APPROXIMATE COMPLEX POLYNOMIAL EVALUATION IN NEAR CONSTANT WORK PER POINT*

JOHN H. REIF†

Abstract. Given the n complex coefficients of a degree $n-1$ complex polynomial, we wish to evaluate the polynomial at a large number $m \geq n$ of points on the complex plane. This problem is required by many algebraic computations and so is considered in most basic algorithm texts (e.g., [A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974]). We assume an arithmetic model of computation, where on each step we can execute an arithmetic operation, which is computed exactly. All previous exact algorithms [C. M. Fiduccia, *Proceedings 4th Annual ACM Symposium on Theory of Computing*, 1972, pp. 88–93; H. T. Kung, *Fast Evaluation and Interpolation*, Carnegie-Mellon, 1973; A. B. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numerical Problems*, American Elsevier, 1975; V. Pan, A. Sadikou, E. Landowne, and O. Tiga, *Comput. Math. Appl.*, 25 (1993), pp. 25–30] cost at least work $\Omega(\log^2 n)$ per point, and previously, *there were no known approximation algorithms for complex polynomial evaluation within the unit circle with work bounds better than the fastest known exact algorithms.* There are known approximation algorithms [V. Rokhlin, *J. Complexity*, 4 (1988), pp. 12–32; V. Y. Pan, J. H. Reif, and S. R. Tate, in *Proceedings 32nd Annual IEEE Symposium on Foundations of Computer Science*, 1992, pp. 703–713] for polynomial evaluation at real points, but these do not extend to evaluation at general points on the complex plane.

We provide approximation algorithms for complex polynomial evaluation that cost, in many cases, near constant amortized work per point. Let $k = \log(|P|/\epsilon)$, where $|P|$ is the sum of the moduli of the coefficients of the input polynomial $P(z)$. Let $\tilde{P}(z_j)$ *be an ϵ-approx of* $P(z)$ if ϵ upper bounds the modulus of the error of the approximation $\tilde{P}(z_j)$ at each evaluation point z_j, that is, $|P(z_j) - \tilde{P}(z_j)| \leq \epsilon$; note that ϵ is an absolute error bound rather than a relative error bound. In many applications (particularly in signal processing) the evaluation points z_j are fixed and require only polylogarithmic $k = \log(|P|/\epsilon) = O(\log^{O(1)} n)$; for these cases we get a surprising reduction in work by use of approximation algorithms, as compared to the fastest known exact algorithms.

We ϵ-approx complex degree $n-1$ polynomial evaluation at $m \geq n \log n / \log^2 k$ fixed points on or within the unit disk in the complex plane in amortized work $O(\log^2 k)$ per point, which is $O(\log^2 \log n)$ for polylogarithmic k. If the m points are not fixed, then we have increased amortized work $O(\log^2 k + \log m)$ per point, which is $O(\log m)$ for polylogarithmic k and $m \geq n \log n / \log k$, and is still substantially below the previous bound of $\Omega(\log^2 m)$ for known exact algorithms. We further reduce our amortized bounds for special sets of evaluation points widely used in signal processing applications. The *chirp transform* is equivalent to evaluating a complex degree $n-1$ polynomial at the *chirp points*, which are $\zeta^j, j = 0, \ldots, m-1$, for some fixed complex number ζ. We ϵ-approx complex degree $n-1$ polynomial evaluation at these m chirp points, where $m \geq n \log n / \log^2 k$ and $|\zeta| \leq 1$ in amortized work $O(\log k)$ per point, whereas the previous best bounds for exact evaluation (via the chirp transform) were $\Omega(\log m)$ per point [A. V. Aho, K. Steiglitz, and J. D. Ullman, *SIAM J. Comput.*, 4 (1975), pp. 533–539]. All these results use an interesting reduction to fast multipole algorithms for solving Trummer's problem.

Using instead a reduction to approximate real polynomial evaluation (by interpolation at the Chebyshev points), in total work $O(n \log k)$, we

- ϵ-approx the evaluation of a degree n polynomial at the first n powers of the n'th root of unity, where $n' \geq \Omega(n^2/k)$, and
- ϵ-approx the n-point DFT for certain inputs with descending coefficient magnitude.

All of our results require polylogarithmic (that is, $\log^{O(1)} n$) depth with the same work bounds. We also provide a lower bound for a wide class of schemes for approximate evaluation of a degree $n-1$ polynomial on the unit circle; namely, we prove that if a scheme uses an approximation polynomial of degree $k-1$, then it can be convergent only over a small fraction $O(k/n)$ of the unit circle. We believe this is the first lower bound of this sort proved, and the proof uses an interesting reduction

---

This work was supported by NSF grant NSF-IRI-91-00681 and Army Research Office contract DAAH-04-96-1-0448. A preliminary version of this paper appeared in 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997.

†Department of Computer Science, Duke University, Durham, NC 27708-0129 (reif@cs.duke.edu, http://www.cs.duke.edu/~reif/HomePage.html).

Preprint of paper appearing in SIAM Journal of Computing (SICOMP), Vol. 28 Number 6 pp. 2059-2089, 1999.

to the approximation of a matrix product by a matrix of reduced rank.

Key words. algebraic computation, discrete Fourier transform (DFT), fast Fourier transform (FFT), multipoint polynomial evaluation, complex plane, approximate algorithm

AMS subject classifications. 12Y05, 12-04, 65D15, 65D05, 41A21, 41A10, 68Q25

PII. S0097539797324291

1. Introduction.

The following subsections give an extended description of motivation, statement of the problem, and comparison with prior work.

1.1. Machine model.

For most of this paper, we assume an arithmetic circuit model of sequential computation, where each basic arithmetic or logical operation such as addition, subtraction, multiplication, division, and comparison over the domain of rational numbers can be exactly computed in one step. The floor and ceiling operations are not allowed in this model. (One of our results assumes an *extended arithmetic model*, where given a real θ, then $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ can be computed in $O(1)$ steps. The assumptions made for the extended arithmetic model will be stated during presentation of our algorithms.) We also assume an arithmetic (EREW) PRAM model of parallel computation, where the processors can execute these arithmetic operations in parallel, with exclusive reading and exclusive writing into locations of the shared memory. The computation is *sequential* if there is only one processor. The resource metrics of this model are: *parallel time* (the time to compute the outputs) and *work*, which is the total number of such steps summed over all processors.

1.2. Multipoint polynomial evaluation and interpolation.

The *multipoint polynomial evaluation problem* over ring $\mathcal{D}$ is defined as follows:
Input: n coefficients $p_0, \ldots, p_{n-1} \in \mathcal{D}$ defining a polynomial

$$P(z) = \sum_{j=0}^{n-1} p_j z^j$$

of degree $n-1$ and m evaluation points

$$z_0, \ldots, z_{m-1} \in \mathcal{D}.$$

Output: The values $P(z_0), \ldots, P(z_{m-1})$.

The *multipoint polynomial interpolation problem* over ring $\mathcal{D}$ is defined as follows:
Input: n distinct interpolation points $z_0, \ldots, z_{n-1} \in \mathcal{D}$ and n values $y_0, \ldots, y_{n-1} \in \mathcal{D}$.
Output: coefficients $p_0, \ldots, p_{n-1} \in \mathcal{D}$ defining the unique polynomial

$$P(z) = \sum_{j=0}^{n-1} p_j z^j$$

of degree $n-1$ such that $y_j = P(z_j)$ for $j = 0, \ldots, n-1$.

The multipoint *complex (real)* polynomial evaluation and interpolation problems restrict $\mathcal{D}$ to the complex numbers $\mathcal{C}$ (real numbers $\mathcal{R}$, respectively). Exact algorithms for multipoint polynomial evaluation and interpolation use modular techniques developed in the initial work of Moenck and Borodin [27], Borodin and Munro [4], Horowitz [25], Fiduccia [16], and Kung [26] and were improved by Borodin and Munro [5] to the best known work bounds: $O((n+m)\log^2(n+m))$ for evaluation and $O(n\log^2 n)$ for interpolation. An error analysis for these algorithms is given by Newbery [28]. Pan et al. [31] also gave polynomial evaluation and interpolation algorithms with at least this same work, which in certain cases provide improved numerical stability.

1.3. Multipoint polynomial evaluation at a small number of points. A special case of the multipoint polynomial evaluation problem is where the polynomial $P(z)$ has degree $n-1$, where n is much larger than the number m of evaluation points $z_0, \ldots, z_{m-1}$. This special case has been well known since the early days of algebraic computation and is frequently used in recursive algorithms for multipoint polynomial evaluation.

PROPOSITION 1.1. *The work to evaluate a polynomial $P(z)$ of degree $n-1$ at $m < n$ evaluation points $z_0, \ldots, z_{m-1}$ is $O(n)$ plus $\lceil n/m \rceil$ times the work to evaluate a polynomial of degree $m-1$ at m evaluation points over the same domain.*

Proof. Given $P(z)$ of degree $n-1$, we define $\lceil n/m \rceil$ polynomials

$$P_0(z), \ldots, P_{\lceil n/m \rceil - 1}(z)$$

of degree at most $m-1$, where

$$P(z) = \sum_{j=0}^{\lceil n/m \rceil - 1} P_j(z) z^{jm}.$$

To initialize, we precompute the (trivial) multipoint evaluation z_k^m for each $k = 0, \ldots, m-1$. Then for each $j = 0, \ldots, \lceil n/m \rceil - 1$ we do the multipoint evaluation $P_j(z_k)$ for $k = 0, \ldots, m-1$ and also compute each z_k^{jm} by multiplication of z_k^m times $z_k^{(j-1)m}$. ∎

By application of Proposition 1.1, combined with previous exact algorithms [27, 16, 26, 5] for m point polynomial evaluation for $\lceil n/m \rceil$ polynomials of degree $m-1$, which require work $O(m \log^2 m)$ each, we have the m point polynomial evaluation problem, for $m \leq n$, which can be computed within work

$$\lceil n/m \rceil (m \log^2 m) \leq O(n \log^2 m).$$

Also, the multipoint polynomial evaluation problem, for $m \geq n$, can be solved within work

$$m/n O(n \log^2 n) \leq O(m \log^2 n).$$

Hence we have the following proposition.

PROPOSITION 1.2. *The multipoint polynomial evaluation problem costs work* $\leq O((m+n) \log^2 \min(n, m))$.

1.4. The DFT and generalizations. An nth root of unity ω satisfies $\omega^n = 1$ and $\omega^j \neq 1$ for $1 \leq j < n$. The *nth root of unity over the complex numbers* $\mathcal{C}$ is $\omega = e^{i2\pi/n}$, where $i = \sqrt{-1}$. The *n roots of unity over* $\mathcal{C}$ are $\omega^0, \omega^1, \ldots, \omega^{n-1}$.

The *discrete Fourier transform (DFT) problem* over the complex numbers $\mathcal{C}$ is defined as follows:

Input: coefficients $p_0, \ldots, p_{n-1} \in \mathcal{D}$ defining a polynomial

$$P(z) = \sum_{j=0}^{n-1} p_j z^j$$

of degree $n-1$.

Output: The values

$$P(\omega^0), P(\omega^1), \ldots, P(\omega^{n-1}).$$

The celebrated *fast Fourier transform (FFT)* algorithm of Cooley and Tukey [9] (which, according to Cooley, Lewis, and Welch [8], originates with early work by Runge and König [36] and had early use in the works of Danielson and Lanczos [11] and Good [20]) provides the DFT in work $O(n \log n)$ (also see Gentleman and Sande [17] and Rabiner and Rader [32] for efficient implementations of the DFT).

The DFT^{-1}, which is the inverse problem to the DFT (that is, interpolation from the n roots of unity), reduces to multiplying $\frac{1}{n}$ by the evaluation of a given polynomial at the inverses of each of the n roots of unity. Since for each of the n roots of unity ω^j, the inverse $\omega^{-j} = \omega^{n-j}$ is also one of the n roots of unity (now reverse ordered), the DFT^{-1} can also be solved via the FFT in work $O(n \log n)$. The chirp transform generalizes the FFT to the evaluation of a polynomial of degree $n-1$ over the points ζ^j, for a complex constant ζ. The chirp transform can be computed in work $O(n \log n)$ by a generalization of the FFT algorithm. By a reduction to convolution and thus DFT, the following has also been shown.

PROPOSITION 1.3 (see Aho, Steiglitz, and Ullman [2]). *For fixed complex constants s_0, s_1, ζ, the generalized chirp transform problem of evaluation of a polynomial of degree $n-1$ over the points $s_0 + s_1\zeta^j$ for $j = 0, \ldots, n-1$ (and also the inverse of this problem, that of interpolation from these chirp points) can be solved within work $O(n \log n)$.*

By application of Proposition 1.1, we have the following proposition.

PROPOSITION 1.4. *For fixed complex constants s_0, s_1, ζ, the generalized chirp transform problem of evaluation of a polynomial of degree $n-1$ for over the points $s_0 + s_1\zeta^j$ for $j = 0, \ldots, m-1$ can be solved within work $O((m+n)\log \min(n, m))$.*

Recently, Dutt and Rokhlin, [14] gave an ϵ-approx algorithm for evaluation of a degree $n-1$ polynomial at $n-1$ points on a unit circle with work $O(n \log n + n \log(1/\epsilon))$ and Dutt, Gu, and Rokhlin [12] gave an ϵ-approx algorithm for interpolation of a degree $n-1$ polynomial from $n-1$ Chebyshev points on a unit circle with work $O(n \log(1/\epsilon))$.

1.5. Organization of this paper. The results of this paper are summarized in the abstract. Section 1 provides the standard definitions of the assumed arithmetic model of computation, multipoint polynomial evaluation and interpolation, the DFT, and generalizations to the chirp transform (see also Aho, Steiglitz, and Ullman [2]).

Section 2 gives our *main result*, Theorem 2.7, an approximate complex polynomial evaluation algorithm using a reduction to Trummer's problem which we approximately solve by multipole algorithms. We first define Trummer's problem in subsection 2.1 and describe Multipole methods for approximately solving Trummer's problem in section 2.2. We then give an algorithm for multipoint complex polynomial re-evaluation by reduction to Trummer's problem in section 2.3, and finally in section 2.4 we use this reduction to do approximate complex polynomial evaluation via DFT and multipole algorithms.

Note. Our computational model assumes each arithmetic operation yields exact results; thus, it is not yet known if our algorithm yields the performance given in theory if each arithmetic operation is computed in approximate floating point. However, implementations (see [22, 15]) of fast multipole algorithms using approximate floating point operation do give excellent performance in practice.

Section 3 describes known results for approximate real polynomial evaluation; section 3.1 defines the Chebyshev point evaluation problem and known algorithms; section 3.2 bounds the errors of interpolation at the Chebyshev points; and section 3.3

describes known methods for approximate real polynomial evaluation via interpolation at the Chebyshev points.

Section 4 has a *secondary (less general) result:* it gives approximate polynomial evaluation on a circle in a number of interesting cases, using classical methods of real approximation by interpolation at the Chebyshev points. Subsection 4.1 describes approximate polynomial evaluation on a circle via interpolation at the Chebyshev angles and section 4.2 gives an alternative method which is proved in Appendix A.

Section 5 proves a lower bound for a wide class of schemes for polynomial evaluation on the unit circle; namely, there is no general approximation method, using a low degree polynomial, that is convergent over a large fraction of the unit circle.

Appendix B gives an algorithm for exact Chebyshev point evaluation in $O(n \log n)$ work, which is useful for the approximate real polynomial evaluation results of section 3 (this can be used as an alternative for a somewhat more complex algorithm for Chebyshev point evaluation of Gerasoulis [18]).

2. Approximate complex polynomial evaluation via the multipole methods.

2.1. Trummer's problem. We define the *(generalized) Trummer's problem* over the complex plane $\mathcal{C}$:

Input: n points $a_0, \ldots, a_{n-1} \in \mathcal{C}$, n weights $c_0, \ldots, c_{n-1} \in \mathcal{C}$, defining a rational function $\psi(z) = \sum_{j=0}^{n-1} \frac{c_j}{z-a_j}$, and also $m \geq n$ evaluation points $z_0, \ldots, z_{m-1} \in \mathcal{C}$.

Output: The values $\psi(z_0), \ldots, \psi(z_{m-1})$.

Trummer's problem has widespread application to calculation of electrostatic and gravitational forces. Gerasoulis [18] (also see Gerasoulis, Grigoriadis, and Sun [19]) gave an exact algorithm for Trummer's problem, requiring $O(m \log^2 m)$ work. For the case where $m = n$ and $a_j = z_j$, for $j = 1, \ldots, n$, Gerasoulis defined a polynomial $\sigma(z) = \Pi_{j=0}^{n-1}(z - a_j)$ and its derivative $\sigma^{(1)}(z) = \frac{d\sigma(z)}{dz}$. By *l'Hôpital's* rule (the limit is preserved by taking derivatives), we have the following proposition.

PROPOSITION 2.1.

$$\psi(a_j) = \frac{c_j}{\sigma^{(1)}(a_j)}$$

for $j = 0, \ldots, n-1$

Thus Gerasoulis gave a reduction of the probem of evaluation of $\psi(z)$ to evaluation of $\sigma^{(1)}(z)$ at the n points $a_0, \ldots, a_{n-1}$, which requires $O(n \log^2 n)$ work for the exact solution of Trummer's problem. The result of Gerasoulis also easily extends (e.g., by use of additional weights c_j with value 0) to allow for exact solution of the generalized Trummer's problem, as defined above, in $O(m \log^2 m)$ work.

2.2. Multipole methods. Let $k = \log(|C|/\epsilon)$, where $|C| = \sum_{j=0}^{n-1} |c_j|$ is the sum of the moduli (the modulus of complex number $re^{i\theta}$ is r) of the weights $c_0, \ldots, c_{n-1}$ and ϵ is a given absolute error bound on each output. Greengard and Rokhlin [21] (also see Carrier, Greengard, and Rokhlin [7]) gave an ϵ-approx algorithm, known as the multipole algorithm, for Trummer's problem using an $O(k)$ term rational series expansion of $O(k)$ terms and requiring $O(mk^2)$ work over the complex plane. They computed ϵ-approx values $\widetilde{\psi}(a_0), \ldots, \widetilde{\psi}(a_m)$ such that $|\psi(a_j) - \widetilde{\psi}(a_j)| \leq \epsilon$ for $j = 0, \ldots, m-1$. Given the n points $a_0, \ldots, a_{n-1} \in \mathcal{C}$ for Trummer's problem, rational function $\psi(z)$, and also $m \geq n$ evaluation points $z_0, \ldots, z_{m-1} \in \mathcal{C}$, the multipole algorithm requires using these $m+n$ points to construct a certain tree-like data structure known as the *well-separated decomposition*; see Callahan and Kosaraju [6] for details.

Given these $m+n$ points, the well-separated decomposition algorithm of Callahan and Kosaraju [6] costs work $O(m \log m)$. Also, Pan, Reif, and Tate [30] (see Reif and Tate [33] for the full paper) gave an $O(m \log\log m)$ algorithm for the well-separated decomposition in the case where the input points have logarithmic bit-precision. Assuming a well-separated decomposition, the Multipole algorithm over the complex plane was improved by Greengard and Rokhlin [22] to $O(mk \log k)$ work by use of the FFT, to do each of the $O(m)$ operations on $O(k)$ term power series required by this multipole algorithm, each within work $O(k \log k)$. Later work by Pan, Reif, and Tate [30] (see Reif and Tate [34] for the full paper) gave a further substantial improvement, which remains the most efficient known algorithm for approximate solution of the Trummer's problem.

LEMMA 2.2. *Given a well-separated decomposition of the set of the input points, a Trummer's problem for m evaluation points can be ϵ-approx on every output within work $O(m \log^2 k)$, where $k = \log(|C|/\epsilon)$.*

The algorithm of [30, 34] uses a reduction from Trummer's problem for m evaluation points to ϵ-approx solution of a (slightly generalized) Trummer's problem of size $m' \leq m/k^c$ for a constant c with cost $O(m'k \log k) \leq O(m)$, and also solves $O(m')$ instances of Trummer's problem; each has $m/m' \leq k^c$ evaluation points costing $O((m/m') \log^2 k)$ per such instance, resulting in the total cost

$$O(m'(m/m') \log^2 k) \leq O(m \log^2 k).$$

Note. In the special case that the m evaluation points are chirp points, the algorithm of [30, 34] can be sped up as follows: again we compute an ϵ-approx solution of a (slightly generalized) Trummer's problem of size $m' \leq m/k^c$ with cost $O(m'k \log k)$ $\leq O(m)$ and also solve $O(m')$ instances of Trummer's problem with $m/m' \leq k^c$ chirp evaluation points, with the reduced cost $O((m/m') \log k)$ for each such instance, resulting in a somewhat reduced total cost $O(m'(m/m') \log k) \leq O(m \log k)$.

2.3. Reduction to Trummer's problem. The *multipoint polynomial re-evaluation problem* over $\mathcal{C}$ is defined as follows:

Input: n distinct points $a_0, \ldots, a_{n-1} \in \mathcal{C}$ and n values $y_0, \ldots, y_{n-1} \in \mathcal{C}$ defining a unique polynomial $P(z)$ of degree $n-1$ such that $y_j = P(a_j)$ for $j = 0, \ldots, n-1$, and m re-evaluation points $z_0, \ldots, z_{m-1} \in \mathcal{C}$, which are distinct from these points $a_0, \ldots, a_{n-1}$.

Output: The values $P(z_0), \ldots, P(z_{m-1})$.

Note that in the polynomial re-evaluation problem, the coefficients of the input polynomial are not explicitly given.

Given as input n distinct points $a_0, \ldots, a_{n-1} \in \mathcal{C}$, and values $y_0, \ldots, y_{n-1}$, let $\sigma(z) = \Pi_{j=0}^{n-1}(z - a_j)$. Our basic approach is as follows. We construct the rational function $\psi(z) = P(z)/\sigma(z)$, which can be expanded as a Trummer's function $\psi(z)$ $= \sum_{j=0}^{n-1} \frac{c_j}{z-a_j}$. The weights $c_0, \ldots, c_{n-1} \in \mathcal{C}$ are determined by reversing the formula of Proposition 2.1, as follows in this proposition.

PROPOSITION 2.3. *Suppose we set $c_j = y_j/\sigma^{(1)}(a_j)$ for $j = 0, \ldots, n-1$, where $\sigma^{(1)}(z) = \frac{d\sigma(z)}{dz}$. Then $P(z) = \psi(z)\sigma(z)$ for all z, where $\psi(z) = \sum_{j=0}^{n-1} \frac{c_j}{z-a_j}$, and $P(z)$ is the unique degree $n-1$ polynomial such that $P(a_0) = y_0, \ldots, P(a_{n-1}) = y_{n-1}$.*

Proof. If $P(z) = \psi(z)\sigma(z)$, then for $j = 0, \ldots, n-1$,

$$c_j = \lim_{z \to a_j} (z - a_j)\psi(z) = \lim_{z \to a_j} (z - a_j)(P(z)/\sigma(z))$$

$$= P(a_j)/\sigma^{(1)}(a_j) = y_j/\sigma^{(1)}(a_j). \qquad \square$$

We will exactly compute the values $\sigma^{(1)}(a_j)$, $j = 0, \ldots, n-1$. We will compute or approximate (as in certain specialized cases described below) the values $\sigma(z_j)$, $j = 0, \ldots, m-1$. Next we apply the efficient multipole method of [30, 34] (Lemma 2.2) to construct an ϵ-approx solution $\widetilde{\psi}(z_0), \ldots, \widetilde{\psi}(z_{m-1})$ of this Trummer's problem, such that $|\psi(z_j) - \widetilde{\psi}(z_j)| \leq \epsilon$ for $j = 0, \ldots, m-1$. If we exactly compute $\sigma(z_j)$, then we approximate each $P(z_j)$ by $\widetilde{P}(z_j) = \widetilde{\psi}(z_j)\sigma(z_j)$. Otherwise, we approximate each $P(z_j)$ by $\widetilde{P}(z_j) = \widetilde{\psi}(z_j)\widetilde{\sigma}(z_j)$, where $\widetilde{\sigma}(z_j)$ is an ϵ^*-approx to $\sigma(z_j)$ (as determined below).

PROPOSITION 2.4. *For $j = 0, \ldots, m-1$, the $\widetilde{P}(z_j)$ are an $\widehat{\epsilon}$-approx to the $P(z_j)$, where if the $\sigma(z_j)$ are exactly computed, then*

$$\widehat{\epsilon} \leq \epsilon \max_j |\sigma(z_j)|,$$

and if each $\sigma(z_j)$ is ϵ^*-approx by $\widetilde{\sigma}(z_j)$, then

$$\widehat{\epsilon} \leq (\epsilon^* + \max_j |\sigma(z_j)|)\epsilon + (2 + 3\epsilon^*/\min_j |\sigma(z_j)|)|P|,$$

where $|P| = \sum_{j=0}^{n-1} |p_j|$.

Proof. First suppose the $\sigma(z_j)$ are exactly computed, so $\epsilon^* = 0$. By definition, $P(z_j) = \psi(z_j)\sigma(z_j)$ and $\widetilde{P}(z_j) = \widetilde{\psi}(z_j)\sigma(z_j)$, so we have

$$\begin{aligned} |P(z_j) - \widetilde{P}(z_j)| &= |\psi(z_j)\sigma(z_j) - \widetilde{\psi}(z_j)\sigma(z_j)| \\ &= |\psi(z_j) - \widetilde{\psi}(z_j)||\sigma(z_j)| \leq \epsilon|\sigma(z_j)|. \end{aligned}$$

Otherwise, suppose each $\sigma(z_j)$ is ϵ^*-approx by $\widetilde{\sigma}(z_j)$. We have $|\psi(z_j)| \leq |P|/|\sigma(z_j)|$, and $|\widetilde{\psi}(z_j)| \leq \epsilon + |\psi(z_j)| \leq \epsilon + |P|/|\sigma(z_j)|$, so

$$|\widetilde{\psi}(z_j)||\sigma(z_j)| \leq |\sigma(z_j)|\epsilon + |P|,$$

and also

$$|\psi(z_j)| + |\widetilde{\psi}(z_j)| \leq \epsilon + 2|\psi(z_j)| \leq \epsilon + 2|P|/|\sigma(z_j)|.$$

Furthermore, $|\sigma(z_j) - \widetilde{\sigma}(z_j)| \leq \epsilon^*$, so $|\widetilde{\sigma}(z_j)| \leq \epsilon^* + |\sigma(z_j)|$, and

$$|\psi(z_j)\widetilde{\sigma}(z_j)| \leq \epsilon^*|\psi(z_j)| + |P| = \epsilon^*(|P|/|\sigma(z_j)|) + |P|.$$

Also,

$$(|\psi(z_j)| + |\widetilde{\psi}(z_j)|)|\sigma(z_j) - \widetilde{\sigma}(z_j)| \leq \epsilon^*(\epsilon + 2|P|/|\sigma(z_j)|).$$

Note that

$$\begin{aligned} P(z_j) - \widetilde{P}(z_j) &= \psi(z_j)\sigma(z_j) - \widetilde{\psi}(z_j)\widetilde{\sigma}(z_j) \\ &= (\psi(z_j) + \widetilde{\psi}(z_j))(\sigma(z_j) - \widetilde{\sigma}(z_j)) + \psi(z_j)\widetilde{\sigma}(z_j) - \widetilde{\psi}(z_j)\sigma(z_j). \end{aligned}$$

Hence

$$\begin{aligned} |P(z_j) - \widetilde{P}(z_j)| &= |\psi(z_j)\sigma(z_j) - \widetilde{\psi}(z_j)\widetilde{\sigma}(z_j)| \\ &\leq (|\psi(z_j)| + |\widetilde{\psi}(z_j)|)|\sigma(z_j) - \widetilde{\sigma}(z_j)| + |\psi(z_j)\widetilde{\sigma}(z_j)| + |\widetilde{\psi}(z_j)\sigma(z_j)| \\ &\leq \epsilon^*(\epsilon + 2|P|/|\sigma(z_j)|) + (\epsilon^*(|P|/|\sigma(z_j)|) + |P|) + (|\sigma(z_j)|\epsilon + |P|) \\ &= (\epsilon^* + |\sigma(z_j)|)\epsilon + (2 + 3\epsilon^*/|\sigma(z_j)|)|P|. \qquad \square \end{aligned}$$

To approximately compute each of the $\sigma(z_j)$, given z_j (and with fixed a_j) for $j = 0, \ldots, m-1$, we define a *unitary* Trummer's problem of size m where we specialize the y_j to 1 but keep the a_j fixed as before. Hence, since the a_j are fixed, the weights are fixed as $c_j = 1/\sigma^{(1)}(a_j)$, and so can be assumed to be precomputed. This corresponds to defining a constant polynomial $P^*(z) = 1 = \psi^*(z)\sigma(z)$; with this specialization, $\psi^*(z) = 1/\sigma(z)$. Next we can apply again Lemma 2.2 to construct an ϵ-approx solution: $\widetilde{\psi}^*(z_0), \ldots, \widetilde{\psi}^*(z_{m-1})$ of the unitary Trummer's problem. Then we define $\widetilde{\sigma}(z) = 1/\widetilde{\psi}^*(z)$, and use $\widetilde{\sigma}(z_j)$ as an approximation to $\sigma(z_j)$.

PROPOSITION 2.5. *For $j = 0, \ldots, n-1$, each*

$$\widetilde{\sigma}(z_j) = 1/\widetilde{\psi}^*(z_j)$$

is an ϵ^*-approx to $\sigma(z_j)$, where

$$\epsilon^* = \frac{\epsilon|\sigma(z_j)|^2}{1 - |\sigma(z_j)|\epsilon}.$$

Proof. The error is bounded as

$$|\widetilde{\sigma}(z_j) - \sigma(z_j)| = |\frac{1}{\widetilde{\psi}^*(z_j)} - \frac{1}{\psi^*(z_j)}| = |\frac{\widetilde{\psi}^*(z_j) - \psi^*(z_j)}{\psi^*(z_j)\widetilde{\psi}^*(z_j)}| = \frac{|\widetilde{\psi}^*(z_j) - \psi^*(z_j)|}{|\psi^*(z_j)\widetilde{\psi}^*(z_j)|}$$
$$\leq \frac{\epsilon}{|\psi^*(z_j)||\widetilde{\psi}^*(z_j)|} \leq \frac{\epsilon}{|\psi^*(z_j)|(|\psi^*(z_j)| - \epsilon)} = \epsilon^*$$

for

$$\epsilon^* = \frac{\epsilon|\sigma(z_j)|^2}{1 - |\sigma(z_j)|\epsilon},$$

since $\sigma(z) = 1/\psi^*(z)$, and

$$|\widetilde{\psi}^*(z_j)| \geq |\psi^*(z_j)| - \epsilon = \frac{1 - |\sigma(z_j)|\epsilon}{|\sigma(z_j)|}. \qquad \square$$

2.4. Approximate complex polynomial evaluation via DFT and multipole algorithms. Let $\omega = e^{i2\pi/n}$ be the nth root of unity $\in \mathcal{C}$. We assume the n roots of unity ω^j, for $j = 1, \ldots, n-1$ are precomputed. Let us specialize the points $a_0, \ldots, a_{n-1}$ to be the n roots of unity, so $a_j = \omega^j$ for $j = 0, \ldots, n-1$. Then it is known (e.g., see Aho, Hopcroft, Ullman [1]) that $\sigma(z) = \Pi_{j=0}^{n-1}(z - \omega^j) = z^n - 1$. In this case, each $\sigma(z_j) = z_j^n - 1$ can be exactly computed by repeated squaring with work only $O(\log n)$. Since $\sigma^{(1)}(z) = nz^{n-1}$ in this case, and $\omega^n = 1$, each

$$\sigma^{(1)}(a_j) = \sigma^{(1)}(\omega^j) = n\omega^{j(n-1)} = n/\omega^j = n\omega^{n-j}.$$

Thus each $\sigma^{(1)}(a_j)$ costs one multiplication to compute from the known (precomputed) root of unity ω^{n-j}. Also,

$$\max_j |\sigma(z_j)| \leq 1 + \max_j |z_j|^n.$$

Note that if all the re-evaluation points z_j, for $j = 0, \ldots, m-1$, are on or within the unit disk,

$$\max_j |{z_j}^n| \leq \max_j |z_j|^n \leq 1.$$

This implies that

$$\max_j |\sigma(z_j)| \leq 1 + \max_j |{z_j}^n| \leq 2,$$

so by Proposition 2.4 (where we fix $\epsilon^* = 0$ since the $\sigma(z_j)$ are exactly computed) we have the following lemma.

LEMMA 2.6. *Let the points $a_0, \ldots, a_{n-1}$ be the n roots of unity, and all the re-evaluation points z_j, $j = 0, \ldots, m-1$ be on or within the unit disk, so $|z_j \leq 1$, and let the z_j be distinct from the n roots of unity. We construct an ϵ-approx solution $\widetilde{\psi}(z_0), \ldots, \widetilde{\psi}(z_{m-1})$ of the resulting Trummer's problem. Then for $j = 0, \ldots, m-1$, each $P(z_j)$ is 2ϵ-approx by*

$$\widetilde{P}(z_j) = \widetilde{\psi}(z_j)\sigma(z_j).$$

Now we bound the work. Given as input the coefficients of a degree $n-1$ polynomial $P(z)$, we can exactly evaluate $P(z)$ at the n roots of unity by applying the DFT algorithm to the coefficients of $P(z)$ in work $O(n \log n)$. Now given m evaluation points $z_0, \ldots, z_{m-1}$ on or within the unit disk, and distinct from the n roots of unity, we apply the above reduction to Trummer's problem. Let $|P| = \sum_{j=0}^{n-1} |p_j|$. Since $\sigma^{(1)}(z) = \frac{d\sigma(z)}{dz} = nz^{n-1}$, it follows that $|\sigma^{(1)}(\omega^j)| = n$. Hence the coefficients $c_j = P(\omega^j)/\sigma^{(1)}(\omega^j)$ of Trummer's problem have magnitude

$$|c_j| = |P(\omega^j)|/n = |P|/n$$

and have summed magnitude

$$|C| = \sum_{j=0}^{n-1} |c_j| = |P|.$$

We compute all the $\sigma(a_j)$ by $O(n)$ multiplications (of n times each of the precomputed roots of unity). If the input evaluation points $z_0, \ldots, z_{m-1}$ are fixed, then we can assume a precomputed well-separated decomposition of the set containing all the n roots of unity and the set of m evaluation points. Also, when the input evaluation points are fixed, we can assume precomputed $\sigma(z_j)$, $j = 0, \ldots, m-1$.

By the efficient multipole algorithm of [30, 34] (Lemma 2.2) the resulting Trummer's problem of size m can be ϵ-approx on every output within work $O(m \log^2 k)$, where $k = O(\log(|C|/\epsilon)) = O(\log(|P|/\epsilon))$. Thus if $m \geq (n \log n)/\log^2 k$, then the total work for m fixed evaluation points is $O(m \log^2 k)$. By Proposition 2.4, for $j = 0, \ldots, m-1$, the $\widetilde{P}(z_j)$ are an $\widehat{\epsilon}$-approx to the $P(z_j)$, where

$$\widehat{\epsilon} = \epsilon \max_j |\sigma(z_j)| \leq 2\epsilon,$$

since $\max_j |\sigma(z_j)| \leq 2$. Thus the approximation errors are upper bounded by 2ϵ, and rescaling (to simplify notation) 2ϵ to ϵ, we have the following theorem.

THEOREM 2.7. *Suppose we are given a complex degree $n-1$ polynomial $P(z)$, m fixed evaluation points on or within the unit disk (with a precomputed well-separated decomposition), and a given $\epsilon > 0$. Let $k = O(\log(|P|/\epsilon))$. Then we can compute an ϵ-approx of this complex polynomial evaluation problem within work $O(m \log^2 k + n \log n)$. If $m \geq (n \log n)/\log^2 k$, the amortized work per evaluation point is bounded by $O(\log^2 k)$.*

Note. For practical implementation of approximate polynomial evaluation via Theorem 2.7, one may substitute a theoretically less efficient multipole algorithm in place of the efficient multipole algorithm of [30, 34] (Lemma 2.2), which costs $O(m \log^2 k)$ work. For example, the FFT-accelerated fast multipole algorithm of Greengard and Rokhlin [22], which requires $O(mk \log k)$ work, in theory gives an improvement over the bounds of $O(n \log^2 n)$ for exact algorithms if $k \leq o((\log^2 n)/\log \log n)$ and moreover is known to be very efficient in practice. This FFT-accelerated fast multipole algorithm was implemented by Elliott and Board [15] on a variety of high performance machines, gives the currently fastest running multipole algorithm implementation known in practice (as opposed to theory) on these machines, and is used in many molecular simulation applications. Another approach would be to do a careful implementation of the multipole algorithm of Reif and Tate [30, 34], which may provide improved performance in practice, over the FFT-accelerated fast multipole algorithm.

Further note. We can reduce the work bounds of Theorem 2.7 for the special case of m chirp evaluation points, a Trummer's problem for m chirp evaluation points can be ϵ-approx on every output within work $O(m \log k)$. Thus the work bounds given in Theorem 2.8 can in this case be reduced by replacing $\log^2 k$ by $\log k$.

The case where the m evaluation points are not fixed. For small $k = o(n)$, the most costly parts of the above approximate complex polynomial evaluation algorithm is the computation of the $\sigma(z_j)$ and the well-separated decomposition, both costing work $O(m \log m)$ in the worst case. We now describe how to reduce this cost by approximate computation of the $\sigma(z_j)$ and, in certain cases, more efficient computation of the well-separated decomposition.

Again, we will assume that the input evaluation points $z_0, \ldots, z_{m-1}$ are on or within the unit disk; however, we redefine $a_j = r\omega^j$ where ω is the nth root of unity. Let $r = (1 + \frac{1}{n})$ and note $r^n \approx e$. Recall (e.g., see Aho, Hopcroft, Ullman [1]) that $\Pi_{j=0}^{n-1}(z - \omega^j) = z^n - 1$. In this case we redefine

$$\sigma(z) = \prod_{j=0}^{n-1}(z - r\omega^j) = r^n \prod_{j=0}^{n-1}\left(\frac{z}{r} - \omega^j\right) = r^n\left(\left(\frac{z}{r}\right)^n - 1\right) = z^n - r^n.$$

Since $|z_j| \leq 1$, we have

$$|\sigma(z_j)| = |z_j^n - r^n| \leq |z_j|^n + r^n \leq 1 + r^n \approx 1 + e.$$

Also,

$$|\sigma(z_j)| = |z^n - r^n| \geq |r^n| - |z^n| \leq r^n - 1 \approx e - 1.$$

Since we again have

$$\sigma^{(1)}(z) = \frac{d\sigma(z)}{dz} = nz^{n-1}$$

in this specialized case, and since $|a_j| = 1$, each

$$\sigma^{(1)}(a_j) = \sigma^{(1)}(r\omega^j) = n(r\omega^j)^{n-1},$$

so

$$|\sigma^{(1)}(a_j)| = nr^{n-1} \approx ne/r.$$

1. *Approximate computation of the $\sigma(z_j)$ via the unitary Trummer's problem.* If the input evaluation points are not fixed, we might exactly compute all the $\sigma(z_j) = z_j^n - r^n$ by powering in $O(m \log n)$ work. Instead, we will approximately compute the $\sigma(z_j)$, $j = 0, \ldots, m-1$. To do this, as described above, define a unitary Trummer's problem of size m where we specialize the weights to be 1, so $\psi^*(z) = 1/\sigma(z)$. Since for each j, the $a_j = r\omega^j$ are fixed, we can assume the $\sigma^{(1)}(a_j) = n(r\omega^j)^{n-1}$ are precomputed, thus providing the weights $c_j = 1/\sigma^{(1)}(a_j)$. Applying Lemma 2.2 again, the resulting unitary Trummer's problem of size n is ϵ-approx on every output $\widetilde{\sigma}(z_j)$, so $|\sigma(z_j) - \widetilde{\sigma}(z_j)| \leq \epsilon$, with the same work $O(m \log^2 k)$. We assume $\epsilon \leq 1/4$. By Proposition 2.5, each $\widetilde{\sigma}(z) = 1/\widetilde{\psi}^*(z_j)$ is an ϵ^*-approx to $\sigma(z_j)$, where

$$\epsilon^* = \max_j \frac{\epsilon|\sigma(z_j)|^2}{1 - |\sigma(z_j)|\epsilon} \leq \frac{\epsilon(1+e)^2}{1-(1+e)\epsilon} \leq O(\epsilon),$$

since we have shown $\max_j |\sigma(z_j)| \leq 1 + e$ and we have assumed $\epsilon \leq 1/4$. By Proposition 2.4, for $j = 0, \ldots, m-1$, each $\widetilde{P}(z_j)$ are an $\widehat{\epsilon}$-approx to $P(z_j)$, where

$$\widehat{\epsilon} \leq (\epsilon^* + \max_j |\sigma(z_j)|)\epsilon + (2 + (3\epsilon^*)/\min_j |\sigma(z_j)|)|P| \leq O(\epsilon|P|).$$

2. *Computation of the well-separated decomposition.* We can compute a well-separated decomposition of the set union of the $m \geq n$ evaluation points and the $a_j = r\omega^j$, $j = 0, \ldots, n-1$ within work $O((n+m)\log(n+m)) \leq O(m \log m)$ by the algorithm of Callahan and Kosaraju [6]. Alternatively, if the points have logarithmic bit-precision, then the algorithm of [30, 34] computes, within work $O((n+m)\log\log(n+m)) \leq O(m \log\log m)$, a well-separated decomposition the set of m evaluation points. Furthermore, since the $a_j = r\omega^j$, $j = 0, \ldots, n-1$ are regularly spaced on the radius r circle, a simple modification of the well-separated decomposition algorithm of [30, 33] can be used to compute a well-separated decomposition of the set union of n roots of unity and the set of m evaluation points, within work $O(m \log\log m)$.

In either case, we proceed as follows. We apply the above construction, for re-evaluation via Trummer's problem, to $P(z)$. Since $|\sigma^{(1)}(r\omega^j)| \approx ne/r$, the coefficients $c_j = P(\omega^j)/\sigma^{(1)}(\omega^j)$ of the Trummer's problem have summed magnitude $|C| \approx |P|r/e \leq O(|P|)$, so we may let

$$k = O(\log(|C|/\epsilon)) = O(\log(|P|/\epsilon)).$$

Again the resulting Trummer's problem of size m can be ϵ'-approx on every output using the efficient multipole algorithm of [30, 34] (Lemma 2.2) within work $O(m \log^2 k)$. For $j = 0, \ldots, m-1$, let

$$\widetilde{P}(z_j) = \widetilde{\psi}(z_j)\widetilde{\sigma}(z_j)$$

be the resulting approximation of $P(z_j)$. In either case, the error of approximation of $P(z_j)$ by $\widetilde{P}(z_j)$ is bounded as $|\widetilde{P}(z_j) - P(z_j)| \leq \widehat{\epsilon}$. Thus, by rescaling (to simplify notation) the total error $O(\widehat{\epsilon})$ to ϵ, we have the following theorem.

THEOREM 2.8. *Suppose we are given a complex degree $n-1$ polynomial $P(z)$, m evaluation points (which are not fixed) on or within the unit disk, and a given $\epsilon, 0 < \epsilon < 1/4$. Let $k = O(\log(|P|/\epsilon))$. Then we can compute an ϵ-approx of this complex*

polynomial evaluation problem within work $O(m(\log m + \log^2 k) + n \log n)$. *Thus, if* $m \geq (n \log n)/(\log m + \log^2 k)$, *then we require amortized work* $O(\log m + \log^2 k)$ *per evaluation point.*

Note. If the m evaluation points have logarithmic bit-precision, then the complexity bounds of Theorem 2.8 can be improved by application of the well-separated decomposition algorithm of [30, 33]; in this case each appearance of $\log m$ in Theorem 2.8 can be replaced with $\log \log m$.

Further note. In the special case that the m evaluation points are chirp points, a Trummer's problem for m chirp evaluation points can be ϵ-approx on every output within work $O(m \log k)$. Thus the work bounds given in Theorem 2.8 can in this case be reduced by replacing $\log^2 k$ by $\log k$.

3. Chebyshev points and approximate real polynomial evaluation.

3.1. The Chebyshev point evaluation problem. Let $\omega = e^{i2\pi/n'}$ be the n'th root of unity over the complex numbers $\mathcal{C}$. Note that

$$\omega^j = \cos(j2\pi/n') + i\sin(j2\pi/n').$$

Let $m \leq n'$. The (n', m)-*Chebyshev points* are the set of m real parts $x_j = \cos(j2\pi/n')$ of the powers ω^j, for $j = 0, 1, \ldots, m-1$. (This is slightly nonstandard notation, but convenient for our purposes.) Given real constants s_0, s_1 the *shifted* (n', m)-*Chebyshev points* are real values

$$x'_j = s_0 + s_1 \cos(j2\pi/n').$$

The *(shifted)* (n', m)-*Chebyshev point evaluation problem* is the multipoint real evaluation problem of evaluating a polynomial of degree $n-1$ at the set of m (shifted, respectively) (n', m)-Chebyshev points. The *(shifted)* (n', n)-*Chebyshev point interpolation problem* is the multipoint real interpolation problem of interpolating a polynomial of degree $n-1$ from the set of n (shifted, respectively) (n', n)-Chebyshev points.

Pan [29] has given an $O(n \log^2 n)$ algorithm for the $(4n, n)$-Chebyshev point evaluation problem and the shifted version of this problem. The results of Gerasoulis [18] imply an $O(n \log n)$ algorithm for (n', n)-Chebyshev point evaluation, for $n' = O(n)$, which uses a reduction to Trummer's problem for Chebyshev points. In the appendix we give an alternative algorithm (using a recursive algorithm and a reduction to the DFT and by application of Proposition 1.1) for (n', n)-Chebyshev point evaluation (and also interpolation) with these same work bounds but yielding a simpler algorithm.

LEMMA 3.1. *For* $n' \leq O(n)$, *the* (n', m)-*Chebyshev point evaluation problem can be solved in work* $O((n+m) \log \min(n, m))$, *and the* (n', n)-*Chebyshev point interpolation problem can be solved in work* $O(n \log n)$. *(The circuit depth is* $O(\log n)$.*) Also, the shifted version of these problems can be solved within the same work bounds.*

3.2. Approximate real evaluation via interpolation at the Chebyshev points. Interpolation at the Chebyshev points is a well-known classical method for approximation of real functions (see, for example, Dahlquist and Björck [10] and Henry [24]). Fix an interval $I = [L, U]$ of the real line of length $|I| = U - L$. Let $f(x)$ be any real function over I. We say $\widetilde{f}(x)$ is an ϵ-approx of $f(x)$ over I if $|f(x) - \widetilde{f}(x)| \leq \epsilon$ for any $x \in I$. Fix $\widetilde{f}(x)$ to be the degree k polynomial derived by interpolating $f(x)$ at the shifted $(2k, k)$-Chebyshev points $x'_j = \frac{U+L}{2} + \frac{U-L}{2}\cos(\frac{j\pi}{k})$, for $j = 0, \ldots, k-1$,

which are on the interval $I = [L, U]$. Thus $\widetilde{f}(x)$ is the unique degree k polynomial such that $\widetilde{f}(x'_j) = f(x'_j)$ for $j = 0, \ldots, k-1$. Let

$$f^{(k)}(x) = \frac{d^k f(x)}{d^k x}$$

be the kth derivative of $f(x)$ with respect to x.

PROPOSITION 3.2 (see Dahlquist and Björck [10]). *For any $x \in I$, $\widetilde{f}(x)$ is an ϵ-approx of $f(x)$, for*

$$\epsilon = \frac{2(|I|/4)^k}{k!} \max_{y \in I} |f^{(k)}(y)|.$$

3.3. Approximate real polynomial evaluation via interpolation at the Chebyshev points. Rokhlin [35] applied Proposition 3.2 for a fast algorithm for the discrete Laplace transformation. Fix $P(x)$ to be a real polynomial of degree $n-1$. As observed in Pan, Reif, and Tate [30], direct application of the well-known Proposition 3.2 immediately gives an ϵ-approx to the real multipoint evaluation problem for $P(x)$ at any given m real points $x_0, \ldots, x_{m-1}$. By Lemma 3.1, the shifted $(2k, k)$-Chebyshev point evaluation problem for $P(x)$ can be solved within work $O(n \log k)$. Known exact algorithms [16, 26, 5] for k-point polynomial interpolation require work $O(k \log^2 k)$. By Proposition 1.2, the multipoint polynomial evaluation problem for $\widetilde{P}(x)$ at the real points $x_0, \ldots, x_{m-1} \in I$ can be solved within work $O(m \log^2 k)$. Thus an ϵ-approx of the multipoint polynomial evaluation problem for $P(x)$ at any m real points $\in I$ can be solved within work $O(m \log^2 k)$, where $\epsilon = \frac{2(|I|/4)^k}{k!} \max_{y \in I} |P^{(k)}(y)|$. Let $\beta = \log(|P|/\epsilon)$, where $|P|$ is the sum of the moduli of the coefficients of P. Assuming without loss of generality (w.l.o.g.) $k = o(n)$, note that if we fix $I = [-1, 1]$ we have that

$$\max_{y \in I} |P^{(k)}(y)| \leq \max_{y \in I} \max_{k \leq j < n} \frac{j!}{(j-k)!} y^{-(j-k)} |P| \leq \max_{k \leq j < n} \frac{j!}{(j-k)!} 2^{-(j-k)} |P|,$$

which is maximized when $j = k$, so

$$\max_{y \in I} |P^{(k)}(y)| \leq \frac{j!}{(j-k)!} 2^{-(j-k)} |P| \leq k! |P|.$$

Setting the degree of $\widetilde{P}$ to be $k-1$, where $k = \log(|P|/\epsilon)$, we have that

$$\epsilon \leq \frac{2(|I|/4)^k}{k!} \max_{y \in I} |P^{(k)}(y)| \leq |P|/2^{k-1},$$

implying, by Proposition 3.2, the following known result.

LEMMA 3.3 (see Pan, Reif, and Tate [30] and also Bini and Pan [3]). *An ϵ-approx of the multipoint polynomial evaluation problem for a degree $n-1$ polynomial $P(x)$ at any $m \geq n$ real points $\in [-1, 1]$ can be solved within work $O(m \log^2 \min(n, k))$, where $k = \log(|P|/\epsilon)$.*

So far we have assumed that $P(x)$ has only real coefficients. Note that if $P(x)$ has complex coefficients, then we can let $P(x) = P_0(x) + iP_1(x)$ where $P_0(x), P_1(x)$ have only real coefficients. Then application of Lemma 3.3 for each of the $P_0(x), P_1(x)$

implies that for $k = \log(|P|/\epsilon)$, the resulting error of approximation for evaluation of $P(x)$ is a $\sqrt{2}$ factor more, that is, $\sqrt{2}\epsilon$. Hence rescaling (to simplify notation) the error $\sqrt{2}\epsilon$ to ϵ, we have the following lemma.

LEMMA 3.4. *An ϵ-approx of the multipoint polynomial evaluation problem for a degree $n-1$ polynomial $P(x)$ with complex coefficients at any $m \geq n$ real points $\in [-1,1]$ is solved in work*

$$\leq O(m \log^2 \min(n, \log(|P|/\epsilon))).$$

By Lemma 3.1, the (n', m)-Chebyshev point evaluation problem for the degree $k-1$ polynomial $\widetilde{P}(x)$ at shifted (n', m)-Chebyshev points can be solved within work $O(m \log k)$. All Chebyshev points are within the interval $I = [-1, 1]$. Setting $k = \log(|P|/\epsilon)$, we have again that $\epsilon \leq |P|/2^{k-1}$ implying the improved result.

LEMMA 3.5. *An ϵ-approx of the (n', m)-Chebyshev point evaluation problem for a degree $n-1$ polynomial $P(x)$ is solved in work*

$$O((m+n) \log \min(n, \log(|P|/\epsilon))).$$

(The parallel time is $O(\log \min(n, \log(|P|/\epsilon)))$.)

4. Approximate evaluation on a circle.

4.1. Approximate polynomial evaluation on a circle via interpolation at the Chebyshev angles. We further reduce our amortized work bounds for special sets of evaluation points. By reduction to approximate evaluation of Trummer's problem via the Multipole algorithm, we have already shown how to ϵ-approx complex degree $n-1$ polynomial evaluation at $m \geq n \log n$ chirp points $\zeta^j, j = 0, \ldots, m-1$, for some fixed complex number ζ, $|\zeta| \leq 1$ in amortized work $O(\log k)$ per point. Using quite distinct techniques, we give in this section a reduction from ϵ-approx complex degree $n-1$ polynomial evaluation at $m \geq n$ points on a circle of radius r to approximate real polynomial evaluation, again in amortized work $O(\log^2 k)$ per point. Let $i = \sqrt{-1}$. Fix a complex polynomial $P(z) = \sum_{j=0}^{n-1} p_j z^j$ of degree $n-1$ and let

$$|P| = \sum_{j=0}^{n-1} |p_j|.$$

Let $\epsilon > 0$ be a given error bound. Let $z = re^{i\theta}$ range on a circle of radius r, and define the angle of z to be $\theta(z) = \theta$. We now consider the evaluation of $P(z)$ at a set of points $z_j = re^{i\theta_j}$ for $j = 1, \ldots, m$ over an interval of a circle of radius r with angular bounds $\theta_j \in [0, \Delta]$, for a positive real $\Delta \leq 2\pi$. The set of points $re^{i\theta_j}$, for $j = 0, \ldots, m-1$, are *regularly spaced* on the circle of radius r if the θ_j are a linear function of j (for example, the n roots of unity are regularly spaced on the unit circle). The polynomial $P(z)$ is (k, t)-*descending* if the magnitude of the coefficients drops as

$$|p_j| \leq n^{O(1)} |P| \left(\frac{t}{j}\right)^k,$$

for $j = k, \ldots, n-1$. Here we prove the following.

THEOREM 4.1. *Suppose we are given a complex polynomial $P(z) = \sum_{j=0}^{n-1} p_j z^j$ of degree $n-1$, and a fixed set of evaluation points $z_j = re^{i\theta_j}$ for $j = 0, \ldots, m-1$ on a circle of radius r with angular range $\theta_j \in [0, \Delta]$. Let $c = 1$ if the evaluation points are*

regularly spaced and otherwise $c = 2$. *Then an* ϵ*-approx of this multipoint polynomial evaluation problem can be computed within work* $O(m \log^c k)$ *if either* 1. $r = 1$ *and* $\Delta \leq \frac{k}{e(n-1)}$, *or* 2. $r = 1$ *and* $P(z)$ *is* $(k, k/(e\pi))$*-descending, or* 3. $r \leq 1/(e\pi)$. *Here* $k = \frac{1}{2}\log(|P|/\epsilon) + O(1)$ *in cases* 1 *and* 3 *and* $k = O(\log(n|P|/\epsilon))$ *in case* 2.

Proof. Restrict z to the upper half circle of radius r, so $z = re^{i\theta}$, where $i = \sqrt{-1}$, and $0 \leq \theta \leq \pi$. Let $Q(\theta) = P(re^{i\theta}) = \sum_{j=0}^{n-1} q_j e^{ij\theta}$, where $q_j = r^j p_j$, and $P(z) = \sum_{j=0}^{n-1} p_j z^j$. Fix a number k which divides n, to be determined below. The exact evaluation of $P(z)$ at the evaluation points z_j, for $j = 0, \ldots, m-1$, can be done by an exact evaluation of $Q(\theta)$ at the real points $\theta_0, \ldots, \theta_{m-1}$, where $z_j = re^{i\theta_j}$. Instead, we will do an ϵ-approximate evaluation of $P(z)$ at the evaluation points $z_0, \ldots, z_{m-1}$, as follows. We can assume w.l.o.g. that each $0 \leq \theta_j \leq \pi$. Since $P(-z) = \sum_{j=0}^{n-1} p_j(-z)^j = \sum_{j=0}^{n-1} p'_j z^j$, where $p'_j = -p_j$ if j is odd, and else $p'_j = p_j$, the case where the evaluation points $z_j = re^{i\theta_j}$ are on the lower half circle can be reduced to this case by multiplication of the z_j by $e^{i\pi} = -1$, and switching the sign of the coefficients p_j of $P(z)$ where j is odd. We construct a degree $k-1$ polynomial $\widetilde{Q}(\theta)$ which gives an ϵ-approx of polynomial $Q(\theta)$. To construct $\widetilde{Q}(\theta)$, we do an exact evaluation of $Q(\theta)$ at the shifted $(2k, k)$-Chebyshev points $\theta'_j = \pi\cos(j\pi/k)$ for $j = 0, \ldots, k-1$ over the real interval $I = [0, \Delta]$, for a given $\Delta, 0 < \Delta \leq 2\pi$. To do this first step, we can exactly evaluate $P(z)$ at $re^{i\theta'_j}$ for $j = 0, \ldots, k-1$, which in general costs work $O(n \log^2 k)$ by Proposition 1.1; or if the evaluation points are regularly spaced, then by Proposition 1.4 this costs $O(n \log k)$. We could now exactly interpolate a degree $k-1$ polynomial $\widetilde{Q}(\theta)$ from $Q(\theta'_j) = P(re^{i\theta'_j})$, for $j = 0, \ldots, k-1$, which are the values of $Q(\theta)$ at these shifted $(2k, k)$-Chebyshev points $\theta'_0, \theta'_1, \ldots, \theta'_{k-1}$. By the known algorithms of [16, 26, 5], the exact interpolation of polynomial $\widetilde{Q}(\theta)$ at these k points cost work $O(k \log^2 k)$. Also, by known algorithms (see Proposition 1.2), we can exactly evaluate $\widetilde{Q}(\theta_j)$ for $j = 0, \ldots, m-1$, in work $O(m \log^2 k)$, and if these evaluation points are regularly spaced, then this costs work $O(m \log k)$ by the chirp transform (Proposition 1.4).

LEMMA 4.2. *For* $k = o(n)$, *and* $0 \leq \theta \leq \Delta$, *then*

$$|Q^{(k)}(\theta)| = \left|\frac{d^k Q(\theta)}{d^k \theta}\right|$$

is bounded by $O(k!|P|\Delta^{-k})$ *if either* 1. $r = 1$ *and* $\Delta \leq \frac{k}{e(n-1)} \leq \pi$, *or* $\Delta = \pi$ *and either* 2. $r = 1$ *and* $P(z)$ *is* $(k, k/(e\pi))$*-descending, or* 3. $r \leq 1/(e\pi)$.

Proof.

$$\frac{d^k e^{ij\theta}}{d^k \theta} = \frac{j!}{(j-k)!}(ij)^k e^{ij\theta},$$

so

$$Q^{(k)}(\theta) = \frac{d^k Q(\theta)}{d^k \theta} = \sum_{j=k}^{n-1} q_j \frac{j!}{(j-k)!}(ij)^k e^{ij\theta}.$$

Since $|i| = |e^i| = 1$, and $|q_j| = |p_j| r^j$, we have

$$|Q^{(k)}(\theta)| = \sum_{j=k}^{n-1} |q_j| j^k = \sum_{j=k}^{n-1} |p_j| r^j j^k.$$

1. Now suppose $r = 1$ and $\Delta \leq \frac{k}{e(n-1)}$. Then by the Stirling approximation to factorial, $k! \geq (k/e)^k$ and so

$$j^k \leq (k/e)^k \Delta^{-k} \leq k! \Delta^{-k}.$$

Hence we have

$$|Q^{(k)}(\theta)| \leq \sum_{j=k}^{n-1} |p_j| j^k \leq O(|P| k! \Delta^{-k}).$$

2. Next, suppose $r = 1$, and $\Delta = \pi$, and $P(z)$ is $(k, k/(e\pi))$-descending; so by definition, the magnitude of the coefficients of $P(z)$ drops as

$$|p_j| \leq n^{O(1)} |P| (\frac{k}{je\pi})^k,$$

for $j = k, \ldots, n-1$. Then

$$|p_j| r^j j^k \leq n^{O(1)} |P| (k/(e\pi))^k \leq O(n^{O(1)} |P| k! \Delta^{-k}).$$

Thus,

$$|Q^{(k)}(\theta)| \leq \sum_{j=k}^{n-1} |p_j| r^j j^k \leq \sum_{j=k}^{n-1} O(n^{O(1)} |P| k! \Delta^{-k}) \leq O(n^{O(1)} |P| k! \Delta^{-k}).$$

3. Finally, suppose $r \leq 1/(e\pi)$ and $\Delta = \pi$. Since $|Q^{(k)}(\theta)|$ increases with r, to upper bound $|Q^{(k)}(\theta)|$ we can assume w.l.o.g. that r is at its maximum value $r = 1/(e\pi)$. By taking derivatives, it is easy to verify that $1/(e\pi) \leq (se\pi)^{-1/s}$ for any $s \geq 1$. So

$$1/(e\pi) = \min_{s \geq 1} (se\pi)^{-1/s} = \min_{j \leq k} \left(\frac{k}{je\pi} \right)^{k/j}$$

(by the substitution $j = sk$). Hence for each $r \leq 1/(e\pi)$, we can bound $r^j \leq (\frac{k}{je\pi})^k$ and so for all $j \geq k$ we have

$$r^j j^k \leq (k/(e\pi))^k \leq k! \Delta^{-k},$$

since $\Delta = \pi$ and $k! \geq (k/e)^k$. Thus,

$$|Q^{(k)}(\theta)| \leq \sum_{j=k}^{n-1} |p_j| r^j j^k \leq \sum_{j=k}^{n-1} |p_j| k! \Delta^{-k} \leq O(|P| k! \Delta^{-k}). \qquad \square$$

To complete the proof of Theorem 4.1, we need to bound the approximation errors for evaluation of $\widetilde{Q}(\theta)$ at all the given evaluation points; this is implied by the following proof of ϵ-approximate evaluation of $P(z)$. Since $|I| = \Delta$, then by Lemma 4.2 and Proposition 3.2 the error in the approximation of $P(z) = Q(\theta)$ by $\widetilde{Q}(\theta)$, for $\theta \in I = [0, \Delta]$ is upper bounded by

$$\frac{2}{k!} (|I|/4)^k \max_{\theta \in I} |Q^{(k)}(\theta)| \leq O(|P| (\Delta/4)^k \Delta^{-k}) \leq O(|P| 4^{-k})$$

in cases 1 and 3, and in case 2, the error is upper bounded by $O(n^{O(1)}|P|4^{-k})$. Hence the error is upper bounded by $O(\epsilon)$ if in cases 1 and 3 we set $k = \frac{1}{2}\log(|P|/\epsilon) + O(1)$ and in case 2 we set $k = O(\log(n|P|/\epsilon))$. Finally, note that the (empty) restriction $0 \le \theta \le \pi = \Delta$ in cases 2 and 3 allows z to range over any point on the upper half circle of fixed radius r, so we have the following lemma.

LEMMA 4.3. *$\widetilde{Q}(\theta)$ is an ϵ-approx of $P(z)$ at any point $z = re^{i\theta}$ on the upper half circle of fixed radius r, where $0 \le \theta \le \Delta$, if either* 1. *$r = 1$ and $\Delta \le \frac{k}{e(n-1)} \le \pi$, or* 2. *$r = 1$ and $P(z)$ is $(k, k/(e\pi))$-descending, or* 3. *$r \le 1/(e\pi)$.*

Thus, we have proven Theorem 4.1. □

Cases 1 and 2 of Theorem 4.1 imply the following corollary.

COROLLARY 4.4. *In total work $O(n \log k)$ we can*

1. *ϵ-approx the evaluation of a degree n polynomial at the first n powers of the n'th root of unity, where $n' \ge \Omega(n^2/k)$, and*
2. *ϵ-approx the n-point DFT in total work $O(n \log k)$ for inputs with $(k, k/(e\pi))$-descending coefficient magnitude.*

Note. If the input set of evaluation points are not fixed, the result as given in Theorem 4.1 holds with the same work bounds, assuming an *extended arithmetic model* where, given a real θ, then $e^{i\theta} = \cos(\theta) + i\sin(\theta)$ can be computed in $O(1)$ steps. This assumption is not required if the points are fixed, as assumed in Theorem 4.1, or regularly spaced (as in Corollary 4.4). Also this assumption is not required in the alternative construction given in subsection 4.2, even if the input set of evaluation points are not fixed.

4.2. An alternative approach to approximate evaluation on a circle via approximate real polynomial evaluation. We now give an alternative construction for a similar, but slightly weaker, result as Theorem 4.1, by use of a surprisingly simple algorithm (the proof is somewhat more involved, however), which uses complex polynomials rather than real polynomials.

Input: The coefficients of degree $n-1$ complex polynomial $P(z)$, and $m \ge n$ evaluation points $z_j = re^{i\theta_j}$, for $j = 0, \ldots, m-1$ on an interval of a circle of radius $r = 1$ with $|\cos(\theta_j)| \le \Delta$. We assume either 1. $\Delta \le \frac{k}{e(n-k)}$, or 2. $P(z)$ is $(k, \sqrt{2}/8)$-descending. In case 1. let k be the smallest power of 2 which is $\ge \log(\sqrt{2}|P|/\epsilon)$, and in case 2. let k be the smallest power of 2 which is $O(\log(n|P|/\epsilon))$. In any case we assume $k = o(n)$.

[0] Partition the set of evaluation points $\{z_j | j = 1, \ldots, m-1\}$ into 4 sets

$$\{z_{j,h} | j = 0, \ldots, m_h - 1\}, \qquad h = 0, \ldots, 3,$$

such that $|i^h\theta_{j,h} - \pi/2| \le \pi/4$ for each $j = 0, \ldots, m_h$.

Comment: Multiplication by $i = e^{i\pi/2}$ shifts the angle by $\pi/2$.

For $h = 0, \ldots, 3$ **do**

1. Define polynomial $P_h(z) = P(i^h z)$. Evaluate $P_h(z)$ at

$$z'_j = r(x'_j + i\sqrt{1 - (x'_j)^2}),$$

where

$$x'_j = \cos(j\pi/k)/\sqrt{2},$$

for $j = 0, \ldots, k-1$, in work $O(n \log^2 k)$ (by Proposition 1.1).

2. Interpolate the degree $k-1$ complex polynomial $\widetilde{P}_h(z)$ from the values $P_h(z'_j)$, for $j = 0, \ldots, k-1$, in work $O(k \log^2 k)$.

3. Evaluate polynomial $\widetilde{P}_h(z)$ at the given evaluation points $z_{j,h}$ for $j = 0, \ldots, m_h - 1$. If these evaluation points are regularly spaced, then this costs work $O(m_h \log k)$ by the chirp transform (Proposition 1.4) and otherwise costs work $O(m_h \log^2 k)$ by Proposition 1.1. This gives $\widetilde{P}_h(z_j)$ which ϵ-approx $P_h(z_j)$, for $j = 0, \ldots, m_h - 1$.

Output: $\widetilde{P}(z_j)$ which ϵ-approx $P(z_j)$, for $j = 0, \ldots, m-1$.

We prove the following in Appendix A.

THEOREM 4.5. *$\widetilde{P}(z)$ is an ϵ-approx of $P(z)$ at any point $z = r(x+iy)$ on the unit circle of fixed radius $r = 1$ where $|\theta(z) - \pi/2| \leq \pi/4$ and $|x| \leq \Delta$ if either* 1. $\Delta \leq \frac{k}{e(n-k)}$ *or* 2. *$P(z)$ is $(k, \sqrt{2}/8)$-descending.*

The work is again $O(n \log^2 k) + O(m \log^c k) \leq O(n \log^2 k + m \log^c k)$. Also, the circuit depth can be seen to be $O(\log^c k)$ with this same work bound.

4.3. Dutt and Rokhlin's approximation of a polynomial on the unit circle. A further method for approximation of a polynomial over an interval of the unit circle is proposed by Dutt and Rokhlin [14], who give an approximation to a polynomial $P(z)$, as follows. Fix a real constant α, $0 < \alpha < \pi/3$. They define a mapping τ from complex numbers $z = e^{i\theta}$ on the unit circle to real $x = \tau(z) = 3 \tan\alpha \cot(\theta/2)$ and use this mapping to define the real function $f(x) = P(z)$. They use a degree $k-1$ polynomial $\widetilde{f}(x)$ to approximate $P(z)$ at angular positions θ, $6\alpha \leq \theta \leq 2\pi - 6\alpha$. The polynomial $\widetilde{f}(x)$ interpolates $f(x)$ at the shifted $(2k, k)$-Chebyshev points $a_j = -\cos((j - 1/2)\pi/k)$ and is defined, for $|x| \leq 1$, as $\widetilde{f}(x) = \sum_{j=1}^{k-1} f(a_j) \prod_{\ell \neq j} (\frac{x - a_\ell}{a_j - a_\ell})$. Note that $\widetilde{f}(a_j) = f(a_j)$ for $j = 1, \ldots, k-1$. Dutt and Rokhlin [14] prove that $\widetilde{f}(x)$ approximates $f_h(x) = P(z)$, with relative error $O(1/5^k)$ over some portion of this interval. Our Theorem 5.1 implies that the length of the interval where these error bounds can be obtained must be very small.

5. Approximation everywhere on the unit circle is not possible. Here we show that it is not possible to approximate an arbitrary polynomial by a small degree polynomial over a large portion of the unit circle. Fix an angular interval $[0, \Delta]$ for $0 \leq \Delta \leq 2\pi$ and an error ϵ, $0 \leq \epsilon < 1$. Let us define an (k, Δ) *circle ϵ-approx evaluation scheme* for a degree $n-1$ polynomial $P(z)$ to be a degree $k-1$ polynomial $\widetilde{f}(x)$ which ϵ-approx $P(z) = f(x)$ at all angular positions $\theta \in [0, \Delta]$, where τ is a mapping from complex numbers $z = e^{i\theta}$ on the interval of the unit circle, $\theta \in [0, \Delta]$ to real $x = \tau(z)$. We now prove that any (k, Δ) circle ϵ-approx evaluation scheme in general only can be convergent for angular positions $\theta \in [0, \Delta]$, where $\Delta \leq O(k/n)$.

THEOREM 5.1. *There is no (k, Δ) circle ϵ-approx evaluation scheme for every degree $n-1$ polynomial $P(z)$, where $k\lceil 2\pi/\Delta \rceil < n$ and $\epsilon(\epsilon + 2|P|) < 1$.*

Proof. Let $H = \lceil 2\pi/\Delta \rceil$. We can use the (k, Δ) circle ϵ-approx evaluation scheme to define for each $h, 0 \leq h < H$ a degree $k-1$ polynomial $f_h(x)$ that ϵ-approx $P(z)$ for $x = \tau(z\omega^{-2\pi h/H})$. For a complex number z, define $h(z)$ to be the integer h such that $2\pi h/H \leq |\text{angle}(z)| < 2\pi(h+1)/H$.

PROPOSITION 5.2. *Given a complex degree $n-1$ polynomial $P(z)$, and a (k, Δ) circle ϵ-approx evaluation scheme, we can construct $H = \lceil 2\pi/\Delta \rceil$ polynomials $(f_0(x), \ldots, f_{H-1}(x))$=POLY-APPROX$_k(P)$, each of degree $k-1$, such that for any z on the unit circle, $f_h(x)$ is an ϵ-approx of $P(z)$, for $h = h(z)$ and $x = \tau(z\omega^{-2\pi h/H})$.*

Polynomial Convolution is defined as follows:

Input: Complex coefficients $p_0, \ldots, p_{n-1}, q_0, \ldots, q_{n-1}$, defining degree $n-1$ polyno-

mials $P(z) = \sum_{j=0}^{n-1} p_j x^j$ and $Q(z) = \sum_{j=0}^{n-1} q_j x^j$.
Output: Complex coefficients $q_0, \ldots, q_{2n-2}$, defining degree $2n-2$ product polynomial $R(z) = P(z)Q(z) = \sum_{\ell=0}^{2n-2} r_\ell x^\ell$, where

$$r_\ell = \sum_{j=0, 0 \le \ell - j < n}^{n-1} p_j q_{\ell - j}.$$

Next we show the following.

LEMMA 5.3. *Suppose there is a (k, Δ) circle ϵ-approx evaluation scheme. Then, given complex n-vectors $\mathbf{u}$, $\mathbf{v}$, we can construct vectors $\widetilde{\mathbf{u}} = ROW\text{-}COMPRESS_{\,k}(\mathbf{u})$, $\widetilde{\mathbf{v}}^T = COLUMN\text{-}COMPRESS_k(\mathbf{v}^T)$ of size $O(k)$ (where $\widetilde{\mathbf{u}}$ depends only on $\mathbf{u}$ and not on $\mathbf{v}$, and furthermore $\widetilde{\mathbf{v}}$ depends only on $\mathbf{v}$ and not on $\mathbf{u}$) such that $\widetilde{\mathbf{u}}^T \widetilde{\mathbf{v}}$ is an ϵ_1-approx of $\mathbf{u}^T\mathbf{v}$ for $\epsilon_1 \le \epsilon(\epsilon + |\mathbf{u}| + |\mathbf{v}|)$.*

Proof. Given complex n-vectors $\mathbf{u} = (u_0, \ldots, u_{n-1})^T$, $\mathbf{v} = (v_0, \ldots, v_{n-1})^T$, we can compute the inner product

$$\mathbf{u}^T\mathbf{v} = \sum_{j=0}^{n-1} u_j v_j$$

by computing the nth coefficient

$$r_n = \sum_{j=0}^{n-1} p_j q_{n-j} = \sum_{j=0}^{n-1} u_j v_j = \mathbf{u}^T\mathbf{v}$$

of the product polynomial $R(z) = P(z)Q(z)$ as defined above and where $p_j = u_j$ and $q_j = v_{n-j}$, for $j = 0, \ldots, n-1$. Let $\omega = e^{i\pi/n}$ be the $(2n)$th root of unity, where $i = \sqrt{-1}$. By the convolution theorem, $\mathbf{u}^T\mathbf{v} = r_n = \frac{1}{2n} \sum_{j=0}^{2n-1} \omega^{-nj} P(\omega^j) Q(\omega^j)$. For each of $P(z), Q(z)$ defined above, by Proposition 5.2 we can construct $H = \lceil 2\pi/\Delta \rceil$ degree $k-1$ polynomials $(f_0(x), \ldots, f_{H-1}(x)) =$ POLY-APPROX$_k(P)$, and $(g_0(x), \ldots, g_{H-1}(x)) =$ POLY-APPROX$_k(Q)$, such that for any z on the unit circle, $\widetilde{P}_h(z) = f_h(x)$ is an ϵ-approx of $P(z)$, and $\widetilde{Q}_h(z) = g_h(x)$ is an ϵ-approx of $Q(z)$, for $h = h(z)$ and $x = \tau(z\omega^{-2\pi h/H})$. For $h = 0, \ldots, H-1$ let $f_h(x) = \sum_{a=0}^{k-1} f_{h,a} x^a$ and $g_h(x) = \sum_{b=0}^{k-1} g_{h,b} x^b$. Let $\delta = n/H$. For each integer $j, 0 \le j < 2n$, let h_j be the number $h \in \{0, \ldots, H-1\}$ such that $\delta h \le j < \delta(h+1)$. Then we have that $\widetilde{P}_{h_j}(\omega^j)$ is an ϵ-approx of $P(\omega^j)$, and $\widetilde{Q}_{h_j}(\omega^j)$ is an ϵ-approx of $Q(\omega^j)$. We will approximate $r_n = \frac{1}{2n} \sum_{j=0}^{2n-1} \omega^{-nj} P(\omega^j) Q(\omega^j)$ by $\widetilde{r}_n = \frac{1}{2n} \sum_{j=0}^{2n-1} \omega^{-nj}\ \widetilde{P}_{h_j}(\omega^j) \widetilde{Q}_{h_j}(\omega^j)$.

PROPOSITION 5.4. $|r_n - \widetilde{r}_n| \le \epsilon(\epsilon + |P| + |Q|)$.

Proof. Since $|\omega^{-nj}| = 1$ and for $h = h_j$,

$$|P(\omega^j)Q(\omega^j) - \widetilde{P}_h(\omega^j)\widetilde{Q}_h(\omega^j)| \le |P(\omega^j)(Q(\omega^j) - \widetilde{Q}_h(\omega^j)) + (P(\omega^j) - \widetilde{P}_h(\omega^j))\widetilde{Q}_h(\omega^j)|$$

$$\le |P(\omega^j)||Q(\omega^j) - \widetilde{Q}_h(\omega^j)| + |P(\omega^j) - \widetilde{P}_h(\omega^j)||\widetilde{Q}_h(\omega^j)| \le \epsilon(|P| + \epsilon + |Q|).$$

Thus we have

$$|r_n - \widetilde{r}_n| \le \frac{1}{2n} \sum_{j=0}^{2n-1} |\omega^{-nj}||P(\omega^j)Q(\omega^j) - \widetilde{P}_{h_j}(\omega^j)\widetilde{Q}_{h_j}(\omega^j)|$$

$$\leq \frac{2n}{2n}\epsilon(\epsilon + |P| + |Q|) = \epsilon(\epsilon + |P| + |Q|). \qquad \square$$

Surprisingly, we can contract the expansion of $\widetilde{r}_n$ as follows in this proposition.

PROPOSITION 5.5. *$\widetilde{r}_n = \sum_{h=0}^{H-1} \sum_{a=0}^{k-1} f_{h,a} s_{h,a}$, where $s_{h,a} = \sum_{b=0}^{k-1} g_{h,b} c_{h,a+b}$, and the $c_{h,\ell}$ are complex scalar constants dependent only on h, ℓ, n, and δ.*

Proof. The $(2n)$th root of unity is defined $\omega = e^{i\pi/n}$. For a fixed small constant α, each power ω^j, for $0 \leq j < n$, is mapped to the real $d_j = \tau(\omega^j)$ which is a constant (that is, the d_j need not be computed, and instead are provided by the algebraic circuit that we construct) for fixed n and α. For $h = h_j$, we have defined $\widetilde{P}_h(\omega^j) = f_h(d_{j-\delta h}) = \sum_{a=0}^{k-1} f_{h,a} d_{j-\delta h}^a$ and similarly,

$$\widetilde{Q}_h(\omega^j) = g_h(d_{j-\delta h}) = \sum_{b=0}^{k-1} g_{h,b} d_{j-\delta h}^b.$$

Thus for $h = h_j$, we can expand

$$\widetilde{P}_h(\omega^j)\widetilde{Q}_h(\omega^j) = \left(\sum_{a=0}^{k-1} f_{h,a} d_{j-\delta h}^a\right)\left(\sum_{b=0}^{k-1} g_{h,b} d_{j-\delta h}^b\right) = \sum_{a,b=0}^{k-1} f_{h,a} g_{h,b} d_{j-\delta h}^{a+b}.$$

Interchanging the order of summation (bringing the summation of j inside and a summation of h outside), we get

$$\widetilde{r}_n = \frac{1}{2n}\sum_{j=0}^{2n-1} \omega^{-nj} \widetilde{P}_{h_j}(\omega^j)\widetilde{Q}_{h_j}(\omega^j) = \sum_{h=0}^{H-1}\sum_{a=0}^{k-1} f_{h,a} s_{h,a},$$

where we define the *associated prefix sums* to be

$$\begin{aligned} s_{h,a} &= \frac{1}{2n}\sum_{b=0}^{k-1} g_{h,b} \sum_{j=\delta h}^{\delta(h+1)-1} \omega^{-nj} d_{j-\delta h}^{a+b} \\ &= \sum_{b=0}^{k-1} g_{h,b}\left(\frac{1}{2n}\sum_{j=\delta h}^{\delta(h+1)-1} \omega^{-nj} d_{j-\delta h}^{a+b}\right) = \sum_{b=0}^{k-1} g_{h,b} c_{h,a+b}, \end{aligned}$$

and where we define the *associated scalar constants* to be

$$c_{h,\ell} = \frac{1}{2n}\sum_{j=\delta h}^{\delta(h+1)-1} \omega^{-nj} d_{j-\delta h}^{a+b}.$$

Thus, Proposition 5.5 follows. $\square$

Note that the $s_{h,a}$ are defined independently of the $f_{h,a}$. To complete the proof of Lemma 5.3, we define $\widetilde{\mathbf{u}} =$ ROW-COMPRESS$_k(\mathbf{u})$ and $\widetilde{\mathbf{v}}^T =$ COLUMN-COMPRESS$_k(\mathbf{v}^T)$, where for each $h = 0, \ldots, H-1$ and for each $a = 0, \ldots, k-1$,

$$\widetilde{\mathbf{u}}_{hk+a} = f_{h,a}$$

and

$$\widetilde{\mathbf{v}}_{hk+a} = s_{h,a}.$$

Thus by the convolution theorem and Proposition 5.5,

$$\widetilde{r}_n = \sum_{h=0}^{H-1}\sum_{a=0}^{k-1} f_{h,a}s_{h,a} = \sum_{\ell=0}^{Hk-1} \widetilde{\mathbf{u}}_\ell \widetilde{\mathbf{v}}_\ell = \widetilde{\mathbf{u}}^T\widetilde{\mathbf{v}}.$$

Note that

$$|\mathbf{u}| = \sum |u_j| = \sum |p_j| = |P|$$

and

$$|\mathbf{v}| = \sum |v_j| = \sum |q_j| = |P|.$$

Now, given an error bound $\epsilon_1 > 0$, let

$$\epsilon = \epsilon_1/(\epsilon + |P| + |Q|) = \epsilon_1/(\epsilon + |\mathbf{u}| + |\mathbf{v}|).$$

By Proposition 5.4, approximation error $\epsilon_1 \le \epsilon(\epsilon + |\mathbf{u}| + |\mathbf{v}|)$, so we have proved Lemma 5.3. ∎

Given two $n \times n$ matrices A, B (with rows and columns indexed from 0 to $n-1$), we now approximate the inner product AB, where $(AB)_{\ell,m} = \sum_{j=0}^{n-1} A_{\ell,j}B_{j,m}$; this is the inner product of the ℓth row of A times the mth column of B. Now again fix an error bound $\epsilon_1 > 0$. Let

$$k = O(\log((\|A\|_\infty + \|B^T\|_\infty)/\epsilon_1),$$

where $\|A\|_\infty$ is the maximum, for any row of A, of the sum of the moduli of elements of the row, and $\|B^T\|_\infty$ is the maximum, for any column of B, of the sum of the moduli of elements of the column. We can precompute approximation polynomials of degree $k-1$ and their associated prefix sums for each of the rows of A and each of the columns of B. That is, we define an $n \times (Hk)$ matrix $\widetilde{A} =$ ROWS-COMPRESS$_k(A)$ and an $(Hk) \times n$ matrix $\widetilde{B} =$ COLUMNS-COMPRESS$_k(B)$ such that for each $\ell = 0, \ldots, n-1$ we define row $\widetilde{A}_\ell =$ ROW-COMPRESS$_k(A_\ell)$ (where $A_\ell, \widetilde{A}_\ell$ denote the ℓth row vectors of matrices $A, \widetilde{A}$, respectively) and for each $m = 0, \ldots, n-1$ we define column $\widetilde{B}_{-,m} =$ COLUMN-COMPRESS$_k(B_{-,m})$ (where $B_{-,m}, \widetilde{B}_{-,m}$ denote the mth columns of matrices $B, \widetilde{B}$, respectively). Then by Lemma 5.3, $\widetilde{A}_\ell\widetilde{B}_{-,m}$ is an ϵ-approx to $A_\ell B_{-,m}$. Hence, for each $0 \le \ell, m < n$, $(\widetilde{A}\widetilde{B})_{\ell,m}$ is an ϵ-approx to $(AB)_{\ell,m}$. Hence we have the following lemma.

LEMMA 5.6. *Suppose there is a (k, Δ) circle ϵ-approx evaluation scheme. Then given two $n \times n$ matrices A, B, we can ϵ_1-approximate the inner product AB by $\widetilde{A}\widetilde{B}$ for*

$$\epsilon_1 \le \epsilon(\epsilon + |\mathbf{u}| + |\mathbf{v}|).$$

PROPOSITION 5.7. *An $n \times n$ matrix of rank $< n$ cannot ϵ_1-approximate an $n \times n$ identity matrix, for any $\epsilon_1 < 1$.*

Proof. Suppose, for the sake of contradiction, that an $n \times n$ matrix $\widetilde{M}$ of rank $< n$ is an ϵ_1-approx of $n \times n$ matrix I, so $\|\widetilde{M} - I\|_\infty \le \epsilon_1$. Then since $\widetilde{M}$ has rank $< n$, there is an $x \ne 0$ such that $\|x\|_\infty = 1$ and $\widetilde{M}x = 0$. Thus

$$\|\widetilde{M}x - Ix\|_\infty = \|Ix\|_\infty = \|x\|_\infty = 1,$$

so

$$\|\widetilde{M} - I\|_\infty \geq \|\widetilde{M}x - Ix\|_\infty / \|x\|_\infty$$

$$= 1 > \epsilon_1,$$

a contradiction. □

We now consider the case where A, B are $n \times n$ identity matrices. Their product $M = AB$ is also an identity matrix. But $\widetilde{M} = \widetilde{A}\widetilde{B}$ is the product of an $n \times (kH)$ matrix and a $(kH) \times n$ matrix and thus has rank $\leq kH$. Recall that the statement of Theorem 5.1 makes the assumption that $kH = k\lceil 2\pi/\Delta \rceil < n$, so $\widetilde{M}$ has rank $< n$. But Lemma 5.6 states that $\widetilde{M}$ is an ϵ_1-approximation to $I = AB$, a contradiction of Proposition 5.7. Hence Theorem 5.1 follows. □

Appendix A. Proof of an alternative approximate evaluation on a circle.

Proof of Theorem 4.5. We give a proof of this alternative construction in stages, first considering a somewhat more complex algorithm.

An approach to approximate evaluation on a circle via approximate real polynomial evaluation. We will derive and prove the alternative algorithm given in subsection 4.2 by the use of classic real polynomial approximation techniques which provide our error analysis, but we observe at the end of this subsection that we do not need to explicitly construct the real approximation polynomials. Restrict z to the circle of radius r, so $z = r(x + iy)$, where x, y are real and $x + iy$ is on the unit circle, so $y = \sqrt{1 - x^2}$. Note that an expansion of $P(z)$ in terms of x, y gives $P(z) = R(x) + iS(x)y$, for polynomials $R(x), S(x)$ of degree $n - 1$. For simplicity, let us assume, w.l.o.g., $P(z)$ has real coefficients $p_0, \ldots, p_{n-1} \in \mathcal{R}$, so $R(x), S(x)$ are real polynomials. The exact evaluation of $P(z)$ at the evaluation points z_j, for $j = 0, \ldots, n-1$, might be done by an exact evaluation of $R(x), S(x)$ at the real points $x_0, \ldots, x_{n-1}$, where $z_j = x_j + iy_j$, so $P(z_j) = R(x_j) + iS(x_j)y_j$. Instead, we consider (see also Bini and Pan [3]) an approach for an ϵ-approximate evaluation of $P(z)$ at the evaluation points $z_0, \ldots, z_{m-1}$ using approximation of real polynomials (we will later show we can avoid explicit construction of these real polynomials). Fix a number k which divides n, to be determined below. We could construct degree $k-1$ polynomials $\widetilde{R}(x), \widetilde{S}(x)$, which give an ϵ'-approx of polynomials $R(x), S(x)$, where $\epsilon' = \epsilon/\sqrt{2}$. Then $\widetilde{P}(z) = \widetilde{R}(x) + i\widetilde{S}(x)y$ is an ϵ-approx of $P(z)$ over the circle of radius r, since

$$|P(z) - (\widetilde{R}(x) + i\widetilde{S}(x)y)| \leq \sqrt{(R(x) - \widetilde{R}(x))^2 + (S(x) - \widetilde{S}(x))^2} \leq \sqrt{2}\epsilon' = \epsilon.$$

To construct $\widetilde{R}(x), \widetilde{S}(x)$, we could do an exact evaluation of $R(x), S(x)$ at the shifted $(2k, k)$-Chebyshev points $x'_j = \Delta cos(j\pi/k)$, for $j = 0, \ldots, k-1$ over the real interval $I = [-\Delta, \Delta]$, for a positive real $\Delta \leq 1$. To do this first step, we can exactly evaluate $P(z)$ at

$$z'_j = r\left(x'_j + i\sqrt{1 - (x'_j)^2}\right),$$

for $j = 0, \ldots, k-1$, which costs work $O(n \log k)$ by Proposition 1.4. Then, we could exactly interpolate a degree $k-1$ real polynomial $\widetilde{R}(x)$ from the real parts $R(x'_j)$ of $P(z'_j)$, for $j = 0, \ldots, k-1$. We could also interpolate a degree $k-1$ real polynomial $\widetilde{S}(x)$ from the real values $s_0, s_1, \ldots, s_{k-1}$, where

$$is_j\sqrt{1 - (x'_j)^2}$$

is the complex part of $P(z'_j)$. By Lemma 3.1, the exact interpolation of polynomials $\widetilde{R}(x), \widetilde{S}(x)$ at these $(2k, k)$-Chebyshev points cost work $O(k \log k)$.

Exponential coefficient growth: The key difficulty with this approach. Next we need to bound the approximation errors for evaluation of $\widetilde{R}(x), \widetilde{S}(x)$ at all the evaluation points. To apply Proposition 3.2 we need to upper bound, the maximum moduli of $R^{(k)}(x) = \frac{d^k R(x)}{d^k x}$ (the kth derivative of $R(x)$ with respect to x), and $S^{(k)}(x) = \frac{d^k S(x)}{d^k x}$ over the interval $I = [-\Delta, \Delta]$. Unfortunately, straightforward application of the binomial expansion (in terms of x and y) to bound the moduli of coefficients of $R(x)$ and $S(x)$ give bounds on $|R|, |S|$ (the sums of the moduli of the coefficients of $R(x), S(x)$) that grow exponentially with the degree $n-1$. Thus application of Lemma 3.4 to evaluate $\widetilde{R}(x), \widetilde{S}(x)$ at all the evaluation points gives the work bound of the form

$$O(n \log \log(|R|/\epsilon')) = O(n \log(n + \log(1/\epsilon'))),$$

since in this case

$$\log(2^n/\epsilon') = n + \log(1/\epsilon').$$

The same difficulty occurs for ϵ'-approximate evaluation of the polynomial $S(x)$.

Evaluation at restricted intervals. To avoid this difficulty, we will do only an ϵ-approximate evaluation of $P(z)$ at a restricted interval on the radius r circle, where $|\cos(\theta(z))| \leq \Delta$ or, for the case of descending input coefficients,

$$|x| \leq \frac{1}{\sqrt{2}} \leq \Delta = \frac{1}{\max(\sqrt{2}, 8(n-1-k)/\sqrt{2})}.$$

Then we can do an ϵ-approximate evaluation of $P(z)$ over the entire circle of radius r by a number of separate evaluations of $P(z)$, each at a consecutive segment of the circle. The following lemma provides useful bounds on the k derivatives of $R(x), S(x)$.

LEMMA A.1. *Suppose $r = 1$ and either 1. $\Delta \leq \frac{k}{e(n-k)}$ or 2. $P(z)$ is $(k, \sqrt{2}/8)$-descending. For*

$$|x| \leq \Delta \leq \frac{1}{\sqrt{2}},$$

the modulus of the kth derivative (with respect to x) of $z = r(x + iy)$ (where $x + iy$ is on the unit circle) is bounded as

$$\left|\frac{d^k z}{d^k x}\right| \leq O(rk!8^k).$$

Also,

$$\left|\frac{d^k P(z)}{d^k x}\right|, \left|R^{(k)}(x)\right|,$$

and $|S^{(k)}(x)|$ are bounded in case 1 by $O(k!|P|\Delta^{-k})$ and in case 2 by $O(k!n^{O(1)}|P|\Delta^{-k})$.

Proof. We shall show that $\frac{dz}{dx} = r(1 - if_1(x))$ and, for $k \geq 2$, $\frac{d^k z}{d^k x} = -ir f_k(x)$, where $f_k(x)$ is a sum of at most 2^{k-1} terms, each of the form $c(k)\frac{x^a}{y^{2b+1}}$, where $|c(k)| \leq O(k!2^k)$ is the term's coefficient, and $0 \leq a, b \leq k$ are integers. We prove this by

construction of a *derivative tree*, which is a binary tree whose nodes are such terms. Since $y = \sqrt{1-x^2}$, $\frac{dy}{dx} = \frac{-x}{y^3}$, we have $\frac{dz}{dx} = 1 - if_1(x)$ where $f_1(x) = \frac{x}{y^3}$, so we define the root to be $f_1(x)$. Note that for $k > 1$,

$$\frac{d^k z}{d^k x} = -i\frac{d^{k-1} f_1(x)}{d^{k-1} x}.$$

Inductively, assume the label of a node is a term of form $c(k)\frac{x^a}{y^{2b+1}}$ labeling a node, since the derivative of $c(k)\frac{x^a}{y^{2b+1}}$ with respect to x is the sum of terms $c(k)(2b+1)$ $\frac{x^a}{y^{2(b+1)+1}}$ and $c(k)a\frac{x^{a-1}}{y^{2b+1}}$, we let its left and right children be labeled by these two terms, respectively. Note that since we take k derivatives, the tree has depth $k-1$, and so there are at most 2^{k-1} terms at the leaves of the tree. The leaves will be the terms of $\frac{d^{k-1} f_1(x)}{d^{k-1} x}$. The leaf with largest coefficient modulus is reached by a length $k-1$ path of left branches down the tree, so the maximum coefficient modulus of any leaf can be upper bounded as

$$|c(k)| \leq (2k-1)(2k-3)\cdots 3 \cdot 1 \leq k! 2^{k-1}.$$

Since $|x| \leq \frac{1}{\sqrt{2}}$,

$$y \geq \sqrt{1-x^2} \geq 1/\sqrt{2},$$

so

$$1/y^{2b+1} \leq (\sqrt{2})^{2k+1} \leq 2^{k+1}.$$

Thus the modulus of each leaf term $c(k)\frac{x^a}{y^{2b+1}}$ is upper bounded by $|c(k)|2^{k+1} \leq k!2^{k-1}2^{k+1}\ O(k!4^k)$. Hence $|f_k(x)| \leq$ the product of the number 2^{k-1} of terms at the leaves times their maximum modulus $O(k!4^k)$, so $|f_k(x)| \leq O(k!8^k)$. This implies

$$\left|\frac{d^k z}{d^k x}\right| \leq r(1 + |f_k(x)|) \leq O(rk!8^k).$$

Note that for $j > k$,

$$\frac{j!}{(j-k)!} \approx (j-k)^k$$

for $k = o(n)$, since by the Stirling approximation to factorial,

$$j! \approx (j)^j e^{-j}\sqrt{2\pi j},$$

$$(j-k)! \approx (j-k)^{j-k} e^{-(j-k)}\sqrt{2\pi(j-k)},$$

and

$$\left(\frac{j}{j-k}\right)^{j-k} = \left(1 + \frac{k}{j-k}\right)^{j-k} \approx e^k;$$

so

$$\frac{j!}{(j-k)!} \approx \frac{j^j}{(j-k)^{j-k}} e^{-k} \approx (j-k)^k e^{-k} e^k \approx (j-k)^k.$$

Since $|z| \le r$, we have $|\frac{d^k P(z)}{d^k z}| \le \sum_{j=k}^{n-1} \frac{j!}{(j-k)!} r^{j-k} |p_j|$. By the chain rule of derivatives,

$$\frac{d^k P(z)}{d^k x} = \frac{d^k z}{d^k x} \frac{d^k P(z)}{d^k z},$$

so

$$\left| \frac{d^k P(z)}{d^k x} \right| \le \left| \frac{d^k z}{d^k x} \right| \left| \frac{d^k P(z)}{d^k z} \right| \le O(rk!8^k) \sum_{j=k}^{n-1} \frac{j!}{(j-k)!} r^{j-k} |p_j|.$$

Thus

$$\left| \frac{d^k P(z)}{d^k x} \right| \le O(rk!8^k) \sum_{j=k}^{n-1} \frac{j!}{(j-k)!} r^{j-k} |p_j| \le O(rk!8^k) \sum_{j=k}^{n-1} (j-k)^k r^{j-k} |p_j|.$$

1. Now suppose $r = 1$ and $\Delta \le \frac{1}{8(n-k)}$. Then $\sum_{j=k}^{n-1} (8(j-k))^k |p_j| \le |P| \Delta^{-k}$, so

$$(8(j-k))^k |p_j| \le |P| (1/\sqrt{2})^k \le O(|P| \Delta^{-k}).$$

   Thus,

$$\left| \frac{d^k P(z)}{d^k x} \right| \le O(rk!8^k) \sum_{j=k}^{n-1} (j-k)^k r^{j-k} |p_j| \le O(k!) \sum_{j=k}^{n-1} (8(j-k))^k r^{j-k+1} |p_j|$$

$$\le O(k! |P| \Delta^{-k}).$$

   Since $P(z) = R(x) + iS(x)y$, we have

$$\frac{d^k P(z)}{d^k x} = R^{(k)}(x) + i \left(S^{(k)}(x) y + \frac{x S(x)}{y} \right).$$

   Hence for $|x| \le \Delta$,

$$|R^{(k)}(x)| \le \left| \frac{d^k P(z)}{d^k x} \right| \le O(k! |P| \Delta^{-k}).$$

   Also, $|\frac{xS(x)}{y}| \le |P(z)| \le |P|$, so

$$\begin{aligned} |S^{(k)}(x)| &\le \left| \frac{d^k P(z)}{d^k x} \right| + \left| \frac{x S(x)}{y} \right| \le \left| \frac{d^k P(z)}{d^k x} \right| + |S(x)| \left| \frac{x}{y} \right| \\ &\le \left| \frac{d^k P(z)}{d^k x} \right| + O(|P|) \\ &\le O(k! |P| \Delta^{-k}). \end{aligned}$$

2. Next suppose $r = 1$. If $P(z)$ is $(k, \sqrt{2}/8)$-descending, then, by definition, the magnitude of the coefficients of $P(z)$ drops as

$$|p_j| \le n^{O(1)} |P| \left(\frac{\sqrt{2}}{j8} \right)^k$$

for $j = k, \ldots, n-1$. Then $\sum_{j=k}^{n-1}(8(j-k))^k|p_j| \leq |P|\Delta^{-k})$, so

$$(8(j-k))^k|p_j| \leq n^{O(1)}|P|(1/\sqrt{2})^k \leq O(n^{O(1)}|P|\Delta^{-k}).$$

Thus in this case,

$$\left|\frac{d^k P(z)}{d^k x}\right| \leq O(rk!8^k)\sum_{j=k}^{n-1}(j-k)^k|p_j| \leq O(k!)\sum_{j=k}^{n-1}(8(j-k))^k|p_j|$$
$$\leq O(k!n^{O(1)}|P|\Delta^{-k}).$$

Also,

$$|S^{(k)}(x)| \leq O(k!n^{O(1)}|P|\Delta^{-k}).$$

Since $|I| = 2\Delta$, for $I = [-\Delta, \Delta]$, by Propositions 3.2 and A.1, the error in the approximation of $R(x)$ by $\widetilde{R}(x)$, for $|x| \leq \Delta$, is upper bounded in case 1 by

$$\frac{2}{k!}(|I|/4)^k \max_{x\in I} R^{(k)}(x) \leq O(|P|(\Delta/2)^k\Delta^{-k}) \leq O(|P|2^{-k})$$

and in case 2 by $O(n^{O(1)}|P|2^{-k})$. This error is upper bounded by $O(\epsilon')$ if we set $k = \log(|P|/\epsilon) + O(1)$ in case 1 or $k = O(\log(n|P|/\epsilon))$ in case 2. Also, the error in the approximation of $S(x)$ by $\widetilde{S}(x)$, for $|x| \leq \Delta$, is upper bounded by

$$\frac{2}{k!}(|I|/4)^k \max_{y\in I} S^{(k)}(y) \leq \epsilon'.$$

Hence

$$|P(z) - (\widetilde{R}(x) + i\widetilde{S}(x)y)| \leq ((R(x) - \widetilde{R}(x))^2 + (S(x) - \widetilde{S}(x))^2)^{1/2} \leq \sqrt{2}\epsilon' = \epsilon.$$

Thus we have proved the error bounds given in Theorem 4.5 for the case

$$\widetilde{P}(z) = \widetilde{R}(x) + i\widetilde{S}(x)y.$$

A simplification avoiding construction of real polynomials. Finally, we observe that we do not need to explicitly construct real approximation polynomials defined above. For simplicity, we assume, w.l.o.g., for each evaluation point z_j that $|\theta(z_j) - \pi/2| \leq \pi/4$. Instead, we do the following:

- In work $O(n\log^2 k)$, interpolate the degree $k-1$ complex polynomial $\widetilde{P}(z)$ from the values $P(z'_j)$, for $j = 0, \ldots, k-1$, where $z'_j = r(x'_j + i\sqrt{1-(x'_j)^2})$, and $x'_j = \cos(j\pi/k)/\sqrt{2}$, for $j = 0, \ldots, k-1$.
  Comment: Thus we construct $\widetilde{P}(z) = (\widetilde{R}(x) + i\widetilde{S}(x)y)$, without explicitly constructing $\widetilde{R}(x), \widetilde{S}(x)$.
- Evaluate polynomial $\widetilde{P}(z)$ exactly at the given evaluation points $z_0, \ldots, z_{m-1}$, within work $O(m\log^c k)$, where $c = 2$ by Proposition 1.1 (or $c = 1$ by Proposition 1.4 if these evaluation points are regularly spaced).
  Comment: Thus we evaluate $\widetilde{P}(z) = (\widetilde{R}(x)+i\widetilde{S}(x)y)$, at the given evaluation points, without explicitly evaluating $\widetilde{R}(x), \widetilde{S}(x)$.

Thus we have derived the algorithm as defined in subsection 4.2, which provides a somewhat simpler implementation of Theorem 4.5. ∎

Appendix B. Exact Chebyshev point evaluation in $O(n \log n)$ work.

Let $P(x)$ be a polynomial of degree $n-1$. Let $\omega = e^{i2\pi/n'}$ be the n'th root of unity over the complex numbers $\mathcal{C}$, for some $n' \geq n$, such that n divides n'. We wish to exactly evaluate $P(x)$ at the (n', n)-Chebyshev points, which we defined to be the real parts $x_{0,j} = \cos(j2\pi/n')$ of the powers $z_{0,j} = \omega^j = e^{ji2\pi/n'}$, for $j = 0, \ldots, n-1$.

For simplicity, we assume n is a power of 2 and $n' = 2n$. For each $\ell = 0, \ldots, \log n$,

- let $n_\ell = n/2^\ell, n'_\ell = n'/2^\ell$,
- let $z_{\ell,j+n_\ell} = -z_{\ell,j}$, and if $\ell > 0$ then let $z_{\ell,j} = 2z^2_{\ell-1,j} - 1$, and
- let $x_{\ell,j} = \text{ real}(e^{ji2\pi/n'_\ell}) = \cos(j2\pi/n'_\ell)$ denote the (n'_ℓ, n_ℓ)-Chebyshev points.

By the identities $\cos(2\theta) = 2\cos^2(\theta) - 1$ and $\cos(\theta + \pi) = -\cos(\theta)$, we have a recursive definition of these (n'_ℓ, n_ℓ)-Chebyshev points.

PROPOSITION B.1. *For each $\ell = 0, \ldots, \log n$ and all $j = 0, \ldots, n_\ell - 1$, $x_{\ell,j+n_\ell} = -x_{\ell,j}$ and if $\ell > 0$ then $x_{\ell,j} = 2x^2_{\ell-1,j} - 1$.*

The key idea behind our algorithm is as follows:

- Starting with the polynomial P to be evaluated, we define two polynomials P_0 and P_1 as the unique polynomials such that $P(x) = P_0(2x^2-1) + xP_1(2x^2-1)$.
- The map from x to $2x^2 - 1$ sends $\cos\theta$ to $\cos 2\theta$, that is, it maps Chebyshev points to Chebyshev points. If the original set of Chebyshev points contains both x and $-x$, then this map collapses them, so the number of points to be evaluated is halved. Thus the problem of evaluating P at the (n', n)-Chebyshev points reduces to evaluating P_0 and P_1 at $(n'/2, n/2)$-Chebyshev points.
- To apply this argument recursively, we compute the polynomials at each node of a logarithmic depth tree, by their evaluations on suitably defined points. This process can be initiated at the leaves, since there evaluation is the same as the coefficient itself.

We define a full binary tree of depth $\log n$ which we call the (n', n)-*Chebyshev evaluation tree*, whose nodes of depth $\ell = 0, \ldots, \log n$ are the set $\{0,1\}^\ell$ of binary strings of length ℓ. The root is labeled with the input polynomial $P_\lambda(x) = P(x)$, where λ is the empty string. Each node $\beta \in \{0,1\}^\ell$ of the tree is labeled with a degree $n_\ell - 1$ polynomial $P_\beta(x)$ and if $\ell < \log n$, node β has two children $\beta 0, \beta 1$ with labels consisting of the unique degree $n_{\ell+1} - 1$ polynomials $P_{\beta 0}(x), P_{\beta 1}(x)$ such that $P_\beta(x) = P_{\beta 0}(2x^2 - 1) + xP_{\beta 1}(2x^2 - 1)$. In contrast with the $O(n \log^2 n)$ algorithm of Pan [29], we will not actually compute these polynomials, but they will aid us in definition of our algorithm. However, we will compute for each node $\beta \in \{0,1\}^\ell$ two sets of values

$$\{F_{\beta,j} = P_\beta(z_{\ell,j}) | j = 0, \ldots, n_\ell - 1\}$$

and

$$\{\phi_{\beta,j} = P_\beta(x_{\ell,j}) | j = 0, \ldots, n_\ell - 1\}.$$

PROPOSITION B.2. *For each $\beta \in \{0,1\}^{\log n}$, $\phi_{\beta,0} = F_{\beta,0}$. For each $\ell = 0, \ldots, (\log n) - 1$, $\beta \in \{0,1\}^\ell$ and $j = 0, \ldots, n_\ell - 1$,*

1. *$F_{\beta,j} = F_{\beta 0,j} + z_{\ell,j}F_{\beta 1,j}$, $F_{\beta,j+n_\ell} = F_{\beta 0,j} - z_{\ell,j}F_{\beta 1,j}$, and so*
2. *$F_{\beta 0,j} = (F_{\beta,j} + F_{\beta,j+n_\ell})/2$, $F_{\beta 1,j} = (F_{\beta,j} - F_{\beta,j+n_\ell})/(2z_{\ell,j})$, and similarly*
3. *$\phi_{\beta,j} = \phi_{\beta 0,j} + x_{\ell,j}\phi_{\beta 1,j}$, $\phi_{\beta,j+n_\ell} = \phi_{\beta 0,j} - x_{\ell,j}\phi_{\beta 1,j}$, and so*

4. $\phi_{\beta 0,j} = (\phi_{\beta,j} + \phi_{\beta,j+n_\ell})/2$, $\phi_{\beta 1,j} = (\phi_{\beta,j} - \phi_{\beta,j+n_\ell})/(2x_{\ell,j})$.

Proof. For $\ell = |\beta| = \log n$, since P_β is a degree 0 polynomial, we have $\phi_{\beta,0} = P_\beta = F_{\beta,0}$.

1. By definition, for $\ell \geq 0$, $F_{\beta,j} = P_\beta(z_{\ell,j}) = P_{\beta 0}(z_{\ell+1,j}) + z_{\ell,j}P_{\beta 1}(z_{\ell+1,j}) = F_{\beta 0,j} + z_{\ell,j}F_{\beta 1,j}$. Also by definition, $z_{\ell,j+n_\ell} = -z_{\ell,j}$, so $F_{\beta,j+n_\ell} = P_\beta(z_{\ell,j+n_\ell}) = P_\beta(-z_{\ell,j}) = P_{\beta 0}(2(-z_{\ell,j})^2 - 1) - z_{\ell,j}P_{\beta 1}(2(-z_{\ell,j})^2 - 1) = P_{\beta 0}(z_{\ell+1,j}) - z_{\ell,j}P_{\beta 1}(z_{\ell+1,j}) = F_{\beta 0,j} - z_{\ell,j}F_{\beta 1,j}$.
2. The reverse formulas for $F_{\beta 0,j}, F_{\beta 1,j}$ follow from linear solution of the previous two equations.
3. By definition, for $\ell \geq 0$, $\phi_{\beta,j} = P_\beta(x_{\ell,j}) = P_{\beta 0}(x_{\ell+1,j}) + x_{\ell,j}P_{\beta 1}(x_{\ell+1,j}) = \phi_{\beta 0,j} + x_{\ell,j}\phi_{\beta 1,j}$. By Proposition B.1, $x_{\ell,j+n_\ell} = -x_{\ell,j}$, so $\phi_{\beta,j+n_\ell} = P_\beta(x_{\ell,j+n_\ell}) = P_\beta(-x_{\ell,j}) = P_{\beta 0}(2(-x_{\ell,j})^2 - 1) - x_{\ell,j}P_{\beta 1}(2(-x_{\ell,j})^2 - 1) = P_{\beta 0}(x_{\ell+1,j}) - x_{\ell,j}P_{\beta 1}(x_{\ell+1,j}) = \phi_{\beta 0,j} - x_{\ell,j}\phi_{\beta 1,j}$.
4. The reverse formulas for $\phi_{\beta 0,j}, \phi_{\beta 1,j}$ follow from linear solution of the previous two equations. □

Our algorithms for exact Chebyshev point evaluation and interpolation. We now show how to do (n', n)-Chebyshev point evaluation by a recursive algorithm and reduction to the DFT.

For each $\ell = 0, \ldots, (\log n) - 1$, we precompute $x_{\ell,j} = \cos(j2\pi/n')$, and $z_{\ell,j}$ for $j = 0, \ldots, n_\ell - 1$ as defined above. We then compute the $(2n, n)$-Chebyshev point evaluation of a given $P(x)$ as follows:

$(2n, n)$-Chebyshev point evaluation algorithm.

Input: The n coefficients $p_0, \ldots, p_{n-1} \in \mathcal{C}$ polynomial $P(x)$ of degree $\leq n - 1$.

1. In work $O(n \log n)$, compute the values $F_{\lambda,j} = P(\omega^j), j = 0, \ldots, n - 1$, where λ is the empty string (this can be done as stated in Proposition 1.3 within work $O(n \log n)$ by the chirp transform, which generalizes the DFT to this case).
2. Traverse the Chebyshev evaluation tree from root to leaves:
   For each $\ell = 0, \ldots, (\log n) - 1$ **do** apply Proposition B.2, part 2:
   For each $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0, 1\}^\ell$ **do**
   let $F_{\beta 0,j} = (F_{\beta,j} + F_{\beta,j+n_\ell})/2$, $F_{\beta 1,j} = (F_{\beta,j} - F_{\beta,j+n_\ell})/(2z_{\ell,j})$.
3. **For each** $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0, 1\}^\ell$ **do**
   let $\phi_{\beta,0} = F_{\beta,0}$, as given by Proposition B.2.
4. Traverse the Chebyshev evaluation tree from leaves to root:
   For each $\ell = (\log n) - 1, \ldots, 0$ **do** apply Proposition B.2, part 3:
   For each $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0, 1\}^\ell$ **do**
   let $\phi_{\beta,j} = \phi_{\beta 0,j} + x_{\ell,j}\phi_{\beta 1,j}$, $\phi_{\beta,j+n_\ell} = \phi_{\beta 0,j} - x_{\ell,j}\phi_{\beta 1,j}$.

Output: $\phi_{\lambda,j} = P(x_{\lambda,j}), j = 0, \ldots, n - 1$, which is the $(2n, n)$-Chebyshev evaluation of $P(x)$.

For each $\ell = 0, \ldots, \log n$, the work is $O(n_\ell) = O(n/2^\ell)$ for each of the 2^ℓ nodes $\beta \in \{0, 1\}^\ell$ of depth ℓ, so the total work is $O(n_\ell 2^\ell \log n) = O(n \log n)$, plus $O(n \log n)$ work for computing the chirp transform by Proposition 1.3.

We can apply this $(2n, n)$-Chebyshev point evaluation algorithm to compute an (n', n)-Chebyshev point evaluation of degree $n-1$ polynomial $P(x)$, for any $n' = O(n)$ such that n divides n'. In the case $n' \geq 2n$, we can view the polynomial $P(x)$ to be of degree $(n'/2) - 1$, and apply the $(n', n'/2)$-Chebyshev point evaluation algorithm. In the case $n' < 2n$, since n divides n', we can decompose $P(x) = \sum_{j=0}^{c-1} P_j(x)x^{jn}$, where $c = n'/(2n) = O(1)$ and each polynomial $P_j(x)$ is of degree $\leq (n/c) - 1$, and then apply the $(2n/c, n/c)$-Chebyshev point evaluation algorithm a constant c number of

times.

Thus we have proved that the (n', n)-Chebyshev point evaluation problem for a polynomial of degree $n-1$ can be solved within work $O(n \log n)$ for any $n' = O(n)$ such that n divides n'. Since the Chebyshev evaluation tree has depth $\log n$, and there are $2 \log n$ stages, each taking $O(1)$ time with n work, and the DFT takes $O(\log n)$ with $O(n \log n)$ work, it follows that the total circuit depth is $O(\log n)$ with $O(n \log n)$ work. This completes the first part of the proof of Lemma 3.1. We next show how to do (n', n)-Chebyshev point interpolation by reduction to the DFT. We compute the interpolation of a polynomial from a set of n values at $(2n, n)$-Chebyshev points, as follows:

$(2n, n)$-Chebyshev point interpolation algorithm.

Input: The values $v_0, \ldots, v_{n-1} \in \mathcal{C}$ at the $(2n, n)$-Chebyshev points: $x_{0,0}, \ldots, x_{0,n-1}$.

1. Let $\phi_{\lambda,j} = v_j), j = 0, \ldots, n-1$.
2. Traverse the Chebyshev evaluation tree from root to leaves:
   For each $\ell = 0, \ldots, (\log n) - 1$ **do** apply Proposition B.2, part 4:
   For each $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0,1\}^\ell$ **do**
   let
$$\phi_{\beta 0,j} = (\phi_{\beta,j} + \phi_{\beta,j+n_\ell})/2,$$
$$\phi_{\beta 1,j} = (\phi_{\beta,j} - \phi_{\beta,j+n_\ell})/(2x_{\ell,j}).$$
3. **For each** $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0,1\}^\ell$ **do**
   let $F_{\beta,0} = \phi_{\beta,0}$, as given by Proposition B.2.
4. Traverse the Chebyshev evaluation tree from leaves to root:
   For each $\ell = (\log n) - 1, \ldots, 0$ **do** apply Proposition B.2, part 1:
   For each $j = 0, \ldots, n_\ell - 1$ and $\beta \in \{0,1\}^\ell$ **do**
   let
$$F_{\beta,j} = F_{\beta 0,j} + z_{\ell,j} F_{\beta 1,j},$$
$$F_{\beta,j+n_\ell} = F_{\beta 0,j} - z_{\ell,j} F_{\beta 1,j}.$$
5. In work $O(n \log n)$, compute the inverse DFT of the values $F_{\lambda,j} = P(\omega^j), j = 0, \ldots, n-1$ (this can be done within work $O(n \log n)$ as stated in Proposition 1.3), giving:

Output: The n coefficients of the unique interpolated polynomial $P(x)$.

For each $\ell = 0, \ldots, \log n$, the work is again $O(n_\ell) = O(n/2^\ell)$ for each of the 2^ℓ nodes $\beta \in \{0,1\}^\ell$ of depth ℓ, so the total work is again $O(n_\ell 2^\ell \log n) = O(n \log n)$, plus $O(n \log n)$ work for computing the DFT as in Proposition 1.3.

We can also apply this $(2n, n)$-Chebyshev point interpolation algorithm to compute an (n', n)-Chebyshev point interpolation of degree $n-1$ polynomial $P(x)$ for any $n', n \le n' < 2n$, such that n divides n'. We apply the $(2\bar{n}, \bar{n})$-Chebyshev point interpolation algorithm, for $\bar{n} = n'/c$, to interpolate c polynomials of degree $\bar{n}-1$ from the given (n', n)-Chebyshev points, and then construct $P(x)$ in further work $O(n \log n)$ by the well-known divide-and-conquer interpolation methods of [16, 26, 5, 1]. Thus we have proved that the (n', n)-Chebyshev point interpolation problem for a polynomial of degree $n-1$ can be solved within work $O(n \log n)$ for any $n' \le 2n$ such that n divides n'. The shifted version of these evaluation and interpolation problems can

be solved within the same work bounds using the same techniques by initializing the recurrences at the leaves to the appropriate values corresponding to the shifts. Again, since the Chebyshev evaluation tree has depth $\log n$, and there are $2 \log n$ stages, each taking $O(1)$ time with n work, and the inverse DFT takes $O(\log n)$ with $O(n \log n)$ work, it follows that the total circuit depth is $O(\log n)$ with $O(n \log n)$ work. To complete the proof of Lemma 3.1, we apply the technique of Proposition 1.1 to solve the (n', m)-Chebyshev point evaluation problem, for $n' = O(n)$, in work $O(n + m \log \min(n, m))$. □

Acknowledgments. The author sincerely thanks Chris Lambert, Victor Pauca, Kathy Redding, Ken Robinson, Steven Tate, and especially Deganit Armon, Octavian Procopiuc, and D. Sivakumar for comments improving the presentation of the results in this paper. Don Coppersmith suggested the reduction to the approximation of a matrix by small rank used in the proof of Theorem 5.1. Donald Rose described to us a simple proof of Proposition 5.7.

REFERENCES

[1] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA, London, 1974.
[2] A. V. Aho, K. Steiglitz, and J. D. Ullman, *Evaluating Polynomials at Fixed Sets of Points*, SIAM J. Comput., 4 (1975), pp. 533–539.
[3] D. Bini and V. Pan, *Polynomial and Matrix Computations*, 1, Birkhauser, Boston, MA, 1994.
[4] A. B. Borodin and I. Munro, *Evaluating polynominals at many points*, Inform. Process. Lett., 1 (1971), pp. 660–68.
[5] A. B. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numerical Problems*, American Elsevier, New York, 1975.
[6] P. B. Callahan and S. R. Kosaraju, *A decomposition of multi-dimensional point sets with applications to k-nearest-neighbors and n-body potential fields*, J. ACM, 42 (1995), pp. 67–90.
[7] J. Carrier, L. Greengard, and V. Rokhlin, *A fast adaptive multipole algorithm for particle simulations*, SIAM J. Sci. Comput., 9 (1998), pp. 669–686.
[8] J. M. Cooley, P. A. Lewis, and P. D. Welch, *History of the fast Fourier transform*, Proc. IEEE, 55 (1967), pp. 1675–1677.
[9] J. M. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comp., 19 (1965), pp. 297–301.
[10] G. Dahlquist and A. Björck, *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[11] G. C. Danielson and C. Lanczos, *Some improvements in practical Fourier analysis and their application to X-ray scattering from liquids*, J. Franklin Inst., 233 (1942), pp. 365–380 and pp. 435–452.
[12] A. Dutt, M. Gu, and V. Rokhlin, *Fast algorithms for polynomial interpolation, integration, and differentiation*, SIAM J. Numer. Anal., 33 (1996), pp. 1689–1711.
[13] A. Dutt, *Fast Fourier Transforms for Nonequispaced Data*, Technical Report 980, Department of Computer Science, Yale University, New Haven, CT, 1993.
[14] A. Dutt and V. Rokhlin, *Fast Fourier Transforms for Nonequispaced Data II*, in Applied and Computational Harmonic Analysis, Academic Press, 2 (1995), pp. 85–100.
[15] W. D. Elliott and J. A. Board, *Fast fourier transform accelerated fast multipole algorithm*, SIAM J. Sci. Comput., 17 (1996), pp. 398–415.
[16] C. M. Fiduccia, *Polynomial evaluation via the division algorithm—the fast Fourier transform revisited*, in Proceedings of the 4th Annual ACM Symposium on Theory of Computing, 1972, pp. 88–93.
[17] W. M. Gentleman and G. Sande, *Fast Fourier transforms for fun and profit*, in Proceedings of the AFIPS 1966 Fall Joint Computer Conference, 29, 1966, pp. 563–578.
[18] A. Gerasoulis, *A fast algorithm for the multiplication of generalized Hilbert matrices with vectors*, Math. Comp., 50 (1988), pp. 179–188.
[19] A. Gerasoulis, M. D. Grigoriadis, and Liping Sun, *A fast algorithm for Trummer's problem*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. s135–s138.
[20] I. J. Good, *The interaction algorithm and practical Fourier series*, J. Roy. Statist. Soc. Ser.

A, 20 (1958), pp. 361–372. Addendum, 22 (1960), pp. 372–375.
[21] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.
[22] L. Greengard and V. Rokhlin, *On the efficient implementation of the fast multipole algorithm*, Technical Report RR-602, Dept. of Computer Science, Yale University, New Haven, CT, 1988.
[23] P. Henrici, *Applied and Computational Complex Analysis*, III, John Wiley, New York, 1986.
[24] M. S. Henry, *Approximation by polynomials: Interpolation and optimal nodes*, Amer. Math. Monthly, 91 (1984), pp. 497–499.
[25] E. Horowitz, *A fast method for interpolation using preconditioning*, Inform. Process. Lett., 1 (1972), pp. 157–163.
[26] H. T. Kung, *Fast evaluation and interpolation*, Department of Computer Science, Carnegie-Mellon University., Pittsburgh, PA, 1973, unpublished manuscript.
[27] R. Moenck and A. B. Borodin, *Fast modular transforms via division*, in Conf. Record, IEEE 13th Ann. Symposium on Switching and Automata Theory, 1972, pp. 90–96.
[28] A. C. R. Newbery, *Error analysis for polynomial evaluation*, Math. Comp., 28 (1979), pp. 789–793.
[29] V. Pan, *Fast Evaluation and interpolation at the Chebyshev sets of points*, Appl. Math. Lett., 2 (1989), pp. 255–258.
[30] V. Y. Pan, J. H. Reif, and S. R. Tate, *The power of combining the techniques of algebraic and numerical computing: Improved approximate multipoint polynomial evaluation and improved multipole algorithms*, in Proceedings of the 32nd Annual IEEE Symposium Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 703–713.
[31] V. Pan, A. Sadikou, E. Landowne, and O. Tiga, *A new approach to fast polynomial interpolation and multipoint evaluation*, Comput. Math. Appl., 25 (1993), pp. 25–30.
[32] L. R. Rabiner and C. M. Rader, eds., *Digital Signal Processing*, IEEE Press, New York, 1972.
[33] J. H. Reif and S. R. Tate, *Fast Spatial Decomposition and Closest Pair Computation for Limited Precision Input*, Technical Report N-96-001, Department of Computer Science, University of North Texas, Denton, TX, 1996; also available online from http://www.cs.duke.edu/∼reif/paper/Sep.ps.
[34] J. H. Reif and S. R. Tate, *N-Body Simulation* I: *Fast Algorithms for Potential Field Evaluation and Trummer's Problem*, Technical Report N-96-002, Department of Computer Science, University of North Texas, Denton, TX, 1996; also available online from http://www.cs.duke.edu/∼reif/paper/Multipole.ps.
[35] V. Rokhlin, *A fast algorithm for the discrete Laplace transformation*, J. Complexity, 4 (1988), pp. 12–32.
[36] C. Runge and H. König, *Die Grundlehren der mathematischen Wissenschaften*, 11, Springer, Berlin, 1924.