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

Hongkang Ni[a], Rahul Sarkar[b], Lexing Ying[a], and Lin Lin[b*]

Background: The Quantum Singular Value Transformation (QSVT) [2] which constitutes one of the most important algorithmic developments in the past decade, has a wide range of applications in quantum computation, such as Hamiltonian simulation [3, 4](QIP 2017, QIP 2019), linear system of equations [4–6], eigenvalue problems [7, 8], Gibbs states preparation [4], Petz recovery channel [9], and benchmarking quantum systems [10, 11], to name a few. The key ingredient of QSVT is a polynomial representation called Quantum Signal Processing (QSP) [3], which encodes a target polynomial of definite parity as the $(1,1)$-entry of the product of a sequence of parameterized SU(2) matrices. Given a target polynomial f of degree n, the corresponding parameters, denoted by $\Psi = (\psi_0, \ldots, \psi_n)$, are called the QSP phase factors. Despite significant algorithmic advancements in evaluating phase factors in recent years, the following question remained open:

(1) Is it possible to evaluate QSP phase factors with time complexity scaling linearly or near-linearly in n? If so, is the algorithm numerically stable[1]?

Main contribution: In this work, we provide a positive answer to the above question by developing an efficient and numerically stable algorithm, which we call the inverse nonlinear fast Fourier transform (INLFFT), for computing the phase factors of all functions that admit a QSP representation or its generalized version, GQSP [12]. The time complexity is $\mathcal{O}(n\log^2 n)$, and the space complexity is $\mathcal{O}(n)$. This effectively closes a chapter in quantum signal processing, after a long sequence of works for phase factor evaluation [4, 12–23] (see Table I for a comparison with selected related works).

The divide and conquer structure of the INLFFT algorithm may also be useful in other generalizations of quantum signal processing, such as SU(1,1) [24], SU(N) [25], or multi-variable QSPs [26–28]. The INLFFT algorithm may also find applications beyond quantum computation.

Algorithm	Time complexity	Space complexity	Bit requirement
Riemann-Hilbert [22]	$\mathcal{O}(n^4 + n\eta^{-1}\log^2(n\eta^{-1}\epsilon^{-1}))$	$\mathcal{O}(n^2)$	$\mathcal{O}(\log(n\eta^{-1}\epsilon^{-1}))$
Half-Cholesky [23]	$\mathcal{O}(n^2 + n\eta^{-1}\log^2(n\eta^{-1}\epsilon^{-1}))$	$\mathcal{O}(n)$	$\mathcal{O}(\log(n\eta^{-1}\epsilon^{-1}))$
Layer stripping [29]	$\mathcal{O}(n^2 + n\eta^{-1}\log^2(n\eta^{-1}\epsilon^{-1}))$	$\mathcal{O}(n)$	$\mathcal{O}(\log(n\eta^{-1}\epsilon^{-1}))$
Inverse nonlinear FFT	$\mathcal{O}(n\log^2 n + n\eta^{-1}\log^2(n\eta^{-1}\epsilon^{-1}))$	$\mathcal{O}(n)$	$\mathcal{O}(\log^2 n + \log n \log \eta^{-1} + \log \epsilon^{-1})$

TABLE I. Comparison of QSP phase factor finding methods. Items marked in blue are the contributions of this work.

Problem Setup: In *quantum signal processing* [4, 30], for a phase factor sequence $\Psi := \{\psi_0, \ldots, \psi_n\}$, with each $\psi_k \in [-\pi/2, \pi/2]$, we consider the SU(2) valued map, for $x \in [-1, 1]$:

$$U_n(\Psi, x) = \begin{pmatrix} u_n(\Psi,x) & i\,v_n(\Psi,x) \\ i\,\overline{v_n}(\Psi,x) & \overline{u_n}(\Psi,x) \end{pmatrix} := e^{i\psi_0 Z}\prod_{k=1}^{n}\left(W(x)e^{i\psi_k Z}\right),\ W(x) := \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix},\ Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1)$$

The goal of QSP phase factor finding is the following problem: given $f \in \mathbb{R}[x]$ of degree n and appropriate parity (determined by [4, Theorem 3]), satisfying $\|f\|_\infty := \sup_{x\in[-1,1]} |f(x)| \le 1$, find Ψ such that $\mathrm{Im}(u_n(\Psi, x)) = f(x)$. Based on this, in addition to the open question above, another key question we address in this work is:

(2) What is a set of sufficient conditions that guarantee numerical stability of the algorithm?

Our solution to questions (1)-(2) will rely on the nonlinear Fourier transform (NLFT), which we briefly introduce. The SU(2) *nonlinear Fourier transform* (NLFT), for a given sequence $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_n)$ of complex

[a] Stanford University
[b] University of California, Berkeley
[*] This submission is based on the technical paper [1].
[1] We say that an algorithm is *numerically stable* if the number of bits r needed to implement it scales as $r = \mathcal{O}(\mathrm{polylog}(n, 1/\epsilon))$, where n denotes the problem size, and ϵ is the target accuracy (here accuracy of each phase factor).

numbers, also called the NLFT coefficients, is defined via the ordered product

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

Here $b(z)$ and $a^*(z)$ are polynomials of degree at most n, and we define $a^*(z) := \overline{a(\overline{z}^{-1})}$, and similarly $b^*(z)$. We sometimes abbreviate this map as $\overbrace{\boldsymbol{\gamma}} = (a, b)$. When $z \in \mathbb{T}$ (the unit circle), the image of NLFT is an element of SU(2). The NLFT map is bijective [1, Lemma 2.2], and hence given (a, b), the problem of *inverse* NLFT is to determine $\boldsymbol{\gamma}$ from (a, b).

Connection between QSP(GQSP) and NLFT: The QSP map in (1) and the NLFT map in (2) are closely related, first identified in [21, 22], where such a connection was only established for the symmetric QSP case. In our work, we go beyond the symmetric case and establish such a connection even for the non-symmetric setting. Specifically, if each $\psi_k \in (-\pi/2, \pi/2)$, then with the substitutions $\gamma_k = \tan\psi_k$ and $\cos\theta = x$, one has the following relation [1, Lemma 3.1]:

$$\begin{pmatrix} 1 & \\ & i \end{pmatrix} H U_n(\Psi, \cos\theta) H \begin{pmatrix} 1 & \\ & -i \end{pmatrix} = \overbrace{\boldsymbol{\gamma}}(e^{2i\theta}) \begin{pmatrix} e^{in\theta} & 0 \\ 0 & e^{-in\theta} \end{pmatrix}, \quad H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{3}$$

This allows us to establish a 2-step process to find QSP phase factors [1, Theorem 3.2]. In Step 1, given $f \in \mathbb{R}[x]$ with $\|f\|_\infty < 1$, we construct a SU(2) matrix $\left(\begin{smallmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{smallmatrix}\right)$, where $b(z)$ is defined using $b(e^{2i\theta}) := e^{in\theta} f(\cos\theta)$, and $a^*(z)$ is then determined by the Weiss algorithm [22, Algorithm 2] (QIP 2025) (see also [31]). Then in Step 2, we solve the inverse NLFT problem to determine a sequence $\boldsymbol{\gamma}$ such that $\overbrace{\boldsymbol{\gamma}} = (a, b)$, from which the QSP phase factors can be recovered as $\psi_k = \arctan(\gamma_k)$. We also establish a similar connection between the *generalized quantum signal processing* (GQSP) protocol [12] and NLFT, which analogously yields a process to determine the GQSP phase factors (the reader is referred to [1, Section 3.2, Theorem 3.3] for details). Note that such a connection between GQSP and NLFT also appears in [32].

Main Results: Using the QSP-NLFT connection, we provide positive answers to both questions mentioned above. Since $\|f\|_\infty < 1$ in Step 1 above, it implies $\|b\|_{L^\infty(\mathbb{T})} := \sup_{z\in\mathbb{T}} |b(z)| \le 1 - \eta$, for some $\eta \in (0, 1)$. The Weiss algorithm is already known to be numerically stable under this assumption [22, Section 5.1], and has time complexity $\mathcal{O}(n\eta^{-1}\log^2(n\eta^{-1}\epsilon^{-1}))$ and bit requirement $\mathcal{O}(\log(n\eta^{-1}\epsilon^{-1}))$, where ϵ is the desired accuracy. Thus for fixed ϵ, η, the runtime of Step 1 is $\mathcal{O}(n\log(n))$. For Step 2, with both $a^*(z)$ and $b(z)$ determined from Step 1, the well-known layer stripping algorithm in the literature [29] for computing the inverse NLFT has time complexity $\mathcal{O}(n^2)$, which fails to achieve near-linear time complexity. In our work, we propose a different algorithm, the inverse nonlinear FFT, that computes the inverse NLFT in time complexity $\mathcal{O}(n\log^2(n))$. Using the inverse nonlinear FFT, the combined time complexity of both steps for determining all the QSP phase factors is $\mathcal{O}(n\log^2(n))$, which is linear in n, up to polylog(n) factors.

In the attached manuscript [1, Theorem 5.5, Theorem 5.6], it is shown that both the layer stripping and inverse nonlinear FFT algorithms are numerically stable when two conditions are met: (i) $\|b\|_\infty \le 1 - \eta$ for some $\eta \in (0, 1)$, and (ii) a^* has no zeros in the closed unit disk[2]. In the case of finding the QSP phase factors, condition (i) is already met, and the Weiss algorithm guarantees that the a^* obtained satisfies condition (ii), and moreover, it is the unique such a^* [22, Theorem 4]. Thus, the numerical stability of the complete algorithm for determining the QSP phase factors follows, and this is summarized below in Theorem 1, stated informally. The theorem follows by combining the guarantees of the Weiss algorithm [22, Section 5.1], and [1, Theorem 5.5, Theorem 5.6]. Note that the specific choice of b given f, as mentioned above, and the uniqueness of a^* satisfying condition (ii) ensure that we can unambiguously talk about the QSP phase factor Ψ in the theorem below, irrespective of which algorithm is used in Step 2.

Theorem 1 (QSP-Stability, [1, Alternative version of Theorem 5.5, Theorem 5.6])**.** *Let $f \in \mathbb{R}[x]$ satisfy $\|f\|_\infty \le 1 - \eta$ for some $\eta \in (0, 1)$, and let Ψ be the solution to the QSP phase factor problem. Then one can compute Ψ in a numerically stable manner using the Weiss algorithm, together with either the inverse nonlinear FFT or the layer stripping algorithm. Moreover,*

[2] This technical condition is called the *outerness* condition of a^*.

(a) *Let $\widehat{\Psi}$ be the computed phase factors using the inverse nonlinear FFT algorithm. To achieve an accuracy $\left\|\Psi - \widehat{\Psi}\right\|_\infty < \epsilon$, it suffices to use $\mathcal{O}(\log^2 n + \log n \log \eta^{-1} + \log \epsilon^{-1})$ bits of precision. The time complexity of the total algorithm is $\mathcal{O}(n \log^2(n) + n\eta^{-1} \log^2(n\eta^{-1}\epsilon^{-1}))$.*

(b) *Let $\widehat{\Psi}$ be the computed phase factors using the layer stripping algorithm. To achieve an accuracy $\left\|\Psi - \widehat{\Psi}\right\|_\infty < \epsilon$, it suffices to use $\mathcal{O}(\log(n\eta^{-1}\epsilon^{-1}))$ bits of precision. The time complexity of the total algorithm is $\mathcal{O}(n^2 + n\eta^{-1} \log^2(n\eta^{-1}\epsilon^{-1}))$.*

Complete implementations for finding the QSP factors using both the inverse nonlinear FFT and layer stripping algorithms are available at QSPPACK.

Technical overview of INLFFT: If we split the product on the right-hand side of Equation (2) at index $k = m$ ($m \le n$) and move the left portion to the left-hand side, we obtain

$$\prod_{k=m-1}^{0} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & -\gamma_k z^k \\ \overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right] \begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \begin{pmatrix} a_m(z) & z^m b_m(z) \\ -z^{-m} b_m^*(z) & a_m^*(z) \end{pmatrix}. \tag{4}$$

where a_m^* and b_m are polynomials of degree $n - m$. One key observation is that the first m NLFT coefficients $\gamma_0, \ldots, \gamma_{m-1}$ only depend on the first k coefficients of a^* and b, and the remaining $\gamma_m, \ldots, \gamma_n$ only depend on a_m^* and b_m. This helps us to develop a divide-and-conquer strategy for our INLFFT algorithm.

Assuming we already know the first $m = \lceil \frac{n}{2} \rceil$ NLFT coefficients from the first m coefficients of a^* and b, we may follow the following 3 steps to get the remaining $n + 1 - m$ NLFT coefficients:

1. Use a divide-and-conquer type fast algorithm to calculate the product $\prod_{k=m-1}^{0} \left[\frac{1}{\sqrt{1+|\gamma_k|^2}} \begin{pmatrix} 1 & -\gamma_k z^k \\ \overline{\gamma_k} z^{-k} & 1 \end{pmatrix} \right]$.
2. Construct (a_m^*, b_m) by calculating the matrix product of the RHS of (4), where we use FFT to do the polynomial multiplications.
3. Retrieve the last half of the NLFT coefficients $\gamma_m, \ldots, \gamma_n$ from (a_m^*, b_m).

The important observation is that steps 1 and 3 can be done recursively within the same framework, as they are the same problems of size $\frac{n}{2}$ (assume n is a power of 2 for simplicity). As the recursion is of depth $\log_2(n)$, and the product calculations in steps 1 and 2 cost $\mathcal{O}(n \log n)$ time, we can conclude that the total complexity of the INLFFT algorithm is $\mathcal{O}(n \log^2 n)$. The space complexity of the algorithm remains $\mathcal{O}(n)$, which is the same as the layer stripping algorithm. We refer the reader to [1, Section 4.2] for a detailed explanation of these facts.

As for the layer stripping process [29] to solve the inverse NLFT problem, given the complementary polynomial pair $(a^*(z), b(z))$, we sequentially find the coefficients $\gamma_0, \ldots, \gamma_n$. Starting from (a^*, b), it is easy to find the unique γ_0 such that

$$\frac{1}{\sqrt{1+|\gamma_0|^2}} \begin{pmatrix} 1 & -\gamma_0 \\ \overline{\gamma_0} & 1 \end{pmatrix} \begin{pmatrix} a(z) & b(z) \\ -b^*(z) & a^*(z) \end{pmatrix} = \begin{pmatrix} a_1(z) & z b_1(z) \\ -z^{-1} b_1^*(z) & a_1^*(z) \end{pmatrix}, \tag{5}$$

for some $(n-1)$-degree polynomial pair (a_1^*, b_1). Then one can show that the NLFT coefficient of (a_1^*, b_1) is $(\gamma_1, \ldots, \gamma_n)$. Therefore, what we have done in this step is strip off the first coefficient γ_0. We can repeat this process to strip off γ_1 and get the polynomial pair (a_2^*, b_2), and so on. As a result of iterating this process, we end up constructing the sequence of polynomial pairs (a_k^*, b_k), for $k = 1, \ldots, n$, as well as the NLFT coefficients $\gamma_0, \ldots, \gamma_n$ sequentially.

The numerical stability of the layer stripping algorithm and the INLFFT algorithm is highly nontrivial to prove due to their recursive nature. In fact, we show that for certain complementary polynomial choices, these processes can be numerically unstable [1, Section 5.2], which justifies the analysis in [13]. A central contribution of our work is proving that when the complementary polynomial $a^*(z)$ is obtained via the Weiss algorithm, both algorithms are stable. The stability proofs are established in an indirect but powerful way: by connecting layer stripping to Schur's algorithm for factoring matrices with displacement structure, and then leveraging the stability results of the corresponding Cholesky factorization (see [1, Section 5] for details).

[1] H. Ni, R. Sarkar, L. Ying, and L. Lin, arXiv preprint arXiv:2505.12615 (2025).
[2] A. Gilyén, S. Arunachalam, and N. Wiebe, in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (2019) pp. 1425–1444.
[3] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).
[4] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019) p. 193–204.
[5] L. Lin and Y. Tong, Quantum **4**, 361 (2020).
[6] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, PRX Quantum **2**, 040203 (2021).
[7] L. Lin and Y. Tong, Quantum **4**, 372 (2020).
[8] Y. Dong, L. Lin, and Y. Tong, PRX Quantum **3**, 040305 (2022).
[9] A. Gilyén, S. Lloyd, I. Marvian, Y. Quek, and M. M. Wilde, Phys. Rev. Lett. **128**, 220502 (2022).
[10] Y. Dong and L. Lin, Phys. Rev. A **103**, 062412 (2021).
[11] Y. Dong, K. B. Whaley, and L. Lin, npj Quantum Information **8**, 131 (2022).
[12] D. Motlagh and N. Wiebe, PRX Quantum **5**, 020368 (2024).
[13] J. Haah, Quantum **3**, 190 (2019).
[14] R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy, arXiv preprint arXiv:2003.02831 (2020).
[15] Y. Dong, X. Meng, K. B. Whaley, and L. Lin, Phys. Rev. A **103**, 042419 (2021).
[16] L. Ying, Quantum **6**, 842 (2022).
[17] J. Wang, Y. Dong, and L. Lin, Quantum **6**, 850 (2022).
[18] Y. Dong, L. Lin, H. Ni, and J. Wang, Quantum **8**, 1558 (2024).
[19] Y. Dong, L. Lin, H. Ni, and J. Wang, SIAM Journal on Scientific Computing **46**, A2951 (2024), https://doi.org/10.1137/23M1598192.
[20] B. K. Berntson and C. Sünderhauf, arXiv preprint arXiv:2406.04246 (2024).
[21] M. Alexis, G. Mnatsakanyan, and C. Thiele, Revista Matemática Complutense **37**, 655 (2024).
[22] M. Alexis, L. Lin, G. Mnatsakanyan, C. Thiele, and J. Wang, Commun. Pure Appl. Math. in press (2024).
[23] H. Ni and L. Ying, arXiv preprint arXiv:2410.06409 (2024).
[24] Z. M. Rossi, V. M. Bastidas, W. J. Munro, and I. L. Chuang, arXiv preprint arXiv:2304.14383 (2023).
[25] X. Lu, Y. Liu, and H. Lin, arXiv preprint arXiv:2408.01439 (2024).
[26] Z. M. Rossi and I. L. Chuang, Quantum **6**, 811 (2022).
[27] B. Németh, B. Kövér, B. Kulcsár, R. B. Miklósi, and A. Gilyén, arXiv preprint arXiv:2312.09072 (2023).
[28] N. Gomes, H. Lim, and N. Wiebe, arXiv preprint arXiv:2408.03254 (2024).
[29] Y.-J. Tsai, *SU(2) nonlinear Fourier transform* (University of California, Los Angeles, 2005).
[30] G. H. Low, T. J. Yoder, and I. L. Chuang, Physical Review X **6**, 041067 (2016).
[31] B. K. Berntson and C. Sünderhauf, Communications in Mathematical Physics **406**, 161 (2025).
[32] L. Laneve, arXiv preprint arXiv:2503.03026 (2025).