

# 科学计算中的量子算法：线性方程组的量子算法

安冬

北京大学北京国际数学研究中心 (BICMR)

*andong@bicmr.pku.edu.cn*

25-26 学年第 2 学期

# 大纲

- ▶ HHL
- ▶ LCU
- ▶ QSVT

## 量子线性方程组问题

经典:  $A$  是一个  $N$  乘  $N$  的厄米矩阵,  $b$  是一个  $N$  维向量, 求

$$x = A^{-1}b$$

▶ 非厄米情况: 考虑  $\begin{pmatrix} 0 & A \\ A^\dagger & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$

量子: 制备一个量子态, 以不超过  $\epsilon$  的误差逼近

$$|x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}$$

▶ 假设  $\|A\| = 1$ , 并且我们已知  $A$  的  $(1, a, 0)$ -block-encoding 和  $b$  的态制备 oracle

参数: 误差  $\epsilon$ , 条件数  $\kappa = \|A\|\|A^{-1}\|$

# Harrow-Hassidim-Lloyd (HHL)<sup>1</sup>

**HHL:** 首个量子线性方程组算法

**核心思路:** 记  $A$  的特征值和特征向量为  $(\lambda_j, |v_j\rangle)$ , 右端项  $|b\rangle = \sum_{j=0}^{N-1} \beta_j |v_j\rangle$

$$A^{-1} |b\rangle = \left( \sum_{j=0}^{N-1} \lambda_j^{-1} |v_j\rangle \langle v_j| \right) \left( \sum_{j=0}^{N-1} \beta_j |v_j\rangle \right) = \sum_{j=0}^{N-1} \frac{\beta_j}{\lambda_j} |v_j\rangle$$

**核心步骤:**

- ▶ “并行” 计算  $\lambda_j$  的信息
- ▶ “并行” 在  $|v_j\rangle$  前乘以  $\lambda_j^{-1}$

---

<sup>1</sup>Harrow-Hassidim-Lloyd [arXiv:0811.3171]

**核心步骤:**

- ▶ “并行” 计算  $\lambda_j$  的信息
- ▶ “并行” 在  $|v_j\rangle$  前乘以  $\lambda_j^{-1}$

**子程序:**

- ▶ 哈密顿量模拟:  $U = e^{iA}$
- ▶ 量子相位估计 (QPE):

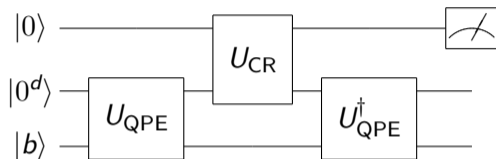
$$U_{\text{QPE}} |0\rangle |v_j\rangle = |\lambda_j\rangle |v_j\rangle$$

(暂时假设  $A$  的特征值有  $d$  位二进制表示)

- ▶ 控制旋转:

$$U_{\text{CR}} |0\rangle |\theta\rangle = \left( \frac{C}{\theta} |0\rangle + \sqrt{1 - |C/\theta|^2} |1\rangle \right) |\theta\rangle$$

# HHL 算法



- ▶ 测量前输出:

$$C|0\rangle|0^d\rangle A^{-1}|b\rangle + |\perp\rangle$$

- ▶  $C \sim 1/\kappa$

## HHL 误差分析

误差来源：哈密顿量模拟，QPE

$$U_{\text{QPE}} |0\rangle |v_j\rangle = |\tilde{\lambda}_j\rangle |v_j\rangle, \quad |\tilde{\lambda}_j - \lambda_j| \leq \epsilon' |\lambda_j|$$

$$U_{\text{HHL}} |0\rangle |b\rangle = |0\rangle \sum \frac{C\beta_j}{\tilde{\lambda}_j} |v_j\rangle + |\perp\rangle$$

记  $\tilde{x} := \sum \frac{C\beta_j}{\tilde{\lambda}_j} |v_j\rangle$ ,  $x := \sum \frac{C\beta_j}{\lambda_j} |v_j\rangle = CA^{-1} |b\rangle$ , 并记误差  $\tilde{\lambda}_j = \lambda_j(1 + e_j)$ ,

$$\|\tilde{x} - x\| = C \left\| \sum \beta_j \left( \frac{1}{\tilde{\lambda}_j} - \frac{1}{\lambda_j} \right) |v_j\rangle \right\| = C \left\| \sum \frac{\beta_j}{\lambda_j} \frac{-e_j}{1 + e_j} |v_j\rangle \right\| \leq 2\epsilon' \|x\|$$

$$\| |\tilde{x}\rangle - |x\rangle \| \leq \frac{2\|\tilde{x} - x\|}{\|x\|} \leq 4\epsilon'$$

可以取  $\epsilon' = \epsilon/4$ , 但注意  $\epsilon'$  是 QPE 中的相对误差, 因此绝对误差可以取为  $\mathcal{O}(\epsilon/\kappa)$

## HHL 复杂度分析

单次 HHL 运算：需要 QPE 达到  $\mathcal{O}(\epsilon/\kappa)$  的精度

- ▶ QPE 中关于  $U = e^{iA}$  的访问复杂度： $\mathcal{O}(\kappa/\epsilon)$
- ▶ 总的关于  $A$  的访问复杂度： $\mathcal{O}((\kappa/\epsilon) \log(\kappa/\epsilon)) = \tilde{\mathcal{O}}(\kappa/\epsilon)$

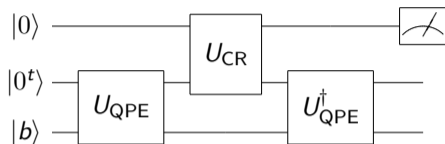
重复次数：注意到  $U_{\text{HHL}} |0\rangle |b\rangle = |0\rangle \|\tilde{x}\| |\tilde{x}\rangle + |\perp\rangle$ ,

$$p_0 = \|\tilde{x}\|^2 \geq (\|x\| - \|\tilde{x} - x\|)^2 \geq \|x\|^2 (1 - 2\epsilon')^2 \geq \frac{1}{4} C^2 \|A^{-1} |b\rangle\|^2 \sim \frac{\|A^{-1} |b\rangle\|^2}{\kappa^2} \geq \frac{1}{\kappa^2}$$

总复杂度：

	$A$ 的访问复杂度	$ b\rangle$ 的访问复杂度	线路深度
无振幅放大	$\tilde{\mathcal{O}}(\kappa^3/\epsilon)$	$\mathcal{O}(\kappa^2)$	$\tilde{\mathcal{O}}(\kappa/\epsilon)$
有振幅放大	$\tilde{\mathcal{O}}(\kappa^2/\epsilon)$	$\mathcal{O}(\kappa)$	$\tilde{\mathcal{O}}(\kappa^2/\epsilon)$

## HHL 小结



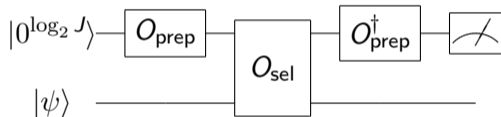
总访问复杂度:  $\tilde{O}(\kappa^2/\epsilon)$

- ▶ 实际复杂度与具体问题有关:  $\tilde{O}(\kappa^2/(\epsilon\|A^{-1}|b\rangle\|))$
- ▶ 与经典算法的比较: 多项式级别更差的  $\kappa$  依赖, 指数级别更差的  $\epsilon$  依赖, 或许指数级别更好的  $N$  的依赖 (取决于输入模型的构造)
- ▶ 输出仍为量子态
- ▶ 下界: 关于  $A$  为  $\Omega(\kappa \log(1/\epsilon))$ , 关于  $|b\rangle$  为  $\Omega(\kappa)$
- ▶ 通过修改控制旋转, 可以求广义逆或更一般的厄米矩阵函数

# LCU 算法

目标：改进线性方程组算法中关于  $\epsilon$  的依赖

思路：考虑函数  $1/x$  的展开，并利用 LCU 实现



$$O_{\text{prep}} : |0\rangle \mapsto \frac{1}{\sqrt{\|\vec{c}\|_1}} \sum_{j=0}^{J-1} \sqrt{c_j} |j\rangle, \quad O_{\text{sel}} = \sum_{j=0}^{J-1} |j\rangle \langle j| \otimes U_j$$

## LCU 算法: Fourier

思路: 将  $1/x$  展开成  $e^{-ixt}$  的线性组合的形式

任取一个奇函数  $f(y)$ , 满足  $\int_0^{\infty} f(y) dy = 1$ , 那么

$$\frac{1}{x} = \int_0^{\infty} f(xy) dy$$

取  $f(y) = ye^{-y^2/2}$ , 并利用 Fourier 变换

$$f(y) = \frac{i}{\sqrt{2\pi}} \int_{-\infty}^{\infty} ze^{-z^2/2} e^{-iyz} dz,$$

我们有

$$\frac{1}{x} = \frac{i}{\sqrt{2\pi}} \int_0^{\infty} dy \int_{-\infty}^{\infty} dz ze^{-z^2/2} e^{-ixyz}$$

## LCU 算法: Fourier

考虑一阶积分离散, 可用 LCU 实现

$$\begin{aligned} A^{-1} &= \frac{i}{\sqrt{2\pi}} \int_0^\infty dy \int_{-\infty}^\infty dz ze^{-z^2/2} e^{-iyzA} \\ &\approx \frac{i}{\sqrt{2\pi}} \int_0^Y dy \int_{-Z}^Z dz ze^{-z^2/2} e^{-iyzA} \\ &\approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \sum_{k=-K}^{K-1} z_k e^{-z_k^2/2} e^{-iy_j z_k A} \Delta y \Delta z \end{aligned}$$

- ▶  $Y, Z$ : 两个正实数
- ▶  $J, K$ : 两个正整数
- ▶  $\Delta y = Y/J, \Delta z = Z/K, y_j = j\Delta y, z_k = k\Delta z$

**LCU 输出:**  $A^{-1}$  的一个  $(\alpha, a, \epsilon')$ -block-encoding

**访问复杂度:** 与  $\alpha, Y, Z$  和逼近误差  $\epsilon'$  有关 (与  $J, K$  的选取无关, 我们选充分大的  $J, K$  使得积分离散的误差充分小)

## LCU 算法: Fourier

$$\alpha = \frac{1}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \sum_{k=-K}^{K-1} |z_k| e^{-z_k^2/2} \Delta y \Delta z \approx \frac{1}{\sqrt{2\pi}} \int_0^Y dy \int_{-Z}^Z dz |z| e^{-z^2/2} = \mathcal{O}(Y)$$

## LCU 算法: Fourier

$$\begin{aligned} & \left| \frac{1}{x} - \frac{i}{\sqrt{2\pi}} \int_0^Y dy \int_{-Z}^Z dz ze^{-z^2/2} e^{-ixyz} \right| = \left| \frac{1}{x} - \frac{1}{\sqrt{2\pi x}} \int_{-Z}^Z dz e^{-z^2/2} (1 - e^{-ixYz}) \right| \\ & \lesssim \frac{1}{|x|} \int_Z^\infty e^{-z^2/2} dz + \frac{1}{|x|} \left| \int_{-\infty}^\infty e^{-z^2/2} e^{-ixYz} dz \right| \\ & \lesssim \frac{1}{|x|} e^{-Z^2/2} + \frac{1}{|x|} e^{-(xY)^2/2} \\ & \lesssim \kappa e^{-Z^2/2} + \kappa e^{-(Y/\kappa)^2/2} \end{aligned}$$

为了使误差小于  $\epsilon'$ , 可取

$$Y = \mathcal{O}(\kappa \sqrt{\log(\kappa/\epsilon')}), \quad Z = \mathcal{O}(\sqrt{\log(\kappa/\epsilon')})$$

## LCU 算法: Fourier

$$A^{-1} \approx \frac{i}{\sqrt{2\pi}} \sum_{j=0}^{J-1} \sum_{k=-K}^{K-1} z_k e^{-z_k^2/2} e^{-iy_j z_k A} \Delta y \Delta z$$

**LCU 输出:**  $A^{-1}$  的一个  $(\alpha, a, \epsilon')$ -block-encoding,  $\alpha = \mathcal{O}(Y) = \mathcal{O}(\kappa \sqrt{\log(\kappa/\epsilon')})$

**构造  $A^{-1}$ :** 需要控制版本的  $e^{-iy_j z_k A} = e^{-i(y_j/Y)(z_k/Z)AYZ}$ , 精度为  $\epsilon'/\kappa$

- ▶ 采用截断 Taylor, 关于  $A$  的访问复杂度为  $\mathcal{O}(YZ \log(\kappa/\epsilon')) = \mathcal{O}(\kappa \log^2(\kappa/\epsilon'))$

## LCU 算法: Fourier

算法: LCU 构造  $A^{-1}$  的 block-encoding, 作用在  $|b\rangle$  上, 测量或振幅放大后测量

$$\frac{1}{\alpha} |0\rangle \|\tilde{x}\| |\tilde{x}\rangle + |\perp\rangle, \quad \tilde{x} \approx A^{-1} |b\rangle, \quad \|\tilde{x}\| \geq 1$$

- ▶ 为了使得  $|A^{-1}b\rangle$  的误差小于  $\epsilon$ , 可取  $\epsilon' \sim \epsilon$

总复杂度:

	$A$ 的访问复杂度	$ b\rangle$ 的访问复杂度	线路深度
无振幅放大	$\tilde{O}(\kappa^3 \log^3(1/\epsilon))$	$\tilde{O}(\kappa^2 \log(1/\epsilon))$	$\tilde{O}(\kappa \log^2(1/\epsilon))$
有振幅放大	$\tilde{O}(\kappa^2 \log^{2.5}(1/\epsilon))$	$\tilde{O}(\kappa \log^{0.5}(1/\epsilon))$	$\tilde{O}(\kappa^2 \log^{2.5}(1/\epsilon))$

- ▶ 结合变时间振幅放大 (variable time amplitude amplification, VTAA), 访问复杂度可改进为几乎最优:

$$\mathcal{O}(\kappa \text{ poly } \log(\kappa/\epsilon))$$

## LCU 算法：多项式展开

考虑  $1/x$  在  $[-1, -1/\kappa] \cup [1/\kappa, 1]$  上的多项式逼近

$$\frac{1}{x} \approx \frac{1 - (1 - x^2)^b}{x}$$

为了使得逼近误差小于  $\epsilon'$ ，需要取

$$b \geq \kappa^2 \log(\kappa/\epsilon')$$

若直接实现  $A^k$  并利用 LCU 实现多项式  $\frac{1 - (1 - A^2)^b}{A}$ ，LCU 中系数的 1 范数  $\sim 2^b$

- ▶ 原因：单项式  $x^k$  不是一组好的基底

改进方向：

- ▶ 好的基底
- ▶ 降低多项式次数

## LCU 算法: Chebyshev

### Chebyshev 多项式:

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x),$$

$$T_k(\cos \theta) = \cos(k\theta)$$

- ▶ 正交性 (带  $\frac{1}{\sqrt{1-x^2}}$  权重)
- ▶ 近似最优的 minimax 逼近

## LCU 算法: Chebyshev

$$\frac{1}{x} \approx \frac{1 - (1 - x^2)^b}{x} = 4 \sum_{j=0}^{b-1} (-1)^j \left( \frac{\sum_{i=j+1}^b \binom{2b}{b+i}}{2^{2b}} \right) T_{2j+1}(x)$$

当  $j$  比较大时,  $T_{2j+1}(x)$  的系数很小, 因此可以进一步截断多项式

$$\frac{1}{x} \approx 4 \sum_{j=0}^{J-1} (-1)^j \left( \frac{\sum_{i=j+1}^b \binom{2b}{b+i}}{2^{2b}} \right) T_{2j+1}(x)$$

$$J = \mathcal{O}(\sqrt{b \log(b/\epsilon')}) = \mathcal{O}(\kappa \log(\kappa/\epsilon'))$$

还可计算系数绝对值之和为  $\mathcal{O}(\kappa \log(\kappa/\epsilon'))$

## LCU 算法: Chebyshev

$$A^{-1} \approx 4 \sum_{j=0}^{\mathcal{O}(\kappa \log(\kappa/\epsilon'))} (-1)^j \left( \frac{\sum_{i=j+1}^b \binom{2b}{b+i}}{2^{2b}} \right) T_{2j+1}(A)$$

- ▶ 实现  $T_{2j+1}(A)$ : 可用 qubitization, 访问复杂度为  $\mathcal{O}(j)$

**求解线性方程组的总访问复杂度:**  $\mathcal{O}(\kappa^2 \text{poly log}(\kappa/\epsilon))$

- ▶ 与 Fourier 情况类似, 也可通过 VTAA 改进为  $\mathcal{O}(\kappa \text{poly log}(\kappa/\epsilon))$

## LCU 算法小结

**思路：**将  $1/x$  展开，并应用 LCU 实现

- ▶ Fourier
- ▶ Chebyshev

**访问复杂度：** $\mathcal{O}(\kappa^2 \text{poly log}(\kappa/\epsilon))$

- ▶ 可用 VTAA 改进为  $\mathcal{O}(\kappa \text{poly log}(\kappa/\epsilon))$

$$Ax = b, \quad |x\rangle = \frac{A^{-1}|b\rangle}{\|A^{-1}|b\rangle\|}, \quad \|A\| = 1$$

**SVD:**

$$A = W\Sigma V^\dagger, \quad A^{-1} = V\Sigma^{-1}W^\dagger = f^\diamond(A^\dagger)$$

$$f(x) = \frac{1}{x}, \quad x \in [1/\kappa, 1]$$

**方法:** 寻找一个奇实多项式  $p(x)$  逼近  $f(x)/(2\kappa)$ , 用 QSVT 实现  $p(A^\dagger)$

## Lemma (反比例函数的多项式近似)

对于任意的  $\epsilon \in (0, 1)$ , 存在次数为  $d = \mathcal{O}(\kappa \log(\kappa/\epsilon))$  的奇实多项式  $p(x)$ , 满足

$$\sup_{x \in [1/\kappa, 1]} |p(x) - 1/x| \leq \epsilon, \quad \sup_{x \in [-1, 1]} |p(x)| \leq \mathcal{O}(\kappa)$$

QSVT 可以实现  $A^{-1}$  的  $(\kappa, a+1, \epsilon)$ -block-encoding, 还需把它作用在  $|0\rangle|b\rangle$  上并测量

- ▶ 最终访问复杂度 (有振幅放大): 关于  $A$ :  $\mathcal{O}(\kappa^2 \log(\kappa/\epsilon))$ ; 关于  $b$ :  $\mathcal{O}(\kappa)$
- ▶ 无需扩大矩阵, 相比 LCU 节省了量子比特

## 推广：矩阵幂函数

### Lemma (负幂函数的多项式近似)

对于任意的  $\delta, \epsilon \in (0, 1/2]$ ,  $c > 0$ , 存在次数为  $d = \mathcal{O}(\frac{\max\{1, c\}}{\delta} \log(1/\epsilon))$  的奇或偶实多项式  $p(x)$ , 满足

$$\sup_{x \in [\delta, 1]} \left| p(x) - \frac{\delta^c}{2} x^{-c} \right| \leq \epsilon, \quad \sup_{x \in [-1, 1]} |p(x)| \leq 1$$

## 推广：矩阵幂函数

### Lemma (正幂函数的多项式近似)

对于任意的正整数  $s$  和  $d$ , 存在一个  $d$  次实多项式  $p(x)$ , 满足

$$\sup_{x \in [-1, 1]} |p(x) - x^s| \leq 2e^{-d^2/(2s)}$$

- ▶  $s$  次单项式可以由  $\mathcal{O}(\sqrt{s \log(1/\epsilon)})$  次实多项式逼近

## 复杂度总结

方法	复杂度	
	$A$	$b$
HHL[arXiv:0811.3171]	$\mathcal{O}(\kappa^2/\epsilon)$	$\mathcal{O}(\kappa)$
LCU[arXiv:1511.02306]	$\mathcal{O}(\kappa^2 \text{ poly log}(\kappa/\epsilon))$	$\mathcal{O}(\kappa)$
QSVT[arXiv:1804.01973]	$\mathcal{O}(\kappa^2 \text{ log}(\kappa/\epsilon))$	$\mathcal{O}(\kappa)$
RM[arXiv:1805.10549]	$\mathcal{O}((\kappa/\epsilon) \text{ log}(\kappa))$	$\mathcal{O}((\kappa/\epsilon) \text{ log}(\kappa))$
AQC(p)[arXiv:1909.05500]	$\mathcal{O}((\kappa/\epsilon) \text{ log}(\kappa/\epsilon))$	$\mathcal{O}((\kappa/\epsilon) \text{ log}(\kappa/\epsilon))$
AQC(exp)[arXiv:1909.05500]	$\mathcal{O}(\kappa \text{ poly log}(\kappa/\epsilon))$	$\mathcal{O}(\kappa \text{ poly log}(\kappa/\epsilon))$
dAQC+EF[arXiv:2111.08152]	$\mathcal{O}(\kappa \text{ log}(1/\epsilon))$	$\mathcal{O}(\kappa \text{ log}(1/\epsilon))$
KR[arXiv:2406.12086]	$\mathcal{O}(\kappa \text{ log}(1/\epsilon))$	$\mathcal{O}(\kappa \text{ log}(1/\epsilon))$
Tunable VTAA[arXiv:2410.18178]	$\mathcal{O}(\kappa \text{ log}(\kappa/\epsilon) \text{ log}(1/\epsilon))$	$\mathcal{O}(\kappa)$
⋮	⋮	⋮
下界	$\Omega(\kappa \text{ log}(1/\epsilon))$	$\Omega(\kappa)$

# 阅读

阅读:

- ▶ LL: Chapter 4.3
- ▶ arXiv: 1511.02306