Unitary tools for non-unitary tasks

Zane Rossi

The University of Tokyo

Unitary tools for non-unitary tasks

or, modern methods in quantum numerical linear algebra

Zane Rossi

The University of Tokyo

Quantum algorithms for classical problems

Quantum algorithms for classical problems

It appears generically difficult to efficiently classically simulate certain quantum mechanical systems

Quantum algorithms for classical problems

It appears generically difficult to efficiently classically simulate certain quantum mechanical systems

Similar space and time savings may be possible for more general dynamics[1]

[1] [SBS26, DCTK26]

Quantum algorithms for classical problems

It appears generically difficult to efficiently classically simulate certain quantum mechanical systems

Similar space and time savings may be possible for more general dynamics[1]

Roadblock and question

How can we practically compile ***non-unitary*** *tasks and* ***non-linear*** *transformations to computing devices built around* ***unitarity*** *and* ***linearity****?*

[1] [SBS26, DCTK26]

Quantum algorithms for classical problems

It appears generically difficult to efficiently classically simulate certain quantum mechanical systems

Similar space and time savings may be possible for more general dynamics[1]

Roadblock and question

*How can we practically compile **non-unitary** tasks and **non-linear** transformations to computing devices built around **unitarity** and **linearity**?*

👉 Linear combinations of unitary transformations ...

[1] [SBS26, DCTK26]

👉 Today

A taxonomy for quantum numerical linear algebra

Walking through recently developed algorithmic families[1]

Ultimate bounds on unitary techniques

Numerical tips & tricks in practice

Applications & open problems

[1]Bits & pieces of LCHS, QET, LCHM, G-QSP, ...

Trends in quantum numerical linear algebra

Growing toolkit for matrix methods, built around block encodings

Trends in quantum numerical linear algebra

Growing toolkit for matrix methods, built around block encodings

$$\|A - \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I)\| \leq \varepsilon \tag{1}$$

Trends in quantum numerical linear algebra

Growing toolkit for matrix methods, built around block encodings

$$\|A - \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I)\| \leq \varepsilon \tag{1}$$

Costs split across initial state preparation, block encoding construction, queries, and amplification

Trends in quantum numerical linear algebra

Growing toolkit for matrix methods, built around block encodings

$$\|A - \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I)\| \leq \varepsilon \tag{1}$$

Costs split across initial state preparation, block encoding construction, queries, and amplification

Maps for singular values & eigenvalues [GSLW19, LS26, MW24], general matrix decompositions [QR22], generalized product formulas [CST+21], randomized techniques [MR25], widely applied to to simulation, linear systems, PCA, statistical inference ... [MRTC21]

Trends in quantum numerical linear algebra

Growing toolkit for matrix methods, built around block encodings

$$\|A - \alpha(\langle 0| \otimes I)U(|0\rangle \otimes I)\| \leq \varepsilon \tag{1}$$

Costs split across initial state preparation, block encoding construction, queries, and amplification

Maps for singular values & eigenvalues [GSLW19, LS26, MW24], general matrix decompositions [QR22], generalized product formulas [CST+21], randomized techniques [MR25], widely applied to to simulation, linear systems, PCA, statistical inference … [MRTC21]

Less addressed: compilation for real instances

⚠️ Limitations of current methods

⚠️ Limitations of current methods

Diverse costs: initial state preparation, block encoding construction, amplification, etc. are all relevant!

⚠️ Limitations of current methods

Diverse costs: initial state preparation, block encoding construction, amplification, etc. are all relevant!

Structural mismatch: common problems aren't always easy to state unitarily

⚠️ Limitations of current methods

Diverse costs: initial state preparation, block encoding construction, amplification, etc. are all relevant!

Structural mismatch: common problems aren't always easy to state unitarily

Small toolkit: we have good tools for spectral maps and linear combinations, but not other common matrix functions

⚠️ Limitations of current methods

Diverse costs: initial state preparation, block encoding construction, amplification, etc. are all relevant!

Structural mismatch: common problems aren't always easy to state unitarily

Small toolkit: we have good tools for spectral maps and linear combinations, but not other common matrix functions

Inflexible compilation: when scheduling these families of algorithms to hardware, our choices are limited

Simulating dynamics

[1]Many of which may have quantum computational analogues!

Simulating dynamics

A good testing ground in dynamical simulation

[1]Many of which may have quantum computational analogues!

Simulating dynamics

A good testing ground in dynamical simulation

Diverse input constraints, experimentally motivated, practical ubiquity

[1] Many of which may have quantum computational analogues!

Simulating dynamics

A good testing ground in dynamical simulation

Diverse input constraints, experimentally motivated, practical ubiquity

Long classical history[1] of numerical methods

[1]Many of which may have quantum computational analogues!

Simulating dynamics

A good testing ground in dynamical simulation

Diverse input constraints, experimentally motivated, practical ubiquity

Long classical history[1] of numerical methods

Interpolates simply from classically easy to quantumly hard regimes

[1] Many of which may have quantum computational analogues!

Simulating dynamics

A good testing ground in dynamical simulation

Diverse input constraints, experimentally motivated, practical ubiquity

Long classical history[1] of numerical methods

Interpolates simply from classically easy to quantumly hard regimes

In parallel, the perfect excuse to communicate clearly and effectively with diverse adjacent subfields …

[1]Many of which may have quantum computational analogues!

Linear combination of Hamiltonian simulation (LCHS)

Linear combination of Hamiltonian simulation (LCHS)

Originally formulated to handle *time-dependent, inhomogeneous, linear* differential equations ... [ALL23, ACL26]

$$\dot{u} = -A(t)u(t) + b(t), \quad u(0) \equiv u_0, \tag{2}$$

Linear combination of Hamiltonian simulation (LCHS)

Originally formulated to handle *time-dependent, inhomogeneous, linear* differential equations … [ALL23, ACL26]

$$\dot{u} = -A(t)u(t) + b(t), \quad u(0) \equiv u_0, \tag{2}$$

which boils down to approximating formal solution

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds, \quad U_s(t) \equiv \mathscr{T}\left\{e^{-\int_s^t A(s')\,ds'}\right\} \tag{3}$$

Linear combination of Hamiltonian simulation (LCHS)

Originally formulated to handle *time-dependent, inhomogeneous, linear* differential equations … [ALL23, ACL26]

$$\dot{u} = -A(t)u(t) + b(t), \quad u(0) \equiv u_0, \tag{2}$$

which boils down to approximating formal solution

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds, \quad U_s(t) \equiv \mathcal{T}\left\{e^{-\int_s^t A(s')\,ds'}\right\} \tag{3}$$

and computing state preparation, block encoding, query, amplification, and discretization costs

Linear combination of Hamiltonian simulation (LCHS)

Originally formulated to handle *time-dependent, inhomogeneous, linear* differential equations … [ALL23, ACL26]

$$\dot{u} = -A(t)u(t) + b(t), \quad u(0) \equiv u_0, \tag{2}$$

which boils down to approximating formal solution

$$u(t) = U_0(t)u_0 + \int_0^t U_s(t)b(s)\,ds, \quad U_s(t) \equiv \mathscr{T}\left\{e^{-\int_s^t A(s')\,ds'}\right\} \tag{3}$$

and computing state preparation, block encoding, query, amplification, and discretization costs

Parameters include t, ε, α, κ, and cost is sensitive to properties of A!

Existing work

Existing work

Given kernel function $\hat{f}$ whose FT decays rapidly, for[1] $A = L + iH$, we can use sums of unitaries to approximate non-unitary transforms [ALL23, ACL26]

[1]For $L \succeq 0$, and some additional conditions on $\hat{f}$

Existing work

Given kernel function $\hat{f}$ whose FT decays rapidly, for[1] $A = L + iH$, we can use sums of unitaries to approximate non-unitary transforms [ALL23, ACL26]

$$\forall x \geq 0, \quad e^{-x} = \int_{\mathbb{R}} \hat{f}(k) e^{-ikx}\, dk \quad \Longrightarrow \tag{4}$$

$$\forall t \geq 0, \quad e^{-(L+iH)t} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) e^{-i(kL+H)t}\, dk. \tag{5}$$

[1]For $L \succeq 0$, and some additional conditions on $\hat{f}$

Existing work

Given kernel function $\hat{f}$ whose FT decays rapidly, for[1] $A = L + iH$, we can use sums of unitaries to approximate non-unitary transforms [ALL23, ACL26]

$$\forall x \geq 0, \quad e^{-x} = \int_{\mathbb{R}} \hat{f}(k) e^{-ikx} \, dk \quad \Longrightarrow \tag{4}$$

$$\forall t \geq 0, \quad e^{-(L+iH)t} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) e^{-i(kL+H)t} \, dk. \tag{5}$$

We truncate at R, and LCU says query cost[2] should be $\mathcal{O}(\alpha R t + \log \varepsilon^{-1})$

[1] For $L \succeq 0$, and some additional conditions on $\hat{f}$

[2] To prepare evolved state, we also need amplification by $\|\hat{f}\|_{L^1}$

Existing work

Given kernel function $\hat{f}$ whose FT decays rapidly, for[1] $A = L + iH$, we can use sums of unitaries to approximate non-unitary transforms [ALL23, ACL26]

$$\forall x \geq 0, \quad e^{-x} = \int_{\mathbb{R}} \hat{f}(k) e^{-ikx}\, dk \quad \Longrightarrow \tag{4}$$

$$\forall t \geq 0, \quad e^{-(L+iH)t} = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k) e^{-i(kL+H)t}\, dk. \tag{5}$$

We truncate at R, and LCU says query cost[2] should be $\mathcal{O}(\alpha R t + \log \varepsilon^{-1})$

Even in homogenous, time-independent case, R and $\|\hat{f}\|_{L^1}$ introduces nonideal polylog ε^{-1} dependence, and expensive quadrature

[1] For $L \succeq 0$, and some additional conditions on $\hat{f}$

[2] To prepare evolved state, we also need amplification by $\|\hat{f}\|_{L^1}$

LCHS: toward an ‘optimal’ approach

LCHS: toward an ‘optimal’ approach

Mathematical observation: contour integrals admit *approximate, operator versions* with *flexible* parameters

LCHS: toward an ‘optimal’ approach

Mathematical observation: contour integrals admit *approximate, operator versions* with *flexible* parameters

Utility: linear combinations of Hermitian evolutions can be related to non-Hermitian evolutions weighted by *fuzzed* FT’d coefficients

LCHS: toward an 'optimal' approach

Mathematical observation: contour integrals admit *approximate, operator versions* with *flexible* parameters

Utility: linear combinations of Hermitian evolutions can be related to non-Hermitian evolutions weighted by *fuzzed* FT'd coefficients

$$\forall x \geq 0, \quad |e^{-x} - f(x)| \leq \varepsilon \quad \Longrightarrow \tag{6}$$

$$\forall t \geq 0, \quad e^{-(L+iH)t} \approx \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k)\, e^{-i(kL+H)t}\, dk. \tag{7}$$

LCHS: toward an 'optimal' approach

Mathematical observation: contour integrals admit *approximate, operator versions* with *flexible* parameters

Utility: linear combinations of Hermitian evolutions can be related to non-Hermitian evolutions weighted by *fuzzed* FT'd coefficients

$$\forall x \geq 0, \quad |e^{-x} - f(x)| \leq \varepsilon \quad \Longrightarrow \tag{6}$$

$$\forall t \geq 0, \quad e^{-(L+iH)t} \approx \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(k)\, e^{-i(kL+H)t}\, dk. \tag{7}$$

Contributions of [LS25]: a good choice of contour, simple quadrature, and block encoding to greatly reduce costs!

Contour integration

Contour integration

Basic results in complex analysis relate integration over curves in complex plane to the evaluation of functions[1]

[1]E.g., sums of *residues* enclosed by the *contour*.

Contour integration

Basic results in complex analysis relate integration over curves in complex plane to the evaluation of functions[1]

Common practice: take real integral transform over line, and relate to complex integral over some closed contour

[1] E.g., sums of *residues* enclosed by the *contour*.

Contour integration

Basic results in complex analysis relate integration over curves in complex plane to the evaluation of functions[1]

Common practice: take real integral transform over line, and relate to complex integral over some closed contour

As integral is invariant under contour deformation,[2] and we're interested in evaluating at poles, we have huge freedom in both contour and kernel!

[1]E.g., sums of *residues* enclosed by the *contour*.
[2]And analytic changes away from evaluation point!

Contour integration

Basic results in complex analysis relate integration over curves in complex plane to the evaluation of functions[1]

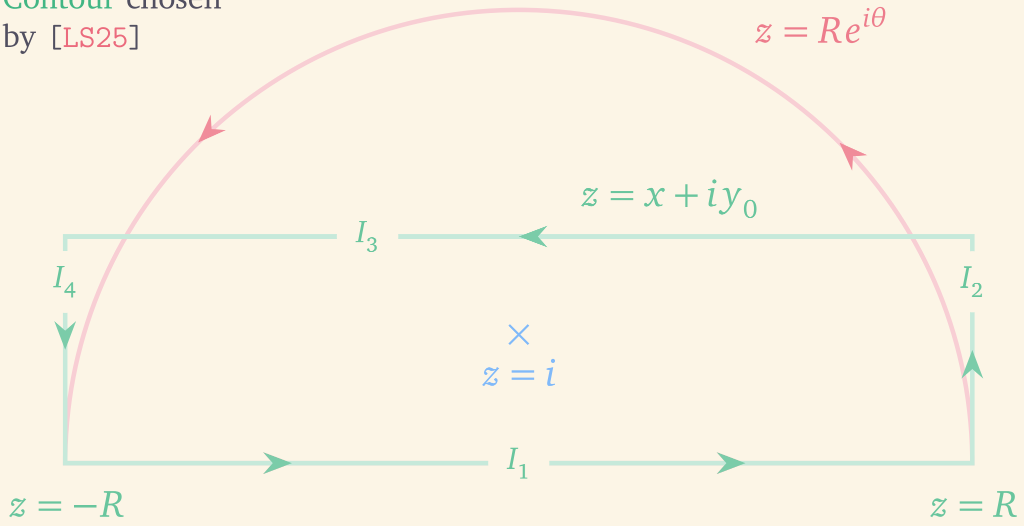
Common practice: take real integral transform over line, and relate to complex integral over some closed contour

As integral is invariant under contour deformation,[2] and we're interested in evaluating at poles, we have huge freedom in both contour and kernel!

Finally, almost all of complex analysis lifts from statements on scalar functions to normed (Banach) spaces — perfect for matrix functions

[1]E.g., sums of *residues* enclosed by the *contour*.
[2]And analytic changes away from evaluation point!

Contour chosen
by [LS25]



Bounding error from the ideal transform

Bounding error from the ideal transform

We relate a finite, real-line integral to a contour integral evaluating desired time-ordered exponential

Bounding error from the ideal transform

We relate a finite, real-line integral to a contour integral evaluating desired time-ordered exponential

$$O_R(t) \equiv \frac{1}{\sqrt{2\pi}} \int_{-R}^{R} \hat{f}(k) U(t;k)\, dk,$$
$$U_0(t) \equiv \frac{1}{\sqrt{2\pi}} \oint_{\Gamma} \hat{f}(z) U(t;z)\, dz = -U(t;i) = \mathcal{T}\left\{ e^{-\int_0^t L(s') + iH(s')\, ds'} \right\}.$$

Bounding error from the ideal transform

We relate a finite, real-line integral to a contour integral evaluating desired time-ordered exponential

Then we argue properties of each piece of the difference

$$U_0(t) = \oint_\Gamma g(z)\,dz = I_1 + I_2 + I_3 + I_4 \tag{8}$$

$$\|O_R(t) - U_0(t)\| \leq \left\| \int_{\mathbb{R}\setminus[-R,R]} g(k)\,dk \right\| + \left\| \int_{\mathbb{R}} g(k+iy_0)\,dk \right\| \tag{9}$$

Bounding error from the ideal transform

We relate a finite, real-line integral to a contour integral evaluating desired time-ordered exponential

Then we argue properties of each piece of the difference

$$U_0(t) = \oint_\Gamma g(z)\, dz = I_1 + I_2 + I_3 + I_4 \tag{8}$$

$$\|O_R(t) - U_0(t)\| \le \left\| \int_{\mathbb{R}\setminus[-R,R]} g(k)\, dk \right\| + \left\| \int_{\mathbb{R}} g(k + iy_0)\, dk \right\| \tag{9}$$

Crucially, I_3 will no longer have to converge to 0 as $y_0 \to \infty$

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}_2(k - i2c\gamma^2; \gamma, c)| \, dk \leq e^{c - c^2\gamma^2} \leq \varepsilon, \tag{10}$$

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}_2(k - i2c\gamma^2; \gamma, c)| \, dk \leq e^{c - c^2\gamma^2} \leq \varepsilon, \tag{10}$$

where $\gamma = \mathcal{O}([1/c] \log^{1/2} [e^c/\varepsilon])$.

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}_2(k - i2c\gamma^2; \gamma, c)| \, dk \leq e^{c - c^2\gamma^2} \leq \varepsilon, \tag{10}$$

where $\gamma = \mathscr{O}([1/c] \log^{1/2} [e^c/\varepsilon])$. Proof follows from the form of $\hat{f}_2$

$$|\hat{f}_2(z; \gamma, c)| = \sqrt{\frac{2}{\pi}} \frac{e^c}{|1 + z^2|} \left| e^{-(z^2+1)/4\gamma^2 - icz} \right|, \tag{11}$$

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}_2(k - i2c\gamma^2; \gamma, c)| \, dk \leq e^{c - c^2\gamma^2} \leq \varepsilon, \tag{10}$$

where $\gamma = \mathcal{O}([1/c] \log^{1/2} [e^c/\varepsilon])$. Proof follows from the form of $\hat{f}_2$

$$|\hat{f}_2(z; \gamma, c)| = \sqrt{\frac{2}{\pi}} \frac{e^c}{|1 + z^2|} \left| e^{-(z^2+1)/4\gamma^2 - icz} \right|, \tag{11}$$

and applying basic inequalities for $z = x + iy_0$ for $y_0 = 2c\gamma^2$ optimally. This, together with $||U(z;t)|| \leq 1$ bounds I_3 from above!

Taking $\hat{f}_2$, for any $c > 0, \gamma \geq 1/\sqrt{c}$, we can show

$$\frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} |\hat{f}_2(k - i2c\gamma^2; \gamma, c)| \, dk \leq e^{c - c^2\gamma^2} \leq \varepsilon, \tag{10}$$

where $\gamma = \mathcal{O}([1/c] \log^{1/2} [e^c/\varepsilon])$. Proof follows from the form of $\hat{f}_2$

$$|\hat{f}_2(z; \gamma, c)| = \sqrt{\frac{2}{\pi}} \frac{e^c}{|1 + z^2|} \left| e^{-(z^2+1)/4\gamma^2 - icz} \right|, \tag{11}$$

and applying basic inequalities for $z = x + iy_0$ for $y_0 = 2c\gamma^2$ optimally. This, together with $\|U(z;t)\| \leq 1$ bounds I_3 from above!

Note $\alpha_{\hat{f}_2} = e^c \mathrm{erfc}(1/2\gamma)$; appearing in integral of $e^{-t^2/\gamma^2}/(1+t^2)\, dt$

Choosing kernels and contours

Choosing kernels and contours

We can choose f to be *entire*[1], e.g., the convolution:

$$f(x) \equiv e^{-|x+c|+c} * e^{-x^2\gamma^2}. \tag{12}$$

[1]This means Fourier coefficients decay superpolynomially.

Choosing kernels and contours

We can choose f to be *entire*[1], e.g., the convolution:

$$f(x) \equiv e^{-|x+c|+c} * e^{-x^2\gamma^2}. \tag{12}$$

I.e., that the FT of our kernel *approximates* exponential decay; more generally we can explore parameterized family:

$$\hat{f}_{j,y}(k;\gamma,c) \equiv \frac{(y+1)^{j-1}}{\sqrt{2\pi}} \frac{e^{c(1-ik)} e^{-(1+k^2)/(4\gamma^2)}}{(1-ik)(y+ik)^{j-1}} \tag{13}$$

For $\forall j \geq 1, y > 0, \gamma > 0, c \in \mathbb{R}$ our choice[2]

[1]This means Fourier coefficients decay superpolynomially.
[2]Usually $y = 1$ is chosen and suppressed.

Choosing kernels and contours

We can choose f to be *entire*[1], e.g., the convolution:

$$f(x) \equiv e^{-|x+c|+c} * e^{-x^2\gamma^2}. \tag{12}$$

I.e., that the FT of our kernel *approximates* exponential decay; more generally we can explore parameterized family:

$$\hat{f}_{j,y}(k;\gamma,c) \equiv \frac{(y+1)^{j-1}}{\sqrt{2\pi}} \frac{e^{c(1-ik)}e^{-(1+k^2)/(4\gamma^2)}}{(1-ik)(y+ik)^{j-1}} \tag{13}$$

For $\forall j \geq 1, y > 0, \gamma > 0, c \in \mathbb{R}$ our choice[2]

Note pole at $k = -i$, and lack of decay over lower half-plane

[1]This means Fourier coefficients decay superpolynomially.
[2]Usually $y = 1$ is chosen and suppressed.

Integral quadrature

[1]Because $U(t;z)$ can be shown to be entire, this result lifts!

Integral quadrature

👉 We have to break our real integral into a sum

[1]Because $U(t;z)$ can be shown to be entire, this result lifts!

Integral quadrature

👉 We have to break our real integral into a sum

Let $A = L + iH$ for L, H finite L_1 norm

[1]Because $U(t; z)$ can be shown to be entire, this result lifts!

Integral quadrature

👉 We have to break our real integral into a sum

Let $A = L + iH$ for L, H finite L_1 norm

Choosing ε, R, γ, and c, one can argue a sufficient *uniform step-size* of $h \le \mathcal{O}([\|L\|_{L^1} + \log(1/\varepsilon')]^{-1})$ allows us to bound error:

[1]Because $U(t;z)$ can be shown to be entire, this result lifts!

Integral quadrature

👉 We have to break our real integral into a sum

Let $A = L + iH$ for L, H finite L_1 norm

Choosing ε, R, γ, and c, one can argue a sufficient *uniform step-size* of $h \leq \mathcal{O}([||L||_{L^1} + \log(1/\varepsilon')]^{-1})$ allows us to bound error:

$$||U_0(t) - I'|| \leq \varepsilon + \varepsilon', \tag{14}$$

where I' is the uniform h-discretization of $O_R(t)$.

[1]Because $U(t;z)$ can be shown to be entire, this result lifts!

Integral quadrature

👉 We have to break our real integral into a sum

Let $A = L + iH$ for L, H finite L_1 norm

Choosing ε, R, γ, and c, one can argue a sufficient *uniform step-size* of $h \le \mathcal{O}([\|L\|_{L^1} + \log(1/\varepsilon')]^{-1})$ allows us to bound error:

$$\|U_0(t) - I'\| \le \varepsilon + \varepsilon', \tag{14}$$

where I' is the uniform h-discretization of $O_R(t)$.

While technical, convergence of this simple discretization is established by [TW14] for functions analytic in finite strip[1]

[1]Because $U(t; z)$ can be shown to be entire, this result lifts!

Optimal LCHS: formal theorem breakdown

[1] Here meaning PSD.

Optimal LCHS: formal theorem breakdown

Thm 1: Let $t \geq 0$, $A = L + iH$ for $\|L\|_{L^1}, \|H\|_{L^1} < \infty$, and[1] $L \succeq 0$. *On strip* assume $\hat{f}(z) \to 0$ for $|z| \to \infty$, and $(z+i)\hat{f}(z)$ analytic with residue $i/\sqrt{2\pi}$. Then finite integral error bound holds!

[1]Here meaning PSD.

Optimal LCHS: formal theorem breakdown

Thm 1: Let $t \geq 0$, $A = L + iH$ for $||L||_{L^1}, ||H||_{L^1} < \infty$, and[1] $L \succeq 0$. *On strip* assume $\hat{f}(z) \to 0$ for $|z| \to \infty$, and $(z+i)\hat{f}(z)$ analytic with residue $i/\sqrt{2\pi}$. Then finite integral error bound holds!

Thm 2: Fix ε and $\hat{f}_{2,1}$ for any $c > 0$, with $\gamma = \mathcal{O}([1/c]\sqrt{c + \log \varepsilon^{-1}})$ and $R = 2c\gamma^2$. Then real integral approximates ideal evolution to precision ε.

[1]Here meaning PSD.

Optimal LCHS: formal theorem breakdown

Thm 1: Let $t \geq 0$, $A = L + iH$ for $\|L\|_{L^1}, \|H\|_{L^1} < \infty$, and[1] $L \succeq 0$. *On strip* assume $\hat{f}(z) \to 0$ for $|z| \to \infty$, and $(z+i)\hat{f}(z)$ analytic with residue $i/\sqrt{2\pi}$. Then finite integral error bound holds!

Thm 2: Fix ε and $\hat{f}_{2,1}$ for any $c > 0$, with $\gamma = \mathcal{O}([1/c]\sqrt{c + \log \varepsilon^{-1}})$ and $R = 2c\gamma^2$. Then real integral approximates ideal evolution to precision ε.

Thms 3 & 4: Fix ε' and parameters above; then choice of h-discretization $h \leq \mathcal{O}([\|L\|_1 + \log(1/\varepsilon')]^{-1})$ bounds error to $\varepsilon + \varepsilon'$.

[1]Here meaning PSD.

Optimal LCHS: formal theorem breakdown

Thm 1: Let $t \geq 0$, $A = L + iH$ for $||L||_{L^1}, ||H||_{L^1} < \infty$, and[1] $L \succeq 0$. *On strip* assume $\hat{f}(z) \to 0$ for $|z| \to \infty$, and $(z+i)\hat{f}(z)$ analytic with residue $i/\sqrt{2\pi}$. Then finite integral error bound holds!

Thm 2: Fix ε and $\hat{f}_{2,1}$ for any $c > 0$, with $\gamma = \mathcal{O}([1/c]\sqrt{c + \log \varepsilon^{-1}})$ and $R = 2c\gamma^2$. Then real integral approximates ideal evolution to precision ε.

Thms 3 & 4: Fix ε' and parameters above; then choice of h-discretization $h \leq \mathcal{O}([||L||_1 + \log(1/\varepsilon')]^{-1})$ bounds error to $\varepsilon + \varepsilon'$.

Moreover, we can block encode $U_0(t)$ to precision ε'' using LCU PREP and SEL with two-qubit gate cost $\mathcal{O}(\log(||L||_{L^1} + \log \varepsilon^{-1}) \log^{5/2} \varepsilon^{-1})$ and $\mathcal{O}(Q \log(||L||_{L^1} + \log \varepsilon^{-1}))$ respectively[2]

[1]Here meaning PSD.

[2]Here meaning more than the full $\pm R$ integration.

On optimality

On optimality

A series of papers on this subject claim successively stronger notions of optimality in simulation and linear systems solving
[ALL23, ACL26, LS25, SBS26]

On optimality

A series of papers on this subject claim successively stronger notions of optimality in simulation and linear systems solving [ALL23, ACL26, LS25, SBS26]

For [LS25], they provide a $\Omega(u\alpha t)$ *query complexity* lower bound[1] for diffeqs,[2] though *multiplicative* lower bound in $\log \varepsilon^{-1}$ is not found

[1] Here $u \equiv |u_0| / |u(t)|$
[2] Basically summarized by imaginary time evolution case

On optimality

A series of papers on this subject claim successively stronger notions of optimality in simulation and linear systems solving [ALL23, ACL26, LS25, SBS26]

For [LS25], they provide a $\Omega(u\alpha t)$ *query complexity* lower bound[1] for diffeqs,[2] though *multiplicative* lower bound in $\log \varepsilon^{-1}$ is not found

Kernel parameters in generic setting can be computed by gradient descent

[1]Here $u \equiv |u_0|/|u(t)|$
[2]Basically summarized by imaginary time evolution case

On optimality

A series of papers on this subject claim successively stronger notions of optimality in simulation and linear systems solving [ALL23, ACL26, LS25, SBS26]

For [LS25], they provide a $\Omega(u\alpha t)$ *query complexity* lower bound[1] for diffeqs,[2] though *multiplicative* lower bound in $\log \varepsilon^{-1}$ is not found

Kernel parameters in generic setting can be computed by gradient descent

⚠️ Statements of optimality are made in time-independent case, while for time-dependent we get logarithmic overhead [LW19]

[1]Here $u \equiv |u_0|/|u(t)|$
[2]Basically summarized by imaginary time evolution case

📌 Sidenote on LCU costs

📌 Sidenote on LCU costs

Beyond time evolution, we have to prepare an approximate block encoding of $A = L + iH$

📌 Sidenote on LCU costs

Beyond time evolution, we have to prepare an approximate block encoding of $A = L + iH$

Both the L^1 norm of $\hat{f}$ and R (which itself depends on kernel!) enters into subnormalization and thus overall complexity

📌 Sidenote on LCU costs

Beyond time evolution, we have to prepare an approximate block encoding of $A = L + iH$

Both the L^1 norm of $\hat{f}$ and R (which itself depends on kernel!) enters into subnormalization and thus overall complexity

The work of [LS25], given uniform quadrature, can also exchange $\log(R/h)$ space for cheaper gate complexity[1] in superpositions of kernel functions

[1]About $R^{3/2} \log(R/\varepsilon)$

📌 Sidenote on LCU costs

Beyond time evolution, we have to prepare an approximate block encoding of $A = L + iH$

Both the L^1 norm of $\hat{f}$ and R (which itself depends on kernel!) enters into subnormalization and thus overall complexity

The work of [LS25], given uniform quadrature, can also exchange $\log(R/h)$ space for cheaper gate complexity[1] in superpositions of kernel functions

Notably: choice of quadrature means $\alpha_{\hat{f},R,h} \approx \alpha_{\hat{f}}$, where the former is the LCU-subnormalization for our discretized integral, and the latter the L^1 norm of our kernel function

[1]About $R^{3/2} \log(R/\varepsilon)$

Limits of simulation methods

Limits of simulation methods

Desiderata in Hamiltonian simulation include polylog$(1/\varepsilon)$ precision dependent, near-linear time-dependence, and good constant factors …

Limits of simulation methods

Desiderata in Hamiltonian simulation include polylog($1/\varepsilon$) precision dependent, near-linear time-dependence, and good constant factors ...

In unstructured case [ZAH26] only coefficient weights and spectral norms matter,[1] and polylog($1/\varepsilon$) isn't permitted

[1]Even here, careful composition of higher order and randomized Trotter

Limits of simulation methods

Desiderata in Hamiltonian simulation include polylog$(1/\varepsilon)$ precision dependent, near-linear time-dependence, and good constant factors ...

In unstructured case [ZAH26] only coefficient weights and spectral norms matter,[1] and polylog$(1/\varepsilon)$ isn't permitted

Commutator [CST$^+$21] versus spectral norm scaling [GSLW19], etc ...

[1] Even here, careful composition of higher order and randomized Trotter

Limits of simulation methods

Desiderata in Hamiltonian simulation include polylog$(1/\varepsilon)$ precision dependent, near-linear time-dependence, and good constant factors …

In unstructured case [ZAH26] only coefficient weights and spectral norms matter,[1] and polylog$(1/\varepsilon)$ isn't permitted

Commutator [CST+21] versus spectral norm scaling [GSLW19], etc …

👉 *Reasonable structure, cheap block encodings, flexible precision, and device characteristics affect method deeply!*

[1]Even here, careful composition of higher order and randomized Trotter

Extensions, insights, and related work

🤔 What's left in quantum simulation?

🤔 What's left in quantum simulation?

Low-energy [KLB+26], symmetry-adapted [BFJ+25], or randomized simulation [ZAH26]

🤔 What's left in quantum simulation?

Low-energy [KLB+26], symmetry-adapted [BFJ+25], or randomized simulation [ZAH26]

End-to-end compilation on specific hardware

🤔 What's left in quantum simulation?

Low-energy [KLB+26], symmetry-adapted [BFJ+25], or randomized simulation [ZAH26]

End-to-end compilation on specific hardware

Methods for generic integral transforms and kernel optimization [WLCL26]

🤔 What's left in quantum simulation?

Low-energy [KLB+26], symmetry-adapted [BFJ+25], or randomized simulation [ZAH26]

End-to-end compilation on specific hardware

Methods for generic integral transforms and kernel optimization [WLCL26]

Clarifying conditions under which lower bounds hold [ZAH26]

🤔 What's left in quantum simulation?

Low-energy [KLB+26], symmetry-adapted [BFJ+25], or randomized simulation [ZAH26]

End-to-end compilation on specific hardware

Methods for generic integral transforms and kernel optimization [WLCL26]

Clarifying conditions under which lower bounds hold [ZAH26]

👉 *Relaxing circuit forms, hybridizing methods, and applying classical numerical tools exposes weakly formulated lower bounds*

Building bridges to higher-order methods

Building bridges to higher-order methods

Everything we've discussed has relied on block encodings

Building bridges to higher-order methods

Everything we've discussed has relied on block encodings

Q1: *What can we do given access to non-unitary dynamics?*

Q2: *What can we do with near-unitary dynamics?*

Q3: *What is the irrep-label equivalent to eigenvalue processing?*

Q4: *Which methods can furnish more fine-grained lower bounds?*

Further reading on new techniques

Further reading on new techniques

Sum-of-squares decomposition techniques for low-energy Hamiltonian simulation [KLB+26], simulating RLC circuit networks [DCTK26], adapted matrix inversion [DLS26]

Further reading on new techniques

Sum-of-squares decomposition techniques for low-energy Hamiltonian simulation [KLB+26], simulating RLC circuit networks [DCTK26], adapted matrix inversion [DLS26]

Each case converts common structure

PSD Hamiltonian in SOS form,
Algebraically constrained differential equation,
Properties of initial state,

to adapted algorithm class and significant improvement

Further reading on new techniques

Sum-of-squares decomposition techniques for low-energy Hamiltonian simulation [KLB+26], simulating RLC circuit networks [DCTK26], adapted matrix inversion [DLS26]

Each case converts common structure

PSD Hamiltonian in SOS form,
Algebraically constrained differential equation,
Properties of initial state,

to adapted algorithm class and significant improvement

Simpler instances [SLBB25] often admit more realistic versions [SBS26]...

🏠 To take home

Simulation lower bounds are over-generic and experimentally un-generous

Adapted simulation methods are still limited: e.g., low-energy, time-independent [KLB+26, ZAH26]

Matrix equations, algebraically constrained dynamics, and operator approximation techniques are still only cursorily applied [DCTK26, SBS26]

👉 *Functional analytic techniques are flexible, well-understood, numerically friendly, and apply to settings well beyond spectral maps*

終

[ACL26] Dong An, Andrew M. Childs, and Lin Lin. Quantum algorithm for linear non-unitary dynamics with near-optimal dependence on all parameters. *Commun. Math. Phys.*, 407(19), 2026.

[ALL23] Dong An, Jin-Peng Liu, and Lin Lin. Linear combination of Hamiltonian simulation for nonunitary dynamics with optimal state preparation cost. *Phys. Rev. Lett.*, 131:150603, Oct 2023.

[BFJ+25] Victor M. Bastidas, Nathan Fitzpatrick, K. J. Joven, Zane M. Rossi, Shariful Islam, Troy Van Voorhis, Isaac L. Chuang, and Yuan Liu. Unification of finite symmetries in the simulation of many-body systems on quantum computers. *Phys. Rev. A*, 111:052433, May 2025.

[CST+21] Andrew M. Childs, Yuan Su, Minh C. Tran, Nathan Wiebe, and Shuchen Zhu. Theory of trotter error with commutator scaling. *Phys. Rev. X*, 11:011020, Feb 2021.

[DCTK26] Arkopal Dutt, Anirban Chowdhury, Kristan Temme, and Hari Krovi. Simulating dynamics of rlc circuits with a quantum differential-algebraic equations solver, 2026.

[DLS26] Alexander M. Dalzell, Jianqiang Li, and Yuan Su. Faster quantum linear system solver beyond the condition number, 2026.

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, 2019.

[KLB+26] Robbie King, Guang Hao Low, Ryan Babbush, Rolando D. Somma, and Nicholas C. Rubin. Quantum simulation with sum-of-squares spectral amplification. *Phys. Rev. Lett.*, 136:110601, Mar 2026.

[LS25] Guang Hao Low and Rolando D. Somma. Optimal quantum simulation of linear non-unitary dynamics. *arXiv preprint, arXiv:2508.19238*, 2025.

[LS26] Guang Hao Low and Yuan Su. Quantum eigenvalue processing. *SIAM Journal on Computing*, 55(1):135–215, 2026.

[LW19] Guang Hao Low and Nathan Wiebe. Hamiltonian simulation in the interaction picture, 2019.

[MR25] John M. Martyn and Patrick Rall. Halving the cost of quantum algorithms with randomization. *npj Quantum Information*, 11(1), March 2025.

[MRTC21] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2(4):040203, 2021.

[MW24] Danial Motlagh and Nathan Wiebe. Generalized quantum signal processing. *PRX Quantum*, 5:020368, Jun 2024.

[QR22] Yihui Quek and Patrick Rebentrost. Fast algorithm for quantum polar decomposition and applications. *Phys. Rev. Res.*, 4:013144, Feb 2022.

[SBS26] Sophia Simon, Dominic W. Berry, and Rolando D. Somma. Efficient quantum algorithm for linear matrix differential equations and applications to open quantum systems. *arXiv preprint, arXiv:2605.16195*, 2026.

[SLBB25] Rolando D. Somma, Guang Hao Low, Dominic W. Berry, and Ryan Babbush. Quantum algorithm for linear matrix equations, 2025.

[TW14] Lloyd N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Review*, 56(3):385–458, 2014.

[WLCL26] Yanqiao Wang, Yixuan Liang, Hongjia Chen, and Jin-Peng Liu. Optimal quantum eigenvalue transformation via linear combinations of hermitian matrices, 2026.

[ZAH26] Alexander Zlokapa, Richard R. Allen, and Aram W. Harrow. Optimal lower bounds for hamiltonian simulation, 2026.