范围	平均泡利字符串数	泡利字符串数量范围
Small Set	220	10	4 - 32	27	6 - 80
Medium Set	36	13	8 - 40	518	200 - 1884
Large Set	23	19	12 - 72	2440	2000 - 368021

表 1 实验数据信息。# 列表示测试用例的数量。

4.1.0.1 基准测试套件

为评估 QBlueBE，我们从实际应用中构建了一个包含 279 个哈密顿量模拟程序的基准测试，这些程序用于模拟与电子行为相关的物理和化学系统。该数据集收集自一个量子材料模拟项目 [92, 25],Genesis [26], , 以及先前哈密顿量模拟项目中的测试程序 [18, 53]。其中包括物理模型，例如一维横向场伊辛模型和 Hubbard模型；还包括量子化学模型，例如对 BeH_2、 $eL42$ 蛋白质和乙烯的粒子模拟。完整列表见我们的工件，我们在表 1 中总结了这些程序的属性。我们对其中 276 个程序运行了所有实验：最大的三个程序（两个包含 368，021 个泡利字符串的苯哈密顿量，以及一个包含 152，371 个泡利字符串的 C_2H_4 片段）超过了两小时的单任务时间限制，因为仅计算它们的步数就需要数小时到数天。

选择基准测试时，我们按泡利串长度将数据集划分为小型、中型和大型三个集合，以保持数据集的多样性，并在多样化的环境中评估 QBlueBE 。小型集合中的程序是用于基于量子计算的哈密顿量模拟的传统基准测试程序，而大型集合中的程序（例如苯和C_2H_4）则是实际化学应用中的小型组成部分。测试数据集时，我们将模拟时间 r 设为 $\frac{\pi}{16}$、 $\frac{\pi}{4}$、 π和 4π，并将误差界设为 0.0002、0.002、0.02 和 0.1。 _

4.1.0.2 基线系统

我们将 QBlueBE 的效率与一个先进框架 PennyLane[11], 进行比较。该框架包含一个采用基于 BE 的算法的量子模拟框架。PennyLane 是一款工业编译器，与 Qiskit Nature [46], 等其他研究工作类似，主要专注于量子化学和量子粒子应用，例如计算基态能量。我们希望对其进行比较，因为它是业界具有代表性的基于 BE 的量子模拟编译器，能够将实际应用编译到量子计算机上，而非专注于某些子程序 [52] 或使用其他类型的机器 [26] 和编译方法 [8, 27]。

4.1.0.3 硬件与实现

我们将 QBlueBE 编译器从 Rocq 提取为 OCaml，并在一台 Ubuntu 计算机上进行了评估。该计算机配备 8 核第 13 代 Intel i9 处理器和 16GB DDR5 内存。

4.2 RQ1：适用性



图 7 BE算法步骤之后，QBlueBE 面向 IBM 量子计算机的编译步骤。

为回答第一个研究问题，我们通过将三个数据集上的哈密顿量程序模拟编译为 IBM 数字电路，并比较 QBlueBE 与 PennyLane 之间的编译差异，来评估 QBlueBE 。

我们的编译遵循图 3 中的编译流水线，并在该流水线之后增加了两个步骤：1）我们应用标准的量子电路优化器 VOQC [44], 来优化生成电路；2）我们执行面向 IBM 量子计算机结构的电路路由步骤，通过插入必要的量子门（例如 SWAP 门），使生成电路能够在 IBM机器上执行。图 8 比较了 QBlueBE 与 PennyLane 在输入误差界 0.0002、0.002、0.02 和 0.1，以及模拟时间段 $\frac{\pi}{16}$、$\frac{\pi}{4}$、π 和 4π 下的编译时间。在每种设置中，对于 PennyLane 能够编译的 223 个程序中的 212–220 个，QBlueBE 的编译速度更快，中位加速比为 2.4–3.4×。此外，PennyLane 未能编译 53 个程序；所有编译失败的程序都包含超过 250 个泡利字符串。

结果表明，QBlueBE 能够在合理的时间内，为所有设置下的程序生成哈密顿量模拟结果。这说明 QBlueBE 适用于不同的程序。

4.3 RQ2：有效性

鉴于在哈密顿量模拟中，编译时间通常是最不需要关注的因素，而优化数字门才是主要因素，我们现在通过将 QBlueBE 与 PennyLane 进行比较来考察其有效性。具体而言，我们主要根据电路门节省量，将 QBlueBE的整体编译性能与 PennyLane 进行比较。我们仅比较 PennyLane 能够编译的程序。为比较整体性能，对于每个程序，我们输入所需的输入误差界和模拟时间段，并让两个平台分别选择最优结果电路。

XX：16 验证与评估综合量子模拟编译器

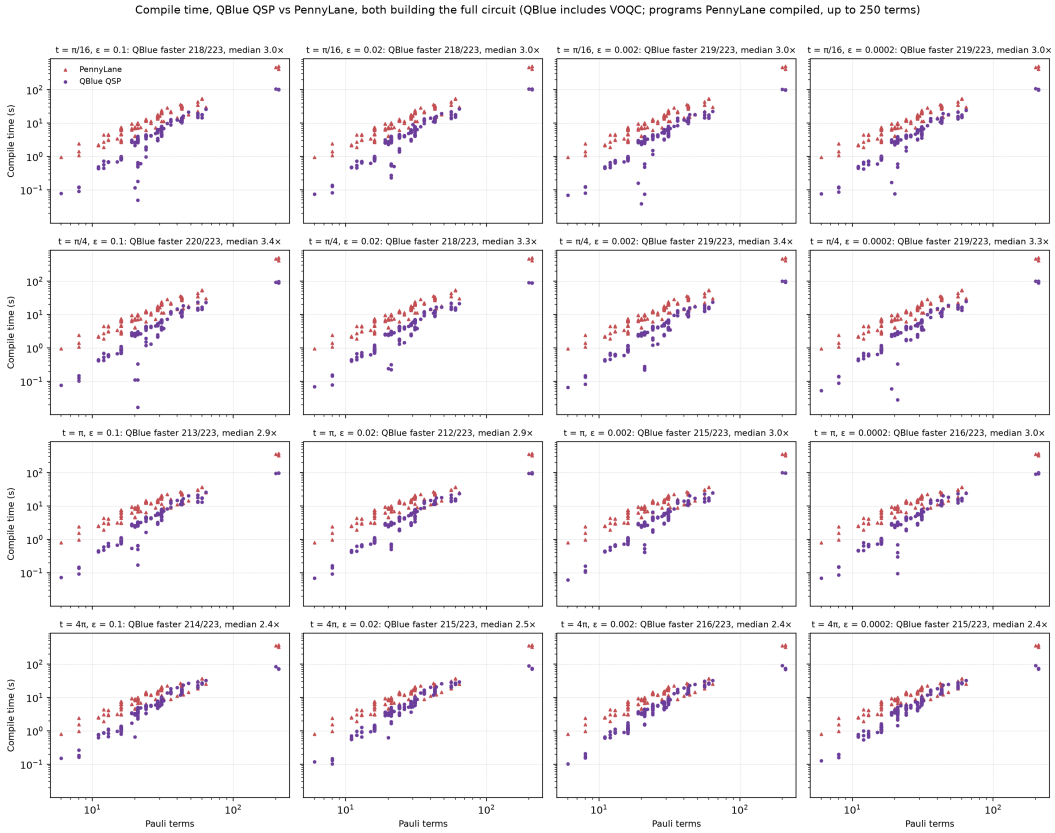


图 8 QBlue 与 PennyLane 的编译时间对比；误差界：0.0002、0.002、0.02 和 0.1，以及模拟时间段： $\pi/16$、 $\pi/4$、 π、 4π。

我们在图 9 中展示了第 4.1.0.1 节所述误差界和时间周期下的实验结果。结果表明，QBlueBE 的编译性能显著优于 PennyLane。编译结果显示，与小型程序相比，QBlueBE 在中型程序上的表现更佳。平均而言，QBlueBE 在中型集合和小型集合上的门数减少量分别约为 98.9 % 和 95.6%；由于 PennyLane 无法编译大型集合中的程序，因此我们不对大型集合进行比较。在我们比较的全部 3,568 个案例中，QBlueBE 生成的量子门数量均少于 PennyLane。在较长模拟时间段和更严格的误差界条件下，中型集合程序的差距最大。对于 SYK 模型，在 4π和 $\epsilon = 0.0002$条件下，QBlueBE 生成的量子电路规模约为 PennyLane 的 $1\,108$ （1.6×10^8 对比 1.7×10^{10} 个量子门）；对于 B3 杂质模型，该比例约为 $1\,107$ （2.4×10^8 对比 2.5×10^{10}个量子门）。在小型集合中，氩程序的比例为 $1\,46$ （$4.5M$ 对比 $207M$）。这一结果证明了 QBlueBE的编译性能，尤其是在规模更大的实际量子应用中。

门数减少几乎不随误差界变化：从 0.1 降至 0.0002 时，其数值基本保持不变。它会随时间周期略有增加，从 $\frac{\pi}{16}$ 时的约 94% 增长到 π 和 4π 时的 96–97%。

在将 QBlueBE 与 PennyLane 进行比较时，我们发现 PennyLane 的量子门估算器存在一个问题。PennyLane 的量子门估算器在不进行分解的情况下比较量子门；例如，它不会将长链控制门（如四量子比特托福利门）分解为单量子比特门和双量子比特门，而 QBlueBE 的估算器会将所有门都分解为单量子比特门和双量子比特门。PennyLane 的量子门估算器可能无法反映量子门的实际成本，因为一些长链控制

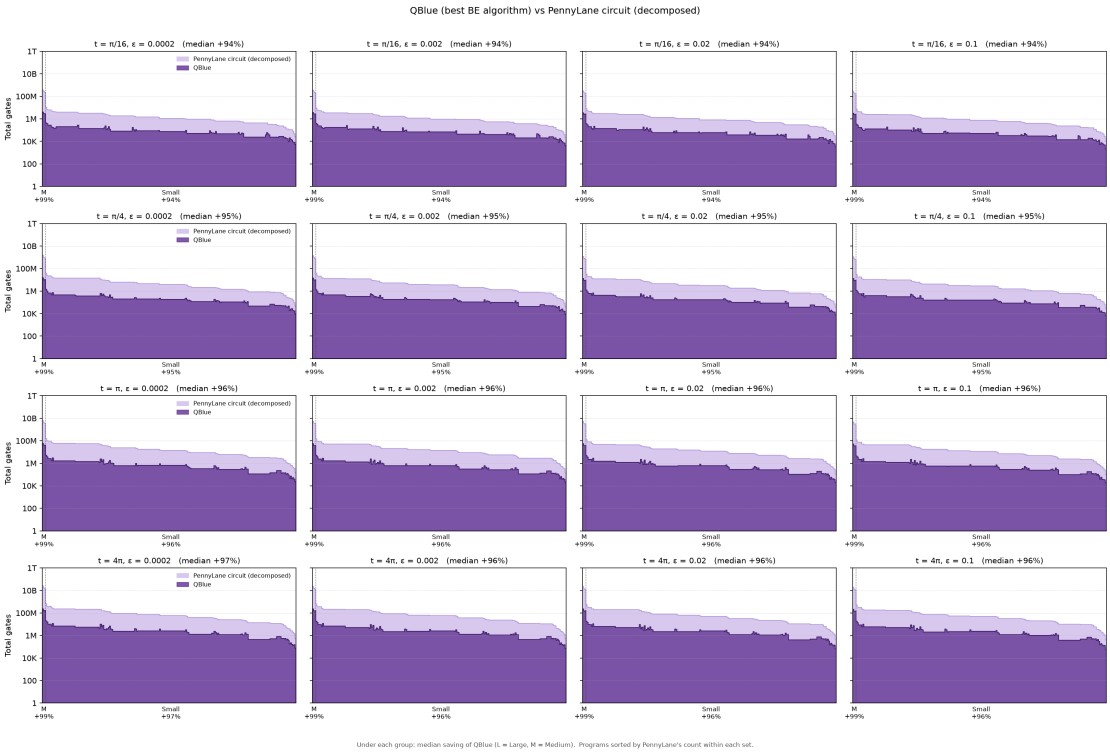


图 9 与 PennyLane 相比，QBlueBE 在误差界 0.0002、0.002 0.02 0.1 4π π _ __ 、和、以及模拟时间：、、$\frac{\pi}{4}$ 和 $\frac{\pi}{16}$ 方面的优化缩减。在每个图中，上方面板将结果与 PennyLane 进行比较。我们比较每个程序中表现最佳的算法。

g门的实现成本可能很高。因此，我们使用 QBlueBE 中的量子门估算器来t比较。

4.4 RQ3：正确性与可靠性

现在，我们通过与 PennyLane 进行比较来考察 QBlue 的正确性与可靠性。正确性衡量编译器在生成的电路中是否存在错误，而可靠性衡量编译器是否无法生成电路。我们在图 10 中记录了 PennyLane 存在的问题。

PennyLane 的量子信号处理算法存在可靠性问题。在量子信号处理中，编译器应生成正确的角度，以近似正确的量子电路；也就是说，该算法包含一个角度求解器。如果求解器未生成正确的角度，编译器就无法生成正确的量子电路。QBlue 中的角度求解器已验证为正确，并能正确处理所有情况。

PennyLane 的角度求解器存在一个主要问题：它依赖于生成多项式次数，而且只检查一个贝塞尔项。我们在全部 4,416 个案例上运行了 PennyLane 的角度求解器（涵盖 4 个误差界、4 个时间周期下的 276 个程序），结果见图 10 顶行。求解器仅在一个案例中生成了可用的角度。在 2,071 个案例中，它因多项式的值超过 1 而拒绝处理；在 2,022 个案例中，它在根分裂这一关键步骤发生崩溃。另有 20 个案例中的多项式过大，无法求解；在剩余的 302 个案例中，它返回的角度所产生的误差超过了用户设定的界限。

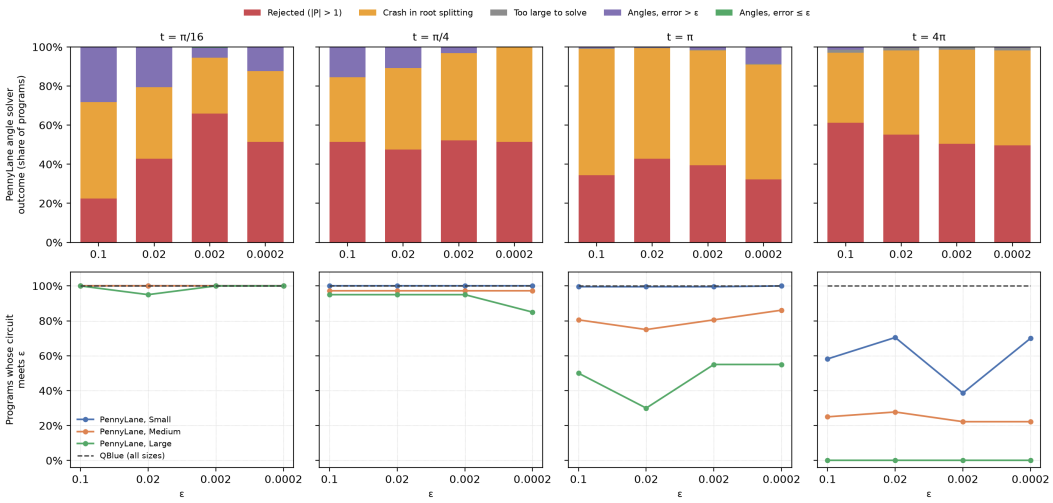


图 10 PennyLane 在误差界 0。1、0。02、0。002 和 0。0002，以及时间周期 $\frac{\pi}{16}$、$\frac{\pi}{4}$、π 和 4π 下，全部 276 个程序的正确性与可靠性。上图：PennyLane 角度求解器的结果。下图：其 PennyLane 电路满足所请求误差界的程序占比；QBlue$_{\text{BE}}$ 在每种情况下都满足要求。

正确性保证的一个关键因素是它如何处理误差界。用户输入误差界后，QBlue 能正确生成满足该误差界要求的量子电路。PennyLane 则无法做到这一点。如图 10 底行所示，在大型程序的 320 个案例中（涵盖不同误差界和 4 个时间周期下的大型程序），PennyLane 仅在 191 个案例中满足误差界要求；在 4π [1] 中则一个也没有。对于其他程序，PennyLane 生成的量子电路不正确，其误差界超过了用户要求的值。在最坏情况下，误差约为请求误差界的 10^4 倍，这意味着该量子电路完全无法近似 $\exp(-ir\hat{}H)$。总的来说，PennyLane 生成的量子电路有 3,785 个案例满足误差界要求，611 个案例生成的量子电路误差大于用户设定的值，还有 20 个案例中 PennyLane 无法处理误差界，也就是说，它会一直运行下去，试图找到正确的量子电路。

5 基于 BE 与 Trotter 的算法的深入比较研究

QBlue 编译器包含用于编译量子模拟程序的基于 BE 的算法和基于 Trotter 的算法，并允许用户设置目标编译机器，选择数字式或模拟式机器。在不同的机器选项、算法和量子电路级优化中，哪种选择最有可能为某个程序带来最佳编译结果，尤其是在资源估算方面？本节比较 QBlue 中不同的编译路径，并回答这一问题。

[1] 由于 PennyLane 的角度求解器无法生成结果，我们使用 QBlue 角度求解器进行实验。

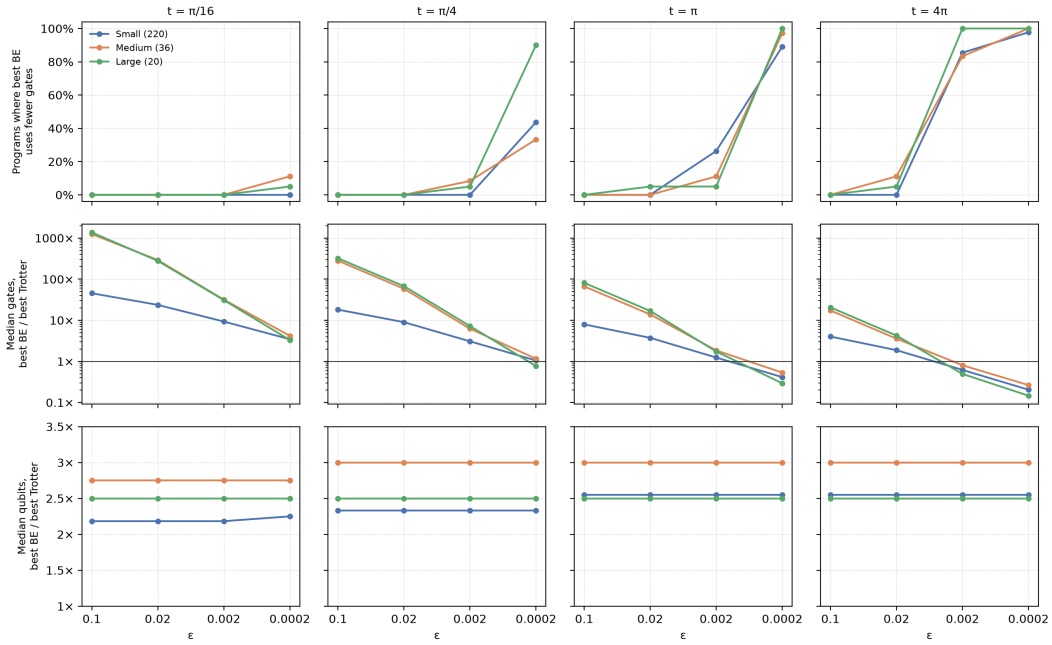


图 11 比较表现最佳的基于 Trotter 的算法和基于 BE 的算法；误差界：0。0002、0。002、0。02 和 0。1，以及模拟时间段： $\pi/16$、 $\pi/4$、 π、 4π。第一行记录基于 BE 的算法表现更佳的百分比；第二行衡量基于 BE 的算法比基于 Trotter 的算法多使用的门数量倍数；第三行显示基于 BE 的算法多使用的量子比特数量倍数。

5.1 比较基于 BE 的算法和基于 Trotter 的算法

我们首先比较基于 BE 的算法和基于 Trotter 的算法，因为长期以来，人们一直在争论哪种算法生成的代码效率最高。比较这两类算法时，我们设置了不同的误差界和模拟时间段。对于每个误差界和时间段，我们针对基于 BE 的算法运行一次 QBlue 编译器，针对基于 Trotter 分解的算法运行多次，并执行表 1 中的每个程序。随后，我们为每个程序选出表现最佳的基于 BE 的算法，并与表现最佳的 Trotter 化算法进行比较。结果汇总于图 11。

总体而言，基于 Trotter 的算法比基于 BE 的算法节省更多量子比特，因为基于 BE 的编译会生成辅助量子比特。平均而言，基于 BE 的算法所需的量子比特数量是表现最佳的基于 Trotter 分解的算法的 2～3 倍。

就门节省量而言，我们得出了一个令人惊讶的结果：基于 BE 的算法应该能生成高效得多的量子电路。然而，这并不总是如此。对于大多数误差界和时间周期组合，基于 Trotter 的算法（通常是二阶 Trotter 分解）能生成最高效的量子电路。对于不超过 π 的时间周期以及 0.002 或更宽松的误差界，基于 Trotter 的算法比基于 BE 的算法节省更多量子门。对于 0.002 的误差界，在许多情况下，基于 Trotter 的算法可节省超过 90% 的量子门，尤其是对于小于 $\pi/4$ 的时间周期。对于时间周期 $\pi/16$ 和误差界 0.1，基于 Trotter 的算法可节省超过 99.9% 的量子门。尤其是，对于中型和大型程序，基于 Trotter 的算法往往能节省更多量子比特。随着时间周期增长且误差界收紧，基于 BE 的算法逐渐占据优势：它们胜出

		QSVT 最优			量子信号处理最优			量子门数 / 最优基于 BE 的算法（中位数）			
t	ϵ	S	M	L	S	M	L	QSVT	QSP	Taylor	Walk
$\pi/16$	0.1	2	14	20	218	22	0	1.2×	1.0×	3.8×	2.2×
$\pi/16$	0.02	3	14	20	217	22	0	1.1×	1.0×	4.0×	4.3×
$\pi/16$	0.002	5	14	20	215	22	0	1.1×	1.0×	4.3×	3.5×
$\pi/16$	0.0002	25	14	20	195	22	0	1.1×	1.0×	4.1×	3.0×
$\pi/4$	0.1	143	34	20	77	2	0	1.0×	1.2×	5.9×	6.0×
$\pi/4$	0.02	139	34	20	81	2	0	1.0×	1.2×	5.4×	5.5×
$\pi/4$	0.002	140	34	20	80	2	0	1.0×	1.2×	6.0×	5.0×
$\pi/4$	0.0002	139	34	20	81	2	0	1.0×	1.3×	6.6×	4.7×
π	0.1	219	36	20	1	0	0	1.0×	1.4×	6.9×	6.7×
π	0.02	220	36	20	0	0	0	1.0×	1.5×	7.8×	6.4×
π	0.002	219	36	20	1	0	0	1.0×	1.6×	8.7×	6.2×
π	0.0002	219	36	20	1	0	0	1.0×	1.6×	9.3×	5.8×
4π	0.1	220	36	20	0	0	0	1.0×	1.5×	8.0×	6.6×
4π	0.02	220	36	20	0	0	0	1.0×	1.5×	8.7×	6.3×
4π	0.002	220	36	20	0	0	0	1.0×	1.6×	9.2×	6.0×
4π	0.0002	220	36	20	0	0	0	1.0×	1.7×	9.7×	6.3×

表 2 比较四种基于 BE 的算法。左：对于每个数据集，QSVT 或量子信号处理产生最少量子门的程序数量（S：220 个，M：36 个，L：20 个程序）；Taylor 级数和量子游走从未是最优算法。右：每种算法的量子门数量除以最优基于 BE 的算法的量子门数量，在全部 276 个程序上取中位数。

在 π 和 4π 上的大多数程序中使用 $\epsilon = 0.0002$，以及在 4π 上使用 $\epsilon = 0.002$。问题在于，大多数现实世界的量子模拟任务并不需要很长的模拟时间，例如，它们通常只需要不到一秒，因此，处理更长模拟时段的能力对用户而言并不十分有吸引力。在大多数量子化学应用中，误差界达到 0.002 即已足够，而量子物理应用则需要高精度误差界（0.0002）。正如我们所见，基于 BE 的算法将在拥有足够辅助量子比特的高精度应用中发挥作用，而基于 Trotter 的算法则适用于许多其他应用。

5.2 不同基于 BE 的算法的效果

现在，我们对比 QBlue 中的四种基于 BE 的算法。对于每个程序、误差界和时间周期，我们记录产生量子门数最少的算法，以及其他算法落后多少，如表 2 所示。

对于每个程序，我们首先找出能够生成最优量子电路的算法，结果发现胜出的只有 QSVT 或量子信号处理，而泰勒级数算法和量子游走算法从未胜出。此外，我们发现，时间周期有助于确定 QSVT 和量子信号处理这两种算法中哪一种表现更好。在时间周期 $\frac{\pi}{16}$，量子信号处理几乎在所有小型程序以及 36 个中型程序中的 22 个上生成了最优量子电路，而 QSVT 则为所有大型程序生成了最优量子电路。从 $\frac{\pi}{4}$ 起，QSVT 开始占优：它为 220 个小型程序中的 140 个生成了最优量子电路，并且在时间周期 π 和 4π 下，除至多一个程序外，均生成了最优量子电路。不过，这两种算法的量子门数差异相对较小；通常，一种算法所需的量子门比另一种少 6–37%。对于中位数集合中的程序，量子信号处理所用的量子门数最多是 QSVT 的 1.7 倍；而在时间周期为 $\frac{\pi}{16}$ 的情况下，它可节省 10%–20% 的量子门。

泰勒级数和量子游走算法则远远落后。它们所需的量子门数量是最佳算法的 2–10 倍，而且差距会随时间周期增长，因此最佳算法比最差算法节省 74–89% 的量子门。量子比特数也有所不同

		单量子比特门数量减少（模拟）			多量子比特门数量减少（模拟）		
t	ϵ	S	M	L	S	M	L
$\pi/16$	0.1	91%	99%	99%	99%	99%	99%
$\pi/16$	0.02	83%	98%	98%	99%	99%	99%
$\pi/16$	0.002	57%	81%	81%	97%	99%	99%
$\pi/16$	0.0002	-23%	-32%	-75%	92%	93%	94%
$\pi/4$	0.1	78%	98%	98%	99%	99%	99%
$\pi/4$	0.02	55%	89%	92%	97%	99%	99%
$\pi/4$	0.002	-33%	-1%	21%	92%	95%	97%
$\pi/4$	0.0002	-296%	-409%	-549%	75%	73%	78%
π	0.1	52%	91%	93%	97%	99%	99%
π	0.02	-3%	55%	67%	94%	98%	99%
π	0.002	-209%	-220%	-220%	81%	84%	88%
π	0.0002	-838%	-950%	-1221%	43%	44%	55%
4π	0.1	-1%	63%	73%	94%	98%	99%
4π	0.02	-120%	-70%	-32%	86%	92%	95%
4π	0.002	-536%	-595%	-774%	60%	62%	70%
4π	0.0002	-1927%	-2028%	-2542%	-15%	-14%	7%

表3 对于演化时间 $t \in \{\pi/16,\ \pi/4,\ \pi,\ 4\pi\}$ 和误差容限 $\epsilon \in \{0.1,\ 0.02,\ 0.002,\ 0.0002\}$，在小型（S）、中型（M）和大型（L）程序中，将印第安纳模拟后端上最佳 Trotter编译与 IBM数字后端上最佳块编码（BE）量子电路进行比较，得出单量子比特门数和多量子比特门数的相对减少比例。每个条目比较的是各数据集的中位数；对于模拟后端，量子门即脉冲。正值表示模拟编译所需的量子门比数字BE电路少相应百分比；负值表示所需量子门更多，例如，-100% 意味着数量是其两倍。

：量子信号处理使用的量子比特最少，QSVT 和量子游走需要多 4–5 个辅助量子比特，而泰勒级数所需的量子比特数是量子信号处理的 2–5 倍，因为每个泰勒项都需要独立的辅助寄存器。

5.3 与基于模拟的量子机器的比较

一个有意思的比较方向是考察基于模拟的量子机器的能力。如上所示，基于 BE 的算法直接生成数字量子门，而基于 Trotter 的算法则基于泡利字符串生成小型单位模拟操作。这意味着基于模拟的量子机器可能只支持基于 Trotter 的算法。这本质上是比较基于 BE 的方法与基于Trotter化的方法的另一种方式：将基于 BE 的算法生成数字电路，与基于 Trotter 的算法生成模拟脉冲进行比较。下面我们比较量子门数，而量子比特数与图11所示的数据一致。

多项研究采用基于模拟的量子计算机进行哈密顿量模拟，认为这种方式能够带来更好的性能 [69]。我们使用 QBlue 检验这一说法，比较将哈密顿量模拟程序编译到数字式和模拟目标机器时的资源使用量，结果见表3。数字机器上的量子门与模拟机器上的脉冲运行速度不同 [50, 5]；因此，我们在此比较它们的原始计数，即每种机器需要执行的操作数量。

实际的模拟机器原型基于印第安纳模拟机器，这是一种新型模拟量子计算机，能够实现高维相互作用。表3比较了印第安纳模拟机器的电路编译与面向 IBM 机器的数字式编译。数字电路由 VOQC 优化。虽然单量子比特项的结果可能更差，但印第安纳模拟后端显著减少了多量子比特

项的操作数量。对于最长为 $\frac{\pi}{4}$–的时间周期，在每个数据集和误差界下，它所需的多量子比特操作都至少减少 73%；在误差界为 0.1 和 0.02 时，操作数量减少 99%。随着时间周期延长、误差界收紧，这一优势会缩小：在时间周期为 π 、误差界为 $\epsilon = 0.0002$ 时，操作数量减少 43–55%；在时间周期为 4π 、误差界为 $\epsilon = 0.002$ 时，则减少 60–70%。即使在时间周期为 4π 、误差界为 0.0002[2] 的情况下，印第安纳原型在小规模程序上的表现仍然较差，但在大规模程序上能够减少操作数量。这是因为印第安纳模拟后端可以一次执行一个多量子比特项，而许多哈密顿量模拟程序都包含此类长程多量子比特项。

在量子计算机执行中，节省多量子比特项比节省单量子比特项重要得多，因为多量子比特门是导致性能下降和噪声的主要来源。此外，印第安纳模拟后端并不采用典型的单量子比特项优化，例如将一系列单量子比特项合并为单个 $SU(2)$ 项[9]。事实上，上述实验展现了门级优化的强大作用 [44], ，而基于模拟的编译缺少此类优化，这表明研究这类优化可能有助于改进哈密顿量模拟的基于模拟的编译。具体而言，印第安纳模拟后端在哈密顿量模拟方面或许能胜过数字方法，但需要复杂的基于模拟的优化才能发挥这一优势。

6 相关工作

本节回顾与我们的方法最相关的工作。尽管还有许多其他研究成果，我们着重介绍与本文讨论的哈密顿量模拟和编译技术直接相关的工作。

基于量子电路的编程语言。 量子电路语言的开发是该领域的核心研究方向。业界已开发出多种量子电路语言，包括 Q# [84], Quilc [81], ScaffCC [47], ProjectQ [82], Cirq [31], 和 Qiskit [4]。尽管这些语言缺乏形式语义，但它们以量子电路的标准化指称语义为基础。与之相对，许多工具专注于量子电路程序的形式化验证，其中包括 Qwire [72], SQIR [43], 和 QBricks [24],，以及量子霍尔逻辑及其扩展 [93, 57, 96],和 Qafny [56]。这些系统已用于验证多种量子算法，包括 Grover搜索和量子相位估计。验证也被应用于量子电路优化，例如 VOQC [44] 和 CertiQ [79], ，以及量子电路编译流程，如 ReVerC [7] 和 ReQWIRE [73] 所示。

已有多种函数式语言用于以形式语义对量子电路建模，包括 [76, 34, 78, 86, 6, 87, 88, 42, 94, 40, 41, 39]。这些语言通常旨在将量子电路编码为函数式结构，例如通过量子模式匹配。Silq [15] 在指称语义框架中引入了自动反计算，用于描述电路行为。最后，ZX演算 [29, 89]提供了一种图示语言，可在不进行测量的情况下表达酉操作，为代数量子语义提供了另一种选择。

模拟量子计算语言与脉冲级编程。 一些脉冲级和目标机器哈密顿量编程接口将量子计算机视为模拟量子模拟器。值得注意的例子包括 IBM Qiskit Pulse [30], 、QuEra Bloqade [1], 和 Pasqal Pulser [80], ，它们均由硬件服务提供商开发。与此同时，

[2] 4π 和 0.0002 的误差界很少使用，因为 0.0002 的误差界通常用于无需长时间模拟 4π 的量子物理应用。

QuTiP [48] 等计算量子物理软件包支持在经典计算机上进行哈密顿量模拟。这些工作受到经典模拟编译技术悠久历史的启发 [2, 3]。SimuQ [69] 通过使用泡利群构造哈密顿量，引入了一种基于此的模拟量子计算形式语言。Cao 等人 [18] 近期提出了一种优于 QDrift 的 Trotter 化算法。

与这些方法相比，QBlue 提供了首个经验证的编译框架，可将以二次量子化形式编写的用户级哈密顿量程序转换为数字式和模拟量子电路。

与量子模拟相关的量子计算方法。人们提出了许多量子计算软件和算法，用于分析粒子系统。此前已有软件工具 [10, 70, 54]和算法 [53, 77, 19, 17]，用于改进基于 Trotter 的哈密顿量模拟编译算法。特别是，QBlue [54]开发了一种用于基于 Trotter 的算法的经验证的编译器。此外，还有用于哈密顿量模拟的基于 BE 的编译算法 [27, 60, 13, 59, 61], , QBlue$_{BE}$ 通过在编译流水线中对所有这些算法进行验证并开展实证研究，对其进行了考察。

许多其他研究 [23, 91, 90, 21] 讨论了在量子计算机中模拟不同粒子系统的算法。由于机器限制，许多算法无法在当前的量子计算机架构上实现。变分量子本征求解器是用于分析粒子系统的最知名算法之一 [85, 22] 。它主要用于计算能量状态，例如估计水分子的基态能量 [66]。

目前已有一些用于计算粒子系统的工业工具，包括 OpenFermion [63], Qiskit Nature [46], 和 PennyLane [11]。这些工具主要用于能态计算，并不一定侧重于哈密顿量模拟。PennyLane 是唯一采用基于 BE 的算法构建编译流水线的工具。

7 结论与未来工作

本文为 QBlue 扩展了基于 BE 的编译算法，并将其命名为 QBlue$_{BE}$；同时，通过将 QBlue$_{BE}$ 与基于 Trotter 的算法进行比较，对这些算法进行了实证评估。我们在 Rocq 中验证了 QBlue$_{BE}$ 的整个编译流水线；也就是说，我们全面验证了面向不同量子计算机后端的哈密顿量模拟程序编译流水线，并在多种设置下对这些后端进行了全面评估，包括误差界、模拟时间和数据集规模。

我们的实证研究表明，QBlue$_{BE}$ 在效率和正确性方面明显优于此前基于 BE 的编译器 PennyLane。我们发现了 PennyLane 中的若干问题和错误。具体而言，我们发现 PennyLane 的角度求解器无法为量子信号处理算法找到正确的角度，而且 PennyLane 无法编译许多包含较长泡利字符串的程序。

我们还发现了一个令人意外的结果：尽管基于 BE 的算法在理论上具有优势，但在大多数时间周期和误差界设置下，它们相较于最优的基于 Trotter 的算法可能并无效率优势。此外，基于 Trotter 的算法可以在模拟量子计算机上实现，而基于 BE 的算法则不能；在模拟机器上实现的基于 Trotter 的算法相较于基于 BE 的算法，效率优势甚至更为明显。

未来的工作中，我们计划开发用于分析基于哈密顿量的程序的工具，并将 QBlue 编译器与用于科学计算的基态能量求解器集成。

此前有一个框架 [55] 能够执行该任务，并可自然地将哈密顿量模拟扩展至绝热量子计算（AQC），即，5he QBlue 的应用语义可以自然扩展到绝热量子计算，以进行能态计算。

参考文献

1 Bloqade.jl：基于中性原子架构的量子计算和量子模拟软件包，2023年。URL: https://github.com/QuEraComputing/Bloqade.jl/。2 Sara Achour 和 Martin Rinard。面向模拟设备的噪声感知动力系统编译：使用 legno。在《第二十五届编程语言与操作系统架构支持国际会议》论文集，ASPLOS '20，第149–166页，纽约，美国，2020年。Association for Computing Machinery。doi:10.1145/3373376.3378449。3 Sara Achour、Rahul Sarpeshkar 和 Martin C. Rinard。使用 arco 为可编程模拟设备进行配置综合。SIGPLAN 通讯，51(6):177–193，2016年6月。 doi:10.1145/2980983.2908116。4 Gadi Aleksandrowicz、Thomas Alexander、Panagiotis Barkoutsos、Luciano Bello、Yael Ben-Haim、David Bucher、Francisco Jose Cabrera-Hernández、Jorge Carballo-Franquis、Adrian Chen、Chun-Fu Chen、Jerry M. Chow、Antonio D. Córcoles-Gonzales、Abigail J. Cross、Andrew Cross、Juan Cruz-Benito、Chris Culver、Salvador De La Puente González、Enrique De La Torre、Delton Ding、Eugene Dumitrescu、Ivan Duran、Pieter Eendebak、Mark Everitt、Ismael Faro Sertage、Albert Frisch、Andreas Fuhrer、Jay Gambetta、Borja Godoy Gago、Juan Gomez-Mosquera、Donny Greenberg、Ikko Hamamura、Vojtech Havlicek、Joe Hellmers、Łukasz Herok、Hiroshi Horii、Shaohan Hu、Takashi Imamichi、Toshinari Itoko、Ali Javadi-Abhari、Naoki Kanazawa、Anton Karazeev、Kevin Krsulich、Peng Liu、Yang Luh、Yunho Maeng、Manoel Marques、Francisco Jose Martín-Fernández、Douglas T. McClure、David McKay、Srujan Meesala、Antonio Mezzacapo、Nikolaj Moll、Diego Moreda Rodríguez、Giacomo Nannicini、Paul Nation、Pauline Ollitrault、Lee James O'Riordan、Hanhee Paik、Jesús Pérez、Anna Phan、Marco Pistoia、Viktor Prutyanov、Max Reuter、Julia Rice、Abdón Rodríguez Davila、Raymond Harry Putra Rudy、Mingi Ryu、Ninad Sathaye、Chris Schnabel、Eddie Schoute、Kanav Setia、Yunong Shi、Adenilton Silva、Yukio Siraichi、Seyon Sivarajah、John A. Smolin、Mathias Soeken、Hitomi Takahashi、Ivano Tavernelli、Charles Taylor、Pete Taylour、Kenso Trabing、Matthew Treinish、Wes Turner、Desiree Vogt-Lee、Christophe Vuillot、Jonathan A. Wildstrom、Jessica Wilson、Erick Winston、Christopher Wood、Stephen Wood、Stefan Wörner、Ismail Yunus Akhalwaya 和 Christa Zoufal。Qiskit：用于量子计算的开源框架，2019年。doi:10.5281/zenodo.2562110。5 Thomas Alexander、Naoki Kanazawa、Daniel J Egger、Lauren Capelluto、Christopher J Wood、Ali Javadi-Abhari 和 David C McKay。Qiskit pulse：通过脉冲和云端编程量子计算机。Quantum Science and Technology，5(4):044006，2020年8月。URL:https://dx.doi.org/10.1088/2058-9565/aba404, doi:10.1088/2058-9565/aba404。6 T. Altenkirch 和 J. Grattage。函数式量子编程语言。在第20届IEEE计算机科学逻辑年度研讨会（LICS'05）上。IEEE。URL: http://dx.doi.org/10.1109/LICS.2005.1, doi:10.1109/lics.2005.1。7 Matthew Amy、Martin Roetteler 和 Krysta M. Svore。空间高效可逆电路的验证编译。在 Rupak Majumdar 和 Viktor Kunčak（编）《计算机辅助验证》，第3–21页，Cham，2017年。Springer International Publishing。8 Juan Miguel Arrazola、Soran Jahangiri、Alain Delgado、Jack Ceroni、Josh Izaac、Antal Száva、Utkarsh Azad、Robert A. Lang、Zeyue Niu、Olivia Di Matteo、Romain Moyard、Jay Soni、Maria Schuld、Rodrigo A. Vargas-Hernández、Teresa Tamayo-Mendoza、Cedric Yen-Yu Lin、Alán Aspuru-Guzik 和 Nathan Killoran。使用 pennylane 的可微量子计算化学，2023年。URL: https://arxiv.org/abs/2111.09967, arXiv:2111.09967。9 Yasar Y. Atas、Jinglei Zhang、Randy Lewis、Amin Jahanpour、Jan F. Haase 和 Christine A. Muschik。在量子计算机上通过变分方法实现 SU(2) 强子。Nature Commu-

nications，12(1)，2021年11月。URL：http://dx.doi.org/10.1038/s41467-021-26825-4, doi:10.1038/s41467-021-26825-4。10 Lindsay Bassman Oftelie、Connor Powers 和 Wibe A. De Jong。Arqtic：用于在量子计算机上模拟材料的全栈软件包。ACM Transactions on Quantu m Computing，3(3)，2022年6月。 doi:10.1145/3511715。11 Ville Bergholm、Josh Izaac、Maria Schuld、Christian Gogolin、Shahnawaz Ahmed、Vishnu Ajith、M. Sohaib Alam、Guillermo Alonso-Linaje、B. AkashNarayanan、Ali Asadi、Juan Miguel Arrazola、Utkarsh Azad、Sam Banning、Carsten Blank、Thomas R Bromley、Benjamin A. Cordier、Jack Ceroni、Alain Delgado、Olivia Di Matteo、Amintor Dusko、Tanya Garg、Diego Guala、Anthony Hayes、Ryan Hill、Aroosa Ijaz、Theodor Isacsson、David Ittah、Soran Jahangiri、Prateek Jain、Edward Jiang、Ankit Khandelwal、Korbinian Kottmann、Robert A. Lang、Christina Lee、Thomas Loke、Angus Lowe、Keri McKiernan、Johannes Jakob Meyer、J. A. Montañez-Barrera、Romain Moyard、Zeyue Niu、Lee James O'Riordan、Steven Oud、Ashish Panigrahi、Chae-Yeun Park、Daniel Polatajko、Nicolás Quesada、Chase Roberts、Nahum Sá、Isidor Schoch、Borun Shi、Shuli Shu、Sukin Sim、Arshpreet Singh、Ingrid Strandberg、Jay Soni、Antal Száva、Slimane Thabet、Rodrigo A. Vargas-Hernández、Trevor Vincent、Nicola Vitucci、Maurice Weber、David Wierichs、Roeland Wiersema、Moritz Willmann、Vincent Wong、Shaoming Zhang 和 Nathan Killoran。PennyLane：量子-经典混合计算的自动微分，2022年。URL：https://arxiv.org/abs/1811.04968, arXiv:1811.04968。12 Dominic W. Berry、Andrew M. Childs、Richard Cleve、Robin Kothari 和 Rolando D. Somma。使用截断泰勒级数模拟哈密顿量动力学。《物理评论快报》，114(9)，2015年3月。13 Dominic W. Berry、Andrew M. Childs 和 Robin Kothari。对所有参数具有近乎最优依赖性的哈密顿量模拟。在2015年IEEE第56届计算机科学基础年会上。IEEE，2015年10月。URL：http://dx.doi.org/10.1109/FOCS.2015.54,doi:10.1109/focs.2015.54。14 Yves Bertot 和 Pierre Castéran。交互式定理证明与程序开发：Coq'Art：归纳构造演算。施普林格科学与商业媒体，2013年。15 Benjamin Bichsel、Maximilian Baader、Timon Gehr 和 Martin Vechev。Silq：具有安全反计算和直观语义的高级量子语言。在第41届ACM SIGPLAN编程语言设计与实现会议论文集，PLDI 2020，第286–300页，纽约，美国，2020年。Association for Computing Machinery。doi:10.1145/3385412.3386007。16 Gilles Brassard、Peter Høyer、Michele Mosca 和 Alain Tapp。量子振幅放大与估计，2002年。URL: http://dx.doi.org/10.1090/conm/305/05215,doi:10.1090/conm/305/05215。17 Earl Campbell。用于快速哈密顿量模拟的随机编译器。《物理评论快报》，123(7)，2019年8月。URL: http://dx.doi.org/10.1103/PhysRevLett.123.070503, doi:10.1103/physrevlett.123.070503。18 Xiuqi Cao、Junyu Zhou、Yuhao Liu、Yunong Shi 和 Gushu Li。MarqSim：在量子模拟的编译器优化中协调确定性与随机性，2024年。URL:https://arxiv.org/abs/2408.03429, arXiv:2408.03429。19 Xiuqi Cao、Junyu Zhou、Yuhao Liu、Yunong Shi 和 Gushu Li。MarqSim：在量子模拟的编译器优化中协调确定性与随机性。ACM《程序设计语言》会议论文集，9(PLDI)，2025年6月。 doi:10.1145/3729269。20 Yudong Cao、Jonathan Romero、Jonathan P. Olson、Matthias Degroote、Peter D. Johnson、Mária Kieferová、Ian D. Kivlichan、Tim Menke、Borja Peropadre、Nicolas P. D. Sawaya、Sukin Sim、Libor Veis 和 Alán Aspuru-Guzik。量子计算时代的量子化学。《化学评论》，119(19):10856–10915，2019年10月。 doi:10.1021/acs.chemrev.8b00803。21 Davide Castaldo、Soran Jahangiri、Alain Delgado 和 Stefano Corni。溶液中的分子量子模拟。《化学理论与计算杂志》，18(12):7457–7469，2022 年。

PMID: 36351289. arXiv:https://doi.org/10.1021/acs.jctc.2c00974, doi:10.1021/acs.jctc.2c00974. 22 M. Cerezo、Kunal Sharma、Andrew Arrasmith 和 Patrick J. Coles。变分量子态本征求解器。npj Quantum Information，8(1):113，2022年9月。 doi:10.1038/s41534-022-00611-6。23 Alba Cervera-Lierta。量子计算机上的伊辛模型精确模拟。Quantum，2:114，2018年12月。URL: http://dx.doi.org/10.22331/q-2018-12-21-114, doi:10.22331/q-2018-12-21-114。24 Christophe Chareton、Sébastien Bardin、François Bobot、Valentin Perrelle 和 Benoît Valiron。面向电路构建量子程序的自动化演绎验证框架。载于 Nobuko Yoshida 编，《编程语言与系统——第30届欧洲编程研讨会（ESOP 2021）论文集》，该会议作为欧洲软件理论与实践联合会议（ETAPS 2021）的一部分，于2021年3月27日至4月1日在卢森堡卢森堡市举行，第12648卷 Lecture Notes in Computer Science，第148–177页。Springer，2021年。 doi:10.1007/978-3-030-72019-3_6。25 I-Chi Chen、Aleksei Khindanov、Carlos Salazar、Humberto Munoz Barona、Feng Zhang、Cai-Zhuang Wang、Thomas Iadecola、Nicola Lanatà 和 Yong-Xin Yao。通过幽灵古茨维勒近似实现量子-经典嵌入，以增强关联电子系统的模拟，2025年。URL: https://arxiv.org/abs/2506.01204, arXiv:2506.01204。26 Zihan Chen、Jiakang Li、Minghao Guo、Henry Chen、Zirui Li、Joel Bierman、Yipeng Huang、Huiyang Zhou、Yuan Liu 和 Eddy Z. Zhang。Genesis：面向混合连续变量-离散变量量子计算机的哈密顿量模拟编译器。载于第52届年度 International Symposium on Computer Architecture 论文集，ISCA '25，第1583–1597页，美国纽约州纽约市，2025年。Association for Computing Machinery。 doi:10.1145/3695053.3731065。27 Andrew M. Childs、Dmitri Maslov、Yunseong Nam、Neil J. Ross 和 Yuan Su。迈向首次实现量子加速的量子模拟。Proceedings of the National Academy of Sciences，115(38):9456–9461，2018年。URL: https://www.pnas.org/doi/abs/10.1073/pnas.1801723115, arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.1801723115, doi:10.1073/pnas.1801723115。28 Andrew M. Childs、Yuan Su、Minh C. Tran、Nathan Wiebe 和 Shuchen Zhu。具有对易子标度的Trotter误差理论。Phys. Rev. X，11:011020，2021年2月。URL: https://link.aps.org/doi/10.1103/PhysRevX.11.011020, doi:10.1103/PhysRevX.11.011020。29 Bob Coecke 和 Ross Duncan。相互作用的量子可观测量：范畴代数与图示法。New Journal of Physics，13(4):043016，2011年4月。URL: http://dx.doi.org/10.1088/1367-2630/13/4/043016, doi:10.1088/1367-2630/13/4/043016。30 Andrew Cross、Ali Javadi-Abhari、Thomas Alexander、Niel De Beaudrap、Lev S. Bishop、Steven Heidel、Colm A. Ryan、Prasahnt Sivarajah、John Smolin、Jay M. Gambetta 和 Blake R. Johnson。OpenQASM 3：更广泛、更深入的量子汇编语言。ACM Transactions on Quantum Computing，3(3)，2022年9月。 doi:10.1145/3505636。31 Cirq Developers。Cirq，2023年12月。 doi:10.5281/zenodo.10247207。32 Yulong Dong、Xiang Meng、K. Birgitta Whaley 和 Lin Lin。量子信号处理中高效的相位因子求值。Phys. Rev. A，103:042419，2021年4月。URL: https://link.aps.org/doi/10.1103/PhysRevA.103.042419, doi:10.1103/PhysRevA.103.042419。33 Weijie Du 和 James P. Vary。通过量子计算研究多核子结构与动力学，2023年。 arXiv:2304.04838。34 Alejandro Díaz-Caro、Gilles Dowek 和 Juan Pablo Rinaldi。λ演算中量子计算的两种线性性。Biosystems，186:104012，2019年。选自2017年 International Conference on the Theory and Practice of Natural Computing论文集。URL: https://www.sciencedirect.com/science/article/pii/S0303264718301102, doi:10.1016/j.biosystems.2019.104012。35 Prashant S. Emani、Jonathan Warrell、Alan Anticevic、Stefan Bekiranov、Michael Gandal、Michael J. McConnell、Guillermo Sapiro、Alán Aspuru-Guzik、Justin T. Baker、Matteo Bastiani,

John D. Murray、Stamatios N. Sotiropoulos、Jacob Taylor、Geetha Senthil、Thomas Lehner、Mark B. Gerstein 和 Aram W. Harrow。生物科学前沿的量子计算。《自然·方法》，18(7):701–709，2021年7月。 doi:10.1038/s41592-020-01004-3。36 Francesco A. Evangelista 和 Victor S. Batista。编者按：用于化学的量子计算。《化学理论与计算杂志》，19(21):7435–7436，2023年11月。 doi:10.1021/acs.jctc.3c01043。37 A. K. Fedorov 和 M. S. Gelfand。迈向量子计算生物学的实际应用。《自然·计算科学》，1(2):114–119，2021年2月。 doi:10.1038/s43588-021-00024-z。38 Richard P Feynman。用计算机模拟物理学。《国际理论物理学杂志》，21(6/7):467–488，1982年。39 Peng Fu、Kohei Kishida、Neil J. Ross 和 Peter Selinger。支持动态提升的 Proto-Quipper。ACM《程序设计语言》会议论文集，7(POPL)，2023年1月。 doi:10.1145/3571204。40 Jonathan Grattage。以 Haskell 的具体实现为例概述量子机器学习。载于第五届量子物理与逻辑联合国际研讨会和第四届计算模型发展研讨会（QPL/DCM 2008）论文集，第270卷，第165–174页，2011年。 arXiv:0806.2735, doi:10.1016/j.entcs.2011.01.015。41 Jonathan James Grattage。一种函数式量子编程语言。博士论文，诺丁汉大学，2006年。URL: http://eprints.nottingham.ac.uk/10250/。42 Alexander Green、Peter LeFanu Lumsdaine、Neil J. Ross、Peter Selinger 和 Benoît Valiron。Quipper：一种可扩展的量子编程语言。载于第34届ACM SIGPLAN编程语言设计与实现会议论文集，PLDI 2013，第333–342页，2013年。 doi:10.1145/2491956.2462177。43 Kesha Hietala、Robert Rand、Shih-Han Hung、Liyi Li 和 Michael Hicks。证明量子程序正确。载于交互式定理证明会议（ITP）论文集，2021年6月。44 Kesha Hietala、Robert Rand、Liyi Li、Shih-Han Hung、Xiaodi Wu 和 Michael Hicks。量子电路验证优化器。ACM程序设计语言与系统汇刊，45(3)，2023年9月。 doi:10.1145/3604630。45 J. Hubbard。窄能带中的电子关联。《伦敦皇家学会会刊A辑：数学与物理科学》，276(1365):238–257，1963年。URL:http://www.jstor.org/stable/2414761。46 Ali Javadi-Abhari、Matthew Treinish、Kevin Krsulich、Christopher J. Wood、Jake Lishman、Julien Gacon、Simon Martiel、Paul D. Nation、Lev S. Bishop、Andrew W. Cross、Blake R. Johnson 和 Jay M. Gambetta。使用 Qiskit 进行量子计算，2024年。 arXiv:2405.08810, doi:10.48550/arXiv.2405.08810。47 Ali JavadiAbhari、Shruti Patil、Daniel Kudrow、Jeff Heckey、Alexey Lvov、Frederic T. Chong 和 Margaret Martonosi。ScaffCC：量子程序的可扩展编译与分析。《并行计算》，45:2–17，2015年6月。URL: http://dx.doi.org/10.1016/j.parco.2014.12.001, doi:10.1016/j.parco.2014.12.001。48 J.R. Johansson、P.D. Nation 和 Franco Nori。QuTiP：用于开放量子系统动力学的开源 Python 框架。《计算物理通讯》，183(8):1760–1772，2012年。URL: https://www.sciencedirect.com/science/article/pii/S0010465512000835, doi:10.1016/j.cpc.2012.02.021。49 Michael Kreshchuk、Shaoyang Jia、William Kirby、Gary Goldstein、James Vary 和 Peter Love。当前量子计算机上的光前场论。《熵》，23(5)，2021年5月。URL:https://www.osti.gov/biblio/1782924, doi:10.3390/e23050597。50 Antonis Kyprianidis、A J Rasmusson 和 Philip Richerme。利用全局驱动工程化囚禁离子量子模拟器中的相互作用图。New Journal of Physics，26(2):023033，2024年2月。URL: http://dx.doi.org/10.1088/1367-2630/ad264d,doi:10.1088/1367-2630/ad264d。

51 B. P. Lanyon、J. D. Whitfield、G. G. Gillett、M. E. Goggin、M. P. Almeida、I. Kassal、J. D. Biamonte、M. Mohseni、B. J. Powell、M. Barbieri、A. Aspuru-Guzik 和 A. G. White。面向量子计算机上的量子化学。《Nature Chemistry》，2(2):106–111，2010年2月。doi:10.1038/nchem.483。52 Lingling Lao 和 Dan E. Browne。2qan：用于二局部量子比特哈密顿量模拟算法的量子编译器。载于第49届年度 International Symposium on Computer Architecture 会议论文集，ISCA '22，第351–365页，美国纽约州纽约市，2022年。Association for Computing Machinery。doi:10.1145/3470496.3527394。53 Gushu Li、Anbang Wu、Yunong Shi、Ali Javadi-Abhari、Yufei Ding 和 Yuan Xie。Paulihedral：用于量子模拟内核的广义分块编译器优化框架。载于第27届 ACM International Conference on Architectural Support for Programming Languages and Operating Systems 会议论文集，ASPLOS '22，第554–569页，美国纽约州纽约市，2022年。Association for Computing Machinery。doi:10.1145/3503222.3507715。54 Liyi Li、Fenfen An、Federico Zahariev、Zhi Xiang Chong、Amr Sabry 和 Mark S. Gordon。用于量子模拟的经验证编译器，2025年。URL：https://arxiv.org/abs/2509.18583,arXiv:2509.18583。55 Liyi Li、David Young、James Bryan Graves、Chandeepa Dissanayake 和 Amr Sabry。一种 Haskell 绝热领域特定语言：在量子硬件上求解经典优化问题。ACM《程序设计语言》会议论文集，9(ICFP)，2025年8月。doi:10.1145/3747521。56 Liyi Li、Mingwei Zhu、Rance Cleaveland、Alexander Nicolellis、Yi Lee、Le Chang 和 Xiaodi Wu。Qafny：量子程序验证器。载于 ECOOP 2024，2024年9月。57 Junyi Liu、Bohua Zhan、Shuling Wang、Shenggang Ying、Tao Liu、Yangjia Li、Mingsheng Ying 和 Naijun Zhan。使用量子霍尔逻辑对量子算法进行形式化验证。载于 Isil Dillig 和 Serdar Tasiran（编），计算机辅助验证，第187–207页，Cham，2019年。Springer International Publishing。58 Seth Lloyd。通用量子模拟器。《Science》，273(5278):1073–1078，1996年。URL：https://www.science.org/doi/abs/10.1126/science.273.5278.1073,arXiv:https://www.science.org/doi/pdf/10.1126/science.273.5278.1073,doi:10.1126/science.273.5278.1073。59 Guang Hao Low 和 Isaac L. Chuang。通过量子信号处理实现最优哈密顿量模拟。《物理评论快报》，118(1)，2017年1月。60 Guang Hao Low 和 Isaac L. Chuang。通过量子比特化实现哈密顿量模拟。《Quantum》，3:163，2019年7月。61 John M. Martyn、Zane M. Rossi、Andrew K. Tan 和 Isaac L. Chuang。量子算法的大统一。《PRX Quantum》，2(4)，2021年12月。URL：http://dx.doi.org/10.1103/PRXQuantum.2.040203,doi:10.1103/prxquantum.2.040203。62 Sam McArdle、Suguru Endo、Alán Aspuru-Guzik、Simon C. Benjamin 和 Xiao Yuan。量子计算化学。《Rev. Mod. Phys.》，92:015003，2020年3月。URL：https://link.aps.org/doi/10.1103/RevModPhys.92.015003,doi:10.1103/RevModPhys.92.015003。63 Jarrod R. McClean、Kevin J. Sung、Ian D. Kivlichan、Yudong Cao、Chengyu Dai、E. Schuyler Fried、Craig Gidney、Brendan Gimby、Pranav Gokhale、Thomas Häner、Tarini Hardikar、Vojtěch Havlíček、Oscar Higgott、Cupjin Huang、Josh Izaac、Zhang Jiang、Xinle Liu、Sam McArdle、Matthew Neeley、Thomas O'Brien、Bryan O'Gorman、Isil Ozfidan、Maxwell D. Radin、Jhonathan Romero、Nicholas Rubin、Nicolas P. D. Sawaya、Kanav Setia、Sukin Sim、Damian S. Steiger、Mark Steudtner、Qiming Sun、Wei Sun、Daochen Wang、Fang Zhang 和 Ryan Babbush。OpenFermion：面向量子计算机的电子结构软件包，2019年。URL：https://arxiv.org/abs/1710.07629,arXiv:1710.07629。64 Filipe V. Melo、Alexandre M. Souza、Ivan S. Oliveira 和 Roberto S. Sarthour。双格点哈伯德哈密顿量的量子模拟。《Physics Open》，6:100053，2021年。URL：https://www.sciencedirect.com/science/article/pii/S2666032620300405,doi:10.1016/j.physo.2020.100053。

65 Danial Motlagh 和 Nathan Wiebe。广义量子信号处理。《PRX Quantum》，5:020368，2024年。 doi:10.1103/PRXQuantum.5.020368。66 Yunseong Nam、Jwo-Sy Chen、Neal C. Pisenti、Kenneth Wright、Conor Delaney、Dmitri Maslov、Kenneth R. Brown、Stewart Allen、Jason M. Amini、Joel Apisdorf、Kristin M. Beck、Aleksey Blinov、Vandiver Chaplin、Mika Chmielewski、Coleman Collins、Shantanu Debnath、Kai M. Hudek、Andrew M. Ducore、Matthew Keesan、Sarah M. Kreikemeier、Jonathan Mizrahi、Phil Solomon、Mike Williams、Jaime David Wong-Campos、David Moehring、Christopher Monroe 和 Jungsang Kim。在囚禁离子量子计算机上估计水分子的基态能量。《npj Quantum Information》，6(1):33，2020年4月。doi:10.1038/s41534-020-0259-3。67 Michael A. Nielsen 和 Isaac L. Chuang。《量子计算与量子信息》。剑桥大学出版社，美国，十周年纪念版，2011年。68 Carlos Outeiral、Martin Strahm、Jiye Shi、Garrett M. Morris、Simon C. Benjamin 和 Charlotte M. Deane。量子计算在计算分子生物学中的前景。《WIREs Computational Molecular Science》，11(1):e1481，2021年。URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1481， arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1481, doi:10.1002/wcms.1481。69 Yuxiang Peng、Jacob Young、Pengyu Liu 和 Xiaodi Wu。SimuQ：用于通过模拟编译进行量子哈密顿量模拟编程的框架。ACM《程序设计语言》会议论文集，8(POPL)，2024年1月。 doi:10.1145/3632923。70 Connor Powers、Lindsay Bassman、Thomas M. Linker、Ken i chi Nomura、Sahil Gulania、Rajiv K. Kalia、Aiichiro Nakano 和 Priya Vashishta。Mistiqs：用于在量子计算机上执行量子动力学模拟的开源软件。《SoftwareX》，14:100696，2021年。URL： https://www.sciencedirect.com/science/article/pii/S2352711021000418, doi:10.1016/j.softx.2021.100696。71 QuEra。QuEra Computing Inc.，2024年。URL: https://github.com/QuEraComputing/。72 Robert Rand。形式化验证的量子编程。博士论文，宾夕法尼亚大学，2018年。73 Robert Rand、Jennifer Paykin、Dong-Ho Lee 和 S. Zdancewic。ReQWIRE：关于可逆量子电路的推理。载于 QPL，2018年。74 Philip Richerme。囚禁离子的多模全局驱动与量子电路综合。《Quantum Science and Technology》，10(3):035046，2025年6月。URL： https://dx.doi.org/10.1088/2058-9565/ade560, doi:10.1088/2058-9565/ade560。75 Philip Richerme。囚禁离子的多模全局驱动与量子电路综合，2025年。URL: https://arxiv.org/abs/2502.13302, arXiv:2502.13302。76 Amr Sabry、Benoît Valiron 和 Juliana Kaizer Vizzotto。从对称模式匹配到量子控制。载于 Christel Baier 和 Ugo Dal Lago（编），软件科学与计算结构基础，第348–364页，Cham，2018年。Springer International Publishing。77 Albert T. Schmitz、Nicolas P. D. Sawaya、Sonika Johri 和 A. Y. Matsuura。低深度 Trotter-Suzuki 分解的图优化视角。《Phys. Rev. A》，109:042418，2024年4月。URL: https://link.aps.org/doi/10.1103/PhysRevA.109.042418, doi:10.1103/PhysRevA.109.042418。78 Peter Selinger 和 Benot Valiron。量子λ演算。《量子计算中的语义技术》，2013年1月。 doi:10.1017/CBO9781139193313.005。79 Yunong Shi、Xupeng Li、Runzhou Tao、Ali Javadi-Abhari、Andrew W. Cross、Frederic T. Chong 和 Ronghui Gu。对现实量子编译器进行基于契约的验证。arXiv 电子预印本，2019年8月。 arXiv:1908.08963。80 Henrique Silvério、Sebastián Grijalva、Constantin Dalyac、Lucas Leclerc、Peter J. Karalekas、Nathan Shammah、Mourad Beji、Louis-Paul Henry 和 Loïc Henriet。Pulser：用于设计可编程中性原子阵列中脉冲序列的开源软件包。《Quantum》，6:629，2022年1月。 doi:10.22331/q-2022-01-24-629。

81 Robert S. Smith、Eric C. Peterson、Mark G. Skilbeck 和 Erik J. Davis。面向量子程序的开源工业级优化编译器，2020年。 arXiv:2003.13961。82 Damian S. Steiger、Thomas Häner 和 Matthias Troyer。ProjectQ：用于量子计算的开源软件框架。Quantum，2:49，2018年1月。URL: http://dx.doi.org/10.22331/q-2018-01-31-49, doi:10.22331/q-2018-01-31-49。83 Masuo Suzuki。分形路径积分的一般理论及其在多体理论和统计物理学中的应用。数学物理学杂志，32(2):400–407，1991年2月。arXiv:https://pubs.aip.org/aip/jmp/article-pdf/32/2/400/19166143/400_1_online.pdf, doi:10.1063/1.529425。84 Krysta Svore、Alan Geller、Matthias Troyer、John Azariah、Christopher Granade、Bettina Heim、Vadym Kliuchnikov、Mariia Mykhailova、Andres Paz 和 Martin Roetteler。Q#：借助高级领域特定语言实现可扩展量子计算与开发。载于《2018年真实世界领域特定语言研讨会论文集》，RWDSL2018。ACM，2018年2月。URL: http://dx.doi.org/10.1145/3183895.3183901, doi:10.1145/3183895.3183901。85 Jules Tilly、Hongxiang Chen、Shuxiang Cao、Dario Picozzi、Kanav Setia、Ying Li、Edward Grant、Leonard Wossnig、Ivan Rungger、George H. Booth 和 Jonathan Tennyson。变分量子本征求解器：方法与最佳实践综述。Physics Reports，986:1–128，2022年11月。URL: http://dx.doi.org/10.1016/j.physrep.2022.08.003,doi:10.1016/j.physrep.2022.08.003。86 André van Tonder。用于量子计算的λ演算。SIAM计算杂志，33(5):1109–1135，2004年1月。URL: http://dx.doi.org/10.1137/S0097539703432165, doi:10.1137/s0097539703432165。87 Juliana Kaizer Vizzotto、Bruno Crestani Calegaro 和 Eduardo Kessler Piveta。用于量子计算的双效应 λ演算。载于 André Rauber Du Bois 和 Phil Trinder 编，《编程语言》，第61–74页，柏林、海德堡，2013年。Springer Berlin Heidelberg。88 Finn Voichick、Liyi Li、Robert Rand 和 Michael Hicks。Qunity：用于量子和经典计算的统一语言。Proceedings of the ACM on Programming Languages，7(POPL):921–951，2023年1月。URL: http://dx.doi.org/10.1145/3571225, doi:10.1145/3571225。89 Quanlong Wang。ZX演算的完备性，2023年。 arXiv:2209.14894。90 Fumiko Yamaguchi 和 Yoshihisa Yamamoto。t–j模型的量子模拟。Superlattices and Microstructures，32(4):343–345，2002年。选自第八届先进物理场国际研讨会“量子计算先进材料”论文。URL: https://www.sciencedirect.com/science/article/pii/S0749603603000399, doi:10.1016/S0749-6036(03)00039-9。91 Bing Yang、Hui Sun、Robert Ott、Han-Yi Wang、Torsten V. Zache、Jad C. Halimeh、Zhen-Sheng Yuan、Philipp Hauke 和 Jian-Wei Pan。在一个含71个格点的玻色–哈伯德量子模拟器中观测到规范不变性。Nature，587(7834):392–396，2020年11月。URL:http://dx.doi.org/10.1038/s41586-020-2910-8, doi:10.1038/s41586-020-2910-8。92 Yongxin Yao、Feng Zhang、Cai-Zhuang Wang、Kai-Ming Ho 和 Peter P. Orth。用于关联材料的Gutzwiller混合量子-经典计算方法。Physical Review Research，3(1)，2021年2月。URL: http://dx.doi.org/10.1103/PhysRevResearch.3.013184, doi:10.1103/physrevresearch.3.013184。93 Mingsheng Ying。量子程序的Floyd–Hoare逻辑。ACM程序设计语言与系统汇刊，33(6)，2012年1月。 doi:10.1145/2049706.2049708。94 Mingsheng Ying。《量子编程基础》。Elsevier Science，马萨诸塞州剑桥，2016年。95 Dan-Bo Zhang、Hongxi Xing、Hui Yan、Enke Wang 和 Shi-Liang Zhu。核物理学量子计算精选专题*。Chinese Physics B，30(2):020306，2021年2月。URL: https://dx.doi.org/10.1088/1674-1056/abd761, doi:10.1088/1674-1056/abd761。

96 Li Zhou、Gilles Barthe、Pierre-Yves Strub、Junyi Liu 和 Mingsheng Ying。Coqq：量子程序的基础验证。*ACM*《程序设计语言》会议论文集，7(POPL)，2023年1月。 doi:10.1145/3571222。

基于 PauliString 的哈密顿量语法和语义

在讨论 QBlue 编译流水线之前，我们先考察其形式语义。本节将介绍 QBlue 在泡利串形式体系中的语法和语义。

A.0.0.1 语法。

ˆ 图 12 定义了 QBlue程序语法。约束表达式 H 表示对系统施加用户定义的约束的符号哈密顿量，而酉表达式$U = \exp(-i\hat{}H)$表示编译后的时间演化，可在量子硬件上执行。

泡利项 $\mathcal{P} ::= X \mid Y \mid Z \mid I$约束表达式 $\hat{}H ::= r\mathcal{P} \mid \hat{}H + \hat{}H \mid \hat{}H \otimes \hat{}_H$酉表达式 $U ::= \exp(-i\,\hat{}H) \mid U \circ U$

图 12 基于 QBlue 泡利字符串的语法。

项 $r\,\mathcal{P}$表示一个泡利操作（$\{X,\ Y,\ Z,\ I\}$），并乘以实振幅r，这足以表示任意厄米哈密顿量；参见 B.1 节。其他表达式还包括张量 $e \otimes e$，用于表示作用于不同量子比特的操作，以及加法组合$e + e$，用于表示各项的线性组合。在酉算子层面，$\exp(-i\,\hat{}H)$执行哈密顿量模拟，而 $U \circ U$ 表示酉复合。程序通常使用带索引记法，作为张量表达式的语法糖。对于具有 n 个量子比特的系统，诸如$\mathcal{P}(j)$这样的带索引算子，表示除 j之外所有量子比特上的恒等算子的张量积；算子$\mathcal{P}$则作用于 j。例如，项$_{1\,2}(X(0) \circ X(1) + Y(0) \circ Y(1))$表示 $(X \otimes I) \circ (I \otimes X) + (Y \otimes I) \circ (I \otimes Y) = X \otimes X + Y$-$\otimes Y$。

A.0.0.2 语义。

QBlue 程序描述有限的 n 量子比特系统，其指称语义见图 13，定义为 $[\![\hat{}H]\!]$，用于解释哈密顿量约束；以及 $[\![U]\!]$，用于解释酉操作。该语义本质上将 n 量子比特系统解释为一个 $2^n \times 2^n$ 矩阵。

图 13 的前两行定义了哈密顿量约束语义，其中第一行将每个泡利算子解释为其矩阵形式。第二行则分别将张量运算和求和运算解释为矩阵张量积（$\otimes$）和线性和（$+$）。

最后一行定义了 QBlue 的动态语义。左侧将表达式$\exp(-i\hat{}H)$解释为矩阵指数，前提是 $\hat{}H$ 为厄米矩阵。指数通过其标准幂级数展开来定义；当 $\hat{}H$ 为厄米矩阵时，该级数收敛到一个酉算子。我们将此结构解释为矩阵指数级数的收敛级数之极限，并根据需要通过计算进行近似。右侧则将幺正算子的顺序复合解释为标准矩阵乘法。因此，QBlue 中的动态程序通过复合酉演化构建而成，每个酉演化都由哈密顿量约束经模拟得出。

B 基于 QBlueTrotter 的编译过程

本节介绍 QBlue 编译步骤，并说明 Rocq 如何构建并验证其正确性证明。编译器将哈密顿量模拟表达式

$$[\![X]\!] := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad [\![Y]\!] := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad [\![Z]\!] := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad [\![I]\!] := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$[\![r\mathcal{P}]\!] := r[\![\mathcal{P}]\!] \qquad [\![\hat{H} \otimes \hat{H}']\!] := [\![\hat{H}]\!] \otimes [\![\hat{H}']\!] \qquad [\![\hat{H} + \hat{H}']\!] := [\![\hat{H}]\!] + [\![\hat{H}']\!]$$

$$[\![\mathsf{exp}(-i\hat{H})]\!] := \sum_{n=0}^{\infty} \frac{(-i[\![\hat{H}]\!]_g)^n}{n!} \qquad [\![U \circ U']\!] := [\![U]\!]([\![U']\!])$$

图 13 QBlue 基于泡利字符串的哈密顿量和幺正算子的语义。 $[\![-]\!]$ 内的项是 QBlue 算子。 紫色 项是量子态向量运算，而 $[\![-]\!]([\![-]\!])$ 表示矩阵乘法。

$\mathsf{exp}(-ir\ \hat{}H)$ 转换为量子电路，其中 e 是 QBlue 哈密顿量程序， r 是时间参数；该电路以用户定义的误差界 ϵ 近似 $\mathsf{exp}(-ir\ \hat{}H)$，并可在量子硬件上执行。我们将编译建模为形式为 $(\hat{}H,\ r) \gg U$ 的判断。QBlue 编译器支持多种编译路径，用户可从四种 Trotter 化算法和三种电路后端中进行选择。我们首先给出将约束表达式 e 的哈密顿量模拟编译为量子电路的正确性定理。

▶ **定理 2** （基于 Trotter 的算法编译正确性）。 给定图 12 中的 $\hat{}H$ 、时间周期 r 和期望误差 ϵ，我们将其模拟 $\mathsf{exp}(-ir\ \hat{}H)$ 编译为量子电路，即 $(\hat{}H, r) \gg U$；对于每个状态 ψ，令 $\psi_1 = [\![U]\!](\psi)$，量子电路与模拟之间的距离由 ϵ 界定，即 $\|\psi_1 - [\![\mathsf{exp}(-ir\ \hat{}H)]\!](\psi)\| < \epsilon$。

证明。 完全机械化证明在 Rocq中完成。 ◀

B.1 规范化

粒子变换后，我们将哈密顿量程序规范化为泡利串形式，其中每个泡利串都具有实振幅。下面首先定义规范化后的形式。

▶ **定义 3** （泡利规范形式）。 设 $\mathcal{P} \in \{I、\ X、\ Y、\ Z\}$ 是一个 n 量子比特厄米哈密顿量 $\hat{}H$，若其中 $\sigma_j = \bigotimes_{k=0}^{n-1} \mathcal{P}_{j、\ k}$ （r_j 为实数），则称其为泡利规范 $\sum_j r_j \sigma_j$。

$\hat{}$ 定理3指出，每个 n 量子比特程序 H 都可以表示为一个二维列表。外层列表对应于求和 $\sum_j r_j\ \hat{}P_j$，而每个内层元素 $\hat{}\ P_j$ 都是一个固定 n 长度的张量积 $\bigotimes_{k=0}^{n-1} \mathcal{P}_k$，表示一个泡利操作列表。在 Rocq 中，我们将基于泡利字符串的哈密顿量及其中间哈密顿量模拟公式实现为如下列表结构。

$$\begin{array}{lllll} \text{Hamiltonian} & \hat{H} & ::= & [(z_1, \sigma_1), \ldots, (z_m, \sigma_m)] & z_j \in \mathbb{C} \\ \text{Real Hamiltonian} & \hat{H}_{\mathbb{R}} & ::= & [(r_1, \sigma_1), \ldots, (r_m, \sigma_m)] & r_j \in \mathbb{R} \\ \text{Rotation sequence} & R & ::= & [(\theta_1, \sigma_1), \ldots, (\theta_m, \sigma_m)] & \theta_j \in \mathbb{R} \end{array}$$

现在，我们逐步证明规范化定理。首先证明，每个厄米哈密顿量都可以改写为仅含实振幅的形式。QBlue 将每个权重替换为其实部，范数$([(z_j,\ \sigma_j)]_j) = [(\mathrm{Re}\, z_j,\ \sigma_j)]_j$。后续阶段需要实权重，因为只有当 θ 为实数时，$\mathsf{exp}(-i\theta[\![\sigma]\!])$才是酉算子。

▶ **引理 4** (实性)。 如果 $[\![\ \hat{}H]\!]$ 是厄米矩阵，则 $[\![$范数$(\ \hat{}H)]\!] = [\![\hat{}H]\!]$。

证明。 每个 $[\![\sigma_j]\!]$ 都是厄米矩阵，因为它是厄米矩阵的张量积，因此 $[\![\hat{}H]\!]^\dagger = \sum_j z_j [\![\sigma_j]\!]$。因此 $[\![\hat{}H]\!] = \frac{1}{2}([\![\hat{}H]\!] + [\![\hat{}H]\!]^\dagger) = \sum_j$ 实部(z_j) $[\![\sigma_j]\!] = [\![$范数$(\hat{}H)]\!]$。已在`QBlueHamiltonian.v`中机械化。 ◀

该假设是必要的： $[(i, Z)]$ 表示 iZ，它不是厄米的，并将其映射到范数 它映射为 $[(0, Z)]$。

第二部分会移除序列操作。请注意，定理3不需要包含序列操作（$\circ$），因为任意单量子比特泡利算子序列都可以规范化为一个单例泡利算子，如图14所示。事实上，任意有效的基于泡利字符串的哈密顿量都可以改写为以下形式。

哈密顿量是一个由带有复权重的泡利字符串组成的列表，它表示这些矩阵的加权和。列表中的字符串可以重复。按照第2节中的记法，我们用 $X(0) \circ Y(1)$表示字符串 XY。这种输入格式具有一般性，因为 n 个量子比特上的 4^n个泡利字符串构成所有 $2^n \times 2^n$ 矩阵的一个基。$\hat{}H$ 在时间 r 内的模拟表示 $\exp(-ir[\![\hat{}H]\!])$，其中 $\exp(M) = \sum_{k\geq 0} M^k/k!$；该级数对每个矩阵 M都收敛，并且当 M 为厄米矩阵时，$\exp(-irM)$是酉算子。Trotter 化会生成一个旋转序列，它表示按列表顺序应用其中各旋转所得的乘积。

	X	Y	Z
X	I	iZ	$-iY$
Y	$-iZ$	I	iX
Z	iY	$-iX$	I

图14 泡利串合并

另一个关键的简化是，单个位置上的一系列泡利操作（$\mathcal{P} \in \{I,\ X,\ Y,\ Z\}$）可以合并为单个泡利操作，即$\mathcal{P}_1 \circ \mathcal{P}_2$ 可以改写为单个泡利操作。将恒等算子 I 与$\mathcal{P}$结合，结果为 $\mathcal{P}$。其他泡利操作（$\mathcal{P}_1$ 和$\mathcal{P}_2$）的应用结果见图14。这两种泡利操作的应用体现了简化过程：一系列泡利群操作最终会得到一个具有特定实数振幅的单个泡利操作。关于规范化正确性，我们有以下引理。

▶ **引理5** （规范编译正确性）。 A $\hat{}H$ 可改写为定理3中的规范化形式。

证明。 证明依赖于定理4以及上述对操作序列的合并。 ◀

B.2 Trotter 化

QBlue 的 Trotter 化步骤的一般流程如下。给定一个时间周期r、期望误差 ϵ以及一个 n量子比特的基于泡利字符串的哈密顿量 $\hat{}H = \sum_{j=0}^{d-1} \theta_j \hat{}P_j$ （其中 $\hat{}P_j = \bigotimes_{k=0}^{n-1} \mathcal{P}_{j,\ k}$），我们选择一个系数 λ，使得 $\exp(-i\frac{r}{\lambda}\theta_0 \hat{}P_0) \circ \ldots \circ \underbrace{\exp(-i\frac{r}{\lambda}\theta_j \hat{}P_j) \circ \ldots}_{N \text{ times}}$ 将 $\exp(-ir\hat{}H)$ 近似为 ϵ。误差与 r 和 θ_j 有关，详述如下。

QBlue 中的四种 Trotter 化算法可分为两组：标准算法组和 QDrift [17] 组。两种标准算法本质上采用 Lie-Trotter 方法。两种 QDrift 方法会随机调整指数运算的顺序，从而加快哈密顿量模拟。我们先陈述一个基于误差界ϵ （针对这两种算法进行了专门化）的一般定理及其在 Rocq 中的证明，然后分别讨论这两种算法，揭示不同设置下的误差界。

▶ **引理 6** （Trotter 化正确性）。 给定 $\hat{}H = \sum_{j=0}^{d-1} \theta_j \hat{}P_j$、时间 r 和所需误差ϵ，存在 λ 和 N，使得 $\exp(-ir\hat{}H)$ 可由 $\Pi_{k=0}^{N-1} U_k(d,\ \lambda,\ N,\ r)$ 近似，且误差不超过给定的误差界：$\|[\![\exp(-ir\hat{}H)]\!] - [\![\Pi_{k=0}^{N-1} U_k(d,\ \lambda,\ N,\ r)]\!]\| < \epsilon$。

证明。 证明依赖于四种不同的算法。在 Rocq 中，证明被拆分为四个不同的定理，分别对应这四种算法，并在 `QBlueTrotterProof.v`、`QBlueQdriftProof.v` 和 `QBlueMarQSimProof.v` 中完成机械化。这些证明通过对 N 进行归纳法得出，关键的误差界引理将在下文讨论。◀

作为贯穿全文的示例，我们使用以下简单的伊辛模型，并展示其特罗特化结果，同时介绍这两类算法组。

$$\hat{H}_{I1} = \sum_j r_h X(j) + \sum_j Z(j) \circ Z(j{+}1) \tag{4.2.1}$$

B.2.1 标准算法。

我们在流程中首先纳入一阶和二阶标准Trotter化。对于一阶Trotter化，我们通过将 $\Pi_{k=0}^{N-1} U_k(d, \lambda, N, r)$ 设为 $\Pi_{k=0}^{d-1}\Pi^m \exp(-i\frac{r}{m}\theta_j \hat{} P_j)$，来近似 $\exp(-ir\hat{}H)$；也就是说，对于每个泡利串 $\theta_j \hat{} P_j$，我们构造一个应用序列 $\Pi^m \exp(-i\frac{r}{m}\theta_j \hat{} P_j)$，其中以缩放后的时间槽 $\frac{r}{m}$，重复模拟 $\theta_j \hat{} P_j$ 共 m 次。每个 $U_k(d, \lambda, N, r)$ 定义为 $\exp(-i\frac{r}{m}\theta_j \hat{} P_j)$，其中 $N = d * m$ 和 $\lambda = m$。例如，为了模拟基于泡利字符串的哈密顿量 H_{I1}，使用时间槽 r 和分割数 m，我们如下计算指数 $\exp(-ir\hat{} H_{I1})$ 的近似值。

$$\begin{aligned}\exp(-ir\hat{H}_{I1}) &= \Pi^m \left(\exp(-i\tfrac{r}{m}(\textstyle\sum_j Z(j)\circ Z(j{+}1) + r_h \sum_j X(j)))\right) \\ &= \Pi^m \left(\exp(-i\tfrac{r}{m}Z(0)\circ Z(1)) \circ \exp(-i\tfrac{r}{m}Z(1)\circ Z(2)) \circ \ldots \circ \exp(-i\tfrac{r}{m}r_h X(0)) \circ \exp(-i\tfrac{r}{m}r_h X(1)) \circ \ldots\right)\end{aligned}$$

二阶 Trotter 分解考虑了单项泡利串之间的关系。将 $\exp(-ir\hat{}H)$ 展开为泰勒级数后，二阶项实质上是将单项泡利串两两相乘。在上面的例子中，除了一阶项之外，对于所有 j，我们还包括以下各项：

$$\frac{1}{2}\cdot X(j)\circ Z(j)\circ Z(j{+}1) \qquad \frac{1}{2}\cdot Z(j)\circ Z(j{+}1)\circ X(j)$$

加入这两项会在相同的误差界基础上改进 m，因为现在 m 与 $O(\frac{r^{1.5}}{\epsilon^{0.5}})$ 相关，而不是像一阶 Trotter 化中那样与 $O(\frac{t^2}{\epsilon})$ 相关。

B.2.1.1 误差界的证明方法。

设 $\hat{H} = \sum_{j=0}^{d-1} \theta_j \hat{} P_j$，其中 $N = d{*}m$。如果通过列出所有泡利字符串来列出所有 Trotter 化的哈密顿量模拟序列，我们将得到一个包含 N 个元素的集合：$\hat{} H_k \in \{\underbrace{\frac{\theta_0}{m}\hat{}P_0、\frac{\theta_0}{m}\hat{}P_0、\ldots\ldots、\frac{\theta_j}{m}\hat{}P_j}_{m}、\frac{\theta_j}{m}\hat{}P_j、\ldots\ldots、\frac{\theta_{d\text{-}1}}{m}\hat{}P_{d\text{-}1}\}$。$\hat{}\hat{}H$ 可视为该集合中 N 个不同元素哈密顿量 H_k 的和。误差界 ϵ 与 $O(r^2)$ 相关；如下方公式所示，我们将误差界与不同分量联系起来（误差界记为 ϵ'）。

$$\epsilon' = \frac{r^2}{2}\sum_{k=1}^{N}\left\|(\sum_{j=k+1}^{N}\hat{H}_j)\circ\hat{H}_k - \hat{H}_k\circ(\sum_{j=k+1}^{N}\hat{H}_j)\right\|. \tag{4.2.2}$$

QBlue 编译器将 ϵ 作为输入，我们通过反复应用上述公式，直到 $\epsilon' < \epsilon$，从而找到相应的 N。

B.2.2 QDrift群。

QDrift 是一种随机采样方法，根据哈密顿量各子项的权重对其进行采样。我们将 $\hat{}$ $H = \sum_{k=0}^{d-1} \theta_k \hat{}P_k$ 中的泡利字符串管理为集合 $\{\theta_0 \hat{}P_0, .., \theta_{d-1} \hat{}P_{d-1}\}$。令 $\lambda = \sum_{k=0}^{d-1} |\theta_k|$；我们从该集合中按概率选取一个泡利字符串，并重复此过程N次；这些选取结果按序列 m_0, $...m_k, ..., m_{N-1}$列出，每个 $m_k \in [0, d)$，而选取项 $\theta_k \hat{}P_k$ 的概率为 $|\theta_k|/\lambda$。QDrift过程生成一个应用序列 $\Pi_{k=0}^{N-1} \exp(-i \frac{r\lambda}{N} \theta_{m_k} \hat{}P_{m_k})$，即将每个 $U_k(d, \lambda, N, r)$设为 $\exp(-i \frac{r\lambda}{N} \theta_{m_k} \hat{}P_{m_k})$，并将 $\exp(-ir\hat{}H)$设为 $\approx \Pi_{k=0}^{N-1} U_k(d, \lambda, N, r)$。

我们使用上面的 $\hat{}H_{I1}$ 示例，展示 QDrift 与标准Trotter化之间的关键区别。为了近似指数 $\exp(-ir\hat{}H_{I1})$，我们从集合 $\{r_h X(0)\ r_h X(1) ...,\ Z(0) \circ Z(1)\ Z(1) \circ Z(2) ...\}$ 中随机选择 N个泡利字符串，然后构造 Trotter化序列。一个选择示例如下。

$$\exp(-ir\hat{H}_{I1}) = \exp(-i\frac{r\lambda}{N} Z(j) \circ Z(j{+}1)) \circ \exp(-i\frac{r\lambda}{N} Z(k) \circ Z(k{+}1)) \circ ... \circ \exp(-i\frac{r\lambda}{N} r_h X(k)) \circ \exp(-i\frac{r\lambda}{N} r_h X(0)) \circ ...$$

MarQSim [18] 通过指出，将泡利串彼此靠近排列，可以提高程序编译为数字电路时在电路优化过程中消去 CX 量子门的概率，从而改进了 QDrift。为了提高将这些相关泡利串分组的概率，MarQSim 使用马尔可夫链图，其中节点对应不同的泡利串，以提高选中这些泡利串的概率。我们的评估表明，在进行数字电路编译时，与 QDrift 相比，MarQSim 能生成经过优化的量子电路。然而，随着哈密顿量规模增大，改进幅度会变小；对于大型哈密顿量（$> 10K$ 泡利字符串），MarQSim 会失效，因为其马尔可夫链难以采样。对于如何处理基于模拟的编译，其效果也值得商榷。

B.2.2.1 误差界的证明方法。

QDrift 中的误差率 ϵ 为 $O(\frac{r^2}{N})$，其中 N 取决于输入 ϵ，如下所示。

$$N = \frac{2\lambda^2 r^2}{\epsilon} \tag{4.2.3}$$

QDrift 正确性的 Rocq证明依赖于一个基于公式 (4.2.3) 中误差界的关键误差界引理。

B.3 综合

综合将由 Trotter 化生成的泡利字符串矩阵指数应用序列应用于数字式或模拟量子电路。要综合应用序列 $\Pi_j \exp(-i\theta_j \hat{}P_j)$，只需说明如何综合单个矩阵指数项 $\exp(-i\theta_j \hat{}P_j)$。下面将讨论数字式和基于模拟的综合。

B.3.0.1 数字综合。

数字综合将泡利串 $\exp(-i\theta_j \hat{}P_j)$ 的模拟综合为基本量子门，例如单量子比特门，如 H、 Rz、 Ry 和 Rx 门，以及双量子比特受控非门 CX (CX(x, y) 以 x 为控制，并在 y 上应用 X 门)。

该过程关注泡利串 $\hat{}P = \bigotimes_{j=0}^{n-1} \mathcal{P}_j$ 中的非-I 项。如果 $\hat{}P$ 只有一个非-I 项，编译会生成一个旋转门，如图15所示。在布洛赫量子比特球面的解释中， [67], 这些旋转门 Rx、 Ry、 Rz 可以表示为

算法2 对泡利串 $\hat{P}$执行 $U = \exp(-i\theta\hat{P})$的数字综合

1：输入：泡利串 $\hat{P} = \bigotimes_{k=0}^{n-1}\mathbf{P}_k$，时间周期 θ2：输出：数字电路 $U = \exp(-i\theta\hat{P})$ 3：4：收集位点 $\{m_1, m_2, \ldots, m_l\}$，其中 $\mathbf{P}_{m_j} \neq I$，且 $l \leq n$5：$U \leftarrow \mathtt{Rz}(2\theta)(m_1)$ 6：$j = m_1$7：对于 $m_j \in \{m_2, \ldots, m_l\}$，执行 8：$U \leftarrow \mathtt{CX}(m_{j-1}, m_j) \circ U \circ \mathtt{CX}(m_{j-1}, m_j)$ 9：结束循环 10：对于 $m_j \in \{m_1, \ldots, m_l\}$，执行 11：如果 $\mathbf{P}_{m_j} = Y$，则 12：$U \leftarrow \mathtt{Rz}\left(\frac{\pi}{2}\right)(m_j) \circ \mathtt{H}(m_j) \circ U \circ \mathtt{H}(m_j) \circ \mathtt{Rz}\left(-\frac{\pi}{2}\right)(m_j)$ 13：否则，如果 $\mathbf{P}_{m_j} = X$，则 14：$U \leftarrow \mathtt{H}(m_j) \circ U \circ \mathtt{H}(m_j)$ 15：结束如果 16：结束循环 17：返回 U

$$\exp(-i\theta X) = \mathtt{RX}(2\theta) = \begin{pmatrix} \cos\theta & -i\sin\theta \\ -i\sin\theta & \cos\theta \end{pmatrix} \qquad \exp(-i\theta Y) = \mathtt{RY}(2\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$
$$\exp(-i\theta Z) = \mathtt{RZ}(2\theta) = \begin{pmatrix} \exp(-i\theta) & 0 \\ 0 & \exp(i\theta) \end{pmatrix}$$

图15 将泡利矩阵模拟为旋转门。$\exp(-i\theta\mathcal{P})$可视为在时间 θ内模拟$\mathcal{P}$。

一些泡利矩阵的哈密顿量模拟（$\mathtt{Rz}(2\theta)(j)$在第j个量子比特上施加绕 Z轴的旋转），本质上表明，对单例泡利算子 X、Y和 Z项的模拟就是上述相位旋转门。例如，$\exp(-i\frac{r}{n}X(j))$的门综合会生成一个 Rx 门，因为 $\mathtt{Rx}(\frac{2r}{n})(j)$。对于包含多个非$I$ 项的泡利串，即满足 $k \geqslant 1$的 k局域泡利串，我们在图15中描述了相应算法。

我们首先展示方程（4.2.1）中简单伊辛模型$\hat{}$ H_{I1} 的门综合，它会生成由 $\exp(-i\frac{r}{n}Z(j) \circ Z(j{+}1))$和 $\exp(-i\frac{r}{n}r_h X(j))$操作构成的序列（序列长度与 n相关）−。如果 n 足够大，该操作序列就能很好地近似 $\exp(-ir\hat{}H_{I1})$。$\exp(-i\frac{r}{n}r_h X(j))$操作会产生一个 Rx 门 [67]，而 $\exp(-i\frac{r}{n}Z(j) \circ Z(j{+}1))$则会产生一个数字电路：在第 j个和第 j+1个量子比特上执行两个受控非门，并在中间施加一个 $\mathtt{Rz}(2\frac{r}{n})$门；也就是说，$\mathtt{Rz}(2\theta)$是一个绕 z轴的相位旋转门。

以上假设非 I 项为 Z 的二局域泡利串。我们在图17中展示了一个长泡利串综合的示例。对 H 进行 Trotter 分解后，我们会生成一系列泡利串的哈密顿量模拟。某些可能的泡利串项可能是四局部的，例如 $Z\otimes Z\otimes Z\otimes Z$（缩写为 $ZZZZ$）、$X\otimes X\otimes X\otimes X$（缩写为 XXXX）和 $X\otimes X\otimes Y\otimes Y$（缩写为 $XXYY$）。图17a中，对 $ZZZZ$ 项的模拟会产生一个四重 Z 相互作用，其中间施加了一个 ZZ 相互作用，如图16所示，周围由受控非门环绕。$\exp(-i\theta XXXX)$和 $\exp(-i\theta XXYY)$的数字门综合基于四重 Z 相互作用，并针对 X 和 Y 转换进行了修正，由此得到图17b和图17c中的电路。下面给出正确性引理。

▶ **引理7**（数字综合正确性）。给定$\hat{}P$, $\exp(-i\theta\hat{}P)$，其已通过基于基本门的算法2正确综合

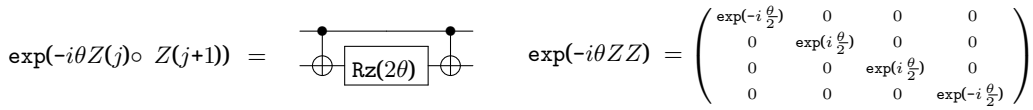
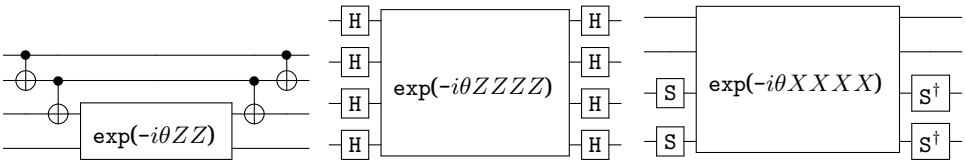


图16 ZZ 相互作用的量子电路和酉算子。左侧量子电路作用于 j 和 j+1 量子比特。



(a) exp(−$i\theta ZZZZ$) 门。 **(b)** exp(−$i\theta XXXX$) 门。 **(c)** exp(−$i\theta XXYY$) 门。

图17 若干长泡利串的综合量子电路。

，其中{H、 Rx、 Ry、 Rz、 CX}。

证明。 我们基于泡利标准形式，通过归纳法给出了这些完全机械化证明，并在 QBlueSynthDigitalProof.v中将其机械化。通过应用算法2，我们生成酉算子 U，对应于 exp(−$i\theta$ ^P)。随后，我们将这两个项解释为 VOQC 中的矩阵表示 [44]并令它们相等。◀

算法3 基于 IBM 的模拟综合：针对二局域泡利串的 $U = \exp(-i\theta\ \hat{}P)$

1：输入：2-局域泡利串 ^$P = \bigotimes_{k=0}^{n-1} \mathbf{P}_k$，时间周期 θ2：输出：模拟哈密顿量模拟 $U = \Pi^m$exp(−ir ^P) 3： 4：识别非恒等位点 m_1 和 m_2，其中 $\mathbf{P}_{m_j} \neq I$5： $U \leftarrow$ exp(−$i\theta Z$(m_1) ∘ Z(m_2)) 6：对于 $m_j \in \{m_1, m_2\}$，执行 7：如果 $\mathbf{P}_{m_j} \neq Z$，则 8：令 $U_1 =$ exp(−$i\frac{\pi}{4}X$(m_j)) ∘ exp(−$i\frac{\pi}{4}Z$(m_j)) ∘ exp(−$i\frac{\pi}{4}X$(m_j)) 9： $U \leftarrow U_1 \circ U \circ U_1$10：结束如果 11：如果 $\mathbf{P}_{m_j} = Y$，则 12： $U \leftarrow$ exp(−$i\frac{\pi}{4}Z$(m_j)) ∘ U ∘ exp(−$i\frac{7\pi}{4}Z$(m_j)) 13：结束如果 14：结束循环 15：返回 U

B.3.0.2 模拟综合。

许多量子计算机为模拟量子计算提供后端；也就是说，我们可以将量子计算机视为按顺序运行小型哈密顿量的哈密顿量模拟器。量子模拟器中的每个模拟步骤都接收一个正实数 θ 和一个泡利串 ^P，并生成模拟结果 exp(−$i\theta$ ^P)。具体过程取决于机器。例如，IBM机器可以模拟集合 $\{X, Z, ZZ\}$ 中的一局域或二局域泡利串（ZZ 表示仅包含 Z 泡利操作的二局域泡利串）。另一台由 Richerme *et al.* [75, 50], 开发的机器（Indiana）允许模拟集合 $\{Z, X^*\}$中的 k局域泡利串，其中 X^* 允许对包含任意数量 X 泡利算符的泡利串进行模拟。我们在算法 3 和 4 中分别展示了两个模拟综合算法，用于将我们的泡利串编译到 IBM机器和 Indiana 机器上。

算法4 Indiana模拟综合 $U = \exp(-i\theta \hat{} P)$

1：输入：泡利串 $\hat{} P = \bigotimes_{k=0}^{n-1} \mathbf{P}_k$**，时间周期** θ**2：输出：模拟电路** $U = \exp(-i\theta \hat{} P)$ **3：4：收集满足** $\mathbf{P}_{m_j} \neq I$ **的位点** $\{m_1, m_2, \ldots, m_k\}$**，其中** $l \leq n$**^5：** $P' = I$**6：对于** $m_j \in \{m_1, \ldots, m_k\}$ **中的每个元素，执行 7：** $\hat{} P' \leftarrow \hat{} P' \circ X(m_j)$ **8：结束循环 9：** $U \leftarrow \exp(-i\theta \hat{} P')$ **10：对于** $m_j \in \{m_1, \ldots, m_k\}$ **中的每个元素，执行 11：如果** $\mathbf{P}_{m_j} = Y$**，则 12：** $U \leftarrow \exp(-i\frac{\pi}{4} Z(m_j)) \circ U \circ \exp(-i\frac{7\pi}{4} Z(m_j))$ **13：否则，如果** $\mathbf{P}_{m_j} = Z$ **，则 14：令** $U_1 = \exp(-i\frac{\pi}{4} X(m_j)) \circ \exp(-i\frac{\pi}{4} Z(m_j)) \circ \exp(-i\frac{\pi}{4} X(m_j))$ **15：** $U \leftarrow U_1 \circ U \circ U_1$**16：结束条件 17：结束循环 18：返回** U

为了对 IBM机器执行模拟综合，我们假设通过应用微扰装置实现二局域哈密顿量。我们可以使用图15中的矩阵指数公式来模拟一局域泡利算符。Indiana 机器允许模拟一个 k 局域泡利串。

^ 例如，要模拟上面的简单伊辛系统 H_{I1} ，在 IBM环境中，对于位点 j 和 k，我们可以直接执行模拟 $\exp(-i\frac{r}{n} Z(j) \circ Z(k))$ 和 $\exp(-i\frac{r}{n} r_h X(j))$。如果 $\frac{r}{n}$ 为负数，我们也可以改为模拟一个正数 $\frac{r}{n} + 2m\pi$。直接进行哈密顿量模拟可以实现更高效的电路编译。上述模拟综合步骤实现了图16中的 ZZ相互作用门，并解释了为什么在 IBM机器上某些相互作用门比基本门高效得多，如 Peng等[69]所示。

要在 Indiana 机器中执行模拟 $\exp(-i\theta XXYY)$，我们可以按如下方式构造模拟过程（算法4）：我们构造模拟操作$\exp(-i\theta XXXX)$，并在最后两个量子比特周围使用单量子比特模拟 $\exp(-i\frac{\pi}{4} Z)$ 和 $\exp(i\frac{7\pi}{4} Z)$。

其他量子机器可能允许使用不同的泡利字符串集合，例如 Quera机器[71]。因此，机器哈密顿量模拟综合步骤必须针对每种量子机器采用不同的模拟综合方法。下面我们给出这两种机器的正确性引理。

▶ **引理 8**（模拟综合正确性）。给定 $\hat{} P$, $\exp(-i\theta \hat{} P)$ 分别基于 IBM机器（$\{X、 Z、 ZZ\}$）和 Indiana 机器（$\{Z、 X^*\}$）所允许的泡利串模拟，通过算法 3 和 4 得到了正确综合。

证明。我们使用与定理 7 相同的方法对证明进行了机械化。在此，我们将 $\exp(-i\theta \hat{} P)$ 的两种矩阵解释与通过算法 3 和 4 生成的幺正算子等同起来。证明已在 `QBlueSynthProof.v` 中实现机械化。◀