Inverse Nonlinear Fast Fourier Transform: Closing A Chapter in Quantum Signal Processing

Hongkang Ni[1], Rahul Sarkar[2], Lexing Ying[1], Lin Lin[2]

[1]Stanford University
[2]UC Berkeley

January 2026
QIP 2026, Riga

Paper and Code

- Hongkang Ni, Rahul Sarkar, Lexing Ying, and Lin Lin. "Inverse nonlinear fast Fourier transform on SU (2) with applications to quantum signal processing." arXiv preprint arXiv:2505.12615 (2025).

- `https://github.com/qsppack/QSPPACK`

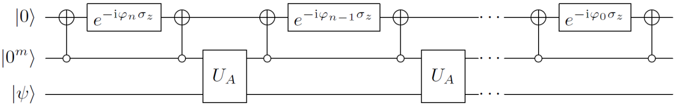
Outline

1. Introduction
2. QSP, GQSP, and NLFT
3. Layer Stripping: Stable or Not?
4. INLFFT: The Fast Algorithm
5. Conclusion

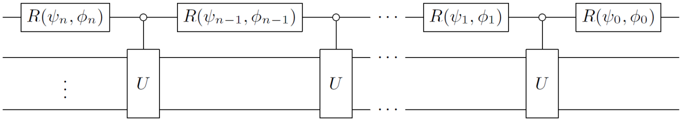
Matrix Function Computation: QSP[1] and GQSP[2]

Numerous applications:

- Hamiltonian simulation
- Linear system solver
- Eigenvalue problems
- Gibbs states preparation
- ...



QSP circuit



GQSP circuit

[1](Low, Chuang, PRL 2017, QIP'17)
[2](Motlagh, Wiebe, PRX Quantum 2024)

QSP Phase Factor Finding Problem

- Qubitization technique: only need to find phase factors for 2×2 matrix!
- Single qubit Pauli matrices

$$\sigma_X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

- For $x = \cos\theta \in [-1, 1]$, rotation matrix

$$W(x) = e^{\mathrm{i}\theta\sigma_X} = \begin{pmatrix} x & \mathrm{i}\sqrt{1-x^2} \\ \mathrm{i}\sqrt{1-x^2} & x \end{pmatrix}$$

Problem (QSP phase factor finding)

Given an even (or odd) real target polynomial $f(x)$ *with* $\|f\|_{L^\infty[-1,1]} \leq 1 - \eta$, *how to find a sequence* $\Phi = (\varphi_0, \ldots, \varphi_n)$ *such that* $f(x) = \mathrm{Im}[P(x)]$, *where* $P(x)$ *is defined by*

$$\begin{pmatrix} P(x) & * \\ * & * \end{pmatrix} = e^{i\varphi_0\sigma_z} W(x) e^{i\varphi_1\sigma_z} W(x) \cdots e^{i\varphi_{n-1}\sigma_z} W(x) e^{i\varphi_n\sigma_z}?$$

GQSP Phase Factor Finding Problem

- Two-parameter single-qubit rotations

$$R(\psi,\phi) := \begin{pmatrix} \cos\psi & e^{i\phi}\sin\psi \\ -e^{-i\phi}\sin\psi & \cos\psi \end{pmatrix} \otimes I$$

- Lift the parity constraint on target polynomial

Problem (GQSP phase factor finding)

Given a target polynomial $b(\cdot)$ of degree n such that $\|b\|_{L^\infty(\mathbb{T})} \le 1$, how to find a sequence of angles

$$\{(\psi_k,\phi_k)\}_{k=0}^{n}$$

such that

$$\begin{pmatrix} * & b(U) \\ * & * \end{pmatrix} = R(\psi_0,\phi_0)\begin{pmatrix} U & \\ & I \end{pmatrix}\cdots\begin{pmatrix} U & \\ & I \end{pmatrix} R(\psi_n,\phi_n)?$$

In the early stages of QSP development..

Toward the first quantum simulation with quantum speedup

Andrew M. Childs[a,b,c,1], Dmitri Maslov[b,c,d], Yunseong Nam[b,c,e], Neil J. Ross[b,c,f], and Yuan Su[a,b,c]

[a]Department of Computer Science, University of Maryland, College Park, MD 20742; [b]Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742; [c]Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742; [d]Division of Computing and Communication Foundations, National Science Foundation, Alexandria, VA 22314; [e]IonQ, Inc., College Park, MD 20740; and [f]Department of Mathematics and Statistics, Dalhousie University, Halifax, NS B3H 4R2, Canada

Edited by John Preskill, California Institute of Technology, Pasadena, CA, and approved August 10, 2018 (received for review January 30, 2018)

Section H.3:

...However, this computation is difficult in practice, so we can only carry it out for very small instances. Specifically, we found the time required to calculate the angles to be prohibitive for values of M greater than about 32...It is a natural open problem to give a more practical method for computing the angles

Key problem 1: Efficiency

- Iterative methods: $\widetilde{\mathcal{O}}(n) \sim \widetilde{\mathcal{O}}(n^2)$. Fails in fully-coherent regime (small η).
- Direct methods:

Algorithm	Time complexity	Space complexity
Riemann-Hilbert[1]	$\widetilde{\mathcal{O}}(n^4 + n\eta^{-1}\log^2(\epsilon^{-1}))$	$\mathcal{O}(n^2)$
Half-Cholesky[2]	$\widetilde{\mathcal{O}}(n^2 + n\eta^{-1}\log^2(\epsilon^{-1}))$	$\mathcal{O}(n)$
Layer stripping[3,4]	$\widetilde{\mathcal{O}}(n^2 + n\eta^{-1}\log^2(\epsilon^{-1}))$	$\mathcal{O}(n)$
Inverse nonlinear FFT	$\widetilde{\mathcal{O}}(n + n\eta^{-1}\log^2(\epsilon^{-1}))$	$\mathcal{O}(n)$

- Easily compute for $n \sim 10^6$

[1](Alexis, Lin, Mnatsakanyan, Thiele, Wang, CPAM 2026, QIP'25)
[2](Ni, Ying, arXiv:2410.06409)
[3](Y.-J. Tsai, Thesis, 2005)
[4](Gilyén, Su, Low, Wiebe, STOC 2019, QIP'19)

Key problem 2: Numerical Stability

What is numerical stability?

- **Intuition:** Small errors should *not* grow uncontrollably as the algorithm proceeds.
- **Practical perspective:** Truncation errors negligible on common devices
  e.g. 52 bits for IEEE double precision
- **Theoretical perspective:** Bit requirement grows only *poly-logarithmically* in problem parameters.
- **In our setting:** Bit requirement scales as

$$r = \mathrm{polylog}\big(n,\ \eta^{-1},\ \epsilon^{-1}\big),$$

Key problem 2: Numerical Stability

Iterative methods:

- Stable when the algorithm converges.

Direct methods:

- Step 1: Complementary polynomial finding:
  - Root finding approach[1]: Unstable
  - Weiss algorithm[2]: Stable
- Step2: Layer stripping: ?

Problem

- *Is it possible to prove the layer stripping part of the direct methods is stable?*
  Yes! (Under certain conditions)
- *Is it possible to develop a more efficient layer stripping type method that preserves stability?*
  Yes!

[1] (Gilyén, Su, Low, Wiebe, STOC 2019, QIP'19), (Haah, Quantum 2019)
[2] (Alexis, Lin, Mnatsakanyan, Thiele, Wang, CPAM 2026, QIP'25)

Outline

1 Introduction

2 QSP, GQSP, and NLFT

3 Layer Stripping: Stable or Not?

4 INLFFT: The Fast Algorithm

5 Conclusion

Nonlinear Fourier Transform (NLFT)

- Given a compactly supported sequence $\gamma = (\gamma_k)_{0\le k\le n}$ of complex numbers, we define its Nonlinear Fourier Transform[1] (NLFT) as

$$\widehat{\gamma} := \begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \prod_{k=0}^{n} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$$

Here, $a^*(z) := \overline{a(\overline{z^{-1}})}$.

- The image of NLFT is

$$\mathcal{S} = \{(a,b) : aa^* + bb^* = 1,\ 0 < a(\infty) < \infty\}.$$

Problem (Inverse NLFT)

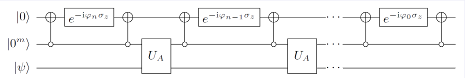
Given a Laurent polynomial $b(z)$ satisfying $\|b\|_{L^\infty(\mathbb{T})} \le 1$, determine a compactly supported sequence γ such that

$$\widehat{\gamma} = \begin{pmatrix} * & b(z) \\ * & * \end{pmatrix}.$$

[1] (Tao, Thiele, arXiv:1201.5129), (Y.-J. Tsai, Thesis, 2005), (Alexis, Mnatsakanyan, Thiele, 2024)

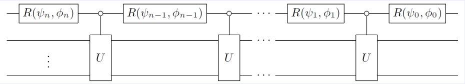
Converting (G)QSP to NLFT[1]

QSP



$$e^{i\varphi_0\sigma_z}W(x)e^{i\varphi_1\sigma_z}W(x)\cdots W(x)e^{i\varphi_n\sigma_z}.$$

GQSP



$$R(\psi_0,\phi_0)\cdot\begin{bmatrix} U & \\ & I \end{bmatrix}\cdot\ldots\cdot\begin{bmatrix} U & \\ & I \end{bmatrix}\cdot R(\psi_n,\phi_n)$$

Nonlinear Fourier Transform

$$\begin{pmatrix} * & b(z) \\ * & * \end{pmatrix} = \prod_{k=0}^{n}\left[\frac{1}{\sqrt{1+|\gamma_k|^2}}\begin{pmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{pmatrix}\right]$$

[1] See details in Section 3 of our paper. See also (Laneve, arXiv:2503.03026)

Outline

1. Introduction
2. QSP, GQSP, and NLFT
3. Layer Stripping: Stable or Not?
4. INLFFT: The Fast Algorithm
5. Conclusion

The Layer Stripping Algorithm

QSP[1,2]

- Given $\mathrm{Im}[P(x)]$
- Find complementary polynomials $\mathrm{Re}[P(x)]$ and $Q(x)$ such that

$$\begin{bmatrix} P(x) & iQ(x)\sqrt{1-x^2} \\ iQ^*(x)\sqrt{1-x^2} & P^*(x) \end{bmatrix}$$

  is unitary for $x \in [-1, 1]$.
- The first phase factor ϕ_0 can be obtained from the leading coefficient of P and Q
- Undo the first rotation and retrieve the following ϕ_k's sequentially using the same method

NLFT[3,4]

- Given $b(z)$
- Find complementary Laurent polynomial $a(z)$ such that

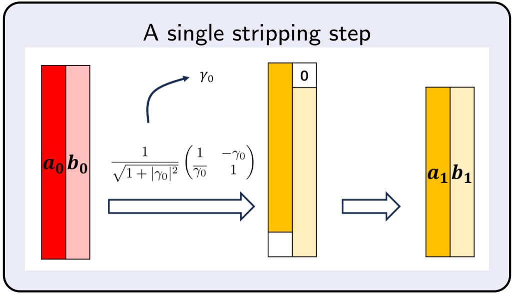
$$\begin{bmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{bmatrix}$$

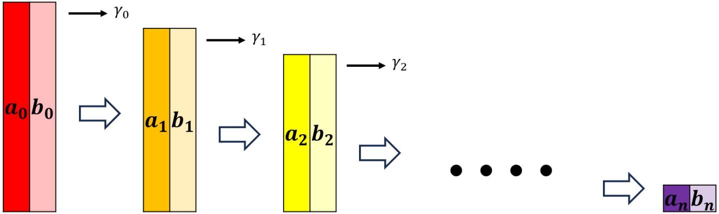
  is unitary for $z \in \mathbb{T}$.
- The first coefficient γ_0 can be obtained from the leading coefficient of a and b
- Undo the first rotation and retrieve the following γ_k's sequentially using the same method

[1] (Gilyén, Su, Low, Wiebe, STOC 2019, QIP'19)
[2] (Haah, Quantum 2019)
[3] (Tao, Thiele, arXiv:1201.5129)
[4] (Alexis, Mnatsakanyan, Thiele, RMC 2024)

Layer Stripping (Matrix Form)

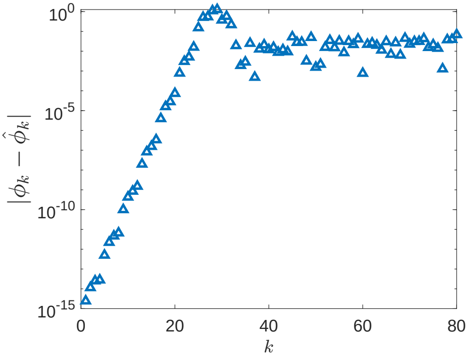
- $\mathbf{a}_0, \mathbf{b}_0$ are the coefficient vectors of $a(z), b(z)$





Instability of layer stripping

- The numerical error may accumulate exponentially during this sequential algorithm.
- Layer stripping works under $\widetilde{\mathcal{O}}(n)$ bits precision floating point arithmetic.[1]
- Instability justified by the following randomly generated example:



Error accumulation of the phase factors for an 80-degree polynomial pair $(a(z), b(z))$

[1](Haah, Quantum 2019)

Stable Layer Stripping Algorithm

- Matrix completion problem: Given $b(z)$, find $a(z)$ to complete $\begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix}$
- Combinatorially many choices[1] of $a(z)$
- Key insight[2]: let $a^*(z) := \overline{a\left(\overline{z^{-1}}\right)}$ be an outer function: roughly, a^* has no zeros in the closed unit disk
- This outer a^* can be constructed efficiently and stably using the Weiss algorithm

Theorem

When $a^*(z)$ is outer, the layer stripping algorithm for the inverse NLFT problem is numerically stable.

---

[1] (Gilyén, Su, Low, Wiebe, STOC 2019, QIP'19), (Haah, Quantum 2019), (Wang, Dong, Lin, Quantum 2022)

[2] (Alexis, Lin, Mnatsakanyan, Thiele, Wang, CPAM 2026, QIP'25)

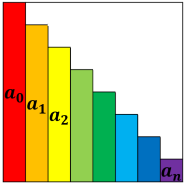
Proof Sketch

- Use the displacement structured matrix theory[1]
- Consider the unique matrix K satisfying

$$K - ZKZ^\dagger = \mathbf{a}_0\mathbf{a}_0^\dagger + \mathbf{b}_0\mathbf{b}_0^\dagger, \qquad Z = \begin{pmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}$$

- K has the triangular factorization

$$K = UU^\dagger, \qquad U =$$



- Doing layer stripping $\approx$ Factorizing matrix K (using Schur's algorithm)
- K is well-conditioned when $a^*(z)$ is outer.

[1](Kailath, Sayed, SIAM Review 1995)

Outline

1 Introduction

2 QSP, GQSP, and NLFT

3 Layer Stripping: Stable or Not?

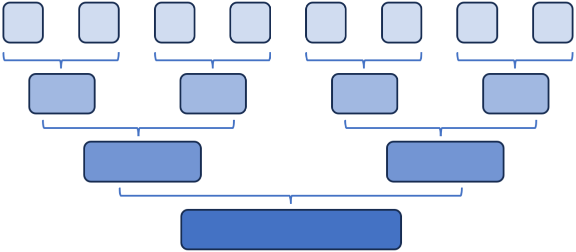
4 INLFFT: The Fast Algorithm

5 Conclusion

Forward NLFT

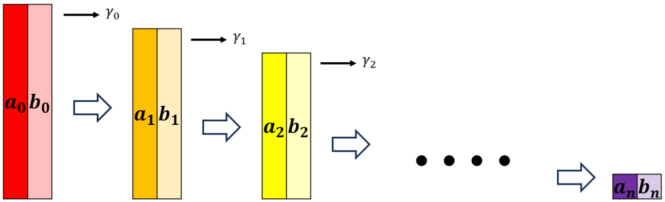
$$\begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \prod_{k=0}^{n} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$$

- Q: How to calculate the polynomial matrix product more efficiently?
- A: Divide and conquer + FFT!

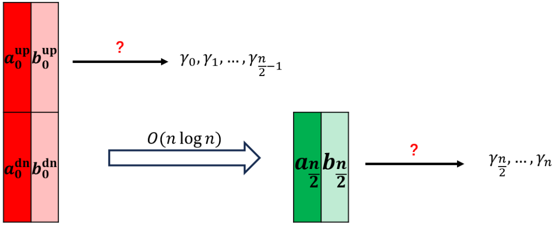


INLFFT

- Layer stripping: $\mathcal{O}(n^2)$



- INLFFT: $\mathcal{O}(n \log^2 n)$



INLFFT

$$\begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \prod_{k=0}^{n} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & \gamma_k z^k \\ -\overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$$

Split the product and take inverse ($m = \lceil \frac{n}{2} \rceil$):

$$\prod_{k=m-1}^{0} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & -\gamma_k z^k \\ \overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right] \begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \begin{pmatrix} a_m(z) & z^m b_m(z) \\ -z^{-m} b_m^*(z) & a_m^*(z) \end{pmatrix}$$

Algorithm Sketch

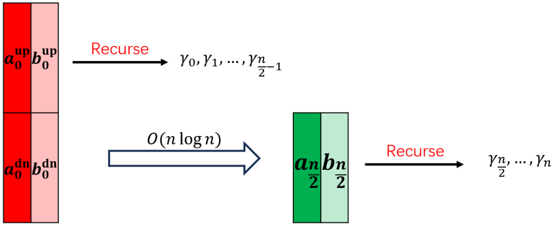
1. *Calculate the first half of the NLFT coefficients* $\gamma_0, \ldots, \gamma_{m-1}$.
2. *Use the divide-and-conquer type fast algorithm to calculate the product* $\prod_{k=m-1}^{0} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & -\gamma_k z^k \\ \overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$.
3. *Construct* (a_m^*, b_m) *by the above formula.*
4. *Retrieve the last half of the NLFT coefficients* $\gamma_m, \ldots, \gamma_n$ *from* (a_m^*, b_m).

INLFFT

Algorithm Sketch

1. *Calculate the first half of the NLFT coefficients* $\gamma_0, \ldots, \gamma_{m-1}$.
2. *Use the divide-and-conquer type fast algorithm to calculate the product* $\prod_{k=m-1}^{0} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & -\gamma_k z^k \\ \overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$.
3. *Construct* (a_m^*, b_m) *by the above formula.*
4. *Retrieve the last half of the NLFT coefficients* $\gamma_m, \ldots, \gamma_n$ *from* (a_m^*, b_m).

- Key insight: $\gamma_0, \ldots, \gamma_{m-1}$ only depends on the first half of $\mathbf{a}_0$ and $\mathbf{b}_0$
- Step 1 and 4 can be done recursively



INLFFT Stability Results

Theorem

When $a^*(z)$ is outer, the INLFFT algorithm is numerically stable.

See Section 5.5 in our paper for proof.

Outline

1. Introduction
2. QSP, GQSP, and NLFT
3. Layer Stripping: Stable or Not?
4. INLFFT: The Fast Algorithm
5. Conclusion

Conclusion

- **Algorithmic Contribution:**
  - Developed INLFFT: A fast, divide-and-conquer implementation of the Inverse NLFT.
  - Achieved near optimal complexity of $\widetilde{\mathcal{O}}(n)$, improving upon the previous $\widetilde{\mathcal{O}}(n^2)$ barrier.
- **Theoretical Contribution:**
  - We resolved the long-standing open question regarding the numerical stability of layer stripping.
  - We further established the stability result of INLFFT.
- **Closing a Chapter in (G)QSP Phase Factoring Finding**
  - We unify QSP and GQSP within the NLFT framework
  - This leads to phase factor finding algorithms that are both efficient and stable.
- **Further Directions:**
  - Multi-variable QSP?
  - NLFT in higher dimensions?

Thank you for your attention!