用于支持量子计算技术应用的符号量子编程

雅罗斯瓦夫·亚当·米什恰克

波兰科学院理论与应用信息学研究所，巴尔蒂茨卡街5号，44-100格利维采，波兰

2023年2月21日

arXiv:2302.09401v1 [quant-ph] 18 Feb 2023

摘要

本文旨在概述当前的技术发展现状，分享开发量子软件工具的经验，并展望如何开发支持符号编程的量子编程工具，以满足量子计算技术的需求。本文主要关注量子计算技术，因为开发能够对量子电路进行符号操作，并提供用于创建、优化和测试量子程序的软件工具，可以最直接地使这些技术受益。我们简要综述了量子软件开发领域最流行的方法，并指出它们各自的优势与不足。这有助于归纳量子计算框架应具备的理想特性。接下来，我们介绍一种软件架构及其初步实现，该架构支持以符号方法开发量子程序，倡导函数式编程范式，同时支持与高性能计算和云计算集成。所介绍的软件由多个软件包组成，分别满足不同需求，但共享共同的设计理念。最后，我们还阐述了所提出的方法如何应用于量子软件工程任务，具体包括量子软件测试和量子电路构建。

关键词：量子技术、量子计算、计算机代数、函数式编程、符号操作

1 引言

量子技术正在蓬勃发展 [9]。量子计算有望显著提升许多重要问题的计算速度，包括物流、医学、化学和物理学领域中至关重要的优化问题。量子霸权的展示印证了这一前景：即使是现有最强大的超级计算机，也确实难以模拟量子计算机。因此，众多公共领域 [3, 4] 和商业领域 [12, 2] 的计划相继启动，以支持量子计算硬件和软件的发展。这些举措推动了可用量子计算资源的快速、系统性进步 [14]。此外，模拟量子计算的需求——这不可避免地与开发量子算法的需求密切相关——也使传统上处于高性能计算前沿的企业对这项技术产生了更大兴趣 [16, 37]。

硬件技术的迅速进步激发了人们对构建软件生态系统的浓厚兴趣，以支持开发基于相关技术的应用，从而发挥量子计算的潜力。量子计算的许多方面都需要专门的软件和技能，因此，自然会让人思考：软件工程在多大程度上能够支持对发挥量子计算潜力所需的逻辑结构和计算结构进行操作 [29, 48]。因此，发展量子软件工程领域需要明确的要求 [1]。此外，量子编程和量子软件开发需要专业知识在软件爱好者和开发者中传播；为此，软件社区需要支持量子计算教育 [52] 工作。

本文结构如下。首先，在第 2 节中，我们回顾了支持量子程序开发的软件工具的最新进展，并举例介绍了一些面向量子软件用户典型需求的软件包。接着，在第 3 节中，我们概述了支持量子程序符号化开发的架构，并介绍该架构若干组成部分的初步实现。为说明开发此类工具的依据，我们在第 4 节中简要讨论量子计算技术中的一些问题，这些问题有望通过符号化方法得到解决。最后，在第 5 节中，我们总结本文的贡献并给出结语。

2 量子软件生态系统的结构

首先，简要回顾量子软件领域的当前发展现状。需要指出的是，本节并不旨在全面列举软件包和工具。各种组织都提供了此类清单，包括 [48] 和[47]中的清单。此外， [60] 和 [6]中还收录了有关该领域进展与趋势的最新综述。此外， [23]对量子计算软件平台进行了全面比较。

另一方面，我们旨在概述用于开发软件的方法，这些软件支持量子计算机科学的各个方面，涵盖从量子物理基础研究，到量子计算与量子信息中研究的结构，再到专门的软件开发框架、领域特定语言和零量子代码解决方案。我们将提供这些类别中软件框架的示例，并借此指出创建量子代码开发框架的不同方法各自的优势与不足。

2.1 高性能模拟器

开发量子软件最直接的方法，是使用专门用于模拟量子物理现象的软件。在近期加入这一支持量子编程的工具家族的新成员中，有几款值得关注。

其中之一是 Qulacs [55]，这是一个 Python/C++ 库，可用于快速模拟大型、含噪或参数化量子电路。与许多此类软件一样， Qulacs 在开发时也将效率放在首位。为此，它以 C++库的形式提供高效方法，专门针对需要反复模拟的小型量子电路，例如计算变分量子算法中目标函数的期望值。 TornadoQSim [54] 也采用了类似的方法来提供高性能模拟引擎，该引擎以 Java开发。在这种情况下，量子模拟器后端通过 TornadoVM 开源虚拟机在异构硬件上加速。 TornadoVM 可在多核 CPU、GPU 和 FPGA 上自动加速 Java 程序，从而实现异构计算。

与上述两种解决方案不同，一种基于成熟编程语言的方法， Yao.jl [24] 旨在充分利用 Julia 编程语言的能力。 Yao.jl 目标是提供一个可扩展、高效的开源框架，用于量子算法设计。由于该软件包采用 Julia 编写，因此往往也继承了这种编程语言的所有优缺点。从本文的角度来看，值得一提的是， Yao.jl通过 YaoSym 组件支持符号计算。在现有量子软件框架中，这种支持并不常见。

2.2 面向研究的软件

另一个支持符号操作的量子软件包是 Tequila [21]，它为（变分）量子算法提供了一个抽象框架。在 Yao.jl 和 Tequila 的案例中，引入符号操作支持都是为了对变分量子算法中用于构建量子电路的目标函数进行自动微分。不过，这两种情况下的支持都受限于提供符号能力的外部库。 Tequila 最有趣的特性是，它可以与多种回弯配合使用，包括上文提到的 Qulacks，以及由大型软件公司支持的 Qiskit 和 Cirq。这种方法用于

例如 ProjectQ [53, 11]，可确保使用特定框架编写的软件不依赖于特定硬件，并能在各种量子计算架构上运行。

开发支持量子元素的量子软件，其动机之一是为量子计算和量子信息理论的基础研究提供支持。因此，已有大量软件包被开发出来，用于支持这些领域中的常见元素、构造和协议。此类软件包的开发者通常主要关注量子态、酉门、量子信道和测量。

在支持量子计算和量子物理研究的热门软件包中， QuTip [17]颇具代表性。该软件包提供了一个用于模拟量子光学和量子信息的 Python库。

近期的一个例子是 toqito [50]，它以 Python 库的形式提供，可用于研究量子信息中的各种对象。除了提供用于操作量子态和量子操作的标准函数库之外， toqito 还集成了一个优化模块，用于求解半定规划，并支持对非局域博弈的研究。

2.2.1 量子计算的专门模型

大量专用于研究的量子软件包聚焦于量子计算的某个狭窄领域。在这些特定领域中，量子游走受到了广泛关注。这在一定程度上是因为这一量子计算范式成功应用于开发新的高效量子算法，并用于界定量子计算的能力。

用于支持量子游走的基础软件包支持对一维和二维晶格上的量子游走进行模拟 [25]。然而，量子游走范式有着丰富多样的推广形式。近期对量子游走的关注主要集中在如何表达图上的量子演化和经典演化的一般形式。基于这一模型，人们开发了一个软件包 QSWalk.jl [7]，可使用 Julia 编程语言对量子随机游走进行高性能分析。该软件包的功能与软件包 QSWalk.m [5]非常相似。与 QSWalk.m相比， QSWalk.jl 有两大优势。从易用性角度来看，它既可用于描述局域情形下的量子随机游走，也可用于描述全局情形下的量子随机游走。更重要的是，借助 Julia 语言的能力，它使用户能够无缝利用并行计算能力。对于任何量子软件框架而言，这第二方面都十分重要。

2.2.2 物理现象的模拟

从量子软件开发的角度来看，另一类重要的软件包专注于模拟量子硬件背后的物理现象。需要注意的是，这类软件不同于通用模拟器，例如 Qulacs 或Yao.jl；通用模拟器主要关注量子程序的逻辑结构模拟。相反，这类专用软件聚焦于特定的物理过程，因此能够精确再现特定现象的物理特性。

这类软件的一个典型示例是用于模拟离子量子计算机物理演化的 IonSim.jl 软件包。该软件包基于 QuantumOptics.jl [22]数值框架构建，提供与 QuTiP类似的功能。不过， IonSim.jl 提供了专用的应用程序接口，用于处理囚禁离子构型与激光相互作用时的动力学。因此，它支持仿真层，可用于开发和测试面向特定硬件架构的量子过程。例如，给定离子构型和激光脉冲的规格后，可以将其用作更高层的虚拟机，包括硬件供应商开发的量子工具包或量子编程语言。

2.3 Quantum领域特定语言

近期量子软件领域观察到的主要趋势，是开发软件库，使创建量子代码所用的元素能够嵌入通用编程语言中。开发专用硬件的公司 [12, 59, 49] 采用了这种方法，因为这使他们能够同时控制硬件和软件栈。许多此类软件框架都基于

on Python 编程语言，这尤其方便，因为量子计算技术经常与数据科学应用相关，而 Python 常常是开发者在这一领域的首选。因此，量子计算常被视为推动数据科学发展的前景技术，这类框架自然也能满足社区的需求。

然而，与 Python 库集成需要付出双重代价。首先，量子软件开发工具包通常缺乏量子电路模拟方面的能力，因此人们往往需要借助专用软件来完成软件开发中计算密集的部分。其次， Python 在符号计算方面的能力也有所不足。如果在流程中依赖从量子计算机获得的结果，这一点并不会造成限制。然而，如果需要应对复杂的优化景观，并希望微调变分量子本征求解器或量子近似优化算法等流程中使用的拟设，这一点就可能构成限制。

ProjectQ [53, 11] 提供了一种软件架构，可将高级语言程序编译为特定于硬件的指令。在某种程度上，这借鉴了专用高级语言的概念，与下文讨论的 LanQ [34] 和 QCL [39, 40, 41] 类似。不过，该实现以 Python 库的形式提供，便于使用该软件。 ProjectQ 最重要的特点是能够编译量子算法，并可面向任何特定的量子硬件实现。因此，它提供了一种硬件无关解决方案。遗憾的是，这也带来了一个问题：此类项目无法指望获得任何量子硬件供应商的专门支持。

2.4 量子编程语言

量子算法和通信协议是在量子电路语言中开发的，其中可逆逻辑运算是酉矩阵。对于简单算法而言，这种方法很方便；但若要处理复合数据类型，或算法需要调用经典子程序（含噪声中型量子算法便属于这种情况），就会非常困难。这种抽象能力的不足推动了量子编程语言领域的发展。迄今为止，人们已提出许多量子编程语言，自然可以将它们分为两类，分别代表量子软件工程中两种根本不同的方法：命令式和函数式。

2.4.1 命令式量子编程

第一类量子编程语言是基于命令式范式的语言。通常，这些语言提供了编程语言 Pascal 和 C 中常见的语法。为了支持量子计算，它们在标准语法中扩展了操作量子存储器所需的元素。

在这一类语言中， QCL （量子计算语言） [39, 40, 41] 是最先进且配有可用解释器的量子编程语言之一。其语法和经典数据类型层次结构基于 C。在最初的实现中，使用 QCL 编写的程序可以通过现有解释器执行。该解释器基于以 C++ 编写的 libqc 模拟库构建，执行速度出色。

基于命令式范式的第二个重要量子编程语言示例是 LanQ [34]。该语言旨在解决因缺乏支持量子通信的元素而产生的问题。此外， LanQ 是第一个完整规定操作语义的量子编程语言 [35]。这使得能够对程序的正确性进行形式化推理，并进一步开发该语言。语义对于优化以 LanQ 编写的程序也至关重要。

值得注意的是，上述两个命令式编程语言的例子都是作为研究项目开发的。遗憾的是，在 NISQ时代，旨在利用容错量子计算资源的专用编程语言这一构想并未得到发展。这令人惊讶，因为 QCL 和 LanQ的许多方面——例如量子条件和量子链接——都可能有助于量子软件开发过程。不过，这表明，广泛采用对于量子软件框架的成功至关重要。而要实现广泛采用，唯有借助某些传统编程语言的流行度。

2.4.2 函数式量子编程 g

函数式编程常被视为科学计算领域中传统方法的一种颇具吸引力的替代方案；这些传统方法主要基于命令式编程范式 [10]。函数式编程之所以更适合科学计算，尤其是用于量子计算系统建模，其中一个原因是函数式语言生成的代码具有透明性，而且易于在并行环境中执行。

早在 [18] 中，就有人讨论了将函数式编程语言用于科学计算，尤其是用于量子力学对象建模的潜力。在将函数式编程风格引入量子领域的早期尝试中，可以举出对 λ演算的扩展—— [26]，中定义的量子λ演算 [57] ，以及开发一种基于函数式编程语言的抽象环境，用于表示量子力学中使用的结构 [19, 51]。 [38]，提出了一种更务实的方法，探讨了在 Mathematica 环境中对量子算法进行符号模拟。然而，与量子力学系统建模方面的工作类似 [18, 20], ，迄今开发的基于函数式范式的量子编程语言 [29] 并不支持量子信息系统建模所需的关键结构，例如量子信道。因此，在量子计算过程中纳入并缓解量子错误方面，这些语言的适用性非常有限。现有基于函数式范式的工具和语言也存在类似缺陷，因而适用性受限。这一点在 NISQ时代尤为重要，因为量子机器实际上被视为数值子程序，经过微调，仅用于处理计算中的某个非常具体的环节。

在已提出的函数式量子编程语言中，有许多实验性项目专注于量子计算过程的理论层面。具体而言，Quipper 于 [8, 56], 提出，旨在成为一种可扩展、富于表达力的高阶函数式量子编程语言。近期，QPCF [61, 43] 也已提出，其量子编程方法以函数式演算为基础。不过，这两种语言都缺少完整的实际实现。这极大地限制了所提方法对当前量子软件生态系统的影响。此外，这些方法大多聚焦于基于量子随机存取机（QRAM）模型的标准量子计算概念，因此难以将其应用于 NISQ量子算法。利用这类工具实现量子优势的相关工作仍在进行中。本文下文将介绍的环境架构，在一定程度上受到函数式量子编程领域进展的启发。

2.5 零代码量子编程

创建量子软件的一种新兴方式基于近期兴起的零代码开发趋势。这类量子软件以法国量子硬件供应商 Pasqal 最近推出的 Pulsar Studio[45, 44] 解决方案为代表。 Pulsar Studio 可用于操控里德堡原子，这是该公司所开发硬件的底层技术。与 IonSim.jl的情况一样，所提供的应用程序接口经过调整，以适配底层物理现象所具备的能力。此外，与前者一样，代码面向特定硬件，因此 PularStudio 会受到硬件平台物理特性的限制。另一方面，程序员无需处理物理演化的内部细节。

3 符号量子编程环境

在回顾量子软件生态系统之后，我们现在将重点转向构建软件环境所需的要素，以解决现有工具包和框架所概述的一些局限。为此，我们首先列出此类环境的要求，这些要求反映了开发此类工具的经验。我们重点关注符号能力，因为这对于利用量子理论的理论层面来开发稳健的量子程序十分重要。我们还将描述该环境所需的架构，尤其会考虑量子变分算法的混合特性。

3.1 量子编程工具包的要求

具体而言，支持量子程序符号化开发的软件环境应解决以下问题：

- 稳定性与长期支持。现有软件包和工具包通常缺少这一要素。在某些情况下，这是由于软件和硬件开发进展迅速所致。尽管如此，缺乏稳定性仍是复现研究结果以及开发量子软件生态系统各项要素（例如过程的外部库）的一大障碍。

- 支持符号计算。这种支持对于处理量子态空间和量子信道空间的结构至关重要。此外，它还有助于支持基于信息几何的方法；在量子-经典混合算法中，这类方法常用于处理能量景观的优化问题。

- 通用编程语言的灵活性。如第 2 节所述，现有的众多解决方案都基于通用语言。就本研究而言，我们倡导采用函数式范式。值得庆幸的是，过去几年间，函数式编程的要素日益普及，如今许多主流编程语言都支持这一编程范式。

- 可扩展性。这一要求有多方面的理由。此类能力对于集成高性能数值库以及与外部服务通信至关重要，包括云端资源提供商 [2] 或其他资源，例如随机性来源 [30, 32]。此外，这些能力对于访问外部资源也至关重要，包括访问量子计算机模拟器（如 IonSim.jl）。从计算流水线的角度来看，这类模拟器被视为专用数值协处理器。

- 支持高效数值过程。这一要素有助于开展小型和中型量子程序的数值模拟。在许多情况下，这一要素依托函数库或由所用的计算机代数系统直接提供。需要指出的是，运行此类模拟的能力对于某些软件工程应用至关重要，包括量子软件测试。

- 支持并行与分布式计算。这种支持为量子系统模拟带来了一些优势。鉴于 IBM 提出的类似量子编织的方法 [13, 46], 意味着可以分布式执行模拟下一代量子计算机所需的计算，因此这一点尤为重要。

3.2 符号量子编程的实现

现在，我们来介绍一种量子软件框架的结构，该框架支持对量子对象进行函数式编程和符号操作 [33]。该框架以软件包的形式开发，使用 Wolfram 语言编写，可在 Mathematica 计算机代数系统中运行。在初始阶段，该软件包仅用于支持对量子态和量子操作进行符号分析 [31, 28]。

在任何量子软件框架所需的结构中，可以区分出三大类基本结构，分别侧重于以下方面：

- 逻辑与计算——用于管理量子存储器和处理相干量子信息的构造；

- 物理学——用于处理退相干并量化系统量子力学性质的要素；

- 可扩展性和互联性——支持与外部量子资源通信，并为量子-经典混合算法提供构建模块的功能。

我们还纳入了框架中使用的构造的三大主要类别：用于处理量子计算中所用基本组件的基础结构；为高级编程构造和物理概念提供支持的计算结构；以及用于处理变分量子算法中常用的混合量子-经典计算模型的变分结构。

下文所述的功能中，大多数基本结构和计算结构都包含在 QI 包中。部分变分结构则在 QINisq 包中实现。此外， QIExtras包还提供了一些额外功能。详细说明见项目代码库 [33]中的文档。

3.2.1 逻辑与计算

基本结构 在最基础的层面上，每个量子软件框架都必须处理量子比特（即量子信息的基本单元）的局部操作，并包含分配量子存储器及操控量子比特状态所需的元素。其中显然包括 Ket、 Bra 和 Proj 函数，用于支持量子存储器的量子比特级操作。然而，符号量子编程还需要实现用于操作一般形式的量子混合态和态矢量的函数，包括 SymbolicMatrix 和 SymbolicVector，以及用于定义量子门的函数，例如 SpecialUnitary。此外，还需要一些专用函数，以便在程序开发过程中利用量子对象的理论结构，例如 SymbolicHermitianMatrix和SymbolicBistochasticMatrix 。使用这类专用类型有利于测试开发出的程序，例如确保演化过程保留输入态的某些性质。此外，通过明确定义所处理对象的域，还可以支持推导有关这些对象的解析结果。

计算结构 在基本结构（包括量子比特和酉门）的基础上，支持计算的逻辑结构由用于管理寄存器、构建复杂量子门以及管理受控量子算符的功能构成。这些功能包括KroneckerProduct、 Gate和 CGate。该组还包括为常见量子算法提供通用模板的功能，其中既有量子傅里叶变换等标准量子算法，也有量子近似优化算法等变分量子算法。

变分结构 这组功能负责处理测量结果、管理目标函数，以及在变分循环中与经典优化器交互。在这种情况下，可以利用符号计算来提高优化结果的质量。此外，得益于符号能力，还可以利用优化景观的结构来优化变分量子算法中所用拟设的结构。

3.2.2 物理学

基本结构 在基础层面，与量子计算背后的物理学相关的功能，涉及量子系统动力学的结构。此类别中的功能包括用于管理 Kraus 表示的功能，包括 SuperoperatorToKraus、 Superoperator和 ApplyChannel。这些功能可用于提供误差模型，并将其纳入计算过程。

计算结构 该领域的功能主要关注量子计算过程中使用的复合系统。支持这一过程的函数示例是 ProductSuperoperator，它可以利用量子寄存器的结构而非单个量子比特来应用噪声模型。这组函数还包括用于处理量子寄存器排列和经典控制的量子操作的函数。在这种情况下，函数式编程能够非常清晰、简洁地实现诸如 PartialTrace 或任何其他作用于量子比特子集或量子子寄存器的操作。更多详情请见 [31]。

变分结构 通过管理硬件供应商提供的规格信息，可以纳入量子硬件的物理特性。或者，也可以假设某种简化的误差模型，并利用上述支持物理学的计算结构来构建一个模拟真实场景的误差模型。管理这些计算特性的另一种方法是采用随机化处理。这可以通过使用提供管理量子随机对象接口的函数来实现，其中包括 RandomKet、 RandomUnitary 和 RandomDynamicalMatrix。这种方法曾用于开发 [28] 中介绍的软件包。

3.2.3 可扩展性与互联性

任何量子软件框架预期提供的最后一组功能，要求支持在宿主编程语言或计算机代数系统中进行扩展。此类模块有许多实例，包括 IBM 开发的 qiskit.providers 应用程序接口 [15]。关于 Mathematica 计算机代数系统，文献 [28, 30] 介绍了可利用外部资源的量子模块。

基本结构 在基础层面上，这组功能包括例如管理外部随机性来源。此类功能不可或缺，因为这些函数提供的大多数特性都可以利用宿主环境的能力来模拟。在所述软件包的背景下，文献 [30] 介绍了为量子统计分析利用远程随机性来源的管理功能。另一个例子是利用离线真随机源。在这种情况下，需要注意管理随机数，同时不损害所生成随机对象的质量 [32]。

计算结构 此功能涵盖用于对量子对象实施非物理变换（即无法在实验室中实现的映射）的函数。最著名的例子是 Resuffling和 PartialTransposition。另一个例子是 TruncatedFidelity，它可在变分量子计算中用作目标函数，并基于量子设备的测量结果。

变分结构 此组中的过程用于管理量子计算机，包括定义和规范用于执行代码的回弯，以及使用经典控制结构管理量子存储器。此组函数提供了管理量子执行的手段，包括 RunGate、 RunCGate和 QRun。它还应提供管理回弯、分析结果、向经典优化子程序提供反馈，并可能修改拟设结构的手段。对于经典优化和拟设优化，通常会使用外部子程序来处理所需功能。在这种情况下，关于当前结构的附加信息可用于改进外部子程序的运行。

4 量子计算技术的应用

上一节介绍的架构和初步实现解决了量子软件用户可能遇到的一些问题。因此，关于现有方法优缺点的部分评论基于作者实现和运行量子代码的经验。不过，为了更全面地探讨这一主题，我们列举了两个量子软件对进一步发展至关重要的应用案例。第一个问题与为给定问题或某类问题寻找最佳拟设的过程有关。第二个问题侧重于测试用例的构建和量子软件测试。

4.1 电路构建与优化

在开发能够实现量子优势的量子算法过程中，主要问题之一是当前可用量子硬件（NISQ 计算机）的局限性。这类

机器在量子比特连通性和相干时间方面存在限制。因此，面向这类机器的任何量子变分算法都必须针对这些限制进行优化。在大多数情况下，这一过程所采用的方法需要由人类专家来完成，而专家必须了解各个量子比特的结构及其相互作用。然而，随着所开发的量子计算机规模迅速扩大，这项任务正变得难以完成。

因此，近期提出了采用机器学习方法的方案。其中一种方案是运用强化学习技术。具体而言， [42] 证明了深度强化学习（DRL）方法能够成功解决量子电路构造问题。

然而，现有的强化学习方法仅适用于非常专门的情况，通常涉及在特定机器上执行固定问题，或采用固定的噪声模型。由于计算开销很高，大多数研究并未考虑环境变化的情形，包括设备噪声的变化。因此，一旦系统-环境相互作用的特性发生变化，就需要重新进行学习。由此可见，机器学习方法若能持续适应不断变化的噪声模式，将会更为理想。为此，需要提供有关变化对所得结果影响的数据，并说明这种影响如何依赖于拟设结构的变化。这通常需要进行大量测试。

将符号计算用于这类问题，与在变分量子计算中探索优化景观的情形非常相似。因此，借助符号计算提供这类信息，有望加快探索可能解空间的过程。此外，重现量子机器模型并获得目标函数的解析形式，可以直接用于模拟量子计算过程，从而显著降低对计算能力的需求。

4.2 量子软件测试

测试量子软件正确性的问题，提供了另一个应用示例，说明量子计算中的符号求值能够带来显著益处。与标准软件一样，测试量子软件旨在提供自动化、系统化的方法，以确保量子代码的正确性。这种保证对于量子软件至关重要，主要有两个原因。

首先，在量子计算中，存储器的内部结构无法被观察，同时，由于可用状态空间远大于经典计算中的状态空间，其内部结构也更加复杂。因此，必须根据有关计算过程的有限信息来推断程序是否正确。这一点尤其棘手，因为对于一大类出现在量子化学现象建模相关算法中的内部状态，量子处理器的状态无法得到高效表示和分析。

其次，即使是规模相对较小的量子计算机，模拟量子信息处理的过程也需要大量计算资源。这个问题与内部状态表示问题有关。这也意味着，无法轻易将中型量子计算机的测试过程交由模拟器完成。因此，可以采用针对量子计算机量子子系统所获得的测试方法的推广形式——部分量子测试——来解决这一问题。

近期提出的一些解决方案旨在应对量子软件测试问题。在 [36] 中，提出了一种量子符号方法，用于生成测试用例，从而帮助发现量子程序中的错误。在 [27], 中，介绍了一种量子变异分析工具 Muskit ，用于 Qiskit 中定义的领域特定语言。另一种工具是 Quito [58]，同样面向 Qiskit，并支持自动生成测试套件，涵盖针对输入和输出定义的三种覆盖准则。 Muskit 和 Quito都要求用户了解量子设备的输入输出特性，而这并非总是可行的。

利用量子程序的符号表示，可以轻松扩展现有量子软件方法的适用性。例如，利用测试用例的符号形式，可以扩展生成的测试用例范围，并纳入具有特定属性的非经典态矢量。此外，通过推断部分演化的形式，可以生成仅针对选定量子子系统的测试用例。这在一定程度上有助于解决对整个量子态空间进行测试的问题，并增强验证结果。

5 结论

本文旨在从量子技术的视角概述量子软件开发，并阐述为充分发挥量子计算的潜力、通过量子软件实现这一目标所需支持的一些关键要素。阅读本文后应当记住的一个主要观点是，量子软件生态系统十分分散，且人们正在探索并实现许多交付量子软件的构想。这可以理解，因为量子计算仍是一个非常新的计算领域。然而，对用户而言，这会带来代码稳定性、文档和社区支持方面的难题。

另一方面，从软件工程的角度来看，量子代码中一些最重要的问题难以理解，其中包括量子态的结构，以及用于开发量子代码的非经典构造。因此，我们希望强调面向量子计算技术的软件工程教育的重要性。为了支持教育，我们需要稳定可靠的量子软件。

利用量子力学实现计算优势，是理论物理学家和计算机科学家最早研究的课题之一。然而，如今不能仅仅声称所提出的算法或协议能够实现理论上的优势；要推动量子计算技术的发展，迫切需要提供能够在现实应用场景中实现量子优势并超越经典算法的算法。不过，这要求量子算法能够在现实硬件上运行，而这类硬件通常在存储容量和精度方面受到显著限制。在此类硬件上执行的量子代码必须经过精细调整，以适应其限制，并经过测试以检查准确性。这些任务都无法通过人工完成，需要有软件来支持量子程序员。

量子计算技术的一个重要方面，涉及现实场景下量子计算能力与量子计算机固有不精确性影响之间的相互作用。量子计算能力体现在使用量子机器可能带来的优势上。这种不精确性通常被称为量子噪声，但也可能与量子计算机的局限性有关，因为在大多数情况下，所需的逻辑操作无法以任意精度执行。这会导致不可避免的误差，并始终存在于执行的量子程序中。因此，了解这种固有不精确性对使用量子计算机获得的结果有多大影响，并提供相应的应对方法，至关重要。量子符号编程使程序员能够深入了解量子程序的结构。这种洞察对于充分利用量子理论的结构以实现量子优势至关重要。

致谢 作者谨向 Mateusz Ostaszewski 和 Akash Kundu 对量子架构搜索问题提出的宝贵意见，以及与 Adam Glos、Ozlem Salehi 和 Ludmila Botelho 就量子软件生态系统展开的富有启发性的讨论表示感谢¨。本研究获得了波兰国家科学中心项目资助，资助协议编号为 2019/33/B/ST6/02011；同时也获得了波兰国家信息处理研究所通过 EuroHPC国家超级计算基础设施（EuroHPC PL）项目提供的支持。该项目由智能增长运营计划（2014—2020）资助，资助协议编号为 POIR.04.02.00-00-D014/20-00。

参考文献

[1] Shaukat Ali、Tao Yue 和 Rui Abreu。软件工程与量子计算相遇。《ACM 通讯》，65(4):84–88，2022年3月。doi:10.1145/3512340。[2]Amazon。Braket——加速量子计算研究，2021年。网址：https://aws.amazon.com/braket/quantum-computers/。[3] 欧盟委员会。《量子技术新战略研究议程》，2020年3月。https://europa.eu/!qmrH3k。[4] 欧洲创新委员会。EIC Pathfinder挑战：量子信息处理、通信与传感的替代方法，2022年6月。https://europa.eu/!Vnrh9X。

[5] Peter E. Falloon、Jeremy Rodriguez 和 Jingbo B. Wang。QSWalk：用于任意图上的量子随机游走的 Mathematica 软件包。《计算机物理通讯》，217:162–170，2017年8月。doi:10.1016/j.cpc.2017.03.014。[6] Sukhpal Singh Gill、Adarsh Kumar、Harvinder Singh、Manmeet Singh、Kamalpreet Kaur、Muham- mad Usman 和 Rajkumar Buyya。量子计算：分类法、系统综述与未来方向。《软件：实践与经验》，52(1):66–114，2021年10月。doi:10.1002/spe.3039。[7] Adam Glos、Jaros law Adam Miszczak 和 Mateusz Ostaszewski。QSWalk.jl：用于量子随机游走分析的 Julia 软件包。《计算机物理通讯》，235:414–421，2019年2月。doi:10.1016/j.cpc.2018.09.001。[8]Alexander S. Green、Peter LeFanu Lumsdaine、Neil J. Ross、Peter Selinger 和 Benoˆıt Val- iron。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大会论文集。ACM，2013年6月。doi:10.1145/2491956.2462177。[9] Aram W. Harrow 和 Ashley Montanaro。量子计算优势。《自然》，549(7671):203–209，2017年9月。doi:10.1038/nature23458。[10] Konrad Hinsen。函数式编程的前景。《计算科学与工程》，11(4):86–90，2009年7月。doi:10.1109/mcse.2009.129。[11] Thomas H¨aner、Damian S Steiger、Krysta Svore 和 Matthias Troyer。量子程序编译的软件方法。《量子科学与技术》，3(2):020501，2018年2月。doi:10.1088/2058-9565/aaa5cc。[12]IBM。IBM量子技术扩展路线图，2020年9月。网址：https://research.ibm.com/blog/ibm-quantum-roadmap。[13] IBM。在小型量子计算机上模拟大型量子电路的代价是多少？，2022年5月。网址：https://research.ibm.com/blog/circuit-knitting-with-classical-communication。[14]IBM。量子中心超级计算：下一波计算浪潮，2022年11月。https://research.ibm.com/blog/next-wave-quantum-centric-supercomputing。[15] IBM Qiskit。提供者接口，2022年。网址：https://qiskit.org/documentation/apidoc/provide

rs.html。

[16]英特尔。量子模拟器（Intel-QS/qHiPSTER），2020。网址：https://github.com/intel/intel-qs.[17] J.R. Johansson、P.D. Nation 和 Franco Nori。QuTiP 2：用于开放量子系统动力学的 Python 框架。《计算机物理通讯》，184(4):1234–1240，2013年4月。doi:10.1016/j.cpc.2012.11.019.[18] Jerzy Karczmarczuk。科学计算与函数式编程。《科学与工程计算》，1(3):64–72，1999。doi:10.1109/5992.764217.[19] Jerzy Karczmarczuk。量子力学的结构与诠释。载于：2003年 ACM SIGPLAN Haskell 研讨会论文集。ACM，2003年8月。doi:10.1145/871895.871901.[20] Jerzy Karczmarczuk。特定的“科学”数据结构及其处理。《理论计算机科学电子论文集》，66:195–209，2011年9月。doi:10.4204/eptcs.66.10.[21] Jakob S Kottmann、Sumner Alperin-Lea、Teresa Tamayo-Mendoza、Alba Cervera-Lierta、Cyrille Lavigne、Tzu-Ching Yen、Vladyslav Verteletskyi、Philipp Schleich、Abhinav Anand、Matthias De-groote、Skylar Chaney、Maha Kesibi、Naomi Grace Curnow、Brandon Solo、Georgios Tsilimigk-ounakis、Claudia Zendejas-Morales、Artur F Izmaylov 和 Al´an Aspuru-Guzik。TEQUILA：用于快速开发量子算法的平台。《量子科学与技术》，6(2):024009，2021年3月。doi:10.1088/2058-9565/abe567。

[22] Sebastian Kr¨amer、David Plankensteiner、Laurin Ostermann 和 Helmut Ritsch。QuantumOptics.jl：用于模拟开放量子系统的 Julia 框架。《计算机物理通讯》，227:109–116，2018年6月。doi:10.1016/j.cpc.2018.02.004.[23] Ryan LaRose。门级量子软件平台概述与比较。《Quantum》，3:130，2019年3月。doi:10.22331/q-2019-03-25-130.[24] Xiu-Zhe Luo、Jin-Guo Liu、Pan Zhang 和 Lei Wang。Yao.jl：用于量子算法设计的可扩展、高效框架。《Quantum》，4:341，2020年10月。doi:10.22331/q-2020-10-11-341.[25] F.L. Marquezino 和 R. Portugal。量子游走模拟器 QWalk。《计算机物理通讯》，179(5):359–369，2008年9月。doi:10.1016/j.cpc.2008.02.019.[26] Philip Maymin。扩展 λ 演算以表达随机化和量子化算法，1996。doi:10.48550/arXiv.qaunt-ph/9612052.[27] Enaut Mendiluze、Shaukat Ali、Paolo Arcaini 和 Tao Yue。Muskit：用于量子软件测试的变异分析工具。载于：2021年第36届IEEE/ACM自动化软件工程国际会议（ASE）论文集。IEEE，2021年11月。doi:10.1109/ase51524.2021.9678563.[28] Jaros law Adam Miszczak。在 Mathematica 中生成和使用真正随机的量子态。《计算机物理通讯》，183(1):118–124，2012年1月。doi:10.1016/j.cpc.2011.08.002。

[29] Jaros law Adam Miszczak。《量子计算的高级结构》。施普林格国际出版公司，2012 年。doi:10.1007/978-3-031-02516-7。

[30] Jaros law Adam Miszczak。使用在线量子随机数生成器在 Mathematica 中生成真正随机的量子态。《计算机物理通讯》，184(1):257–258，2013 年 1 月。doi:10.1016/j.cpc.2012.08.012。

[31]Jaros law Adam Miszczak。量子信息论中的奇异值分解与矩阵重排。《现代物理学国际期刊C辑》，22(09):897–918，2011 年 9 月。doi:10.1142/s0129183111016683。

[32] Jaros law Adam Miszczak 和 Michael Wahl。用于访问基于文件的随机性来源的 Mathematica Randfile 软件包，2013 年。doi:10.48550/arXiv.1302.2738。

[33] Jaros lawAdam Miszczak、Pucha la Zbigniew 和 Piotr Gawron。适用于 Mathematica 计算机代数系统的 QI 包，2011 年至今。网址：https://github.com/iitis/qi。

[34]Hynek Mlnaˇr´Ik。LanQ——一种量子命令式编程语言，2008 年。网址：https://lanq.sourceforge.net/。

[35] Hynek Mlnaˇr´Ik。LanQ 量子编程语言的语义。《量子信息国际期刊》，06(supp01):733–738，2008 年 7 月。doi:10.1142/s0219749908004031。

[36]Jiang Nan、Wang Zichen 和 Wang Jian。量子符号执行，2022年。doi:10.48550/arXiv.2209.08582。[37]NVIDIA。cuQuantum——加速量子计算研究，2022年。URL：https://developer.nvidia.com/cuquantum-sdk。[38]Peter Nyman。用于模拟量子算法的符号经典计算机语言。载于《量子交互》，第158–173页。Springer Berlin Heidelberg，2009年。doi:10.1007/978-3-642-00834-4_14。.[39] Bernard Omer。量子计算的过程形式主义。硕士论文，维也纳工业大学，1998年。.[40] Bernard Omer。QCL 中的量子编程。硕士论文，维也纳工业大学，2000年。

.[41] Bernard Omer。结构化量子编程。博士论文，维也纳工业大学，2003年。[42] Mateusz Ostaszewski、Lea M. Trenkwalder、Wojciech Masarczyk、Eleanor Scerri 和 Vedran Dun-jko。用于优化变分量子电路架构的强化学习。载于 M. Ranzato、A. Beygelzimer、Y. Dauphin、P.S. Liang 和 J. Wortman Vaughan 编，《神经信息处理系统进展》，第34卷，第18182–18194页。Curran Associates, Inc.，2021年。doi:10.48550/arXiv.2103.16089。[43]Luca Paolini、Mauro Piccolo 和 Margherita Zorzi。QPCF：高阶语言与量子电路。《自动推理杂志》，63(4):941–966，2019年3月。doi:10.1007/s10817-019-09518-y。[44]Pasqal SAS。探索 Pulser Studio 的功能，2023年。URL：https://www.pasqal.com/articles/pulser-s-1。[45] Pasqal SAS。Pulser Studio：面向中性原子量子计算的无代码开发平台，2023年。URL：https://pulserstudio.pasqal.cloud/。[46]Christophe Piveteau 和 David Sutter。通过经典通信进行电路编织，2022 年。doi:10.48550/arXiv.2205.00016。[47] Quantiki.org。量子计算模拟器列表，2005年至今。URL：https://www.quantiki.org/wiki/list-qc-simulat

ors.

[48]Quantum Open Software Foundation. 开源量子软件项目，2019 年至今。URL：https://github.com/qosf/awesome-quantum-software.[49] Rigetti Inc. Forest SDK，2023 年。URL：https://qcs.rigetti.com/sdk-downloads.[50]Vincent Russo. toqito——量子信息理论工具包：用于研究量子信息的 Python 软件包。开源软件期刊，6(61):3082，2021 年 5 月。doi:10.21105/joss.03082.[51] Amr Sabry. 使用 Haskell 对量子计算建模。载于《2003 年 ACM SIGPLAN Haskell 研讨会论文集》。ACM，2003 年 8 月。doi:10.1145/871895.871900..[52] Ozlem Salehi、Zeki Seskir 和 Ilknur Tepe. 面向计算机科学的量子计算入门方法。IEEE 教育汇刊，65(1):1–8，2022 年 2 月。doi:10.1109/te.2021.3078552.[53] Damian S. Steiger、Thomas H¨aner 和 Matthias Troyer. ProjectQ：用于量子计算的开源软件框架。Quantum，2:49，2018 年 1 月。doi:10.22331/q-2018-01-31-49.[54] Athanasios Stratikopoulos、Ales Kubicek、Juan Fumero 和 Christos Kotselidis. Tornadoqsim：用于量子电路模拟的开源高性能模块化系统。2022 年。URL：https://github.com/beehive-lab/TornadoQSim.[55] Yasunari Suzuki、Yoshiaki Kawase、Yuya Masumura、Yuria Hiraga、Masahiro Nakadai、Jiabao Chen、Ken M. Nakanishi、Kosuke Mitarai、Ryosuke Imai、Shiro Tamiya、Takahiro Yamamoto、Tennin Yan、Toru Kawakubo、Yuya O. Nakagawa、Yohei Ibe、Youyuan Zhang、Hirotsugu Yamashita、Hikaru Yoshimura、Akihiro Hayashi 和 Keisuke Fujii. Qulacs：面向研究用途的快速、通用量子电路模拟器。Quantum，5:559，2021 年 10 月。doi:10.22331/q-2021-10-06-559.[56]Benoˆıt Valiron、Neil J. Ross、Peter Selinger、D. Scott Alexander 和 Jonathan M. Smith. 编程量子未来。ACM 通讯，58(8):52–61，2015 年 7 月。doi:10.1145/2699415.[57] Andr´e van Tonder. 一种用于量子计算的 λ演算。SIAM 计算杂志，33(5):1109–1135，2004 年 1 月。doi:10.1137/s0097539703432165.[58] Xinyi Wang、Paolo Arcaini、Tao Yue 和 Shaukat Ali. Quito：面向量子程序的覆盖率引导测试生成器。载于《2021年第36届IEEE/ACM自动化软件工程国际会议（ASE）》。IEEE，2021 年 11 月。doi:10.1109/ase51524.2021.9678798.

[59]Xanadu Inc. 用于量子计算机编程的 PennyLane 软件库，2016 年至今。URL: https://xanadu.ai/products/pennylane/.[60]Jianjun Zhao. 量子软件工程：现状与展望，2020 年。doi:10.48550/arXiv.2007.07047.[61] Margherita Zorzi. 量子演算——从理论到语言设计。Applied Sciences，9(24):5472，2019 年 12 月。doi:10.3390/app9245472.