量子计算：简介、应用与前景

潘屹
上海交通大学致远学院 2022 级，上海 201100

论文是否发表：否

摘要 量子计算是基于量子力学理论的全新计算方式，相比起经典计算机的计算模式，量子计算在应对一些特定问题时能带来较大的效率提升。笔者在计算机科学的伟大思想这门课程（CS1950@SJTU）中了解到了基础的量子计算知识，对这一领域产生好奇与兴趣，在课后阅读了量子计算相关材料、书籍，学习量子计算课程（CS294-2@UCBerkeley）的课堂笔记；对量子计算的发展、理论与应用有了初步的认知。本文对当前量子计算的基础理论进行科普性概述，介绍了一些简单的量子算法与应用，并展望量子计算的前景。所有源代码在 https://github.com/Conless/quantum-computing-intro 开源。

关键词 量子计算；量子算法

[引言]

量子计算，指一种运用量子力学现象，例如量子叠加、量子干涉与量子纠缠进行计算的方式，是过去的数十年中兴起的一种全新计算方式，在近几年中也是一个在各类科普中高频出现的热点词汇。下面的内容从经典计算机讲起，简要地介绍量子计算的发展历史与其重要意义[1]。

引入量子计算的初衷是打破经典计算机出现的性能瓶颈。经典计算机的发展领先于量子计算机约半个世纪；Alan Turing 在 1936 年发表的论文中提出了图灵机（Turing Machine）的模型 [2]，宣告了现代计算机科学的诞生。Turing 证明了存在一台通用图灵机，即任何可以在个人电脑上执行的算法，都可以在这台图灵机上完成，这个论断被称为 Church-Turing 命题。随后，von Neumann 设计出了这样的理论模型，用实际元件实现了通用图灵机的全部功能，在随后的几十年里，个人计算机的发展也一直沿用 von Neumann 架构，其发展速度遵从 1965 年 Gordon Moore 所概括的 Moore 定律，即集成电路中单位面积的晶体管数量，以及与之相对应的，计算机计算速度，大约每两年增长一倍 [3]。

自 Moore 定律提出以来，经典计算机硬件发展速度都近似地遵从于该定律；但进入 21 世纪以来，Moore 定律的有效性逐渐下降，许多研究人员认为其将在 21 世纪的前 20 年终结，著名芯片企业，Nvidia 公司的首席执行官 Jensen Huang 就于日前宣称，Moore 定律已死 [4]。其中的重要原因在于传统半导体原件在栅极线宽较小时，可能会出现量子隧穿等效应 [5]。

解决 Moore 定律最终失效的一个可能方案是采用不同的计算模式，量子计算就是其中一种。量子计算这一概念于 1980 年由物理学家 Paul Benioff 首次提出 [6]。随后，Feymann 指出，在经典计算机上有效地模拟量子系统的演化似乎是不可能的，量子计算机可能可以模拟经典计算机无法做到的事情 [7]，并引入了早期版本的量子电路符号 [8]。1985 年，David Desutsch 提出，能否用量子力学原理推导出更强的 Church-Turing 命题，并引导出了现代量子计算机的概念 [9]。他举了一个简单的例子（见节 1.4），表明量子计算机的计算能力确实超过了经典计算机。

在随后十年里，量子算法相关研究不断涌现出新的成果。1994 年，Peter Shor 提出了一种新的量子算法，可以有效地解决大整数的质因数分解问题 [10]，这在经典计算机上被认为是不可解的[2]；Shor 算

[1] 部分内容参考了维基百科词条与 *Quantum Computation and Quantum Information*[1]

[2] 难以在多项式时间内解决

法的出现一度让基于质因数分解的 RSA 加密算法的安全性受到威胁 [11]。1996 年，Lov Grover 证明了，在非结构化搜索空间进行搜索的问题也可以通过量子计算机加速 [12]，这种搜索方法的广泛适用性引起了人们对 Grover 算法的相当关注。

由此可以看出，量子计算与量子算法可以对许多运用经典计算机难以解决的问题进行加速。如今，量子计算领域依旧处于蓬勃发展之中。在下面的章节中，量子计算的相关基础理论将会被介绍。

1 量子计算基础

量子计算理论以量子物理领域的数学物理方法、记号与公式为重要基础；本节将对相关内容进行简单介绍。在学习量子比特与其相关表示的过程中，笔者发现其与本科一年级所学习的线性代数课程有着高度相关性，再一次感受到了理论计算机科学中数学的重要性。

1.1 量子比特

量子比特（qubit）是量子计算和量子信息的基本概念。在经典计算机中，经典比特以 0 和 1 两种状态存在。而量子比特也有相对应的两种形态，在 Dirac 表示法中记作 $|0\rangle, |1\rangle$，任何一个两态的量子系统都可以实现这一点，例如在氢原子中 $|0\rangle, |1\rangle$ 可以代表基态和第一激发态，在质子自旋中可以表示任意方向的 $+\frac{1}{2}, -\frac{1}{2}$ 分量。与经典比特不同的是，量子比特除了 $|0\rangle$ 和 $|1\rangle$ 态，还可以处于叠加态（superposition）；这是 $|0\rangle, |1\rangle$ 两态的一个线性组合，可以记为

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{或列向量形式} \quad |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{1}$$

其中 $\alpha, \beta \in \mathbb{R}$ （实际上取值域为 $\mathbb{C}^2$，方便起见，这里首先以实数系作为研究对象）且 $|\alpha|^2 + |\beta|^2 = 1$。此时，我们不妨从几何的角度理解这一式子：如果将 $|0\rangle$ 看作 x 方向上的单位向量，$|1\rangle$ 看作 y 方向上的单位向量，那么容易发现，$|\psi\rangle$ 就在单位圆上运动，其模长始终为 1。更进一步地，考虑态向量 $|\psi\rangle$ 在这组正交基上的投影，这也就是量子力学中的测量过程：

$$\overrightarrow{\psi_0} = \overrightarrow{0}^T \overrightarrow{\psi}\, \overrightarrow{0} \Rightarrow |\psi\rangle^\dagger |0\rangle|0\rangle = \langle\psi||0\rangle|0\rangle = \langle\psi|0\rangle|0\rangle = \alpha|0\rangle \tag{2}$$

其中，左式是在线性代数中熟知的投影运算形式 $\boldsymbol{p} = \boldsymbol{u}^T \boldsymbol{v}\boldsymbol{u}$，右式是在量子力学中用 Dirac 记号刻画的更加优美的形式，其中将 $|\psi\rangle^\dagger$ 改写为 $\langle\psi|$，进而将 $|\psi\rangle \cdot |0\rangle$ 缩写为 $\langle\psi|0\rangle$，再考虑投影向量模长的平方

$$\alpha\langle 0| \cdot \alpha|0\rangle = \alpha^2. \tag{3}$$

而在 $|1\rangle$ 上的分量模长平方同理则为 β^2，这样一来，约束条件 $\alpha^2 + \beta^2 = 1$ 的意义就逐渐清晰：被测量时，其落在 $|0\rangle, |1\rangle$ 上的概率之和恰好为 1.

在经典比特意义下，n 个比特可以表示 2^n 种不同的状态，在量子比特意义下，也可以通过 n 个量子比特的叠加生成一个 $\dim = 2^n$ 的量子比特空间。这个时候需要引入张量积（tensor product）这一概念，这在线性代数课程中尚未涉及，但是可以用一些简单的例子进行理解：

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} \\ 2 \times \begin{bmatrix} 3 \\ 4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 6 \\ 8 \end{bmatrix}.$$

于是，叠加两个量子比特可以看作对两个向量进行张量积。这意味着，我们可以首先对基求张量积

$$|0\rangle\otimes|0\rangle=|00\rangle,|0\rangle\otimes|1\rangle=|01\rangle,|1\rangle\otimes|0\rangle=|10\rangle,|1\rangle\otimes|1\rangle=|11\rangle$$

上式可以从许多角度进行理解，例如若置 $|0\rangle=\boldsymbol{e}_0=(0,1),|1\rangle=\boldsymbol{e}_1=(1,0)$ 为 $\mathbb{R}^2$ 的标准基，计算可得 $|00\rangle=(0,0,0,1)=\boldsymbol{e}_0,|11\rangle=(1,0,0,0)=\boldsymbol{e}_3$ 是 $\mathbb{R}^4$ 的标准基（这里选取 0-base 可以让单位基的角标恰好为 $|\rangle$ 中二进制数的十进制表示）。而后，计算两个任意单量子比特的张量积，例如

$$|\psi_1\rangle=\begin{bmatrix}\alpha_1\\ \beta_1\end{bmatrix},|\psi_2\rangle=\begin{bmatrix}\alpha_2\\ \beta_2\end{bmatrix} \tag{4}$$

就可以得到

$$|\psi_1\psi_2\rangle=|\psi_1\rangle\otimes|\psi_2\rangle=\alpha_1\alpha_2|00\rangle+\alpha_1\beta_2|01\rangle+\alpha_2\beta_1|10\rangle+\beta_1\beta_2|11\rangle=\begin{bmatrix}\alpha_1\alpha_2\\ \alpha_1\beta_2\\ \beta_1\alpha_2\\ \beta_1\beta_2\end{bmatrix}. \tag{5}$$

特殊地，长度为 n 的全零量子比特表示为 $0^{\otimes n}$，全一量子比特表示为 $1^{\otimes n}$。

1.2　量子比特门

每一个量子比特都可以由一个特定维度的单位向量表示，而量子比特之间的运算同样可以用向量变换的手段，即矩阵，进行刻画。值得注意的是，根据 von Neumann 提出的量子力学公设 [13]，量子比特之间的运算必须是酉变换（unitary transformation），也即，两个沿时间先后顺序出现的量子比特 $|\psi\rangle,|\psi'\rangle$ 满足 $|\psi'\rangle=U|\psi\rangle$，其中变换矩阵 U 满足其共轭转置与自身的乘积 $U^\dagger U=I$。根据线性代数知识，这蕴含了 $U^\dagger=U=U^{-1}$。在此基础上，我们可以构建出几个常用的合法量子运算门

定义矩阵 X 表示非门

$$X=\begin{bmatrix}0&1\\1&0\end{bmatrix},\qquad\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\mapsto\begin{bmatrix}\beta\\ \alpha\end{bmatrix}. \tag{6}$$

显然地，这表示将单量子比特的两位相互交换。这与传统非门是相似的。

定义矩阵 Z,U 表示相位翻转和旋转

$$Z=\begin{bmatrix}1&0\\0&-1\end{bmatrix},\qquad\begin{bmatrix}\alpha\\ \beta\end{bmatrix}\mapsto\begin{bmatrix}\alpha\\ -\beta\end{bmatrix}, \tag{7}$$

$$U=\begin{bmatrix}\cos\theta&-\sin\theta\\ \sin\theta&\cos\theta\end{bmatrix},\qquad\begin{bmatrix}\cos\alpha\\ \sin\alpha\end{bmatrix}\mapsto\begin{bmatrix}\cos(\alpha+\theta)\\ \sin(\alpha+\theta)\end{bmatrix}. \tag{8}$$

定义矩阵 H 表示 Hadamard 门

$$H=\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix},\qquad\begin{bmatrix}\cos\alpha\\ \sin\alpha\end{bmatrix}\mapsto\begin{bmatrix}\cos(\frac{\pi}{4}-\alpha)\\ \sin(\frac{\pi}{4}-\alpha)\end{bmatrix} \tag{9}$$

这代表着将一个在 $\mathbb{R}^2$ 的向量沿 $\theta = \frac{\pi}{8}$ 进行对称。于是

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle, \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle \end{aligned} \tag{10}$$

这两个向量分别对应了 $\pm\frac{\pi}{4}$，前者位于 $|0\rangle, |1\rangle$ 的角平分线上，代表了两者的等概率叠加。因此，Hadamard 门在实际运算中常用于制备叠加态，例如，若需要 n 个量子比特的等概率叠加，即 $|+\rangle$ 的 n 维张量积

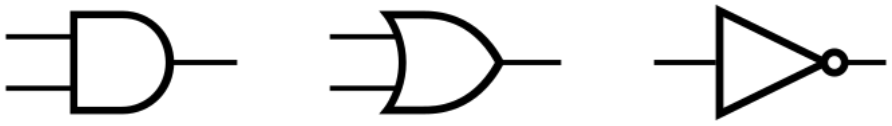
$$|+\rangle^{\otimes n} = \underbrace{|+\rangle \otimes \cdots \otimes |+\rangle}_{n\text{ 个 }|+\rangle} \tag{11}$$

可以考虑对线性算符 H 作张量积并作用于 $|0\rangle$ 上，就得到了

$$H^{\otimes n}(|0\rangle^{\otimes n}) = |+\rangle^{\otimes n}. \tag{12}$$

1.3 量子电路及其简单应用

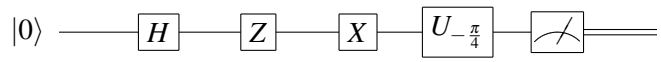
在经典计算机中，逻辑电路由一系列逻辑门与电路元件构成，图 1.1 即为熟知的与或非逻辑门的表示，它们之间相互嵌套组合，构成了经典电子计算机的电路体系。



(a) 逻辑与门 (b) 逻辑或门 (c) 逻辑非门

图 1.1: 与或非门（ANSI 及 IEEE 标准）

而在量子电路中的计算则由一系列逻辑门、测量与赋值操作构成。但是，不同于传统电路是用金属线所连接以传递电压信号或电流信号；在量子线路中，线路是由时间所连接，亦即量子比特的状态随着时间自然演化，一直到遇上逻辑门而被操作。另一方面，经典计算机的大多数基本逻辑门（除了非门）都是不可逆的，例如，对于与门，我们不可能从输出信息的每一位恢复到两个输入信息；而量子计算中的每一步操作都是酉变换。刚刚的几个酉变换在量子电路中都有他们的电路元件表示



上面的过程中，量子比特的变换过程为

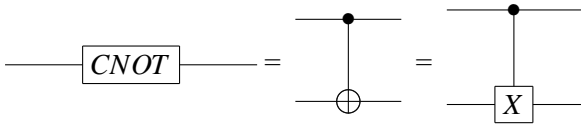
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \to \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \to \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \to \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \to \begin{bmatrix} 1 \\ 0 \end{bmatrix} \xrightarrow{\text{measure}} |\psi\rangle$$

在实际的计算中，仅仅进行单元操作显然是不够的，因此需要考虑将对单量子比特的操作拓展到多量子比特, 从一种简单的情况出发。现在有两个量子比特，但是仅对其中的一个，即目标量子比特进

行操作，另一个作为控制量子比特，决定是否对目标进行否运算, 于是容易得到这一操作的酉矩阵

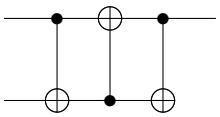
$$U_{\mathrm{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{13}$$

当且仅当控制量子比特为 1 时, 目标量子比特通过非门. 可以被表示为 $|A,B\rangle \to |A,B\oplus A\rangle$。因此也被形象地表示为



理论研究证明，任何多量子比特逻辑门可以由受控非门和单量子门组成 [1]，因此，受控非门具有通用性，这也与经典电路中与非门（XAND）的通用性相对应。

下面介绍几个量子电路的简单应用。交换两个量子比特的电路可以被表示为



这是因为容易验证

$$\begin{aligned} |a,b\rangle &\to |a,a\oplus b\rangle \\ &\to |a\oplus(a\oplus b),a\oplus b\rangle = |b,a\oplus b\rangle \\ &\to |b,(a\oplus b)\oplus b\rangle = |b,a\rangle. \end{aligned} \tag{14}$$

如果在二维 Hadamard 门后面跟着一个 CNOT 门

对四个二维本征态进行计算, 我们会得到

$$\begin{aligned} |00\rangle &\mapsto |\beta_{0,0}\rangle \frac{|00\rangle + |11\rangle}{\sqrt{2}} \\ |01\rangle &\mapsto |\beta_{0,1}\rangle \frac{|01\rangle + |10\rangle}{\sqrt{2}} \\ |10\rangle &\mapsto |\beta_{1,0}\rangle \frac{|00\rangle - |11\rangle}{\sqrt{2}} \\ |11\rangle &\mapsto |\beta_{1,1}\rangle \frac{|00\rangle - |10\rangle}{\sqrt{2}} \end{aligned} \tag{15}$$

概括地

$$|\beta_{x,y}\rangle = \frac{|0,y\rangle + (-1)^x |1,\bar{y}\rangle}{\sqrt{2}}. \tag{16}$$

这就是著名的 Bell 态（或 EPR[3] 对），它描述了一种最简单的量子纠缠示例：上面的四个量子比特

[3] Einstein-Podolsky-Rosen

虽然都具有两个维度，但是在测量中，只要测量了其中的一维，另一维也被唯一地确定了。Bell 态的这一性质使其在超密编码与量子隐形传态的领域扮演着重要的角色。

1.4 量子电路的设计：Deutsch-Jozsa 算法

Desutsch-Jozsa 算法可由一个简单的游戏进行引入：Alice 从 $0-2^n-1$ 中选一个数 x 并将其传送给 Bob。Bob 计算出某个函数 $f(x)$ 的值，可以为 0 或 1，并将它传回给 Alice。已知该函数只有可能有两种情况：要么 $f(x)$ 对于所有的 x 均为常数，要么 $f(x)$ 恰好对于一半的 x 取 0，一半的取 1。Alice 怎样能够最快地判断 $f(x)$ 的类型？

形式化地，已知函数 $f:\{0,1\}^{\otimes n}\to\{0,1\}$ 一定是下列两种极端形式的一种：

1. Constant：$f(x)\equiv 0$ or $f(x)\equiv 1$；
2. Balanced：$f(x)=0$ for half of $\{0,1\}^{\otimes n}$, and $f(x)=1$ for the other half

问如何用最少的查询次数确定 f 属于二者中的哪一种。

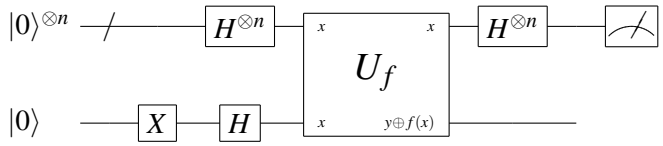


图 1.2: Deustch-Jozsa 算法的量子电路

考虑图 1.2 所示量子电路，输入 U_f 的初态为 $|+\rangle^{\otimes n}|-\rangle$。考虑初态的前 n 位，设其在某一状态下为 x，那么

$$U_f|x\rangle|-\rangle=|x\rangle|-\otimes f(x)\rangle=(-1)^{f(x)}|x\rangle|-\rangle. \tag{17}$$

从而

$$U_f|+\rangle^{\otimes n}|-\rangle=\sum_{x\in\{0,1\}^{\otimes n}}\frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle|-\rangle. \tag{18}$$

另外，容易验证，对于单量子比特门 $H|x\rangle=\sum_{z\in\{0,1\}}(-1)^{xz}|z\rangle/\sqrt{2}$，将这一结果推广到 n 个量子比特上，就有了

$$H^{\otimes n}|x\rangle=\sum_{z\in\{0,1\}^{\otimes n}}\frac{(-1)^{x\cdot z}}{\sqrt{2^n}}|z\rangle. \tag{19}$$

从而

$$H\left(\sum_{x\in\{0,1\}^{\otimes n}}\frac{(-1)^{f(x)}}{\sqrt{2^n}}|x\rangle\right)=\sum_{x,z\in\{0,1\}^{\otimes n}}\frac{(-1)^{x\cdot z+f(x)}}{2^n}|z\rangle\xrightarrow{\text{measure}}|\psi\rangle \tag{20}$$

于是考虑测量结果在 $\langle 0^{\otimes n}|$ 上的分量，即 z 只取 $|0\rangle^{\otimes n}$ 时表达式的值

$$\langle 0^{\otimes n}|\psi\rangle=\sum_{x\in\{0,1\}^{\otimes n}}\frac{(-1)^{f(x)}}{2^n} \tag{21}$$

因此，当 f 为常值函数时，测量结果必为 $|0\rangle^{\otimes n}$；当 f 为平衡函数时，测量结果不可能出现 $|0\rangle^{\otimes n}$。在整个过程中，运用了 3 个 n 量子比特门，总时间复杂度在 $O(n)=O(\log N)$ 级别，相比于朴素算法 $O(N)$ 的时间开销，量子算法在效率上产生了质的飞跃。Deustch-Jozsa 算法是第一个具有重要意义的量子算

法，其出现证明了量子计算在一些方面有着经典计算机无可比拟的性能，为接下来出现的许多量子算法奠基。

2　量子算法应用：以 ACM 班程序设计作业为例

2.1　背景与题目

在量子算法发明之初，以 Deutsch-Jozsa 为例的多种算法应用范围都相对局限，对于多数实际问题都无能为力或无法带来显著变化。但随着量子算法的逐步发展，许多经典计算机能解决的问题都能被量子计算机找到更优的方案。下面以上海交通大学 ACM 班 2022 级程序设计大作业的一道题 P1754 int2048-乘法速度测试 为例，讲解量子算法的简单应用。

题目的大意为：给定若干个大整数，求它们的乘积，所有运算结果范围在 $[1,10^{200000}]$ 内。为简化题意，题目可以改写为，给定正整数 a,b 满足 $a,b\le 10^{100000}$，求 $a\times b$。一种朴素的做法是直接模拟竖式乘法，其时间复杂度为 n^2（其中 $N=\lceil \log_k \max\{a,b\}\rceil$，$k$ 为运算过程中的进制）。进一步地，可以考虑整数的多项式表示：对于一个 k 进制整数

$$a=\overline{a_1\cdots a_N}=\sum_{i\in[N]} a_i k^{N-i} \tag{22}$$

考虑多项式

$$A(z)=\sum_{i\in[N]} a_i z^{N-i} \tag{23}$$

同理可构造出 b 对应的多项式 $B(z)$，那么令

$$C(z)=A(z)B(z)=\sum_{i\in[2N]}\left(\sum_{j\in[i]} a_j b_{i-j}\right) z^i \tag{24}$$

在经典计算机中，利用快速傅里叶变换（Fast Fourier Transform）与其逆变换可以在 $O(N\log N)$ 的时间内求出对应的多项式 C[14]，进行进位就得到了 $a\times b$。接下来，尝试使用量子计算机得到一种更加高效的做法。

2.2　前置知识

在离散傅里叶变换（Discrete Fourier Transform，DFT）中，我们熟知对多项式 $f(x)$ 进行点值 (ω_N^k, y_k) 求值的方法

$$y_k=\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j \omega_N^{jk}. \tag{25}$$

在量子计算机中，我们同样希望进行相同的变换，使得 $N-1$ 维量子态 $|X\rangle=\sum_{j=0}^{N-1} x_j|j\rangle$，经过变换得到 $|Y\rangle=\sum_{k=0}^{N-1} y_k|k\rangle$，其中 y_k 与 x_k 的关系满足上式，$|j\rangle,|k\rangle$ 表示其二进制分解所得结果的张量积。代入可得

$$|Y\rangle=\sum_{k=0}^{N-1} y_k|k\rangle=\sum_{k=0}^{N-1}\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j\omega_N^{jk}|k\rangle\right)=\sum_{j=0}^{N-1} x_j\left(\frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}\omega_N^{jk}|k\rangle\right) \tag{26}$$

对比 $|X\rangle,|Y\rangle$ 的形式，发现变换的实质就是一次基变换

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} |k\rangle \tag{27}$$

两组标准正交基的变换可改写为酉变换

$$U_{\mathrm{QFT}} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \cdots & \omega_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{(N-1)\times 2} & \cdots & \omega_N^{(N-1)(N-1)} \end{bmatrix} \tag{28}$$

这样就完成了一个满足量子计算要求的傅里叶变换算法，也即量子傅里叶变换（Quantum Fourier Transform，QFT）。

2.3 电路设计

上面的内容已经完成了量子傅里叶变换的算法设计，但得到了这样的酉阵，还不能直接设计出所需的量子电路，因此，在这里将式 27 进行进一步演化：考虑直接对 $\omega_{2^n}^{jk} = \exp(\frac{2\pi \mathrm{i} jk}{2^n})$ 指数中的 $\frac{k}{2^n}$ 进行二进制小数分解，也即对 k 进行二进制分解，得到

$$\begin{aligned} |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} |k\rangle &= 2^{-\frac{n}{2}} \sum_{l\in[n], k_l\in\{0,1\}} \exp\left(2\pi \mathrm{i} j \sum_{l\in[n]} k_l 2^{-l}\right) |k_1, \cdots, k_n\rangle \\ &= 2^{-\frac{n}{2}} \bigotimes_{l\in[n]} \left(\sum_{k_l\in[0,1]} \exp(2\pi \mathrm{i} j k_l 2^{-l}) |k_l\rangle \right) \\ &= 2^{-\frac{n}{2}} \bigotimes_{l\in[n]} \left(|0\rangle + \exp(2\pi \mathrm{i} j 2^{-l}) |1\rangle \right) \\ &= 2^{-\frac{n}{2}} \bigotimes_{l\in[n]} \left(|0\rangle + \exp\left(2\pi \mathrm{i} \left\{ \frac{j}{2^l} \right\}\right) |1\rangle \right) \end{aligned} \tag{29}$$

这里的 $\{\frac{j}{2^l}\}$ 表示取 $\frac{j}{2^l}$ 的小数部分，由于 $e^{2\pi \mathrm{i}} = 1$，因此等式是成立的。为了方便构建电路，记

$$\frac{j}{2^n} = \overline{0.j_1 j_2 \cdots j_n} \tag{30}$$

那么式 29 就可以写作

$$2^{-\frac{n}{2}} \bigotimes_{l\in[n]} \left(|0\rangle + \exp(2\pi \mathrm{i} \overline{0.j_{n-l+1} \cdots j_n}) |1\rangle \right)$$

这种张量积意义下的量子电路就可以通过旋转门 R_k 加以控制生成，

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & \exp\left(\frac{2\pi \mathrm{i}}{2^k}\right) \end{bmatrix} \tag{31}$$

特殊地，H 可以直接看作自身受控的 R_1 门，这是因为 $H|0\rangle = |+\rangle, H|1\rangle = |-\rangle$

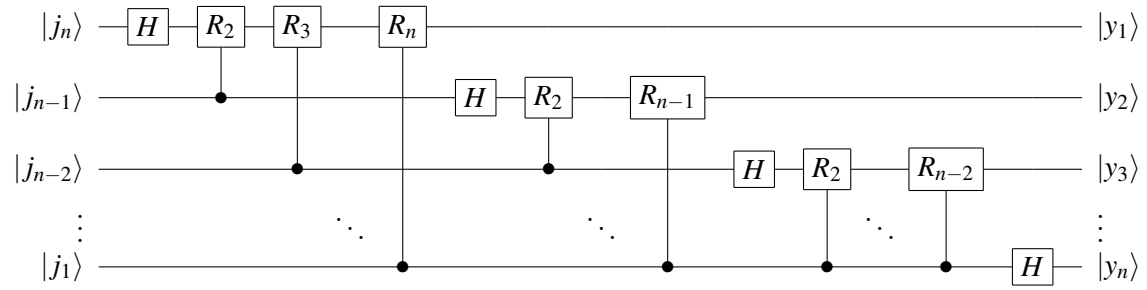
例如，考虑输入态 $|j_1,\cdots,j_n\rangle$，将其首位通过 H 门将得到

$$\frac{1}{\sqrt{2}}\left(|0\rangle+\exp(2\pi \mathrm{i}\overline{0.j_1})|1\rangle\right)|j_2,\cdots,j_n\rangle \tag{32}$$

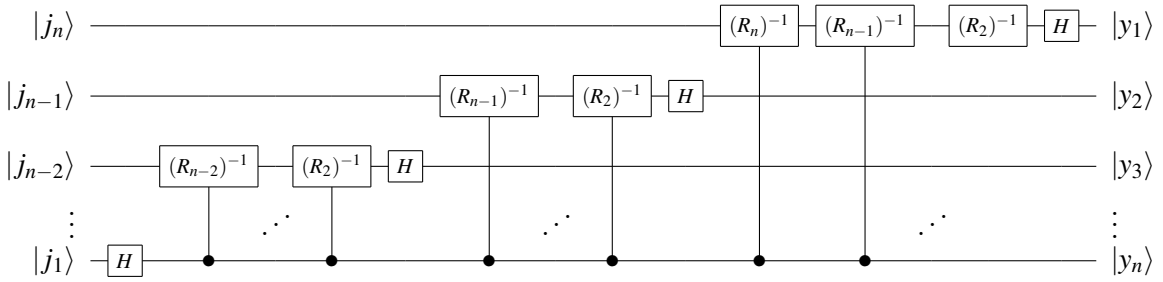
随后，每次让首位通过受 j_k 控的 R_k 门，相位都将增加 $\frac{j_k\pi}{2^{k-1}}$，最终显然会得到

$$\frac{1}{\sqrt{2}}\left(|0\rangle+\exp(2\pi \mathrm{i}\overline{0.j_1\cdots j_n})|1\rangle\right)|j_2,\cdots,j_n\rangle \tag{33}$$

依此类推，每一位需要的状态都可以这样进行制备，这样就得到了量子电路[4]



于是，这样就可以完成量子傅里叶变换，将两个乘数的傅里叶变换点值相乘，就得到了积的多项式点值，进而可以利用量子运算可逆的性质方便地进行量子傅里叶逆变换，只需要将刚刚的操作完全反转即可，其电路为



在此不再列出其电路。在每一个过程中，用到了 $\frac{n(n+1)}{2}$ 个量子门，时间复杂度为 $O(n^2)=O(\log^2 N)$，算法效率较经典计算机中的快速傅里叶变换有了显著的提升。利用 IBM 的 Python 宏包 Qiskit 可以在 Jupyter Notebook 中完成这样的量子乘法器，笔者的代码[5]在 GitHub 开源。

3　思考与展望

在过去的一段时间里，我对量子计算这一领域进行了初步的了解与学习。从一开始，对量子的叠加性特点与对量子比特进行数学计算方法的困惑，后来，通过检索资料、求助同学等方式，我逐渐理解了量子计算的原理，并通过与经典计算机的对比理解了量子计算的诸多特点；下面简要概括我对此的一些思考。

在深入学习之前，我一直不理解的一个问题是：如果说量子计算可以直接由矩阵与向量等传统数学操作完成，那么它和经典计算机中直接进行数学运算的区别在哪里？为什么可以带来效率的提升？后来，通过学习节 1.4 中用 Deustch-Jozsa 算法，我形成了较为直观的理解：考虑一个 n 量子比特门，对于一个酉变换，它能在 $O(n)$ 的时间内完成量子比特的变换，表面上看仅仅是改变了 n 个比特位上的概率

[4] 参考了 Michael Charemza 的 LaTeX 代码 [15]。

[5] 参考了 qiskit-tutorials 中的实现代码 [16]。

分布，实际上却同时改变了关于一组 2^n 个标准正交基的线性组合。这就意味着，如果初态与变换选取得当，量子计算就可以用 $O(n)$ 的时间完成 $O(2^n)$ 的工作。这样的案例，也即刚刚提到的 Deustch-Jozsa 算法，已经在前文中展示了。

在了解了简单的几种量子算法后，我产生了进一步的好奇：目前有哪些类型的量子算法，它们能解决哪些经典计算机科学中的问题？ *Quantum Computation and Quantum Information*[1] 中对现有的量子算法进行了这样的分类：基于傅里叶变换的量子算法，例如 Shor 的素因子分解和离散对数算法；量子搜索算法，例如大名鼎鼎的 Grover 搜索；以及量子模拟算法。此前一段时间里我主要学习的是第一类量子算法。在尝试学习后面两种算法时，遇到的瓶颈在于对复向量空间的变换还没有建立起一个很好的几何直观，因此对例如 Grover 搜索中的 Amplitude Amplification 与迭代操作并没有理解得很透彻，希望在本学期结束后继续这一领域的学习。

展望量子计算领域的未来发展，这一结合物理原理、数学知识与计算机科学思想的计算方法，能带来跨时代意义的进步，具有相当光明的前景。不过在工程领域中，业界许多人认为该领域的前景具有很强的不确定性；很大程度上这是因为量子计算机的物理实现进展较为缓慢，理论领域的许多算法在工程上依旧无法进行实现。为了实现量子计算机，人们也提出了多种方案，目前较为热门的有超导量子计算、量子点量子计算等。回到理论领域，尽管现在已经有许多高效的量子算法，但它们的应用范围还是相对狭窄，学者们一直在致力于拓展量子算法的应用范围，并降低其不确定性；目前，对于很多量子计算相关问题的研究还非常有限 [17]，例如量子计算机多项式时间能否解决经典计算机在多项式时间所不能求解的问题, 即 BQP 是否等于 BPP。希望在不远的未来，量子计算领域的相关工作能取得进一步进展，人们对于量子计算能力的认识能变得更加全面。

参考文献

[1] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002. 1, 1.3, 3

[2] Alan Mathison Turing et al. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936. (document)

[3] Gordon E Moore et al. Cramming more components onto integrated circuits, 1965. (document)

[4] Wallace Witkowski. "moore's law's dead," nvidia ceo jensen huang says in justifying gaming-card price hike, 9 2022. (document)

[5] Suhas Kumar. Fundamental limits to moore's law. *arXiv preprint arXiv:1511.05956*, 2015. (document)

[6] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of statistical physics*, 22(5):563–591, 1980. (document)

[7] Richard P Feynman. Simulating physics with computers, 1981. *International Journal of Theoretical Physics*, 21(6/7), 1981. (document)

[8] Richard P Feynman. Quantum mechanical computers. *Found. Phys.*, 16(6):507–532, 1986. (document)

[9] David Deutsch. Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985. (document)

[10] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994. (document)

[11] David Mermin. Breaking rsa encryption with a quantum computer: Shor's factoring algorithm. *Lecture notes on Quantum computation*, pages 481–681, 2006. (document)

[12] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996. (document)

[13] John Von Neumann. *Mathematical foundations of quantum mechanics: New edition*. Princeton university press, 2018. 1.2

[14] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965. 2.1

[15] Michal Charemza. Examples of quantum circuit diagrams. *Warwick University, Apr*, 2006. 4

[16] Qiskit: An open-source framework for quantum computing, 2021. 5

[17] 孙晓明. 量子计算若干前沿问题综述. 中国科学: 信息科学, 46(8):982–1002, 2016. 3

Quantum Computing: Introduction, Application and Prospect

Conless Pan

2022, Zhiyuan College, Shanghai Jiao Tong University, Shanghai 201100

Abstract: Quantum computing is a new computing paradigm based on the theory of quantum mechanics. Compared to the computing model of classical computers, quantum computing can bring significant efficiency improvements in dealing with specific problems. In the course The Great Ideas of Computer Science (CS1950@SJTU), I learned basic knowledge of quantum computing, getting curious and interested in this field, and read quantum computing related materials and books after class, as well as studying lecture notes from the course Quantum Computing (CS294-2@UCBerkeley). This has provided me with a preliminary understanding of the development, theory, and applications of quantum computing. This article provides a popular science overview of the basic theory of current quantum computing, introduces some simple quantum algorithms and applications, and looks forward to the prospects of quantum computing. All source code is open-sourced on GitHub.

Key words: Quantum Computing; Quantum Algorithm