Approximating the quotient of two polynomials by means of evaluation and interpolation

by

Elliott Landowne

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1996

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Nov 14, 1996
Date

Victor Pan
Chair of Examining Committee

Nov 14, 1996
Date

Executive Officer

Prof. Stathis Zachos

Prof. Stephen Tate

Supervisory Committee

The City University of New York

Preface

This dissertation presents research concerning the derivation of an algorithm for approximating the quotient of two polynomials. In this preface, we motivate the need for careful study of solutions to the problem of polynomial division (with a remainder).

Polynomial division is known to play a central role in the area of algebraic algorithm design. A subroutine for polynomial division is included in every library and package of subroutines for algebraic/symbolic computation. A solution of this problem must be clearly understood by algorithm designers, as well as researchers in the theory of algebraic computations. It turns out that full analysis of this elementary operation inevitably includes rather sophisticated mathematics.

It is known that polynomial division is utilized in the solution of numeric problems, as well as algebraic problems. Yet the requirements of these areas are quite different. Algorithms which reduce the asymptotic complexity of an algebraic computation may introduce devastating numerical instability. Some points of this analysis can be rather subtle.

In another direction, the recent advances of parallel computer architecture require the re-examination of polynomial division from a new point of view, that is, designing most effective algorithms for parallel implementation. This problem also turned out to involve some additional advanced techniques of study.

In order to understand these issues when designing polynomial division subroutines for algebraic and numeric computations, it is important to thoroughly understand the algorithm derivations. We hope that the clarification of the approximation method, introduced by our research, will be especially useful in all these respects.

Acknowledgements

I would like to thank Prof. Stathis Zachos and Prof. Stephen Tate for serving on my examining committee, and Prof. Stanley Habib for his patience with me, as Executive Officer, during the writing of this dissertation.

I especially appreciate the guidance of my thesis advisor, Prof. Victor Pan, who helped shape so much of this work.

Contents

0 Introduction

0.1 Review of polynomial division algorithms

We briefly state the problem of polynomial division with a remainder: given the coefficients of two polynomials s and t over the field $\mathbf{C}$ of complex numbers, $\deg s = m$ and $\deg t = n$, compute the coefficients of two polynomials, quotient q and remainder r, $\deg q = m - n = K - 1$ and $\deg r < n$, such that

$$s(z) = q(z) \times t(z) + r(z). \tag{0.0}$$

Early work in improving on the (well-known) high school polynomial division algorithm culminated with the work of Sieveking and Kung in the early '70s. Their algorithms utilized Newton's method (in analogy to Cook's earlier integer division algorithm) and the fast Fourier transform, with total work $O(K \log K)$ and $O(\log^2 K)$ parallel time-complexity (we assume any model of parallel computation that supports computation of the fast Fourier transform with total work $O(K \log K)$ and $O(\log K)$ parallel time-complexity (see [BiniPan, ch. 4] for an introduction to parallel algorithms)).

Subsequent work has concentrated primarily on improving the bounds for parallel time-complexity. The best result achieved so far by modification of Newton's method is due to Bini and Pan (1990), where they improved the bound on parallel time-complexity to $O(\log K \log^* K)$ (define $\log^* K = \max(\ell | \log^{(\ell)} K \geq 0)$). Another approach, most closely related to our research, is due to [Bini] and [Schönhage]. Their algorithm computes an approximation to the quotient s/t, with $O(\log K)$ parallel time-complexity. By comparison, a variation of the algorithm of Bini and Pan has parallel time-complexity $O(\log K)$, but with total work $O(K \log K \log^* K)$.

We present a brief table to simplify the comparison of the algorithms that we have discussed (see [BiniPan, ch. 6] for a thorough survey of polynomial division algorithms).

polynomial division algorithm summary

method	total work	|| time-complexity
High school	$O(K \min(K,n))$	$O(K)$
Newton		
Sieveking, Kung	$O(K \log K)$	$O(\log^2 K)$
Bini & Pan	$O(K \log K)$	$O(\log K \log^* K)$
	$O(K \log K \log^* K)$	$O(\log K)$
Approximation		
Bini, Schönhage	$O(K \log K)$	$O(\log K)$

0.2 Summary of the dissertation

In section 1, we study multipoint evaluation and interpolation of polynomials at roots of 1, our basic algebraic method. The discrete Fourier transform (dft) is defined as multipoint polynomial evaluation at roots of 1, hence the inverse dft is seen as a polynomial interpolation formula ([BiniPan] is a good reference for this approach). We illustrate our algorithm derivation method by application to polynomial multiplication and division (with no remainder).

The core of the dissertation is section 2, where we derive our evaluation-interpolation algorithm for approximating the quotient of two polynomials, along with an informal correctness proof of the derivation [PanSL].

In section 3, we consider the role of roots of 1 in polynomial computations, with a discussion of a Toeplitz operator associated with polynomial multiplication. We then relate our method to the equivalent, linear algebra algorithm derivation method of [Bini]. We also study the relationship between complex integrals and Fourier series, to demonstrate the equivalence of our method to the algorithm derivation method of [Schönhage].

Additional applications of the evaluation-interpolation method are presented in section 4, before we analyze the computation error in section 5. We conclude in section 6 by considering a future research direction.

1 Discrete Fourier transform (dft) and the evaluation-interpolation method

We define dft, discuss an efficient dft computing method (fft), and provide a motivated derivation of the inverse dft. We also define the evaluation-interpolation method and the basic polynomial computation of multiplication (convolution), and illustrate the use of evaluation-interpolation in constructing convolution algorithms.

1.1 Definition of dft and derivation of fft

In [BiniPan, ch. 1], dft is defined as multipoint polynomial evaluation at dth roots of 1. If p is a polynomial ($\deg p < d$),

$$p(z) = \sum_{\ell=0}^{d-1} p_\ell z^\ell,$$

and if $\omega := \exp(2\pi i/d)$ is a primitive dth root of 1 ($\omega^d = 1$ and $\omega^k \neq 1$, for $0 < k < d$), then for $0 \leq k < d$, dft_d computes $[P_k := p(\omega^k)]$. Note that some authors (see [Cooley]) use a different primitive root in the definition of dft, namely $\omega := \exp(-2\pi i/d)$.

If $p^{\natural}$ is a polynomial ($\deg p^{\natural} < 2d$),

$$p^{\natural}(z) = \sum_{\ell=0}^{2d-1} p^{\natural}_\ell z^\ell,$$

and if $\psi := \exp(\pi i/d)$ is a primitive $2d$th root of 1, then for $0 \leq k < 2d$, dft_{2d} computes $[P^{\natural}_k := p^{\natural}(\psi^k)]$.

The key step in our fft derivation is an even-odd decomposition of the powers of ψ, to obtain a relationship between dft_{2d} and dft_d. We use the notation $p^{\natural}$ for polynomials of degree less than $2d$ to help clarify this relationship.

Define

$$p^+(z) = \sum_{\ell=0}^{d-1} p_\ell^+ z^\ell = \sum_{\ell=0}^{d-1} (p_\ell^\natural + p_{\ell+d}^\natural) z^\ell,$$

$$p^-(z) = \sum_{\ell=0}^{d-1} p_\ell^- z^\ell = \sum_{\ell=0}^{d-1} (p_\ell^\natural - p_{\ell+d}^\natural) z^\ell,$$

and

$$\widehat{p^-}(z) = \sum_{\ell=0}^{d-1} \widehat{p_\ell^-} z^\ell = \sum_{\ell=0}^{d-1} \psi^\ell p_\ell^- z^\ell.$$

For $0 \le k < d$,

$$p^\natural(\psi^{2k}) = \sum_{\ell=0}^{2d-1} p_\ell^\natural \omega^{k\ell} = \sum_{\ell=0}^{d-1} (p_\ell^\natural + p_{\ell+d}^\natural) \omega^{k\ell} = p^+(\omega^k)$$

$$p^\natural(\psi^{2k+1}) = \sum_{\ell=0}^{2d-1} p_\ell^\natural \psi^\ell \omega^{k\ell} = \sum_{\ell=0}^{d-1} \psi^\ell (p_\ell^\natural - p_{\ell+d}^\natural) \omega^{k\ell} = \widehat{p^-}(\omega^k).$$

In other words, we can see that evaluation of a polynomial with $2d$ coefficients at $2d$th roots of 1 can be reduced to evaluation of two polynomials with d coefficients at dth roots of 1. It follows that fft computes P_k, at dth roots of 1, with complexity $O(d \log d)$, provided that d is a power of 2.

1.2 Lagrange interpolation formula

Assume, with $\mathbf{C}$ the field of complex numbers, that z_k are d distinct points, and f is a function on $\mathbf{C}$ with $f(z_k) = A_k$. The elementary Lagrange interpolation formula provides a polynomial p, with $\deg p < d$, that agrees with f at the specified points z_k, that is, $p(z_k) = A_k$, $0 \le k < d$.

Before defining the Lagrange interpolation formula, we define a set of auxiliary, degree $d-1$, polynomials ℓ_k, $0 \le k < d$, by

$$\ell_k(z) = \prod_{\substack{j=0 \\ j \ne k}}^{d-1} (z - z_j).$$

The polynomials ℓ_k have the property that, for $j \neq k$, $\ell_k(z_j) = 0$.

The Lagrange interpolation formula

$$p(z) = \sum_{k=0}^{d-1} A_k \frac{\ell_k(z)}{\ell_k(z_k)},$$

provides a polynomial of degree less than d satisfying the required condition, $p(z_k) = A_k$, $0 \leq k < d$.

1.3 A motivated derivation of inverse dft

We provide a motivated derivation of the inverse dft that dispenses with any reference to Fourier transforms. This derivation removes the surprise of the inverse dft by starting from the Lagrange interpolation formula, yielding an elementary algebraic framework for the inverse dft.

Throughout this section, $\rho \neq 0$ will be a complex number. Assume that $z_k = \rho\omega^k$, for $0 \leq k < d$, where ω is a primitive dth root of 1. Note that these primitive roots are algebraic numbers, computable using iterated square roots, when d is a power of 2.

For our proof of the inverse dft, we will need the following lemma.

Lemma 1.0:

$$\prod_{\substack{j=0 \\ j\neq k}}^{d-1} \left(\frac{z}{\rho} - \omega^j\right) = \frac{1}{\omega^k} \sum_{j=0}^{d-1} \rho^{-j}\omega^{-kj} z^j.$$

Proof: Since, for $0 \leq j < d$, each ω^j is a distinct dth root of 1, we know that

$$\prod_{j=0}^{d-1} \left(\frac{z}{\rho} - \omega^j\right) = \left(\frac{z}{\rho}\right)^d - 1.$$

For $z \neq \rho\omega^k$,

$$\left(\frac{1}{(z/\rho) - \omega^k}\right) \prod_{j=0}^{d-1} \left(\frac{z}{\rho} - \omega^j\right) = \frac{(z/\rho)^d - 1}{(z/\rho) - \omega^k} = \frac{1}{\omega^k} \frac{(z/\rho\omega^k)^d - 1}{(z/\rho\omega^k) - 1},$$

since $(\omega^k)^d = 1$. Hence, for all z, by the sum of a geometric series formula,

$$\prod_{\substack{j=0 \\ j\neq k}}^{d-1} \left(\frac{z}{\rho} - \omega^j\right) = \frac{1}{\omega^k} \sum_{j=0}^{d-1} (z/\rho\omega^k)^j \quad \blacksquare$$

Assume there is a function f that has been specified at $z_k = \rho\omega^k$, that is $f(\rho\omega^k) = A_k$. We have a theorem giving a simple formula for its interpolating polynomial coefficients.

Theorem 1.0: (polynomial interpolation at scaled roots of 1)

Given $f(\rho\omega^k) = A_k$, then there is an interpolating polynomial p, with $\deg p < d$, $p(\rho\omega^k) = A_k$, and coefficients

$$p_\ell = \frac{1}{\rho^\ell d} \sum_{k=0}^{d-1} A_k \omega^{-k\ell}.$$

Proof: Using the Lagrange interpolation formula, with

$$\ell_k(z) = \prod_{\substack{j=0 \\ j\neq k}}^{d-1} \rho \left(\frac{z}{\rho} - \omega^j\right),$$

and the previous lemma,

$$\begin{aligned} p(z) &= \sum_{k=0}^{d-1} A_k \frac{\ell_k(z)}{\ell_k(\rho\omega^k)} \\ &= \sum_{k=0}^{d-1} \frac{A_k}{d} \sum_{\ell=0}^{d-1} \rho^{-\ell} \omega^{-k\ell} z^\ell \\ &= \sum_{\ell=0}^{d-1} \left(\frac{1}{\rho^\ell d} \sum_{k=0}^{d-1} A_k \omega^{-k\ell}\right) z^\ell \quad \blacksquare \end{aligned}$$

Corollary 1.0: (scaled inverse discrete Fourier transform)

Assume $[\widehat{P}_k := p(\rho\omega^k)]$ has been computed, with $\rho \neq 0$ a complex number, where p is a polynomial ($\deg p < d$), with coefficients p_ℓ. Then

$$p_\ell = \frac{1}{\rho^\ell d} \sum_{k=0}^{d-1} \widehat{P}_k \omega^{-k\ell} \quad \blacksquare$$

Corollary 1.1: (inverse discrete Fourier transform)

Assume $[P_k := p(\omega^k)]$ has been computed, where p is a polynomial ($\deg p < d$), with coefficients p_ℓ. Then

$$p_\ell = \frac{1}{d}\sum_{k=0}^{d-1} P_k \omega^{-k\ell} \quad \blacksquare$$

Define the polynomials

$$\widehat{P}(z) = \sum_{k=0}^{d-1} \widehat{P}_k z^k \quad \text{and} \quad P(z) = \sum_{k=0}^{d-1} P_k z^k.$$

The conclusions of the previous corollaries can be more concisely expressed as $p_\ell = (1/\rho^\ell d)\widehat{P}(\omega^{-\ell})$ and $p_\ell = (1/d)P(\omega^{-\ell})$.

We conclude that the coefficients of the solution polynomial provided by the [Lagrange] interpolation formula coincide with the inverse discrete Fourier transform formula. Though not a new conclusion, the derivation provides a simpler way of understanding the inverse dft formula.

1.4 Evaluation-interpolation method

The evaluation-interpolation (e.-i.) method, for polynomial computation algorithms, is an indirect method. Let

$$u(z) = \sum_{\ell=0}^{d-1} u_\ell z^\ell, \quad v(z) = \sum_{\ell=0}^{d-1} v_\ell z^\ell, \text{ and } \quad w(z) = \sum_{\ell=0}^{d-1} w_\ell z^\ell$$

be polynomials over $\mathbf{C}$ of degree less than d. Assume a binary arithmetic operation $\diamond$ defined on pairs U_k and V_k of complex numbers, for $0 \le k < d$. The e.-i. method induces a binary operation on pairs of polynomials u and v over $\mathbf{C}$.

Throughout the dissertation, to simplify matters, we assume d is a power of 2, and the points x_k are dth roots of 1, ω^k, where ω is a primitive dth root of 1.

Algorithm 1.0: {e.-i. method at dth roots of 1}

{a: evaluate - two dft_d} $[U_k := u(\omega^k)]$; $[V_k := v(\omega^k)]$;

{b: compute - d op's} $[W_k := U_k \diamond V_k]$;

{c: interpolate - inv. dft_d} $[w_\ell := (1/d)W(\omega^{-\ell})]$ ∎

The complexity of these computations is bounded by $O(d \log d)$.

We will soon illustrate this e.-i. method with two applications, after discussing polynomial multiplication and convolution.

1.5 Convolution

When two polynomials of degree less than d, u and v, are multiplied, the resulting polynomial $w^{\natural}$ will be of degree less than $2d-1$,

$$w^{\natural}(z) = \sum_{\ell=0}^{2(d-1)} w_\ell^{\natural} z^\ell.$$

For $0 \le \ell < d$,

$$w_\ell^{\natural} = \sum_{j=0}^{\ell} u_j v_{\ell-j},$$

$$w_{\ell+d}^{\natural} = \sum_{j=\ell+1}^{d-1} u_j v_{\ell+d-j}.$$

This coefficient formula, for the product of two polynomials, is called the convolution of the coefficients of the given polynomials (note that $w_{2d-1}^{\natural} = 0$, as expected).

Our discussion of computing convolution begins with a review of the fft derivation of section 1.1, applied to $w^{\natural}$.

Define $w^+(z) = \sum_{\ell=0}^{d-1} w_\ell^+ z^\ell$ and $w^-(z) = \sum_{\ell=0}^{d-1} w_\ell^- z^\ell$, where

$$\begin{aligned} w_\ell^+ &= w_\ell^{\natural} + w_{\ell+d}^{\natural}, \\ w_\ell^- &= w_\ell^{\natural} - w_{\ell+d}^{\natural}. \end{aligned} \tag{1.0}$$

Note that the coefficients

$$w_\ell^+ = \sum_{j=0}^{\ell} u_j v_{\ell-j} + \sum_{j=\ell+1}^{d-1} u_j v_{\ell+d-j}$$

and

$$w_\ell^- = \sum_{j=0}^{\ell} u_j v_{\ell-j} - \sum_{j=\ell+1}^{d-1} u_j v_{\ell+d-j}$$

are called the positive and negative wrapped convolutions of the coefficients of u and v.

We will find useful a key relationship between the different types of convolution. For $0 \le \ell < d$, solving the equations (1.0) for $w_\ell^{\mathbf{I}}$ and $w_{\ell+d}^{\mathbf{I}}$ yields

$$\begin{aligned} w_\ell^{\mathbf{I}} &= \tfrac{1}{2}(w_\ell^+ + w_\ell^-), \\ w_{\ell+d}^{\mathbf{I}} &= \tfrac{1}{2}(w_\ell^+ - w_\ell^-). \end{aligned} \tag{1.1}$$

1.6 Polynomial multiplication using e.-i. method

If we were to utilize Algorithm 1.0, with multiplication as the binary operation, degree of u and v less than d, and $2d$ evaluation points, we would obtain a correct polynomial multiplication algorithm. Our objective is to use the e.-i. method for polynomial multiplication, utilizing dft and inverse dft on d points. In order to meet our objective of working with d evaluation points, we follow the fft derivation of section 1.1, and partition the set of $2d$th roots of 1 into even and odd powers of $\psi := \exp(\pi i/d)$, a primitive $2d$th root of 1.

Following the fft derivation, with $p^{\mathbf{I}} = w^{\mathbf{I}}$, for $0 \le k < d$,

$$\begin{aligned} w^{\mathbf{I}}(\psi^{2k}) &= w^+(\omega^k), \\ w^{\mathbf{I}}(\psi^{2k+1}) &= \widehat{w^-}(\omega^k). \end{aligned}$$

By definition of $u \times v$, and since $\psi^2 = \omega$,

$$\begin{aligned} (u \times v)(\psi^{2k}) &= u(\psi^{2k}) \times v(\psi^{2k}) = u(\omega^k) \times v(\omega^k), \\ (u \times v)(\psi^{2k+1}) &= u(\psi^{2k+1}) \times v(\psi^{2k+1}) = \hat{u}(\omega^k) \times \hat{v}(\omega^k). \end{aligned}$$

Since $w^{\natural} = u \times v$, we conclude

$$\begin{aligned} w^{+}(\omega^k) &= u(\omega^k) \times v(\omega^k), \\ \widehat{w^{-}}(\omega^k) &= \hat{u}(\omega^k) \times \hat{v}(\omega^k). \end{aligned} \tag{1.2}$$

Hence we obtain the following algorithms, utilizing dft_d instead of dft_{2d}:

Algorithm 1.1: {positive wrapped convolution using the e.-i. method}

{a} $[U_k := u(\omega^k)]$; $[V_k := v(\omega^k)]$;
{b} $[W_k^{+} := U_k \times V_k]$;
{c} $[w_\ell^{+} := (1/d)W^{+}(\omega^{-\ell})]$∎

Algorithm 1.2: {negative wrapped convolution using the e.-i. method}

{a} $[\widehat{U}_k := \hat{u}(\omega^k)]$; $[\widehat{V}_k := \hat{v}(\omega^k)]$;
{b} $[\widehat{W_k^{-}} := \widehat{U}_k \times \widehat{V}_k]$;
{c} $[w_\ell^{-} := (1/\psi^{\ell} d)\widehat{W^{-}}(\omega^{-\ell})]$∎

The complexity of these algorithms is bounded by $O(d \log d)$.

With these two algorithms, the coefficients of $w^{\natural}$ are computed, using equation (1.1).

Note that Algorithm 1.2, step (c), uses Theorem 1.0 with $\rho = \psi$.

1.7 Polynomial division with no remainder using e.-i. method

Assume that w and u are polynomials with the property that w is known to be divisible by u, with quotient v and no remainder. In order to use the e.-i. method at dth roots of 1 (as described above) for deriving a polynomial division algorithm, it is necessary that $\deg w < 2d - 1$ and $\deg u < d$, to guarantee that $\deg(w - u) = \deg v < d$, where d is a power of 2.

We can show that choosing $d = 2^{(1+\lfloor \log_2 \max(\deg u, \deg v) \rfloor)}$ will assure the applicability of the e.-i. method at dth roots of 1 (as described above) for polynomial division algorithms.

We may now derive algorithms for polynomial division with no remainder utilizing the e.-i. method. With $w^1 = w$, solve equations (1.2), from the polynomial multiplication derivation of the previous section, for v and $\hat{v}$. For $0 \leq k < d$, we obtain

$$v(\omega^k) = w^+(\omega^k)/u(\omega^k),$$
$$\hat{v}(\omega^k) = \widehat{w^-}(\omega^k)/\hat{u}(\omega^k),$$

providing two ways to implement our polynomial division algorithm with no remainder.

Algorithm 1.3:

{polynomial division with no remainder using the e.-i. method}

{a} $[W_k^+ := w^+(\omega^k)]$; $[U_k := u(\omega^k)]$;

{b} $[V_k := W_k^+/U_k]$;

{c} $[v_\ell := (1/d)V(\omega^{-\ell})]$ ∎

Algorithm 1.4:

{polynomial division with no remainder using the e.-i. method}

{a} $[\widehat{W_k^-} := \widehat{w^-}(\omega^k)]$; $[\widehat{U}_k := \hat{u}(\omega^k)]$;

{b} $[\widehat{V}_k := \widehat{W_k^-}/\widehat{U}_k]$;

{c} $[v_\ell := (1/\psi^\ell d)\widehat{V}(\omega^{-\ell})]$ ∎

The complexity of each algorithm is bounded by $O(d \log d)$.

Note that Algorithm 1.4, step (c), uses Theorem 1.0, with $\rho = \psi$.

Remark: There will be difficulty with this method if $v(\omega^k) = 0$ and $\hat{v}(\omega^k) = 0$ (see section 2.2 for a remedy).

2 Approximating the quotient of two polynomials using the evaluation-interpolation method

We concentrate on the problem of approximating the quotient of two polynomials: given the coefficients of two polynomials s and t over the field $\mathbf{C}$, $\deg s = m$ and $\deg t = n$, with $m \geq n$ and $n > 0$,

$$s(z) = \sum_{\ell=0}^{m} s_\ell z^\ell, \quad t(z) = \sum_{\ell=0}^{n} t_\ell z^\ell \quad (t_n = 1),$$

compute the coefficients of a polynomial q^*, an approximate quotient,

$$q^*(z) = \sum_{\ell=0}^{K-1} q_\ell^* z^\ell,$$

$\deg q^* = m - n = K - 1$, such that

$$s(z) \approx q^*(z) \times t(z).$$

2.1 The e.-i. algorithm derivation and correctness proof

In [PanSL], we developed an elementary algorithm for approximating the quotient of two polynomials, equivalent (except for simple power shifts) to the algorithm of [Bini] and [Schönhage] for approximating the reciprocal of a polynomial, with a simpler derivation, based on the evaluation-interpolation (e.-i.) method of [Toom].

We started with equation (0.0), solved for $q(z)$,

$$q(z) = \frac{s(z)}{t(z)} - \frac{r(z)}{t(z)}. \tag{2.0}$$

We knew that as $|z| \to \infty$, $r(z)/t(z) \to 0$ (since $\deg r < \deg t$), hence equation (2.0) implies that $s(z)/t(z) \to q(z)$. We also knew that a polynomial of degree $K - 1$ is determined by its values at K distinct points.

To construct our solution, we interpolated q^* at $H\omega^k$, $0 \le k < K$ (scaled roots of 1), to s/t. Utilizing Theorem 1.0 (with $\rho = H$), we obtained an approximate quotient q^*, with coefficients

$$q_\ell^* = \frac{1}{H^\ell K} \sum_{k=0}^{K-1} \frac{s(H\omega^k)}{t(H\omega^k)} \omega^{-k\ell}. \tag{2.1}$$

We will prove this approximate quotient q^* converges to q in the following lemma. In order to simplify the proof, let $H = 2^h$ for $h > 0$.

Lemma 2.0: $|q_\ell^* - q_\ell| = O(1/H^{\ell+1})$, for $0 \le \ell < K$.

Proof: Using equation (2.0) and Corollary 1.0, with $\rho = H$, we have a formula for q_ℓ,

$$q_\ell = \frac{1}{H^\ell K} \sum_{k=0}^{K-1} \left(\frac{s(H\omega^k)}{t(H\omega^k)} - \frac{r(H\omega^k)}{t(H\omega^k)} \right) \omega^{-k\ell}.$$

Similarly, using Theorem 1.0, with $A_k = s(H\omega^k)/t(H\omega^k)$, we obtain the approximation formula (2.1) for q^*.

Assume $\tau_h \le \min_{0 \le k < K} |t(H\omega^k)|$ and $\varrho \ge \max_{0 \le j < n} |r_j|$ (independent of H). Then

$$\begin{aligned}
|q_\ell^* - q_\ell| &= \left| \frac{1}{H^\ell K} \sum_{k=0}^{K-1} \frac{r(H\omega^k)}{t(H\omega^k)} \omega^{-k\ell} \right| \\
&\le \frac{1}{H^\ell K} \sum_{k=0}^{K-1} \frac{|r(H\omega^k)|}{|t(H\omega^k)|} \le \frac{1}{H^\ell K \tau_h} \sum_{k=0}^{K-1} |r(H\omega^k)| \\
&\le \frac{1}{H^\ell K \tau_h} \sum_{k=0}^{K-1} \left(\sum_{j=0}^{n-1} |r_j (H\omega^k)^j| \right) = \frac{1}{H^\ell K \tau_h} \sum_{k=0}^{K-1} \left(\sum_{j=0}^{n-1} |r_j| H^j \right) \\
&\le \frac{\varrho}{H^\ell \tau_h} \sum_{j=0}^{n-1} H^j = \frac{\varrho}{H^\ell \tau_h} \frac{H^n - 1}{H - 1}.
\end{aligned}$$

We have reduced the proof of this lemma to showing $1/\tau_h = O(1/H^n)$.

Make the additional assumption that

$$2\sum_{j=0}^{n-1} |t_j| H^j < H^n. \tag{2.2}$$

For instance, this assumption will be satisfied when $H \geq 3 \max_{0 \leq j < n} |t_j|^{1/(n-j)}$, since when $|t_j| \leq (H/3)^{n-j}$, for $0 \leq j < n$, then

$$2\sum_{j=0}^{n-1} |t_j| H^j \leq (1 - (1/3)^n)\, H^n < H^n.$$

From the definition of t, we know $t_n = 1$. From the triangle inequality and the assumption (2.2), we have

$$|t(H\omega^k)| \geq H^n - \sum_{j=0}^{n-1} |t_j| H^j > H^n/2.$$

In other words, $\min_{0 \leq k < K} |t(H\omega^k)| > H^n/2$. Hence $1/\tau_h = O(1/H^n)$. Thus

$$|q_\ell^* - q_\ell| < \frac{\varrho}{H^\ell} \frac{H^n}{H-1} \frac{2}{H^n} = \frac{2\varrho}{H^\ell (H-1)} = O(1/H^{\ell+1}) \quad \blacksquare$$

Further details, relating ϱ to the polynomial coefficients $|s_\ell|$ and $|t_\ell|$, are found in [BiniPan86] or [BiniPan, ch. 6].

2.2 The approximation algorithm

Recall Algorithm 1.3 and its complexity estimate (section 1.7), which was derived using the e.-i. method at dth roots of 1. We extend this algorithm for polynomial division with no remainder, to solve the problem of approximating the quotient of two polynomials.

Assume d satisfies the condition of section 1.7, with $u = t$ and $v = q^*$. In order to compute $s(H\omega^k)$ and $t(H\omega^k)$, $0 \leq k < d$, define polynomials $\hat{s}^{\sharp}$ and $\hat{t}$, with $\deg \hat{s}^{\sharp} < 2d-1$ and $\deg \hat{t} < d$, and with coefficients

$$\hat{s}_\ell^{\sharp} = \begin{cases} H^\ell s_\ell, & 0 \leq \ell \leq m, \\ 0, & m < \ell < 2d-1, \end{cases} \quad \text{and} \quad \hat{t}_\ell = \begin{cases} H^\ell t_\ell, & 0 \leq \ell \leq n, \\ 0, & n < \ell < d, \end{cases} \tag{2.3}$$

computed by shifts. As in section 1.7, with $w = s$, we define

$$\hat{s}^+(z) = \sum_{\ell=0}^{d-1} \hat{s}_\ell^+ z^\ell = \sum_{\ell=0}^{d-1} (\hat{s}_\ell^1 + \hat{s}_{\ell+d}^1) z^\ell.$$

We then compute the coefficients $\hat{q}_\ell^*$ of the polynomial

$$\hat{q}^*(z) = \sum_{\ell=0}^{d-1} \hat{q}_\ell^* z^\ell,$$

where

$$\hat{q}_\ell^* = \frac{1}{d} \sum_{k=0}^{d-1} \frac{\hat{s}^+(\omega^k)}{\hat{t}(\omega^k)} \omega^{-\ell k}. \tag{2.4}$$

Algorithm 2.0:

{approximating the quotient of two polynomials using e.-i. method}

{a} $[\widehat{S}_k^+ := \hat{s}^+(\omega^k)]$; $[\widehat{T}_k := \hat{t}(\omega^k)]$;

{b} $[\widehat{Q}_k^* := \widehat{S}_k^+ / \widehat{T}_k]$;

{c} $[\hat{q}_\ell^* := (1/d)\widehat{Q}^*(\omega^{-\ell})]$ ∎

The computational cost of this algorithm is bounded by $O(d \log d)$.

The coefficients q_ℓ^*, $0 \le \ell < K$, of the approximate quotient q^*, can be computed by shifts, utilizing the equation derived from Theorem 1.0, $q_\ell^* = \hat{q}_\ell^* / H^\ell$. Note that $q_\ell^* = 0$ for $K \le \ell < d$.

Remark: This approximation algorithm is most effective for $m \approx \alpha n$, for a small constant α, in which case $d \approx n$.

We summarize our analysis with a theorem, originally proved independently by [Bini] and [Schönhage]:

Theorem 2.0: The approximate quotient of two polynomials can be computed, to an arbitrary accuracy, with the arithmetic computational cost $O(d \log d)$, utilizing $O(1)$ dfts ∎

Note: The parallel implementation of this approximation algorithm is an improvement over other $O(d \log d)$ algorithms, since no other known algorithm with total work $O(d \log d)$ has parallel time-complexity $O(\log d)$.

3 Comparison of the e.-i. method with the linear algebra and complex integration methods

We discuss the relationship of our algorithm derivation method to previous methods, after introducing the linear algebra and complex integration methods.

3.1 Definition of the circular shift operator

Assume we are working with $\mathbf{C}^{\mathbf{d}}$, the d-dimensional vector space over $\mathbf{C}$, with basis vectors e_k for $0 \leq k < d$. For instance, if the vector space is interpreted as d-tuples of complex numbers, then

$$e_{k_j} = \delta_{kj} = \begin{cases} 1, & k = j, \\ 0, & k \neq j. \end{cases}$$

A general vector v can be written as a linear combination of basis vectors

$$v = \sum_{k=0}^{d-1} v_k e_k,$$

where v_k is called the kth coordinate of v with respect to the basis e_k.

In our current work, it would be helpful to adopt another interpretation of $\mathbf{C}^{\mathbf{d}}$, as the vector space of polynomials over $\mathbf{C}$ of degree less than d. Instead of defining the basis vectors e_k in the d-tuple fashion, we simply interpret e_k as z^k, where z is the indeterminate.

Define the circular shift operator T by

$$T(v) = \sum_{\ell=0}^{d-1} v_{(\ell-1) \bmod d} z^{\ell}$$

(for the importance of shift operators in Fourier Analysis, see, for instance, the introduction of [Wiener]).

3.2 Eigenvalues of the Toeplitz operator

The key observation is that T satisfies the operator equation $T^d = I$. Hence T has characteristic polynomial $\lambda^d - 1 = 0$, with eigenvalues $\lambda_k = \omega^k$, $0 \leq k < d$, where ω is a primitive dth root of 1.

Corresponding to each eigenvalue ω^k, we need to determine an eigenvector P_k, satisfying $T(P_k) = \omega^k P_k$, or

$$\sum_{\ell=0}^{d-1} P_{k_{(\ell-1) \bmod d}} z^\ell = \omega^k \sum_{\ell=0}^{d-1} P_{k_\ell} z^\ell.$$

Equating coordinates, we obtain

$$P_{k_{(\ell-1) \bmod d}} = \omega^k P_{k_\ell}.$$

We can restate the equation slightly, to put it into more familiar recurrence relation form, by extending the domain of P_k,

$$P_{k_{\ell+1}} = \omega^{-k} P_{k_\ell}, \quad P_{k_d} = P_{k_0}.$$

This equation, when regarded as a first-order, constant coefficient recurrence relation in ℓ, with boundary condition, has a general solution

$$P_{k_\ell} = c(\omega^{-k})^\ell.$$

Checking the boundary condition, $P_{k_0} = c$, and $P_{k_d} = c(\omega^{-k})^d = c$, shows that we have determined, up to the usual constant, the eigenvector

$$P_k = c \sum_{\ell=0}^{d-1} \omega^{-k\ell} z^\ell,$$

corresponding to the eigenvalue ω^k.

The vector P_k is also an eigenvector for T^j, defined by

$$T^j(v) = \sum_{\ell=0}^{d-1} v_{(\ell-j) \bmod d} z^\ell,$$

for any integer j, with corresponding eigenvalue ω^{jk}.

We now have a lemma concerning eigenvalues of the Toeplitz operator, $U = u(T)$, associated with the polynomial

$$u(z) = \sum_{j=0}^{d-1} u_j z^j.$$

Lemma 3.0:

Assume U is the Toeplitz operator associated with the polynomial u,

$$u(z) = \sum_{j=0}^{d-1} u_j z^j, \quad U(v) = \sum_{j=0}^{d-1} u_j \left(\sum_{\ell=0}^{d-1} v_{(\ell-j) \bmod d} z^\ell \right).$$

Then the vectors P_k form an eigenbasis for $\mathbf{C}^\mathbf{d}$, with corresponding eigenvalues λ_k,

$$P_k = c \sum_{\ell=0}^{d-1} \omega^{-k\ell} z^\ell, \quad \lambda_k = \sum_{j=0}^{d-1} u_j \omega^{jk} \quad \blacksquare$$

This operator U, formally a polynomial in T, is called a Toeplitz operator because the associated matrix is a circulant matrix, $[u_{\ell j}] = [u_{(\ell-j) \bmod d}]$, which is a special case of a Toeplitz matrix (for definitions, see [BiniPan, ch. 2] or [GohbergO]).

Note that the Toeplitz operator U associated with a polynomial u, applied to a (vector) polynomial v, results in the (vector) polynomial w^+, whose coefficients are given by the positive wrapped convolution of the

coefficients of u and v,

$$U(v) = \sum_{j=0}^{d-1} u_j T^j(v) = \sum_{j=0}^{d-1} u_j \sum_{\ell=0}^{d-1} v_{(\ell-j) \bmod d} z^\ell$$
$$= \sum_{\ell=0}^{d-1} \sum_{j=0}^{d-1} u_j v_{(\ell-j) \bmod d} z^\ell = \sum_{\ell=0}^{d-1} w_\ell^+ z^\ell.$$

Historical note: Otto Toeplitz (1881-1940) was a mathematician who also studied the history of mathematics.

3.3 The Toeplitz operator eigenvector expansion

We express the Toeplitz operator, $U = u(T)$, in terms of its eigenvector basis.

To determine the coefficients V_k in the eigenvector expansion of

$$v = \sum_{k=0}^{d-1} V_k P_k,$$

we must solve the polynomial equation

$$\sum_{\ell=0}^{d-1} v_\ell z^\ell = \sum_{k=0}^{d-1} V_k \left(c \sum_{\ell=0}^{d-1} \omega^{-k\ell} z^\ell \right)$$

for cV_k, or equivalently, we must solve

$$v_\ell = \sum_{k=0}^{d-1} cV_k \omega^{-k\ell}.$$

This can be viewed as an interpolation problem for the polynomial

$$V(z) = \sum_{k=0}^{d-1} cV_k z^k :$$

given $V(1/\omega^\ell) = v_\ell$, compute cV_k.

By Corollary 1.1, we know

$$cV_k = \frac{1}{d}\sum_{\ell=0}^{d-1} v_\ell \left(\frac{1}{(1/\omega^\ell)}\right)^k$$
$$= \frac{1}{d}\sum_{\ell=0}^{d-1} v_\ell \omega^{k\ell}.$$

Choose $c = 1/d$, then we can see that the coefficients in the expansion of an arbitrary polynomial v into eigenvectors of the Toeplitz operator are obtained by evaluation of v at dth roots of 1.

We summarize our analysis with a lemma.

Lemma 3.1:

Assume that the Toeplitz operator U is expressed with respect to the eigenvector basis P_k, $0 \le k < d$, with $c = 1/d$ in the definition of P_k.

Then

$$U(v) = \sum_{k=0}^{d-1} \lambda_k V_k P_k$$
$$= \sum_{k=0}^{d-1} \left(\left(\sum_{\ell=0}^{d-1} u_\ell \omega^{k\ell} \right) \left(\sum_{j=0}^{d-1} v_j \omega^{kj} \right) P_k \right) \quad \blacksquare$$

These computations involve two dfts on d points and d multiplications. With

$$U(v) = w^+(z) = \sum_{k=0}^{d-1} W_k^+ P_k,$$

we will then recover w_ℓ^+, $0 \le \ell < d$, the coefficients of $w^+(z) = \sum_{\ell=0}^{d-1} w_\ell^+ z^\ell$, (positive wrapped convolution of u and v), by interpolation (an inverse dft).

3.4 A variation of the Toeplitz operator

It will be instructive to state a variation of Lemma 3.1. Following closely the presentation of the previous section, we define the Toeplitz operator $\widehat{U}$ associated with the polynomial $\hat{u}$ by $\widehat{U} = \hat{u}(T)$, and let

$$\hat{v} = \sum_{k=0}^{d-1} \hat{v}_k z^k$$

be a vector (note $\rho = \psi$ in the definition of $\hat{u}$ and $\hat{v}$).

Lemma 3.2:

Assume that the Toeplitz operator $\widehat{U}$ is expressed with respect to the eigenvector basis P_k, $0 \leq k < d$, with $c = 1/d$ in the definition of P_k.

Then

$$\begin{aligned}\widehat{U}(\hat{v}) &= \sum_{k=0}^{d-1} \hat{\lambda}_k \widehat{V}_k P_k \\ &= \sum_{k=0}^{d-1} \left(\left(\sum_{\ell=0}^{d-1} \hat{u}_\ell \omega^{k\ell} \right) \left(\sum_{j=0}^{d-1} \hat{v}_j \omega^{kj} \right) P_k \right) \quad \blacksquare\end{aligned}$$

These computations involve two dfts on d points and d multiplications. With

$$\widehat{U}(\hat{v}) = \widehat{w^-}(z) = \sum_{k=0}^{d-1} \widehat{W_k^-} P_k,$$

we will then recover w_ℓ^-, $0 \leq \ell < d$, the coefficients of $w^-(z) = \sum_{\ell=0}^{d-1} w_\ell^- z^\ell$, (the negative wrapped convolution of u and v), by interpolation (a scaled inverse dft).

3.5 Equivalence of e.-i. method with the linear algebra method

Following the pattern of Lemma 3.2, we can prove, assuming ($\rho \neq 0$) $\hat{u}_j = \rho^j u_j$, $\hat{v}_j = \rho^j v_j$ and $\widehat{w}_j^+ = \rho^j w_j^! + \rho^{j+d} w_{j+d}^!$, that the operator equation $\widehat{U}(\hat{v}) = \widehat{w}^+$ has a solution

$$v(z) = \sum_{\ell=0}^{d-1} v_\ell z^\ell,$$

with

$$v_\ell = \frac{1}{\rho^\ell d} \sum_{k=0}^{d-1} \left(\frac{\sum_{j=0}^{d-1} \widehat{w}_j^+ \omega^{kj}}{\sum_{j=0}^{d-1} \widehat{u}_j \omega^{kj}} \right) \omega^{-\ell k},$$

provided $\hat{u}(\omega^k) \neq 0$ for $0 \leq k < d$. Then Theorem 2.0 is proved as in section 2.1, since the formula for v_ℓ, following the pattern of equation (2.4), can be related to q_ℓ^* of Lemma 2.0, following the pattern of equation (2.3). This is the essence of the linear algebra method of [Bini].

Now that we have demonstrated the equivalence of the e.-i. method with the linear algebra method, we will discuss the complex integration method.

3.6 Euler-Fourier coefficients and complex integration

We state two basic lemmas from the elementary theory of rational functions of a complex variable (see, for instance [Knopp]), and then prove a lemma which is a simplified version of the Cauchy integral formula. In these lemmas, and subsequently, we assume that C, the path of integration in $\mathbf{C}$, is a circle centered at the origin.

In order to derive polynomial coefficient formulas, we need to integrate power functions on circular paths in **C**.

Lemma 3.3: Let C be a circle, and ℓ an integer. Then

$$\oint_C z^\ell dz = \begin{cases} 2\pi i & \ell = -1, \\ 0 & \ell \neq -1 \end{cases} \quad \blacksquare$$

In order to estimate approximation errors, we need an inequality concerning integrals of rational functions on circular paths in **C**.

Lemma 3.4: Let C be a circle of radius $H = 2^h$, and s and t be polynomials. Assume $|s(z)| \leq \sigma_h$ and $|t(z)| \geq \tau_h$ for all z of C. Then

$$\left| \oint_C \frac{s(z)}{t(z)} dz \right| \leq 2\pi H \frac{\sigma_h}{\tau_h} \quad \blacksquare$$

We need a formula relating a polynomial and its coefficients using complex integration.

Lemma 3.5: Let p be a polynomial, $p(z) = \sum_{\ell=0}^{d-1} p_\ell z^\ell$, and C be any circle. Then

$$p_\ell = \frac{1}{2\pi i} \oint_C \frac{p(z)}{z^{\ell+1}} dz.$$

Proof: Multiply $p(z)$ by $(1/2\pi i) z^{-(\ell+1)}$, $0 \leq \ell < d$, and integrate:

$$\begin{aligned} \frac{1}{2\pi i} \oint_C \frac{p(z)}{z^{\ell+1}} dz &= \frac{1}{2\pi i} \oint_C \left(\sum_{j=0}^{d-1} p_j z^{j-(\ell+1)} \right) dz \\ &= \sum_{j=0}^{d-1} \frac{p_j}{2\pi i} \oint_C z^{j-(\ell+1)} dz = p_\ell, \end{aligned}$$

by Lemma 3.3 $\blacksquare$

Let C be a circle of radius H. Writing z in polar form, we have from the previous lemma, if

$$p(He^{i\theta}) = \sum_{\ell=0}^{d-1} p_\ell H^\ell e^{i\ell\theta},$$

then

$$p_\ell = \frac{1}{2\pi i}\int_0^{2\pi} \frac{p(He^{i\theta})}{H^{\ell+1}e^{i(\ell+1)\theta}} iHe^{i\theta}d\theta = \frac{1}{2\pi H^\ell}\int_0^{2\pi} p(He^{i\theta})e^{-i\ell\theta}d\theta.$$

Letting $f(\theta) = p(He^{i\theta})$, if

$$f(\theta) = \sum_{\ell=0}^{d-1} f_\ell e^{i\ell\theta},$$

we have deduced a special case of the Euler-Fourier coefficient formula,

$$f_\ell = \frac{1}{2\pi}\int_0^{2\pi} f(\theta)e^{-i\ell\theta}d\theta.$$

3.7 Complex integration and polynomial division

Using the notation of section 3.6, we would like to take a second look at Lemma 2.0, this time from the point of view of complex integration.

Start with equation (2.0), and then multiply both sides by $1/z^{\ell+1}$, for $0 \le \ell < d$. Assume all zeroes of t are within a disc of radius D. Let C be a circle with radius $H > D$. Then

$$\oint_C \frac{q(z)}{z^{\ell+1}}dz = \oint_C \frac{s^\natural(z)}{z^{\ell+1}t(z)}dz - \oint_C \frac{r(z)}{z^{\ell+1}t(z)}dz.$$

Using Lemma 3.5, we have $q_\ell = \frac{1}{2\pi i}\oint_C \frac{s^\natural(z)}{z^{\ell+1}t(z)}dz - \frac{1}{2\pi i}\oint_C \frac{r(z)}{z^{\ell+1}t(z)}dz.$

We would like to show that as $H \to \infty$,

$$q_\ell^* = \frac{1}{2\pi i}\oint_C \frac{s^\natural(z)}{z^{\ell+1}t(z)}dz \to q_\ell, \tag{3.0}$$

by proving the following lemma.

Lemma 3.6: $|q_\ell^* - q_\ell| = \left|\frac{1}{2\pi i}\oint_C \frac{r(z)}{z^{\ell+1}t(z)}dz\right| = O(1/H^{\ell+1})$.

Proof: Use Lemma 3.4 ($\tau_h = \min_{0\le\theta<2\pi} |t(He^{i\theta})|$ and ϱ as in Lemma 2.0):

$$\begin{aligned}\left|\frac{1}{2\pi i}\oint_C \frac{r(z)}{z^{\ell+1}t(z)}dz\right| &\le \frac{1}{H^\ell \tau_h}\max_{0\le\theta<2\pi}|r(He^{i\theta})| \\ &\le \frac{1}{H^\ell\tau_h}\sum_{j=0}^{n-1}|r_j|H^j \le \frac{\varrho}{H^\ell\tau_h}\sum_{j=0}^{n-1}H^j \\ &= \frac{\varrho}{H^\ell\tau_h}\frac{H^n-1}{H-1}.\end{aligned}$$

From the proof of Lemma 2.0, we can see again that $1/\tau_h = O(1/H^n)$, since

$$\min_{0\le\theta<2\pi}|t(He^{i\theta})| = \min_{0\le k<K}|t(H\omega^k)|.$$

Hence $|q_\ell^* - q_\ell| = O(1/H^{\ell+1})$ ∎

3.8 Equivalence of e.-i. method with complex integration method

The method from [Schönhage] is to estimate q_ℓ, $0 \le \ell \le K$, in the integral formula (3.0), after change of variable in the contour integral, by numerical integration.

Let $z = He^{i\theta}$, for $0 \le \theta < 2\pi$. Then

$$\begin{aligned}q_\ell^* &= \frac{1}{2\pi i}\oint_C \frac{s^\ell(z)}{z^{\ell+1}t(z)}dz = \frac{1}{2\pi H^\ell}\int_0^{2\pi}\frac{s^\ell(He^{i\theta})}{t(He^{i\theta})}e^{-i\ell\theta}d\theta \\ &\approx \frac{1}{dH^\ell}\sum_{k=0}^{d-1}\frac{s^\ell(He^{2\pi ik/d})}{t(He^{2\pi ik/d})}e^{-2\pi i\ell k/d},\end{aligned}$$

for $d \ge K$, by standard numerical integration.

Thus we arrive at the same formula (2.1) as the e.-i. algorithm (with $d = K$ and $\omega = e^{2\pi i/d}$), utilizing numerical integration of the Euler-Fourier coefficient formula.

4 Additional applications of the e.-i. method

The previous discussion of the e.-i. method was focused on preparation for the derivation of Algorithm 2.0. This section will discuss applications of the e.-i. method that are of independent interest, and will be useful when we discuss the computation error analysis of that algorithm.

4.1 Correlation

When a polynomial of degree less than d, u, is multiplied by a reciprocal polynomial of degree less than d, v,

$$v(z) = \sum_{j=0}^{d-1} v_j z^{-j},$$

the result is the sum of a polynomial and reciprocal polynomial,

$$w^{\flat}(z) = \sum_{\ell=-(d-1)}^{d-1} w_{\ell}^{\flat} z^{\ell}.$$

For $0 \leq \ell < d$,

$$w_{\ell}^{\flat} = \sum_{j=0}^{d-1-\ell} u_{j+\ell} v_j,$$

$$w_{\ell-d}^{\flat} = \sum_{j=d-\ell}^{d-1} u_{j+\ell-d} v_j.$$

This coefficient formula, for the product of a polynomial with a reciprocal polynomial, is called the correlation of the coefficients of the given polynomials (note that $w_{-d}^{\flat} = 0$, as expected).

The notation $w^{\flat}$ is contrasted with $w^{\natural}$, since $d-1$ of the $2d-1$ coefficients have now shifted below, as opposed to above, the range $0 \leq \ell < d$.

Our discussion of computing correlation begins by following our discussion of convolution from section 1.5.

Define $w^+(z) = \sum_{\ell=0}^{d-1} w_\ell^+ z^\ell$ and $w^-(z) = \sum_{\ell=0}^{d-1} w_\ell^- z^\ell$, where

$$\begin{aligned} w_\ell^+ &= w_\ell^\flat + w_{\ell-d}^\flat, \\ w_\ell^- &= w_\ell^\flat - w_{\ell-d}^\flat. \end{aligned} \tag{4.0}$$

Note that the coefficients

$$\begin{aligned} w_\ell^+ &= \sum_{j=0}^{d-1-\ell} u_{j+\ell} v_j + \sum_{j=d-\ell}^{d-1} u_{j+\ell-d} v_j, \\ w_\ell^- &= \sum_{j=0}^{d-1-\ell} u_{j+\ell} v_j - \sum_{j=d-\ell}^{d-1} u_{j+\ell-d} v_j, \end{aligned}$$

are called the positive and negative wrapped correlations of the coefficients of u and v, respectively.

We will find useful a key relationship between the different types of correlation. For $0 \le \ell < d$, solving the equations (4.0) for $w_\ell^\flat$ and $w_{\ell-d}^\flat$ yields

$$\begin{aligned} w_\ell^\flat &= \tfrac{1}{2}(w_\ell^+ + w_\ell^-), \\ w_{\ell-d}^\flat &= \tfrac{1}{2}(w_\ell^+ - w_\ell^-). \end{aligned} \tag{4.1}$$

4.2 Correlation and polynomial multiplication using e.-i. method

Given a polynomial of degree less than d, v, define the reverse polynomial $v^{\leftarrow}$ by $v^{\leftarrow}(z) = z^d v(1/z)$ (see [BiniPan, ch. 1]). It is not difficult to see that if u, a polynomial of degree less than d, is multiplied by $v^{\leftarrow}$, the resulting polynomial will have the same coefficients as the correlation of the previous section, except for the power shift of z^d.

Hence the discussion of section 1.6, where the e.-i. method is used to compute positive and negative wrapped convolutions, can be applied with no essential change to compute the positive and negative wrapped correlations of the previous section.

When using the e.-i. method at dth roots of 1 to compute correlations, we see that the power shift z^d is irrelevant (since $\omega^{kd} = 1$). Given a polynomial v, define the conjugate polynomial $\overline{v}$ by

$$\overline{v}(z) = \sum_{j=0}^{d-1} \overline{v}_j z^j.$$

We see that $v^{\sim}(\omega^k) = v(1/\omega^k) = \overline{\overline{v}(\omega^k)}$, since $\overline{\overline{v}(z)} = \overline{\overline{v}}(\overline{z}) = v(\overline{z})$, and $\overline{\omega^k} = 1/\omega^k$.

Hence we obtain the following algorithms:

Algorithm 4.0: {positive wrapped correlation using the e.-i. method}

{a} $[U_k := u(\omega^k)]$; $[V_k := \overline{\overline{v}(\omega^k)}]$;

{b} $[W_k^+ := U_k \times V_k]$;

{c} $[w_\ell^+ := (1/d)W^+(\omega^{-\ell})]$ ∎

Algorithm 4.1: {negative wrapped correlation using the e.-i. method}

{a} $[\widehat{U}_k := \widehat{u}(\omega^k)]$; $[\widehat{V}_k := \overline{\overline{\widehat{v}}(\omega^k)}]$;

{b} $[\widehat{W_k^-} := \widehat{U}_k \times \widehat{V}_k]$;

{c} $[w_\ell^- := (1/\psi^\ell d)\widehat{W^-}(\omega^{-\ell})]$ ∎

The complexity of these algorithms is bounded by $O(d \log d)$.

With these two algorithms, the coefficients of w^b are computed, using equation (4.1).

4.3 A Hankel operator eigenvector expansion

Following the discussion of section 3.2, we note that the vector P_k from that section (see lemma below) is also an eigenvector for T^{-j}, defined by

$$T^{-j}(u) = \sum_{\ell=0}^{d-1} u_{(\ell+j) \bmod d} z^{\ell}.$$

We now have a lemma concerning eigenvalues of the Hankel operator, $V = v(T)$, associated with the reciprocal polynomial v.

Lemma 4.0:

Assume V is the Hankel operator associated with the reciprocal polynomial v,

$$v(z) = \sum_{j=0}^{d-1} v_j z^{-j}, \quad V(u) = \sum_{j=0}^{d-1} v_j \left(\sum_{\ell=0}^{d-1} u_{(\ell+j) \bmod d} z^{\ell} \right).$$

Then the vectors P_k form an eigenbasis for $\mathbf{C}^{\mathbf{d}}$, with corresponding eigenvalues λ_k,

$$P_k = c \sum_{\ell=0}^{d-1} \omega^{-k\ell} z^{\ell}, \quad \lambda_k = \sum_{j=0}^{d-1} v_j \omega^{-jk} \quad \blacksquare$$

This operator V, formally a reciprocal polynomial in T, is called a Hankel operator (for the German mathematician Hermann Hankel (1839-1873) (see [BiniPan, ch. 2] or [PanSLT] for definitions)). Note that the Hankel operator V associated with a reciprocal polynomial v, applied to a polynomial u, results in the polynomial w^+, whose coefficients are given by the positive wrapped correlation of the coeffcients of u and v,

$$\begin{aligned} V(u) &= \sum_{j=0}^{d-1} v_j T^{-j}(u) = \sum_{j=0}^{d-1} v_j \sum_{\ell=0}^{d-1} u_{(\ell+j) \bmod d} z^{\ell} \\ &= \sum_{\ell=0}^{d-1} \sum_{j=0}^{d-1} v_j u_{(\ell+j) \bmod d} z^{\ell} = \sum_{\ell=0}^{d-1} w_{\ell}^{+} z^{\ell}. \end{aligned}$$

Similarly, following the discussion of section 3.3, we obtain a lemma concerning the Hankel operator eigenvector expansion.

Lemma 4.1:

Assume that the Hankel operator V is expressed with respect to the eigenvector basis P_k, $0 \le k < d$, with $c = 1/d$ in the definition of P_k.

Then

$$V(u) = \sum_{k=0}^{d-1} \lambda_k U_k P_k = \sum_{k=0}^{d-1} \left(\left(\sum_{\ell=0}^{d-1} v_\ell \omega^{-k\ell} \right) \left(\sum_{j=0}^{d-1} u_j \omega^{kj} \right) P_k \right) \quad \blacksquare$$

These computations involve two dfts on d points, d multiplications and $2d$ complex conjugations. With

$$V(u) = w^+(z) = \sum_{k=0}^{d-1} W_k^+ P_k,$$

we then recover w_ℓ^+, $0 \le \ell < d$, the coefficients of $w^+(z) = \sum_{\ell=0}^{d-1} w_\ell^+ z^\ell$, (positive wrapped correlation of u and v), by interpolation (an inverse dft).

The analogous result for negative wrapped correlation, following the discussion of section 3.4, is also true.

4.4 Parseval's theorem and positive wrapped correlation

We establish the basic result that $\left(1/\sqrt{d}\right) \mathrm{dft}_d$ preserves the 2-norm in $\mathbf{C}^d$.

Recall that for a complex number z, we know that its norm $|z|$ has the property that $|z|^2 = z\overline{z}$. For a polynomial p, with $\deg p < d$, we define the 2-norm $\|p\|_2$ to be

$$\|p\|_2 = \sqrt{\sum_{\ell=0}^{d-1} |p_\ell|^2}.$$

We first prove a more general result, which is a discrete version of the Wiener-Khinchin theorem [Wiener, ch. 4].

Theorem 4.0: Assume $w^b(z) = p(z)\overline{p}(1/z)$. Then

$$w_\ell^+ = \sum_{j=0}^{d-1} p_j \overline{p}_{(j+\ell) \bmod d} = \frac{1}{d} \sum_{k=0}^{d-1} |P_k|^2 \omega^{-\ell k}.$$

Proof: It follows from Lemma 4.1 that

$$\begin{aligned} w_\ell^+ &= \sum_{j=0}^{d-1} p_j \overline{p}_{(j+\ell) \bmod d} = \frac{1}{d} \sum_{k=0}^{d-1} p(\omega^k) \overline{p}(\omega^{-k}) \omega^{-\ell k} \\ &= \frac{1}{d} \sum_{k=0}^{d-1} p(\omega^k) \overline{p(\omega^k)} \omega^{-\ell k} = \frac{1}{d} \sum_{k=0}^{d-1} |p(\omega^k)|^2 \omega^{-\ell k} = \frac{1}{d} \sum_{k=0}^{d-1} |P_k|^2 \omega^{-\ell k} \quad \blacksquare \end{aligned}$$

Note: The coefficients w_ℓ^+ of $p(z)\overline{p}(1/z)$ are called the positive wrapped auto-correlation coefficents of p.

Corollary 4.0: (Parseval) $\|P\|_2 = \sqrt{d}\|p\|_2$.

Proof: The corollary follows from the previous theorem, with $\ell = 0$, since

$$\|P\|_2 = \sqrt{\sum_{k=0}^{d-1} |P_k|^2} = \sqrt{d \sum_{j=0}^{d-1} p_j \overline{p}_j} = \sqrt{d \sum_{j=0}^{d-1} |p_j|^2} = \sqrt{d}\|p\|_2 \quad \blacksquare$$

Note that this corollary is called Parseval's theorem due to its similarity to the result

$$\frac{1}{2\pi} \int_0^{2\pi} |f(\theta)|^2 d\theta = \sum_{j=-\infty}^{\infty} |f_j|^2,$$

for square-summable, 2π-periodic, functions and their Euler-Fourier coefficients (this can easily be proved, along the lines of section 3.6, when $f(\theta) = p(e^{i\theta})$, for a polynomial p).

5 Computation error analysis

The proof of Lemma 2.0 showed that the error of approximation, for a rational function by a polynomial, could be reduced to any required tolerance. That proof did not consider the implementation of the computation.

Our goal is to show that the same is true for the computation error, when multiple precision "floating point" numbers are utilized to represent the polynomial coefficients. Basic references for this section are [BiniPan86] and [BiniPan, ch. 6].

5.1 The triangle inequality and other 2-norm inequalities on $\mathbf{C}^d$

Throughout the e.-i. method computation error analysis, we make frequent use of the triangle inequality and other 2-norm inequalities. Most proofs of the triangle inequality in the vector space $\mathbf{C}^d$ use a complex inner product. So far, we have discussed only the 2-norm on $\mathbf{C}^d$, since we feel this is a substantially more elementary concept than the complex inner product. Therefore, we sketch a proof, though similar to the typical proofs, which utilizes only the 2-norm. We begin with the simplest case.

Recall that $\Re(z) = \frac{1}{2}(z + \overline{z})$ satisfies the inequality $\Re(z) \leq |z|$.

Lemma 5.0: (triangle inequality for $\mathbf{C}$) $|u + v| \leq |u| + |v|$.

Proof: Take square roots of all expressions:

$$\begin{aligned} |u+v|^2 &= (u+v)(\overline{u+v}) = (u+v)(\overline{u}+\overline{v}) = |u|^2 + u\overline{v} + v\overline{u} + |v|^2 \\ &= |u|^2 + (u\overline{v} + \overline{u\overline{v}}) + |v|^2 = |u|^2 + 2\Re(u\overline{v}) + |v|^2 \\ &\leq |u|^2 + 2|u||v| + |v|^2 = (|u| + |v|)^2 \quad \blacksquare \end{aligned}$$

Lemma 5.1: (similar to Cauchy et al. inequality) $\sum_{\ell=0}^{d-1} |u_\ell|\,|v_\ell| \le \|u\|_2\,\|v\|_2$.

Proof: Clearly, it suffices to prove $(\|u\|_2\,\|v\|_2)^2 - \left(\sum_{\ell=0}^{d-1} |u_\ell|\,|v_\ell|\right)^2 \ge 0$,

since this implies $\|u\|_2\,\|v\|_2 - \sum_{\ell=0}^{d-1} |u_\ell|\,|v_\ell| \ge 0$.

$$(\|u\|_2\,\|v\|_2)^2 - \left(\sum_{\ell=0}^{d-1} |u_\ell|\,|v_\ell|\right)^2 = \left(\sum_{\ell=0}^{d-1} |u_\ell|^2 \sum_{j=0}^{d-1} |v_j|^2\right) - \left(\sum_{\ell=0}^{d-1} |u_\ell|\,|v_\ell|\right)^2$$

$$= \sum_{\ell=0}^{d-1}\sum_{j=0}^{d-1} \left(|u_\ell|^2\,|v_j|^2 - |u_\ell|\,|v_\ell|\,|u_j|\,|v_j|\right)$$

$$= \sum_{\ell=0}^{d-1}\sum_{j=\ell+1}^{d-1} \left(|u_\ell|^2\,|v_j|^2 - 2\,|u_\ell|\,|v_\ell|\,|u_j|\,|v_j| + |u_j|^2\,|v_\ell|^2\right)$$

$$= \sum_{\ell=0}^{d-1}\sum_{j=\ell+1}^{d-1} (|u_\ell|\,|v_j| - |u_j|\,|v_\ell|)^2 \ge 0 \quad \blacksquare$$

Theorem 5.0: (triangle inequality for $\mathbf{C^d}$) $\|u+v\|_2 \le \|u\|_2 + \|v\|_2$.

Proof: Take square roots of all expressions:

$$\|u+v\|_2^2 = \sum_{\ell=0}^{d-1} |u_\ell + v_\ell|^2 \le \sum_{\ell=0}^{d-1} (|u_\ell| + |v_\ell|)^2$$

$$= \sum_{\ell=0}^{d-1} \left(|u_\ell|^2 + 2\,|u_\ell|\,|v_\ell| + |v_\ell|^2\right) = \|u\|_2^2 + 2\sum_{\ell=0}^{d-1} (|u_\ell|\,|v_\ell|) + \|v\|_2^2$$

$$\le \|u\|_2^2 + 2\,\|u\|_2\,\|v\|_2 + \|v\|_2^2 = (\|u\|_2 + \|v\|_2)^2,$$

where each inequality step was justified by the corresponding previous lemma $\blacksquare$

Other 2-norm inequalities on $\mathbf{C}^d$, for the error analysis of step (b) of the e.-i. method, will be used to estimate $\|U_k \diamond V_k\|_2$.

Lemma 5.2: (inequalities needed for e.-i. method, step (b) error estimates)

(a) $\|[U_k V_k]\|_2 \leq \|U\|_2 \, \|V\|_2$.

(b) For $\alpha > 0$, assume $|W_k| \leq \alpha \, |U_k|$, with $0 \leq k < d$.
Then $\|W\|_2 \leq \alpha \, \|U\|_2$.

Proof: Take square roots of all expressions in both equation sequences.

(a)
$$\|[U_k V_k]\|_2^2 = \sum_{k=0}^{d-1} |U_k|^2 \, |V_k|^2 \leq \sum_{k=0}^{d-1} \sum_{j=0}^{d-1} |U_k|^2 \, |V_j|^2$$
$$= \sum_{k=0}^{d-1} |U_k|^2 \sum_{j=0}^{d-1} |V_j|^2 = \|U\|_2^2 \, \|V\|_2^2 \, .$$

(b)
$$\|W\|_2^2 = \sum_{k=0}^{d-1} |W_k|^2 \leq \sum_{k=0}^{d-1} \alpha^2 \, |U_k|^2 = \alpha^2 \, \|U\|_2^2 \quad \blacksquare$$

Note: For $\alpha = \max\limits_{0 \leq k < d} \dfrac{1}{|V_k|}$, we have $\left| \dfrac{U_k}{V_k} \right| \leq \alpha \, |U_k|$, hence

$$\left\| \left[\frac{U_k}{V_k} \right] \right\|_2 < \max_{0 \leq k < d} \frac{1}{|V_k|} \, \|U\|_2 \, .$$

5.2 Results used in e.-i. method computation error analysis

Let φ be the precision, in binary digits, with which the multiple precision computations are implemented.

Define $\widetilde{w} :\cong u \diamond v$ to mean that the computation $u \diamond v$ has been implemented with at least φ binary digits precision, with the result assigned to the variable $\widetilde{w}$.

To provide examples of the notation used in e.-i. computation error analysis, we restate Algorithm 1.0, assuming that the computation is implemented with φ binary digits precision.

Algorithm 5.0:

{e.-i. method at dth roots of 1, using "floating point" computations}

{a} $\left[\widetilde{U}_k :\cong u(\omega^k)\right]$; $\left[\widetilde{V}_k :\cong v(\omega^k)\right]$;

{b} $\left[\widetilde{W}_k :\cong \widetilde{U}_k \diamond \widetilde{V}_k\right]$;

{c} $\left[\widetilde{w}_\ell :\cong (1/d)\widetilde{W}(\omega^{-\ell})\right]$ ∎

Analyzing the complexity of this e.-i. algorithm will now involve consideration of the precision φ with which the multiple precision computations are implemented.

We will estimate the computation error $\|\widetilde{w} - w\|_2$, utilizing the triangle inequality, by estimating the error generated at (and propagated from) each step separately:

$$\|\widetilde{w} - w\|_2 \leq \left\|\widetilde{w} - \left[\sum_{k=0}^{d-1} \widetilde{W}_k \omega^{-\ell k}\right]\right\|_2 \qquad \text{(a)}$$

$$+ \left\|\left[\sum_{k=0}^{d-1} \left(\widetilde{W}_k - \widetilde{U}_k \diamond \widetilde{V}_k\right) \omega^{-\ell k}\right]\right\|_2 \qquad \text{(b)}$$

$$+ \left\|\left[\sum_{k=0}^{d-1} \left(\widetilde{U}_k \diamond \widetilde{V}_k\right) \omega^{-\ell k}\right] - w\right\|_2 . \qquad \text{(c)}$$

For the computation error analysis of step (a) and step (c) of the e.-i. method, we use a proposition estimating the dft and inverse dft computation error bounds (see [BiniPan, ch. 3] for derivation of these bounds (or [GSande] for similar bounds)).

Let $\gamma(d) =$ [illegible] d.

Assume $\left[\widetilde{P}_k :\cong p(\omega^k)\right]$ and $\left[\widetilde{p}_\ell :\cong (1/d)P(\omega^{-\ell})\right]$.

Proposition 5.0: (dft and inverse dft computation error estimates)

(a) $$\|\widetilde{P} - P\|_2 \leq d\gamma(d)2^{-\varphi}\|p\|_2,$$

(b) $$\|\tilde{p} - p\|_2 \leq \gamma(d)2^{-\varphi}\|P\|_2 \quad \blacksquare$$

Corollary 5.0: Assume that $\varphi \geq \log_2(\sqrt{d}\gamma(d))$, then $\|\widetilde{P}\|_2 \leq 2\sqrt{d}\|p\|_2$.

Proof: By Proposition 5.0(a), the triangle inequality and Parseval's Theorem (Corollary 4.0), and using the condition on φ, we have

$$\|\widetilde{P}\|_2 \leq \|P\|_2 + d\gamma(d)2^{-\varphi}\|p\|_2 \leq \left(\sqrt{d} + d\gamma(d)2^{-\varphi}\right)\|p\|_2 \leq 2\sqrt{d}\|p\|_2 \quad \blacksquare$$

In the following inequalities, β will depend on the method used to implement complex multiplication or division (typically, $\beta \leq 4$).

Lemma 5.3: (more inequalities for e.-i. method, step (b) error estimate)

Assume that $\left|\widetilde{W}_k - \widetilde{U}_k \diamond \widetilde{V}_k\right| \leq \beta 2^{-\varphi}\left|\widetilde{U}_k \diamond \widetilde{V}_k\right|$, for $0 \leq k < d$.

Then (a) $\left\|\left[\widetilde{W}_k - \widetilde{U}_k \diamond \widetilde{V}_k\right]\right\|_2 \leq \beta 2^{-\varphi}\left\|\left[\widetilde{U}_k \diamond \widetilde{V}_k\right]\right\|_2$.

In addition, assume $\varphi > \log_2 \beta$, as will be done implicitly hereafter.

Then (b) $\left\|\widetilde{W}\right\|_2 < 2\left\|\left[\widetilde{U}_k \diamond \widetilde{V}_k\right]\right\|_2$.

Proof: (a) Use Lemma 5.2(b), with $\alpha = \beta 2^{-\varphi}$. (b) Use (a), triangle inequality and condition on φ $\blacksquare$

We now state a lemma which analyzes the e.-i. method computation error by analyzing each step separately, as indicated after Algorithm 5.0.

Lemma 5.4: (computation error estimates for each e.-i. method step)

$$\text{(a)} \qquad \left\| \widetilde{w} - \left[\sum_{k=0}^{d-1} \widetilde{W}_k \omega^{-\ell k} \right] \right\|_2 < 2\gamma(d) 2^{-\varphi} \left\| \left[\widetilde{U}_k \diamond \widetilde{V}_k \right] \right\|_2 .$$

$$\text{(b)} \qquad \left\| \left[\sum_{k=0}^{d-1} \left(\widetilde{W}_k - \widetilde{U}_k \diamond \widetilde{V}_k \right) \omega^{-\ell k} \right] \right\|_2 \leq \beta 2^{-\varphi} \left\| \left[\widetilde{U}_k \diamond \widetilde{V}_k \right] \right\|_2 .$$

$$\text{(c)} \qquad \left\| \left[\sum_{k=0}^{d-1} \left(\widetilde{U}_k \diamond \widetilde{V}_k \right) \omega^{-\ell k} \right] - w \right\|_2 \leq \frac{1}{\sqrt{d}} \left\| \left[\widetilde{U}_k \diamond \widetilde{V}_k - U_k \diamond V_k \right] \right\|_2 .$$

Proof: (a) Use Prop. 5.0(b) and Lemma 5.3(b). (b) Use Parseval and Lemma 5.3(a). (c) Use Parseval ∎

5.3 Computation error analysis of polynomial multiplication

Proposition 5.1: (e.-i. method computation error analysis illustrated)

Assume $\varphi \geq \log_2 \left(\max \left(\sqrt{d}\gamma(d), (11\gamma(d) + 4\beta)\, d(1/\varepsilon) \|u\|_2 \|v\|_2 \right) \right)$.

Then $\|\widetilde{w}^+ - w^+\|_2 \leq \varepsilon$.

Proof: Begin by using Lemma 5.4:

$$\begin{aligned}
\|\widetilde{w}^+ - w^+\|_2 &\leq (2\gamma(d) + \beta)\, 2^{-\varphi} \left\| \left[\widetilde{U}_k \widetilde{V}_k \right] \right\|_2 \\
&\quad + \frac{1}{\sqrt{d}} \left\| \left[\widetilde{U}_k \left(\widetilde{V}_k - V_k \right) + V_k \left(\widetilde{U}_k - U_k \right) \right] \right\|_2 \\
\text{(a)} \qquad &\leq (2\gamma(d) + \beta)\, 2^{-\varphi} \left\| \widetilde{U} \right\|_2 \left\| \widetilde{V} \right\|_2 \\
&\quad + \frac{1}{\sqrt{d}} \left(\left\| \widetilde{U} \right\|_2 \left\| \widetilde{V} - V \right\|_2 + \|V\|_2 \left\| \widetilde{U} - U \right\|_2 \right) \\
\text{(b)} \qquad &\leq (2\gamma(d) + \beta)\, 2^{-\varphi} \left(2\sqrt{d} \right)^2 \|u\|_2 \|v\|_2 \\
&\quad + \frac{1}{\sqrt{d}} \left(2\sqrt{d} + \sqrt{d} \right) d\gamma(d) 2^{-\varphi} \|u\|_2 \|v\|_2 \\
&= (11\gamma(d) + 4\beta)\, d 2^{-\varphi} \|u\|_2 \|v\|_2 .
\end{aligned}$$

Inequality (a) uses Lemma 5.2(a) three times and the triangle inequality, and inequality (b) uses Corollary 5.0 three times, Proposition 5.0(a) twice, and Parseval's theorem. Then with the assumption on φ,

$$\left\|\widetilde{w}^{+} - w^{+}\right\|_2 \leq \frac{(11\gamma(d) + 4\beta)\, d}{(11\gamma(d) + 4\beta)\, d(1/\epsilon)\, \|u\|_2\, \|v\|_2} \|u\|_2\, \|v\|_2 = \epsilon \quad \blacksquare$$

5.4 Computation error analysis of Algorithm 2.0

Note that throughout this section, H satisfies inequality (2.2), a relationship to the coefficients of t, from the approximation error analysis.

Lemma 5.5: (estimates derived from approximation error analysis)

(a) $\left\|\hat{t}\right\|_2 < \frac{3}{2} H^{d-1}$, (b) $\max\limits_{0 \leq k < d} \frac{1}{\left|\widehat{T}_k\right|} < \frac{2}{H^{d-1}}$.

(c) If $\varphi > 6d\gamma(d)$, then $\max\limits_{0 \leq k < d} \frac{1}{\left|\widehat{\widetilde{T}}_k\right|} < \frac{4}{H^{d-1}}$.

Proof: (a) Follows from the proof of Lemma 2.0, by taking square roots of all expressions:

$$\left\|\hat{t}\right\|_2^2 = \sum_{\ell=0}^{d-1} \left|\hat{t}_\ell\right|^2 \leq \sum_{\ell=0}^{d-1} \sum_{j=0}^{d-1} \left|\hat{t}_\ell\right| \left|\hat{t}_j\right| = \left(\sum_{\ell=0}^{d-1} \left|\hat{t}_\ell\right|\right)^2 < \left(\frac{3}{2} H^{d-1}\right)^2.$$

(b) Proved in Lemma 2.0. (c) Clearly, in general, $\max\limits_{0 \leq k < d} |P_k| \leq \|P\|_2$. Using Proposition 5.0(a), part (a) above, and the assumption on φ,

$$\begin{aligned} \max_{0 \leq k < d} \left|\widehat{\widetilde{T}}_k - \widehat{T}_k\right| \leq \left\|\widehat{\widetilde{T}} - \widehat{T}\right\|_2 &\leq d\gamma(d) 2^{-\varphi} \left\|\hat{t}\right\|_2 \\ &< \frac{3}{2} H^{d-1} d\gamma(d) 2^{-\varphi} < \frac{H^{d-1}}{4}. \end{aligned}$$

From the proof of Lemma 2.0, we know $\min\limits_{0 \leq k < d} \left|\widehat{T}_k\right| > \frac{H^{d-1}}{2}$.

Thus, from the triangle inequality,

$$\min_{0\le k<d}\left|\widetilde{\widehat{T}}_k\right| > \frac{H^{d-1}}{4} \quad \text{or} \quad \max_{0\le k<d}\frac{1}{\left|\widetilde{\widehat{T}}_k\right|} < \frac{4}{H^{d-1}} \quad \blacksquare$$

Proposition 5.2: (e.-i. method error analysis for Algorithm 2.0)

Assume $\varphi > \log_2(6d\gamma(d))$ and

$$\varphi > \log_2\left(\left(24d\gamma(d) + 20\sqrt{d}\gamma(d) + 8\sqrt{d}\beta\right) H^{d-1}(1/\epsilon)\left\|s^+\right\|_2\right).$$

Then $\left\|\widetilde{\hat{q}}^* - \hat{q}^*\right\|_2 < \epsilon$.

Proof: Begin by using Lemma 5.4:

$$\begin{aligned}
\left\|\widetilde{\hat{q}}^* - \hat{q}^*\right\|_2 &\le (2\gamma(d)+\beta)\,2^{-\varphi}\left\|\begin{bmatrix}\widetilde{\widehat{S}}_k^+ \\ \widetilde{\widehat{T}}_k\end{bmatrix}\right\|_2 \\
&\quad + \frac{1}{\sqrt{d}}\left\|\begin{bmatrix}\dfrac{\widetilde{\widehat{S}}_k^+ - S_k^+}{\widetilde{\widehat{T}}_k} + \dfrac{\widehat{S}_k^+(\widehat{T}_k - \widetilde{\widehat{T}}_k)}{\widehat{T}_k\widetilde{\widehat{T}}_k}\end{bmatrix}\right\|_2 \\
\text{(a)} \qquad &< \frac{4}{H^{d-1}}(2\gamma(d)+\beta)\,2^{-\varphi}\left\|\widetilde{\widehat{S}}^+\right\|_2 \\
&\quad + \frac{4}{\sqrt{d}H^{d-1}}\left(\left\|\widetilde{\widehat{S}}^+ - S^+\right\|_2 + \frac{2}{H^{d-1}}\left\|\widetilde{\widehat{S}}^+\right\|_2\left\|\widehat{T} - \widetilde{\widehat{T}}\right\|_2\right) \\
\text{(b)} \qquad &\le \frac{4}{H^{d-1}}(2\gamma(d)+\beta)\,2^{-\varphi}2\sqrt{d}\left\|\hat{s}^+\right\|_2 \\
&\quad + \frac{4}{\sqrt{d}H^{d-1}}d\gamma(d)2^{-\varphi}\left\|\hat{s}^+\right\|_2 \\
&\quad + \frac{4}{\sqrt{d}H^{d-1}}\frac{2}{H^{d-1}}2\sqrt{d}\left\|\hat{s}^+\right\|_2 d\gamma(d)2^{-\varphi}\left\|\hat{t}\right\|_2 \\
\text{(c)} \qquad &< \left((16\gamma(d)+8\beta)\sqrt{d} + 4\sqrt{d}\gamma(d) + 24d\gamma(d)\right)H^{d-1}2^{-\varphi}\left\|s^+\right\|_2 \\
&= \left(24d\gamma(d) + 20\sqrt{d}\gamma(d) + 8\sqrt{d}\beta\right)H^{d-1}2^{-\varphi}\left\|s^+\right\|_2
\end{aligned}$$

Inequality (a) uses the triangle inequality, the note after Lemma 5.2(b) three times, then Lemma 5.5(c) three times, Lemma 5.5(b) once, and Lemma 5.2(a) once.

Inequality (b) uses both Proposition 5.0(a) and Corollary 5.0 twice.

Inequality (c) uses Lemma 5.2(b) three times ($\alpha = H^{2(d-1)}$) and Lemma 5.5(a) once.

Then, with the assumption on φ,

$$\left\| \tilde{\hat{q}}^* - \hat{q}^* \right\|_2 < \frac{\left(24d\gamma(d) + 20\sqrt{d}\gamma(d) + 8\sqrt{d}\beta\right) H^{d-1}}{\left(24d\gamma(d) + 20\sqrt{d}\gamma(d) + 8\sqrt{d}\beta\right) H^{d-1}(1/\varepsilon) \left\| s^+ \right\|_2} \left\| s^+ \right\|_2$$

$$= \varepsilon \quad \blacksquare$$

5.5 Boolean complexity estimates

Since our computations are essentially the same as those of [Bini] and [Schönhage], our algorithm supports the Boolean complexity estimates of their algorithm. We refer to [BiniPan, ch. 6] or [BiniPan86] (utilizing the computation error analysis of the previous section), for further details.

6 A direction for future research and conclusion

We are currently studying an application of the e.-i. method to the problem of multivariate polynomial division with a remainder. We sketch this work to show that the methods of this dissertation may help to provide a direction for future research on multivariate polynomial division.

We first state the problem, which is a slight generalization of the univariate problem. Reasoning by analogy, we generalize several of our methods to derive a formula for the approximate quotient of two multivariate polynomials. Though our method is so far only proved to be valid in a special case, we sketch that proof, a generalization of Lemma 2.0.

6.1 Multivariate polynomial division with a remainder

Let $\mathbf{z} = (z_1, \ldots, z_n)$ be an element of $\mathbf{C}^n$ (an n-dimensional vector over $\mathbf{C}$), and let $\mathbf{j} = (j_1, \ldots, j_n)$, $\mathbf{k} = (k_1, \ldots, k_n)$ and $\mathbf{l} = (\ell_1, \ldots, \ell_n)$ be index vectors.

We briefly state a formulation of the problem of multivariate polynomial division with a remainder, over the field $\mathbf{C}$ of complex numbers:

Given the coefficients of two multivariate polynomials s and t,

$$s(\mathbf{z}) = \sum_{j_1=0}^{\mu_1} \cdots \sum_{j_n=0}^{\mu_n} s_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n} \quad \text{and} \quad t(\mathbf{z}) = \sum_{j_1=0}^{\nu_1} \cdots \sum_{j_n=0}^{\nu_n} t_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

with total $\deg s = \mu_1 + \cdots + \mu_n = \mathrm{M}$, total $\deg t = \nu_1 + \cdots + \nu_n = \mathrm{N}$, and $\mu_1 \geq \nu_1, \cdots, \mu_n \geq \nu_n$, compute the coefficients of the quotient q,

$$q(\mathbf{z}) = \sum_{\mathbf{j} | J=0}^{J=M-N} q_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

with $J = j_1 + \cdots + j_n$, so that the remainder,

$$r(\mathbf{z}) = \sum_{\mathbf{j}|J=0}^{J=N-1} r_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

defined by the equation

$$r(\mathbf{z}) = s(\mathbf{z}) - q(\mathbf{z}) \times t(\mathbf{z}), \tag{6.0}$$

satisfies the condition $r_{\mathbf{j}} = 0$ for $J \geq N$.

Note that we allow the solution to be a finite power series (some $j < 0$ are allowed), so that the condition $r_{\mathbf{j}} = 0$, for $J \geq N$, may be satisfied. With the customary definition, where the solution is a polynomial in the ordered variables, the remainder satisfies a weaker condition (see, for instance, [DST]).

We illustrate the two definitions with an example.

Assume

$$s(z,w) = s_{22}z^2w^2 + s_{21}z^2w + s_{12}zw^2 + s_{20}z^2 + s_{11}zw + s_{02}w^2,$$
$$t(z,w) = zw + t_{10}z + t_{01}w + t_{00}.$$

Then, using our definition, we obtain

$$\begin{aligned} q(z,w) = {} & s_{22}zw + (s_{21} - s_{22}t_{10})z + (s_{12} - s_{22}t_{01})w \\ & + (s_{20} - s_{21}t_{10} + s_{22}t_{10}^2)zw^{-1} \\ & + (s_{11} - s_{21}t_{01} - s_{12}t_{10} - s_{22}t_{00} + 2s_{22}t_{10}t_{01}) \\ & + (s_{02} - s_{12}t_{01} + s_{22}t_{01}^2)z^{-1}w, \end{aligned}$$

$$
\begin{aligned}
r(z,w) = &-(s_{20}t_{10} - s_{21}t_{10}^2 + s_{22}t_{10}^3)z^2w^{-1} \\
&- (s_{11}t_{10} + s_{20}t_{01} + s_{21}t_{00} - s_{12}t_{10}^2 \\
&\quad - 2s_{21}t_{10}t_{01} - 2s_{22}t_{10}t_{00} + 3s_{22}t_{01}t_{10}^2)z \\
&- (s_{11}t_{01} + s_{02}t_{10} + s_{12}t_{00} - s_{21}t_{01}^2 \\
&\quad - 2s_{12}t_{10}t_{01} - 2s_{22}t_{01}t_{00} + 3s_{22}t_{10}t_{01}^2)w \\
&- (s_{02}t_{01} - s_{12}t_{01}^2 + s_{22}t_{01}^3)z^{-1}w^2 \\
&- (s_{20}t_{00} - s_{21}t_{10}t_{00} + s_{22}t_{10}^2t_{00})zw^{-1} \\
&- (s_{11}t_{00} - s_{21}t_{01}t_{00} - s_{12}t_{10}t_{00} - s_{22}t_{00}^2 + 2s_{22}t_{10}t_{01}t_{00}) \\
&- (s_{02}t_{00} - s_{12}t_{10}t_{00} + s_{22}t_{01}^2t_{00})z^{-1}w.
\end{aligned}
$$

Note that r satisfies equation (6.0), and that $r_{(j_1,j_2)} = 0$ for $j_1 + j_2 \geq 2$ $(=$ total deg $t)$.

It may be instructive to rewrite this solution in a different form, in preparation for computation of q in a more general case:

$$
\begin{aligned}
q_{(1,1)} &= s_{(2,2)}, \\
q_{(1,0)} &= s_{(2,1)} - q_{(1,1)}t_{(1,0)}, \\
q_{(0,1)} &= s_{(1,2)} - q_{(1,1)}t_{(0,1)}, \\
q_{(1,-1)} &= s_{(2,0)} - q_{(1,0)}t_{(1,0)}, \\
q_{(0,0)} &= s_{(1,1)} - q_{(0,1)}t_{(1,0)} - q_{(1,0)}t_{(0,1)} - q_{(1,1)}t_{(0,0)}, \\
q_{(-1,1)} &= s_{(0,2)} - q_{(0,1)}t_{(0,1)}.
\end{aligned}
$$

In contrast, utilizing the customary definition, we obtain

$$
\begin{aligned}
q(z,w) = &\, s_{22}zw + (s_{21} - s_{22}t_{10})z + (s_{12} - s_{22}t_{01})w \\
&+ (s_{11} - s_{21}t_{01} - s_{12}t_{10} - s_{22}t_{00} + 2s_{22}t_{10}t_{01}),
\end{aligned}
$$

$$\begin{aligned} r(z, u_r(z,w)) = &-(s_{21}t_{10} - s_{22}t_{10}^2)z^2 - (s_{12}t_{01} - s_{22}t_{01}^2)w^2 \\ &- (s_{11}t_{10} + s_{21}t_{00} - s_{12}t_{10}^2 - s_{21}t_{10}t_{01} - 2s_{22}t_{10}t_{00} + 2s_{22}t_{01}t_{10}^2)z \\ &- (s_{11}t_{01} + s_{12}t_{00} - s_{21}t_{01}^2 - s_{12}t_{10}t_{01} - 2s_{22}t_{01}t_{00} + 2s_{22}t_{10}t_{01}^2)w \\ &- (s_{11}t_{00} - s_{21}t_{01}t_{00} - s_{12}t_{10}t_{00} - s_{22}t_{00}^2 + 2s_{22}t_{10}t_{01}t_{00}). \end{aligned}$$

Note that this remainder does not satisfy the condition that our example does (since r_{20} and r_{02} are non-zero), because of the restriction to non-negative exponents in the quotient.

6.2 Multivariate discrete Fourier transforms

Reasoning by analogy from the derivation of section 1.3, we derive the multivariate inverse discrete Fourier transform from the multivariate Lagrange interpolation formula.

Assume that we have a function f of n variables that has been specified at d^n particular points. The multivariate Lagrange interpolation formula provides a polynomial p in n variables that agrees with f at the specified values.

With $f(\omega^{k_1}, \ldots, \omega^{k_n}) = A_{\mathbf{k}}$ for $0 \le k_1 < d, \ldots, 0 \le k_n < d$, we obtain

$$p(\mathbf{z}) = \sum_{k_1=0}^{d-1} \cdots \sum_{k_n=0}^{d-1} A_{\mathbf{k}} \prod_{\substack{j_1=0 \\ j_1 \neq k_1}}^{d-1} \frac{z_1 - \omega^{j_1}}{\omega^{k_1} - \omega^{j_1}} \cdots \prod_{\substack{j_n=0 \\ j_n \neq k_n}}^{d-1} \frac{z_n - \omega^{j_n}}{\omega^{k_n} - \omega^{j_n}},$$

satisfying $p_{\mathbf{k}} := p(\omega^{k_1}, \ldots, \omega^{k_n}) = A_{\mathbf{k}}$ for $0 \le k_1 < d, \ldots, 0 \le k_n < d$.

Following the derivation in section 1.3, for each variable $z_1, \ldots, z_n$,

$$p(\mathbf{z}) = \sum_{k_1=0}^{d-1} \cdots \sum_{k_n=0}^{d-1} A_{\mathbf{k}} \left(\frac{1}{d} \sum_{\ell_1=0}^{d-1} \omega^{-k_1\ell_1} z_1^{\ell_1} \right) \cdots \left(\frac{1}{d} \sum_{\ell_n=0}^{d-1} \omega^{-k_n\ell_n} z_n^{\ell_n} \right).$$

Multiplying out and interchanging summations yields

$$p(\mathbf{z}) = \sum_{\mathbf{l}} \left(\frac{1}{d^n} \sum_{\mathbf{k}} A_{\mathbf{k}} \omega^{-\mathbf{k}\cdot\mathbf{l}} \right) z_1^{\ell_1} \cdots z_n^{\ell_n},$$

for $0 \le k_1 < d, \ldots, 0 \le k_n < d$ and $0 \le \ell_1 < d, \ldots, 0 \le \ell_n < d$, with $\mathbf{k} \cdot \mathbf{l}$ the inner product of the two index vectors.

We conclude, in analogy to section 1.3, that the coefficients of the solution polynomial provided by the multivariate Lagrange interpolation formula coincide with the multivariate inverse discrete Fourier transform formula.

6.3 Approximating the quotient of two multivariate polynomials

We concentrate on the problem of approximating the quotient of two multivariate polynomials, with certain simplifying assumptions:

Given the coefficients of two multivariate polynomials s and t over the field $\mathbf{C}$,

$$s(\mathbf{z}) = \sum_{j_1=0}^{\mu} \cdots \sum_{j_n=0}^{\mu} s_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

$$t(\mathbf{z}) = \sum_{j_1=0}^{\nu} \cdots \sum_{j_n=0}^{\nu} t_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

$M = n\mu$, $N = n\nu$, $\mu \ge \nu$, $t_{(\nu,\ldots,\nu)} = 1$, compute the coefficients of q^*, an approximate quotient,

$$q^*(\mathbf{z}) = \sum_{\mathbf{j} \mid J=0}^{J=M-N} q^*_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

so that

$$s(\mathbf{z}) \approx q^*(\mathbf{z}) \times t(\mathbf{z}).$$

Reasoning by analogy, we start with equation (6.0), solved for $q(\mathbf{z})$,

$$q(\mathbf{z}) = \frac{s(\mathbf{z})}{t(\mathbf{z})} - \frac{r(\mathbf{z})}{t(\mathbf{z})}. \tag{6.1}$$

We know that as $|\mathbf{z}| \to \infty$, $r(\mathbf{z})/t(\mathbf{z}) \to 0$, since $r_l = 0$ for $L \geq N$.

To construct our solution, we interpolate q^* at $(H\omega^{k_1}, \ldots, H\omega^{k_n})$, for $0 \leq k_1 < \kappa, \ldots, 0 \leq k_n < \kappa$, with $\kappa = \mu - \nu + 1$. Utilizing our analogy with Theorem 1.0, proved along the lines of the previous section, we obtain an approximate quotient q^*, with coefficients

$$q_{\mathbf{j}}^* = \frac{1}{H^J \kappa^n} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} \frac{s(H\omega^{k_1}, \ldots, H\omega^{k_n})}{t(H\omega^{k_1}, \ldots, H\omega^{k_n})} \omega^{-\mathbf{k}\cdot\mathbf{j}}, \tag{6.2}$$

for all $\mathbf{j}$ satisfying $0 \leq J \leq M - N$.

We will sketch a proof that this approximate quotient q^* converges to q in the following lemma. In order to simplify the proof, let $H = 2^h$ for $h > 0$.

Lemma 6.0: $|q_{\mathbf{j}}^* - q_{\mathbf{j}}| = O(1/H^{J+1})$, for all $\mathbf{j}$ satisfying $0 \leq J \leq M - N$.

Proof sketch: Using equation (6.1) and the generalization of Corollary 1.0, we have a formula for $q_{\mathbf{j}}$,

$$q_{\mathbf{j}} = \frac{1}{\kappa^n H^J} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} q(H\omega^{k_1}, \ldots, H\omega^{k_n}) \omega^{-\mathbf{k}\cdot\mathbf{j}}.$$

Similarly, using the analogy of Theorem 1.0, with

$$A_{\mathbf{k}} = \frac{s(H\omega^{k_1}, \ldots, H\omega^{k_n})}{t(H\omega^{k_1}, \ldots, H\omega^{k_n})},$$

we obtain the approximation formula (6.2) for q^*.

Assume $\tau_h \leq \min_{\mathbf{k}} |t(H\omega^{k_1}, \ldots, H\omega^{k_n})|$, and

$$\varrho \geq \max_{0 \leq J' < N} \sum_{\mathbf{j} | J = J'} |r_{\mathbf{j}}|, \quad \text{(independent of } H\text{)}.$$

Then

$$
\begin{aligned}
|q_{\mathbf{j}}^* - q_{\mathbf{j}}| &= \left| \frac{1}{H^J \kappa^n} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} \frac{r(H\omega^{k_1}, \ldots, H\omega^{k_n})}{t(H\omega^{k_1}, \ldots, H\omega^{k_n})} \omega^{-\mathbf{k}\cdot\mathbf{j}} \right| \\
&\leq \frac{1}{H^J \kappa^n} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} \frac{|r(H\omega^{k_1}, \ldots, H\omega^{k_n})|}{|t(H\omega^{k_1}, \ldots, H\omega^{k_n})|} \\
&\leq \frac{1}{H^J \kappa^n \tau_h} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} |r(H\omega^{k_1}, \ldots, H\omega^{k_n})| \\
&\leq \frac{1}{H^J \kappa^n \tau_h} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} \left(\sum_{|\mathbf{l}|_L=0}^{L=N-1} |r_{\mathbf{l}}| \, H^L \right) \\
&\leq \frac{\varrho}{H^J \tau_h} \sum_{L=0}^{N-1} H^L = \frac{\varrho}{H^J \tau_h} \frac{H^N - 1}{H - 1}.
\end{aligned}
$$

We have reduced the proof of this lemma to showing $1/\tau_h = O(1/H^N)$. With $T_{L'} = \sum_{|\mathbf{l}|_L = L'} |t_{\mathbf{l}}|$, make the additional assumption that

$$
2 \sum_{L=0}^{N-1} T_L H^L < H^N. \tag{6.3}
$$

For instance, this assumption will be satisfied when $H > 3 \max_{0 \leq L < N} T_L^{1/(N-L)}$, as we showed with assumption (2.2) of section 2.1.

From the definition of t, we know $t_N = 1$. From the triangle inequality and the assumption (6.3), we have

$$
|t(H\omega^{k_1}, \ldots, H\omega^{k_n})| \geq H^N - \sum_{L=0}^{N-1} T_L H^L > H^N/2.
$$

This implies $\min_{\mathbf{k}} |t(H\omega^{k_1}, \ldots, H\omega^{k_n})| > H^N/2$. Hence $1/\tau_h = O(1/H^N)$. Thus, as in section 2.1,

$$
|q_{\mathbf{j}}^* - q_{\mathbf{j}}| = O(1/H^{J+1}).
$$

All that remains in the proof of Lemma 6.0 is an elementary proof of the relationship between the special condition $t_{(\nu,\ldots,\nu)} = 1$ and the existence of $r(\mathbf{z})$ satisfying (6.0), with $r_{\mathbf{j}} = 0$ for $j \geq N$. It can be proved in a fairly straightforward way, using functions of several complex variables (see next section), but we would like to sketch an algebraic proof.

Assume s and t are defined as in section 6.3, with an additional simplifying assumption that $\mu = 2\nu$, so that $M = 2N$. For index vectors $\mathbf{j}$ and $\mathbf{k}$, with $j_1 + \cdots + j_n = J$ and $k_1 + \cdots + k_n = K$, assume

$$-\nu \leq j_1 \leq \nu, \ldots, -\nu \leq j_n \leq \nu, \quad \text{with } 0 \leq J \leq N,$$
$$0 \leq k_1 \leq \nu, \ldots, \quad 0 \leq k_n \leq \nu, \quad \text{with } 0 \leq K \leq N.$$

Then, we construct a quotient using the recurrence relations:

$$q_{\mathbf{j}} = s_{(\nu,\ldots,\nu)+\mathbf{j}} - \sum_{\mathbf{k}|K=0}^{K=N-1} q_{(\nu,\ldots,\nu)+\mathbf{j}-\mathbf{k}}\, t_{\mathbf{k}}, \tag{6.4}$$

obtained by solving the convolution equation

$$\sum_{\mathbf{k}|K=0}^{K=N} q_{(\nu,\ldots,\nu)+\mathbf{j}-\mathbf{k}}\, t_{\mathbf{k}} = s_{(\nu,\ldots,\nu)+\mathbf{j}}$$

for $q_{\mathbf{j}}$, using the assumption $t_{(\nu,\ldots,\nu)} = 1$ from section 6.3.

With this quotient q, we can show that r, defined as in equation (6.0), satisfies the remainder condition $r_{\mathbf{j}} = 0$ for $J \geq N$.

Note that, with $n = 2$ and $\nu = 1$, equations (6.4) reduce to the recurrence relations of the example in section 6.1.

For now, we will not complete the algebraic proof, but we can say that equation (6.4) and the example of section 6.1 will be useful in a formal induction proof.

6.4 Complex integration and multivariate polynomial division

We state two basic lemmas concerning rational functions of several complex variables (see, for instance, [BochnerM]), and then state a lemma which is a simplified version of the multivariate Cauchy integral formula. In these lemmas, and subsequently, we assume that for $C \times \cdots \times C$, the domain of integration in $\mathbf{C^n}$, the path of integration in each complex dimension is a circle centered at the origin. No proofs are provided, especially due to the close similarity with the lemmas of section 3.6.

In order to derive multivariate polynomial coefficient formulas, we need to integrate multivariate power functions on $C \times \cdots \times C$ in $\mathbf{C^n}$.

Lemma 6.1: Let C_j be circles, and $\mathbf{l}$ an index vector. Then

$$\oint_{C_n} \cdots \oint_{C_1} z_1^{\ell_1} \cdots z_n^{\ell_n} \, dz_1 \cdots dz_n = \begin{cases} (2\pi i)^n, & \text{all } \ell = -1, \\ 0, & \text{some } \ell \neq -1. \end{cases}$$

In order to estimate approximation errors, we need an inequality concerning integrals of rational functions on $C \times \cdots \times C$ in $\mathbf{C^n}$.

Lemma 6.2: Let C_j be circles of radius $H = 2^h$, and s and t be multivariate polynomials. Assume $|s(\mathbf{z})| \leq \sigma_h$ and $|t(\mathbf{z})| \geq \tau_h$ for all $\mathbf{z}$ on $C_1 \times \cdots \times C_n$. Then

$$\left| \oint_{C_n} \cdots \oint_{C_1} \frac{s(\mathbf{z})}{t(\mathbf{z})} \, dz_1 \cdots dz_n \right| \leq (2\pi H)^n \sigma_h / \tau_h.$$

We need a formula relating a multivariate polynomial and its coefficients using complex integration.

Lemma 6.3: Let p be a multivariate polynomial, with total deg $= D-1$, and $\mathbf{j}$ an index vector,

$$p(\mathbf{z}) = \sum_{\mathbf{j}|J=0}^{J=D-1} p_{\mathbf{j}} z_1^{j_1} \cdots z_n^{j_n},$$

with each C_ℓ in $C_1 \times \cdots \times C_n$ a circle centered at the origin. Then

$$p_{\mathbf{j}} = \frac{1}{(2\pi i)^n} \oint_{C_n} \cdots \oint_{C_1} \frac{p(\mathbf{z})}{z_1^{j_1+1} \cdots z_n^{j_n+1}} dz_1 \cdots dz_n.$$

As in section 3.6, we may deduce a special case of the multivariate Euler-Fourier coefficient formula from the previous lemma. If

$$f(\boldsymbol{\Theta}) = \sum_{\mathbf{j}|J=0}^{J=D-1} f_{\mathbf{j}} e^{i(\mathbf{j}\cdot\boldsymbol{\Theta})},$$

then

$$f_{\mathbf{j}} = \frac{1}{(2\pi)^n} \int_0^{2\pi} \cdots \int_0^{2\pi} f(\boldsymbol{\Theta}) e^{-i(\mathbf{j}\cdot\boldsymbol{\Theta})} d\theta_1 \cdots d\theta_n.$$

We would now like to take a second look at Lemma 6.0, this time from the point of view of complex integration. In analogy with section 3.7, we wish to show that as $H \to \infty$,

$$q_{\mathbf{j}}^* = \frac{1}{(2\pi i)^n} \oint_{C_n} \cdots \oint_{C_1} \frac{s(\mathbf{z})}{z_1^{j_1+1} \cdots z_n^{j_n+1} t(\mathbf{z})} dz_1 \cdots dz_n \to q_{\mathbf{j}},$$

by proving the following lemma:

Lemma 6.4:

$$|q_{\mathbf{j}}^* - q_{\mathbf{j}}| = \left| \frac{1}{(2\pi i)^n} \oint_{C_n} \cdots \oint_{C_1} \frac{r(\mathbf{z})}{z_1^{j_1+1} \cdots z_n^{j_n+1} t(\mathbf{z})} dz_1 \cdots dz_n \right|$$
$$= O(1/H^{J+1}).$$

Proof sketch: Use Lemma 6.2 ($\tau_h = \min_{\Theta} |t(He^{i\theta_1}, \ldots, He^{i\theta_n})|$ and ϱ as in Lemma 6.0):

$$|q_{\mathbf{j}}^* - q_{\mathbf{j}}| = \left| \frac{1}{(2\pi i)^n} \oint_{C_n} \cdots \oint_{C_1} \frac{r(\mathbf{z})}{z_1^{j_1+1} \cdots z_n^{j_n+1} t(\mathbf{z})} dz_1 \cdots dz_n \right|$$
$$\leq \frac{1}{H^J \tau_h} \max_{\Theta} |r(He^{i\theta_1}, \ldots, He^{i\theta_n})|$$
$$\leq \frac{1}{H^J \tau_h} \sum_{|\mathbf{l}|L=0}^{L=N-1} |r_{\mathbf{l}}| H^L$$
$$\leq \frac{\varrho}{H^J \tau_h} \sum_{L=0}^{N-1} H^L = O(1/H^{J+1}),$$

as in Lemma 6.0.

Finally, in analogy with Section 3.8, we show that the e.-i. method for approximating the quotient of two multivariate polynomials is equivalent to the complex integration method.

Let $\mathbf{z} = (He^{i\theta_1}, \ldots, He^{i\theta_n})$, $0 \leq \theta_\ell \leq 2\pi$. Then

$$q_{\mathbf{j}}^* = \frac{1}{(2\pi i)^n} \oint_{C_n} \cdots \oint_{C_1} \frac{s(\mathbf{z})}{z_1^{j_1+1} \cdots z_n^{j_n+1} t(\mathbf{z})} dz_1 \cdots dz_n$$
$$= \frac{1}{(2\pi)^n H^J} \int_0^{2\pi} \cdots \int_0^{2\pi} \frac{s(He^{i\theta_1}, \ldots, He^{i\theta_n})}{t(He^{i\theta_1}, \ldots, He^{i\theta_n})} e^{-i(\mathbf{j} \cdot \boldsymbol{\Theta})} d\theta_1 \cdots d\theta_n$$
$$\approx \frac{1}{\kappa^n H^J} \sum_{k_1=0}^{\kappa-1} \cdots \sum_{k_n=0}^{\kappa-1} \frac{s(He^{2\pi i k_1/\kappa}, \ldots, He^{2\pi i k_n/\kappa})}{t(He^{2\pi i k_1/\kappa}, \ldots, He^{2\pi i k_n/\kappa})} e^{-(2\pi i/\kappa)(\mathbf{k} \cdot \mathbf{j})},$$

which results in the same formula as (6.2), with $\omega = e^{2\pi i/\kappa}$, utilizing numerical integration of the multivariate Euler-Fourier coefficient formula.

6.5 Conclusion

We feel that our derivation of the Bini-Schönhage algorithm for univariate polynomial division with remainder substantially simplifies the presentation of this algorithm, by relying on the elementary Lagrange interpolation formula. In this way, we avoid both complex integration and complex inner product (unitary) spaces, requiring substantially more complicated mathematics.

We also hope that these methods will be useful in further study of multivariate polynomial division.

Bibliography

[Bini] Bini, D., Parallel Solution of Certain Toeplitz Linear Systems, *SIAM J. of Computing* **13**, 268–276, 1984, also *I.E.I. of C.N.R.*, Tech. Report B82-04, Pisa, 1982.

[BiniPan86] Bini, D., and Pan, V., Polynomial Division and Its Computational Complexity, *J. of Complexity* **2**, 179–203, 1986.

[BiniPan] Bini, D., and Pan, V.Y., *Polynomial and Matrix Computations* vol. 1, Birk-häuser, Boston, 1994, vol. 2 (to appear).

[BochnerM] Bochner, S., and Martin, W.T. *Several Complex Variables*, Princeton University, Princeton, 1948.

[Cooley] Cooley, J., Lanczos and the FFT: A Discovery Before its Time, *Proceed. Lanczos Int'l Centenary Conf.*, 3–9, SIAM, 1994.

[DST] Davenport, J.H., Siret, Y., and Tournier, E., *Computer Algebra: Systems and Algorithms for Algebraic Computations*, Academic, New York, 1988.

[GSande] Gentleman, W., and Sande, G., Fast Fourier Transforms for Fun and Profit, *Proceed. AFIPS Fall Joint Computer Conf.* **29**, 563–578, 1966.

[GohbergO] Gohberg, I., and Olshevsky, V., Complexity of Multiplication with Vectors and Structured Matrices, *Lin. Algebra Appl.* **202**, 163–192, 1994.

[Knopp] Knopp, K. *Theory of Functions* vol. 1, Dover, New York, 1945.

[Lagrange] Lagrange, J., Recherches sur la nature, et la propagation du son (1759), *Ouvres* **1**, 39–148, Paris, 1867.

[PanSL] Pan, V., Sadikou, A., and Landowne, E., Polynomial division with a remainder by means of evaluation and interpolation, *Info. Process. Lett.* **44**, 149–153, 1992, also *Proceed. 3rd Symp. on Parallel and Distrib. Process.*, 212–218, IEEE Computer Society, 1991.

[PanSLT] Pan, V., Sadikou, A., Landowne, E., and Tiga, O., A New Approach to Fast Polynomial Interpolation and Multipoint Evaluation, *Computers Math. Applic.* **25**, 25–30, 1993.

[Schönhage] Schönhage, A., Asymptotically Fast Algorithms for the Numerical Multiplication and Division of Polynomials with Complex Coefficients, *Proceed. EuroCAM-82*, LNCS **144**, 3–15, Springer, Berlin, 1982.

[Toom] Toom, A., The Complexity of a Scheme of Functional Elements Realizing the Multiplication of Integers, *Soviet Math. Doklady* **3**, 714–716, 1963.

[Wiener] Wiener, N., *The Fourier Integral and Certain of Its Applications*, Dover, New York, 1958.