Fast modular composition using spiroids*†

JORIS VAN DER HOEVEN[a], GRÉGOIRE LECERF[b]

Laboratoire d'informatique de l'École polytechnique (LIX, UMR 7161)
CNRS, École polytechnique, Institut Polytechnique de Paris
Bâtiment Alan Turing, CS35003
1, rue Honoré d'Estienne d'Orves
91120 Palaiseau, France

a. Email: vdhoeven@lix.polytechnique.fr
b. Email: lecerf@lix.polytechnique.fr

Preliminary version of November 24, 2025

Given three univariate polynomials P, Q, and R with coefficients in a prime finite field, we present a new algorithm for computing $P \circ Q$ modulo R in time close to linear and with an asymptotic complexity smaller than the one of the Kedlaya–Umans algorithm. As a novelty, our method mostly performs fast floating point Fourier transforms, while previously known ones rely on ad hoc algebraic constructions of finite fields.

1. INTRODUCTION

Let $\mathbb{A}$ be an effective commutative ring, so that we have algorithms for the ring operations. Given a monic polynomial $R \in \mathbb{A}[x]$ of degree $D \geqslant 1$ and polynomials P and Q in $\mathbb{A}[x]$ of degree $<D$ the computation of the remainder of $P \circ Q$ in the division by R, written $P \circ Q \operatorname{rem} R$, is called the problem of *modular composition*. Modular composition is a central operation in computer algebra, especially for irreducible polynomial factorization; see [12] for instance. It is still unknown whether modular composition can be achieved in time nearly linear in D or not for any ground ring $\mathbb{A}$.

1.1. Main result

In this paper, we prove the following new complexity bound for when $\mathbb{A}$ is a finite ring $\mathbb{Z}/r\mathbb{Z}$ and where $\tilde{O}(\Phi)$ is a common abbreviation for $O(\Phi(\log \Phi)^{O(1)})$.

THEOREM 1. *The bit cost of degree D modular composition over $\mathbb{Z}/r\mathbb{Z}$ is bounded by*

$$D^{5/\sqrt{\log D}}\, \tilde{O}(D \log r).$$

This result improves upon the best previously known bound

$$D^{\left(28\frac{\log\log D}{\log D}\right)^{1/2}\left(1+O\left(\frac{1}{\log\log D}\right)\right)}\, \tilde{O}(D \log r)$$

*. Grégoire Lecerf has been supported by the French ANR-22-CE48-0016 NODE project. Joris van der Hoeven has been supported by an ERC-2023-ADG grant for the ODELIX project (number 101142171).
Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

†. This article has been written using GNU TEXMACS [8].

given in [11, Theorem 4], which is derived from an algorithm due to Kedlaya and Umans [14]. The novelty of our approach is a reduction of modular composition to the floating point evaluation of a multivariate polynomial at a special set of points, a so-called *spiroid*. We take advantage of fast Fourier transforms to perform these evaluations. In contrast, previously known methods, in the vein of [14], rely on number theoretic constructions.

1.2. Related work

Let $\mathsf{M}(D)$ denote a cost function that bounds the number of operations in $\mathbb{K}$ required to multiply two polynomials of degree $\leqslant D$ in $\mathbb{K}[x]$. Over any ring $\mathbb{A}$, modular composition can be performed with $\tilde{O}(D\,\mathsf{M}(D))$ operations in $\mathbb{A}$ by applying Horner's rule to evaluate $P(Q(x))$ in $\mathbb{A}[x]/(R(x))$. In 1978, Brent and Kung [3] gave a faster algorithm with cost $O(D^2+\sqrt{D}\,\mathsf{M}(D))$, that uses the *baby-step giant-step* technique [17]. Their algorithm even yielded a sub-quadratic cost $O(D^{\omega_2/2}+\sqrt{D}\,\mathsf{M}(D))$ when combined with fast linear algebra; see [13, p. 185]. Here, the constant ω_2 is a real value between 3 and 4 such that the product of a $n\times n^2$ matrix by a $n^2\times n$ matrix takes $O(n^{\omega_2})$ operations; one may take $\omega_2<3.250385$ according to [19]. At present time, the fastest known modular composition method over any ground field $\mathbb{K}$ is due to Neiger, Salvy, Schost, and Villard [16]: it is probabilistic of Las Vegas type and takes an expected number of

$$\tilde{O}(D^{\kappa}),\ \text{where } \kappa := 1+\frac{1}{\frac{1}{\omega-1}+\frac{2}{\omega_2-2}}$$

operations in $\mathbb{K}$; see [16, Theorem 1.1]. Here, the constant ω denotes any real value between 2 and 3 such that two $n\times n$ matrices over a commutative ring can be multiplied with $O(n^{\omega})$ ring operations. The current best known value is $\omega<2.371552$ [19], so we may take $\kappa<1.43$.

A major breakthrough for the modular composition problem is due to Kedlaya and Umans [14] in the case where $\mathbb{A}$ is a finite field $\mathbb{F}_q$ (and even more generally a finite ring of the form $(\mathbb{Z}/r\mathbb{Z})[z]/(\theta(z))$ for any integer r and θ monic). For any fixed real value $\varepsilon>0$, they showed that $P\circ Q \operatorname{rem} R$ can be computed using $O((D\log q)^{1+\varepsilon})$ bit operations. The dependency in ε is analyzed in [11]. Recent improvements of the Kedlaya–Umans approach can be found in [1, 2], but the dependency in ε is not detailed.

In [9, 10] it was shown how the knowledge of factorizations of R can be exploited to speed up modular composition. An important special case concerns the composition of power series, which corresponds to taking $R(x)=x^D$. Recently, Kinoshita and Li [15] showed how to accomplish this task using $O(\mathsf{M}(D)\log D)$ operations in $\mathbb{K}$.

1.3. Overview of the paper

Our modular composition algorithm will be presented in section 5: Theorem 1 follows from the sharper complexity bound given in Theorem 24. As in [14], the problem will be reduced to the multi-point evaluation of a multivariate polynomial, using Kronecker segmentation. But, instead of relying on finite field arithmetic, we benefit from floating point arithmetic and reduce the multi-point evaluation to a deformation of the fast Fourier transform. The deformed evaluation point set is presented in section 3. Corresponding evaluation and interpolation algorithms are given in section 4. The next section gathers prerequisites.

2. PREREQUISITES

For complexity analyses, we will consider bit complexity models such as computation trees or RAM machines [4, 5]. The function $\mathsf{I}(m)$ will bound the bit cost for multiplying two integers of bit size $\leqslant m$. We will assume that $\mathsf{I}(m)/m$ is nondecreasing. At the end, we will make use of the fast product of [6], that yields $\mathsf{I}(m) = O(m \log m)$.

2.1. Fixed point arithmetic

Numbers will be represented in radix 2, with a finite number of digits. We will represent fixed point numbers by a signed mantissa and a fixed exponent. More precisely, given a precision parameter $p \in \mathbb{N}$, we denote by $\mathbb{C}_p$ the set of complex numbers of the form $z = m_z 2^{-p}$, where $m_z \in \mathbb{Z}[\mathrm{i}]$ and $|z| \leqslant 1$. We write $\mathbb{C}_p 2^e$ for the set of complex numbers of the form $u 2^e$, where $u \in \mathbb{C}_p$ and $e \in \mathbb{Z}$; in particular, for $z \in \mathbb{C}_p 2^e$ we always have $|z| \leqslant 2^e$. At every stage of our algorithms, the exponent e will be specified, so the exponents do not have to be stored or manipulated explicitly.

In the error analyses of our numerical algorithms, each $z \in \mathbb{C}_p 2^e$ is really the approximation of some genuine complex number $\hat{z} \in \mathbb{C}$. So each such z comes with an implicit error bound $\varepsilon_z \geqslant 0$; this is a real number for which we can guarantee that $|z - \hat{z}| \leqslant \varepsilon_z$. A *truncation* of $\hat{z}$ is an approximation z of $\hat{z}$ that is rounded towards zero in the sense that $|z| \leqslant |\hat{z}|$. Given $a, b \in \mathbb{C}_p$, the sum $a + b$ can be computed exactly in time $O(p)$. The product $a\,b$ can be computed exactly in time $O(\mathsf{I}(p))$.

LEMMA 2. *Given $z \in \mathbb{C}_p$ and $p' \leqslant p$, we can compute an approximation of z in $\mathbb{C}_{p'}$ with error $\leqslant 2^{-p'-1/2}$ in time $O(p)$.*

Proof. Rounding $z = u + v\,\mathrm{i} \in \mathbb{C}_p$ into $\mathbb{C}_{p'}$ can be done as follows: let u' and v' be the truncations to the nearest of u and v into $\mathbb{C}_{p'}$ and let $z' := u' + v'\,\mathrm{i} \in \mathbb{C}_{p'}$; Then we have $|z - z'| \leqslant 2^{-p'-1/2}$. □

LEMMA 3. *Given $z \in \mathbb{C}_p$ and $p' \leqslant p$, we can compute a truncation of z in $\mathbb{C}_{p'}$ with error $\leqslant 2^{-p'+1/2}$ in time $O(p)$.*

Proof. Rounding $z = u + v\,\mathrm{i} \in \mathbb{C}_p$ towards zero into $\mathbb{C}_{p'}$ can be done as follows: let u' and v' be the truncations of u and v at precision $\leqslant 2^{-p'}$ and let $z' := u' + v'\,\mathrm{i} \in \mathbb{C}_{p'}$; Then we have $|z'| \leqslant |z|$ and $|z - z'| \leqslant 2^{-p'+1/2}$. □

2.2. Fast Fourier transform

Let Δ be a power of 2 and let $\boldsymbol{a} = (a_0, \ldots, a_{\Delta-1})$ be a vector of complex numbers. We write $|\boldsymbol{a}| := \max(|a_0|, \ldots, |a|_{\Delta-1})$ and $\omega := \mathrm{e}^{2\pi\mathrm{i}/\Delta}$.

LEMMA 4. *Given $p \in \mathbb{N}$, we can compute truncations of $1, \omega, \omega^2, \ldots, \omega^{\Delta-1}$ in $\mathbb{C}_p$ with error $\leqslant 2^{-p+1}$ in time $O(\Delta\,\mathsf{I}(p + \log \Delta))$.*

Proof. We use [7, Proposition 4] at precision $q = p + \log_2 \Delta + 2$ in order to obtain first approximations of $1, \omega, \omega^2, \ldots, \omega^{\Delta-1}$ in $\mathbb{C}_q$ with error $\leqslant 2^{-q+1}$. We next truncate these results in $\mathbb{C}_p$, which yields an overall error $\leqslant 2^{-p+1/2} + 2^{-q+1} \leqslant 2^{-p+1}$ thanks to Lemma 3. □

We define the fast Fourier transform of $\boldsymbol{a}$ to be

$$\mathrm{FFT}_\omega(\boldsymbol{a}) \;:=\; (P(\omega^0), P(\omega^1), P(\omega^2), \ldots, P(\omega^{\Delta-1})),$$

where $P(x) := \sum_{i=0}^{\Delta-1} a_i x^i$. In particular, we have $|\mathrm{FFT}_\omega(\boldsymbol{a})| \leqslant \Delta\,|\boldsymbol{a}|$.

LEMMA 5. *Given $p \in \mathbb{N}$ and $\boldsymbol{a} \in \mathbb{C}_p^\Delta$, a truncation of $\mathrm{FFT}_\omega(\boldsymbol{a})$ in $(\mathbb{C}_p\Delta)^\Delta$ with error $\leqslant \Delta 2^{-p+1}$ can be computed in time $O(\mathsf{I}(\Delta(p+\log\Delta)))$.*

Proof. Thanks to [18, section 3] an approximation of $\mathrm{FFT}_\omega(\boldsymbol{a})$ in $(\mathbb{C}_{p+O(1)}\,2\Delta)^\Delta$ can be computed with error $\leqslant \Delta 2^{-p-2}$ in time $O(\mathsf{I}(\Delta(p+\log\Delta)))$. Consider an entry $\hat{z} = \hat{x} + \hat{y}\,\mathrm{i}$ of $\mathrm{FFT}_\omega(\boldsymbol{a})$ and let $z = x + y\,\mathrm{i}$ be the corresponding approximation. Let $z' := x' + y'\,\mathrm{i}$, where $x' := x - \mathrm{sign}(x)\,\Delta 2^{-p-1}$ and $y' := y - \mathrm{sign}(y)\,\Delta 2^{-p-1}$. Then $|x'| \leqslant |\hat{x}|$, $|y'| \leqslant |\hat{y}|$, and $|\hat{z} - z'| \leqslant \Delta\,(2^{-p-2} + 2^{-p-1/2}) \leqslant \Delta\,2^{-p}$. We may clearly compute z' in time $O(p+\log\Delta)$. Thanks to Lemma 3, and using $O(p+\log\Delta)$ further operations, we may next compute a truncation z'' of $\hat{z}$ with error $\leqslant \Delta 2^{-p+1}$. □

LEMMA 6. *When $\Delta \geqslant 2$, we have $|1-\omega| \geqslant \frac{4}{\Delta}$.*

Proof. If $\Delta = 2$, then $|1-\omega| = 2 = 4/\Delta$. Otherwise, $\Delta \geqslant 4$ and we have

$$|1-\omega| \;=\; 2\sin\Big(\frac{\pi}{\Delta}\Big) \;\geqslant\; 2\cos\Big(\frac{\pi}{4}\Big)\frac{\pi}{\Delta} \;\geqslant\; \frac{4}{\Delta}.$$ □

2.3. Multivariate polynomials

Given integers $\delta_1 \geqslant 1, \ldots, \delta_n \geqslant 1$, we define

$$\mathbb{K}[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n} \;:=\; \{P \in \mathbb{K}[z_1, \ldots, z_n] : \deg_{z_1} P < \delta_1, \ldots, \deg_{z_n} P < \delta_n\},$$

where $\deg_{z_i} P$ denotes the partial degree of P in z_i, for $i = 1, \ldots, n$. In this paper, a dense representation will be used for univariate and multivariate polynomials. This means that a polynomial $P \in \mathbb{K}[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n}$ is stored as a vector of size $\Delta := \delta_1 \cdots \delta_n$ made of the coefficients of the terms of P of partial degree $< \delta_i$ in z_i, for $i = 1, \ldots, n$. Precisely, when $n = 1$, a polynomial $P = \sum_{i=0}^{\delta_1-1} P_i z_1^i$ is stored as the vector $(P_0, \ldots, P_{\delta_1-1})$. When $n \geqslant 2$, a polynomial P will be regarded as a univariate polynomial in z_n whose coefficients are polynomials in $\mathbb{K}[z_1, \ldots, z_{n-1}]_{\delta_1,\ldots,\delta_{n-1}}$.

Given a polynomial $P = \sum_{i_1,\ldots,i_n} P_{i_1,\ldots,i_n} z_1^{i_1} \cdots z_n^{i_n} \in \mathbb{C}[z_1, \ldots, z_n]$, we define

$$|P| \;:=\; \max_{i_1,\ldots,i_n} |P_{i_1,\ldots,i_n}|.$$

Given polynomials $P, Q \in \mathbb{C}_p[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n}$ the sum $P+Q$ can be computed exactly in time $O(\Delta p)$. The product PQ can be computed exactly in time $O(\mathsf{I}(2^n\Delta\,(p+\log\Delta)))$ using Kronecker substitution; see [5, Chapter 8]. In addition we have $|PQ| \leqslant \Delta$.

3. SPIROID ARITHMETIC

Given integers $\delta_1 \geqslant 1, \ldots, \delta_n \geqslant 1$, we let $\Delta := \delta_1 \cdots \delta_n$ and consider a tuple of points

$$\boldsymbol{\xi} \;=\; (\xi_0, \ldots, \xi_{\Delta-1}) \in (\mathbb{C}^n)^\Delta.$$

We define $|\boldsymbol{\xi}| := \max_{i,j} |(\xi_i)_j|$ and call

$$I_{\boldsymbol{\xi}} \;:=\; \{P \in \mathbb{C}[z_1, \ldots, z_n] : P(\boldsymbol{\xi}) = \mathbf{0}\}$$

the *vanishing ideal* of $\boldsymbol{\xi}$. We also define the evaluation map

$$\begin{aligned}\mathrm{Eval}_{\boldsymbol{\xi}}\colon \mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n} &\longrightarrow \mathbb{C}^\Delta\\ P &\longmapsto P(\boldsymbol{\xi}) := (P(\xi_0),\ldots,P(\xi_{\Delta-1})).\end{aligned}$$

We refer to the computation of $\mathrm{Eval}_{\boldsymbol{\xi}}$ as the problem of *multi-point evaluation*. The computation of $\mathrm{Eval}_{\boldsymbol{\xi}}^{-1}$ is called the *interpolation* problem. $\mathrm{Eval}_{\boldsymbol{\xi}}$ is a bijection for a Zariski dense subset of tuples $\boldsymbol{\xi}\in(\mathbb{C}^n)^\Delta$. Whenever this is the case, the points $\xi_0,\ldots,\xi_{\Delta-1}$ are pairwise distinct, so we have a natural bijection

$$\begin{aligned}\mathbb{C}[z_1,\ldots,z_n]/I_{\boldsymbol{\xi}} &\longrightarrow \mathbb{C}^\Delta\\ P+I_{\boldsymbol{\xi}} &\longmapsto P(\boldsymbol{\xi}).\end{aligned}$$

This allows us to use polynomials in $\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ as canonical representatives of residue classes in $\mathbb{C}[z_1,\ldots,z_n]/I_{\boldsymbol{\xi}}$. In particular, given a polynomial $P\in\mathbb{C}[z_1,\ldots,z_n]$, we define $P \operatorname{rem} I_{\boldsymbol{\xi}}$ to be the unique polynomial in $\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ with $P-(P \operatorname{rem} I_{\boldsymbol{\xi}})\in I_{\boldsymbol{\xi}}$. For special tuples $\boldsymbol{\xi}\in(\mathbb{C}^n)^\Delta$, called *spiroids*, we will show in this section that $P \operatorname{rem} I_{\boldsymbol{\xi}}$ can be computed efficiently.

3.1. Regular spiroids

Assume from now that $\delta_1,\ldots,\delta_n\in 2^{\mathbb{N}}$ are powers of two and let $\omega := \mathrm{e}^{2\pi\mathrm{i}/\Delta}$ be the standard primitive Δ-th root of unity in $\mathbb{C}$. For each $k\in\{0,\ldots,\Delta-1\}$, we let

$$\omega_k \;:=\; (\omega^{k\Delta/\delta_1},\omega^{k\Delta/(\delta_1\delta_2)},\ldots,\omega^k)\in\mathbb{C}^n.$$

We call the tuple $\boldsymbol{\omega} := (\omega_0,\ldots,\omega_{\Delta-1})\in(\mathbb{C}^n)^\Delta$ a *regular spiroid*. The vanishing ideal $I_{\boldsymbol{\omega}}$ of this tuple of points is generated by the polynomials

$$z_1^{\delta_1}-1,\; z_2^{\delta_2}-z_1,\; z_3^{\delta_3}-z_2,\;\ldots,z_n^{\delta_n}-z_{n-1}. \tag{1}$$

A straightforward computation shows that these generators actually form a Gröbner basis for the grevlex monomial ordering. For instance, for $2\leqslant i<j\leqslant n$, the S-polynomial of $z_i^{\delta_i}-z_{i-1}$ and $z_j^{\delta_j}-z_{j-1}$ is

$$(z_i^{\delta_i}-z_{i-1})\,z_j^{\delta_j}-\left(z_j^{\delta_j}-z_{j-1}\right)z_i^{\delta_i}=z_{j-1}z_i^{\delta_i}-z_{i-1}z_j^{\delta_j},$$

which reduces to $z_{j-1}z_{i-1}-z_{i-1}z_{j-1}=0$. In particular, the monomials in $\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ are reduced.

Let $\mathbb{E}_n := \{0,1\}^n$, $\mathbf{0} := (0,\ldots,0)$, and $\mathbb{E}_n^* := \mathbb{E}_n\setminus\{\mathbf{0}\}$. For $\boldsymbol{\epsilon}=(\epsilon_1,\ldots,\epsilon_n)\in\mathbb{E}_n^*$, we have

$$F_{\boldsymbol{\epsilon}} \;:=\; z_1^{\epsilon_1\delta_1}\cdots z_n^{\epsilon_n\delta_n}-z_1^{\epsilon_2}\cdots z_{n-1}^{\epsilon_n}\;\in\; I_{\boldsymbol{\omega}}.$$

For convenience, we also set

$$F_{\mathbf{0}} \;:=\; 1.$$

Given $P\in\mathbb{C}[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$ and a family

$$(A_{\boldsymbol{\epsilon}})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}\;\in\;\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}^{\mathbb{E}_n},$$

we say that $(A_{\boldsymbol{\epsilon}})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$ is an *extended reduction* of P modulo $I_{\boldsymbol{\omega}}$ if

$$P \;=\; \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}}F_{\boldsymbol{\epsilon}}.$$

Since $F_{\mathbf{0}}=1$, we have $A_{\mathbf{0}}=P \operatorname{rem} I_{\boldsymbol{\omega}}$. We further define

$$\|A\| \;:=\; \max_{\epsilon\in\mathbb{E}_n} |A_\epsilon|.$$

Given $P\in\mathbb{C}_p[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$ we aim at computing an extended reduction of P by $I_{\boldsymbol{\omega}}$. As a first step we begin with polynomials P in $\mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_n+1}$.

LEMMA 7. *Let $\boldsymbol{\eta}=(\eta_1,\ldots,\eta_n)\in\mathbb{N}^n$. The map*

$$\begin{aligned}\{z_1^{e_1+\eta_1}\cdots z_n^{e_n+\eta_n} : e_1<\delta_1,\ldots,e_n<\delta_n\} &\longrightarrow \{z_1^{e_1}\cdots z_n^{e_n} : e_1<\delta_1,\ldots,e_n<\delta_n\}\\ z_1^{e_1}\cdots z_n^{e_n} &\longmapsto z_1^{e_1}\cdots z_n^{e_n} \operatorname{rem} I_{\boldsymbol{\omega}}\end{aligned}$$

is well defined and one-to-one.

Proof. Since $I_{\boldsymbol{\omega}}$ is generated by binomial polynomials (1), the reduction of a monomial by $I_{\boldsymbol{\omega}}$ is a monomial. The defining equations (1) of $I_{\boldsymbol{\omega}}$ further imply that $z_1,\ldots,z_n$ are invertible in $\mathbb{C}[z_1,\ldots,z_n]/I_{\boldsymbol{\omega}}$. □

The *support* $\mathcal{S}$ of a polynomial P is the set of its monomials with a non-zero coefficient. We say that the monomials of $\mathcal{S}$ are pairwise distinct modulo $I_{\boldsymbol{\omega}}$ when any two distinct monomials of $\mathcal{S}$ have distinct projections in $\mathbb{C}[z_1,\ldots,z_n]/I_{\boldsymbol{\omega}}$.

LEMMA 8. *Let $P=\sum_{e_1,\ldots,e_n} P_{e_1,\ldots,e_n} z_1^{e_1}\cdots z_n^{e_n}\in\mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_k+1,\delta_{k+1},\ldots,\delta_n}$ be such that the monomials of its support are pairwise distinct modulo $I_{\boldsymbol{\omega}}$. Let $P^{[0]},\ldots,P^{[k]}$ be defined recursively as follows: $P^{[k]}:=P$, and*

$$P^{[i-1]} \;:=\; P^{[i]} - \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_{i-1}\times\{1\}\times\{0\}^{n-i}} A_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}}, \tag{2}$$

where

$$A_{\boldsymbol{\epsilon}} \;:=\; \sum_{e_1 \operatorname{quo}\delta_1=\epsilon_1,\ldots,e_n\operatorname{quo}\delta_n=\epsilon_n} P^{[i]}_{e_1,\ldots,e_n} z_1^{e_1 \operatorname{rem}\delta_1}\cdots z_n^{e_n\operatorname{rem}\delta_n} \;\in\; \mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$$

for $\boldsymbol{\epsilon}\in\mathbb{E}_{i-1}\times\{1\}\times\{0\}^{n-i}$. For $i=0,\ldots,k$, the following properties hold:

- *$P^{[i]}\in\mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_i+1,\delta_{i+1},\ldots,\delta_n}$,*
- *the monomials of the support of $P^{[i]}$ are pairwise distinct modulo $I_{\boldsymbol{\omega}}$,*
- *the coefficients of $P^{[i]}$ are coefficients of P.*

Letting $A_{\boldsymbol{\epsilon}}:=0$ for $\boldsymbol{\epsilon}\notin\mathbb{E}_k\times\{0\}^{n-k}$ and $A_{\mathbf{0}}:=P^{[0]}$, the family $(A_{\boldsymbol{\epsilon}})_{\epsilon\in\mathbb{E}_n}$ is an extended reduction of P.

Proof. The proof is done by induction on i. The properties are clear for $P^{[k]}$. Let us assume that they hold for $i\leqslant k$. We decompose $P^{[i]}$ into $P^{[i]}=L+Hz_i^{\delta_i}$ with

$$\begin{aligned}L &\in \mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_{i-1}+1,\delta_i,\ldots,\delta_n}\\ H &\in \mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_{i-1}+1,1,\delta_{i+1},\ldots,\delta_n}.\end{aligned}$$

We verify that

$$P^{[i]} \;=\; L + \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_{i-1}\times\{1\}\times\{0\}^{n-i}} A_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}} + R$$

where

$$R := \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_{i-1}\times\{1\}\times\{0\}^{n-i}} A_{\boldsymbol{\epsilon}} z_1^{\epsilon_2}\cdots z_{i-1}^{\epsilon_i} \in \mathbb{C}[z_1,\ldots,z_n]_{\delta_1+1,\ldots,\delta_{i-1}+1,\delta_i,\ldots,\delta_n}.$$

Since the monomials of the support of P are pairwise distinct modulo I_ω, the supports of L and R are disjoint. In particular the monomials of the support of $P^{[i-1]} = L + R$ are pairwise distinct modulo I_ω, and the non-zero coefficients of $P^{[i-1]}$ are coefficients of P.

Finally unrolling (2) yields

$$P = \sum_{\epsilon \in \mathbb{E}_k^* \times \{0\}^{n-k}} A_\epsilon F_\epsilon + P^{[0]} = \sum_{\epsilon \in \mathbb{E}_k \times \{0\}^{n-k}} A_\epsilon F_\epsilon,$$

which constitutes the extended reduction of P. □

The reduction of a general $P \in \mathbb{C}_p[z_1, \ldots, z_n]_{2\delta_1, \ldots, 2\delta_n}$ modulo I_ω can be performed efficiently by the following algorithm.

Algorithm 1

Input. $P = \sum_{e_1, \ldots, e_n} P_{e_1, \ldots, e_n} z_1^{e_1} \cdots z_n^{e_n} \in \mathbb{C}_p[z_1, \ldots, z_n]_{2\delta_1, \ldots, 2\delta_n}$.

Output. An extended reduction $(A_\epsilon)_{\epsilon \in \mathbb{E}_n} \in (\mathbb{C}_{p+n} 2^n)[z_1, \ldots, z_n]_{\delta_1, \ldots, \delta_n}^{\mathbb{E}_n}$ of P.

1. For all $\epsilon = (\epsilon_1, \ldots, \epsilon_n) \in \mathbb{E}_n$ set

$$A'_\epsilon := \sum_{e_1 \operatorname{quo} \delta_1 = \epsilon_1, \ldots, e_n \operatorname{quo} \delta_n = \epsilon_n} P_{e_1, \ldots, e_n} z_1^{e_1 \operatorname{rem} \delta_1} \cdots z_n^{e_n \operatorname{rem} \delta_n}$$

and compute

$$P^{[n-1]} := \sum_{\epsilon \in \mathbb{E}_n^*} A'_\epsilon z_1^{\epsilon_2} \cdots z_{n-1}^{\epsilon_n}.$$

2. Let $A''_\epsilon := 0$ for $\epsilon \in \mathbb{E}_{n-1} \times \{1\}$, and for i from $n-1$ down to 1 do:
   - For all $\epsilon \in \mathbb{E}_{i-1} \times \{1\} \times \{0\}^{n-i}$, set

$$A''_\epsilon := \sum_{e_1 \operatorname{quo} \delta_1 = \epsilon_1, \ldots, e_n \operatorname{quo} \delta_n = \epsilon_n} P^{[i]}_{e_1, \ldots, e_n} z_1^{e_1 \operatorname{rem} \delta_1} \cdots z_n^{e_n \operatorname{rem} \delta_n};$$

   - Compute $P^{[i-1]} := P^{[i]} - \sum_{\epsilon \in \mathbb{E}_{i-1} \times \{1\} \times \{0\}^{n-i}} A''_\epsilon F_\epsilon$.

3. Let $\mathbb{A}''_{\mathbf{0}} := P^{[0]}$, compute $A_\epsilon := A'_\epsilon + A''_\epsilon$ for $\epsilon \in \mathbb{E}_n$, and return $(A_\epsilon)_{\epsilon \in \mathbb{E}_n}$.

PROPOSITION 9. *Algorithm 1 is correct and runs in time $O(2^n \Delta (p+n))$.*

Proof. For all $\epsilon = (\epsilon_1, \ldots, \epsilon_n) \in \mathbb{E}_n$, we have $A'_\epsilon \in \mathbb{C}_p[z_1, \ldots, z_n]_{\delta_1, \ldots, \delta_n}$. Let us examine the particular case where $P^{[n-1]}$ restricts to a single term $A'_\epsilon z_1^{\epsilon_2} \cdots z_{n-1}^{\epsilon_n}$ for some $\epsilon \in \mathbb{E}_n^*$. By Lemma 7, the monomials of the support of $P^{[n-1]}$ are pairwise distinct modulo I_ω, so Lemma 8 implies the following properties, for $i = 0, \ldots, n-1$:

- $P^{[i]} \in \mathbb{C}[z_1, \ldots, z_n]_{\delta_1+1, \ldots, \delta_i+1, \delta_{i+1}, \ldots, \delta_n}$,
- the coefficients of $P^{[i]}$ are coefficients of $P^{[n-1]}$, hence of P,
- $(A''_\epsilon)_{\epsilon \in \mathbb{E}_n}$ is an extended reduction of $P^{[n-1]}$.

Since the operations in step 2 are linear with respect to the coefficients of $P^{[n-1]}$, the general case where $P^{[n-1]}$ is a sum of $2^n - 1$ terms of the form $A'_\epsilon z_1^{\epsilon_2} \cdots z_{n-1}^{\epsilon_n}$ satisfies the following properties:

- $P^{[i]} \in \mathbb{C}_p 2^n [z_1, \ldots, z_n]_{\delta_1+1, \ldots, \delta_i+1, \delta_{i+1}, \ldots, \delta_n}$,
- $|P^{[i]}| \leqslant 2^{-n} - 1$,

- $(A''_{\boldsymbol{\epsilon}})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$ is an extended reduction of $P^{[n-1]}$.

On the other hand step 1 yields

$$P \;=\; \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}} + P^{[n-1]}.$$

So the correctness proof is completed by noting that

$$P \;=\; \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}} + \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A''_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}} = \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}},$$

and $|A_{\boldsymbol{\epsilon}}| \leqslant |A'_{\boldsymbol{\epsilon}}| + |A''_{\boldsymbol{\epsilon}}| \leqslant 2^n$. As for the complexity analysis, step 1 takes time $O(2^n \Delta (p+n))$. Step 2 runs in time

$$O\left(\sum_{1\leqslant i<n} 2^{i-1} \Delta (p+n) \right) = O(2^n \Delta (p+n)).$$

The cost of step 3 does not exceed $O(2^n \Delta (p+n))$. □

Example. Let $n=2$ and $P = z_1^{2\delta_1-1} z_2^{2\delta_2-1}$. With the notation as in Algorithm 1, the only non-zero $A'_{\boldsymbol{\epsilon}}$ is $A'_{(1,1)} = z_1^{\delta_1-1} z_2^{\delta_2-1}$, and we have $P^{[1]} = z_1^{\delta_1} z_2^{\delta_2-1}$. It follows that all the $A''_{\boldsymbol{\epsilon}}$ are all 0 but $A''_{(1,0)} = z_2^{\delta_2-1}$, so we have $P^{[0]} = z_2^{\delta_2-1}$. The extended reduction of P is

$$P = z_1^{\delta_1-1} z_2^{\delta_2-1} F_{(1,1)} + z_2^{\delta_2-1} F_{(1,0)} + z_2^{\delta_2-1} F_{(0,0)}.$$

3.2. General spiroids

We now turn to the situation where $\boldsymbol{\xi} \in (\mathbb{C}^n)^\Delta$ is a sufficiently small perturbation of $\boldsymbol{\omega}$. More precisely, we say that $\boldsymbol{\xi}$ is a *spiroid* when there exist polynomials

$$E_{\boldsymbol{\epsilon}} \;\in\; \mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$$

such that

$$G_{\boldsymbol{\epsilon}} \;:=\; F_{\boldsymbol{\epsilon}} + E_{\boldsymbol{\epsilon}} \;\in\; I_{\boldsymbol{\xi}},$$

for all $\boldsymbol{\epsilon} \in \mathbb{E}_n^*$, and

$$\varepsilon \;:=\; \max\left(|\boldsymbol{\xi} - \boldsymbol{\omega}|, \|E\|\right) \;<\; \frac{1}{2^{2n}\Delta},$$

with the convention that $E_{\mathbf{0}} := 0$ and $G_{\mathbf{0}} := 1$. We call n and $\delta_1,\ldots,\delta_n$ the *dimension* and the *degrees* of the spiroid and let

$$\bar{\delta} \;:=\; \max_{i=1,\ldots,n} (\delta_i - 1).$$

By Lemma 6 we have $|\omega_k - \omega_{k'}| \geqslant 4/\Delta$ for all $k \neq k'$. Consequently,

$$|\xi_k - \omega_k| \;<\; \frac{1}{\Delta} \;<\; \frac{2}{\Delta} \;\leqslant\; \frac{1}{2} \min_{k\neq k'} |\omega_k - \omega_{k'}|,$$

so the points ξ_k are pairwise distinct. We let

$$\eta \;:=\; \frac{2^{-\kappa}}{2^{2n}\Delta},$$

where $\kappa \geqslant 0$ is a constant, that will be specified later. We denote by $G_{\boldsymbol{\epsilon};p} \in (\mathbb{C}_q 2)[z_1,\ldots,z_n]$ an approximation of $G_{\boldsymbol{\epsilon}}$ with error $\leqslant \eta 2^{-p}$, where $q = O(p+n+\log\Delta)$ will also be defined later. For convenience we always take $G_{\mathbf{0};p} := 1$.

3.3. Existence of spiroids

We are to show that $\boldsymbol{\xi}$ is a spiroid as soon as $|\boldsymbol{\xi}-\boldsymbol{\omega}|$ is sufficiently small.

LEMMA 10. *Let $\boldsymbol{v}\in\mathbb{C}^\Delta$ and let $P\in\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ be such that $P(\boldsymbol{\omega})=\boldsymbol{v}$. Then we have $|P|\leqslant|\boldsymbol{v}|$.*

Proof. The polynomial $q(x):=P(x^{\Delta/\delta_1},x^{\Delta/(\delta_1\delta_2)},\ldots,x)\in\mathbb{C}[x]$ has degree

$$\begin{aligned}&\leqslant \frac{\Delta}{\delta_1}(\delta_1-1)+\frac{\Delta}{\delta_1\delta_2}(\delta_2-1)+\cdots+\frac{\Delta}{\delta_1\cdots\delta_n}(\delta_n-1)\\&= \Delta((1-\delta_1^{-1})+(\delta_1^{-1}-(\delta_1\delta_2)^{-1})+\cdots+(\delta_1\cdots\delta_{n-1})^{-1}-\Delta^{-1})\\&= \Delta-1,\end{aligned}$$

and satisfies $q(\omega^k)=P(\omega_k)$ for $k=0,\ldots,\Delta-1$. This means that q is the inverse Fourier transform of $\boldsymbol{v}$, that is $\Delta^{-1}\,\mathrm{FFT}_{\omega^{-1}}(\boldsymbol{v})$, following the notation of section 2.2. We deduce that $|P|=|q|\leqslant|\boldsymbol{v}|$. □

The two following technical lemmas will be needed several times in the paper.

LEMMA 11. *Let $a,b\in\mathbb{C}^n$ and let $P\in\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ have $\leqslant T$ non-zero terms. Then we have*

$$|P(a)-P(b)| \leqslant n\,\bar{\delta}\,T\,|P|\max(|a|,|b|,1)^{n\bar{\delta}-1}\,|a-b|.$$

Proof. We set $f(t):=P(t\,a+(1-t)\,b)$ for $t\in[0,1]$ and use the classical inequality

$$|P(a)-P(b)|=|f(1)-f(0)| \leqslant \max_{t\in[0,1]}|f'(t)|.$$

From $f'(t)=\sum_{i=1}^n\frac{\partial P}{\partial z_i}(t\,a+(1-t)\,b)\,(a_i-b_i)$, we obtain

$$\max_{t\in[0,1]}|f'(t)| \leqslant n\,\bar{\delta}\,T\,|P|\max(|a|,|b|,1)^{n\bar{\delta}-1}\,|a-b|.$$ □

LEMMA 12. *With the above notation and $\kappa\geqslant 0$, we have $(1+\eta/\alpha)^{n\alpha}\leqslant 2$ for all $\alpha>0$.*

Proof. We verify that

$$\log_2((1+\eta/\alpha)^{n\alpha}) = n\,\alpha\log_2(1+\eta/\alpha) \leqslant \frac{n\,\eta}{\log 2} \leqslant \frac{2^{-\kappa-n}}{\Delta\log 2} \leqslant 1.$$ □

LEMMA 13. *Assume that $\kappa\geqslant 1$, $\Delta\geqslant 2$, and let $\boldsymbol{\xi}\in(\mathbb{C}^n)^\Delta$ be such that $|\boldsymbol{\xi}-\boldsymbol{\omega}|\leqslant\eta/\bar{\delta}$. Given $\boldsymbol{v}\in\mathbb{C}^\Delta$, there exists a unique $P\in\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that $P(\boldsymbol{\xi})=\boldsymbol{v}$. Moreover, $|P|\leqslant 2\,|\boldsymbol{v}|$.*

Proof. Let $\boldsymbol{v}_0:=\boldsymbol{v}$ and define $\boldsymbol{v}_{i+1}:=\boldsymbol{v}_i-P_i(\boldsymbol{\xi})$, where P_i stands for the unique polynomial in $\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that $P_i(\boldsymbol{\omega}):=\boldsymbol{v}_i$. From Lemma 10 we know that $|P_i|\leqslant|\boldsymbol{v}_i|$. Using Lemma 11, we deduce

$$|\boldsymbol{v}_{i+1}| = |P_i(\boldsymbol{\omega})-P_i(\boldsymbol{\xi})| \leqslant n\,\bar{\delta}\,\Delta\,|P_i|\,(1+|\boldsymbol{\xi}-\boldsymbol{\omega}|)^{n\bar{\delta}-1}\,|\boldsymbol{\xi}-\boldsymbol{\omega}| \leqslant |\boldsymbol{v}_i|\left(1+\frac{\eta}{\bar{\delta}}\right)^{n\bar{\delta}}2^{-\kappa-n}.$$

Using Lemma 12 and $\kappa\geqslant 1$, we obtain $|\boldsymbol{v}_{i+1}|\leqslant 2^{-n}\,|\boldsymbol{v}_i|$, hence $|\boldsymbol{v}_i|\leqslant 2^{-in}\,|\boldsymbol{v}|$ for $i\geqslant 0$. Consequently, $P:=\sum_{i\geqslant 0}P_i$ is well defined and we have

$$\boldsymbol{v} = \boldsymbol{v}_0 = P_0(\boldsymbol{\xi})+\boldsymbol{v}_1 = P_0(\boldsymbol{\xi})+P_1(\boldsymbol{\xi})+\boldsymbol{v}_2 = \cdots = P(\boldsymbol{\xi}).$$

Finally, we verify that

$$|P| \leqslant \sum_{i\geqslant 0} |P_i| \leqslant \sum_{i\geqslant 0} |v_i| \leqslant |v| \sum_{i\geqslant 0} 2^{-in} \leqslant 2|v|. \qquad \square$$

PROPOSITION 14. *Assume that $\kappa \geqslant 1$, $\Delta \geqslant 2$, and let $\xi \in (\mathbb{C}^n)^\Delta$ be such that $|\xi - \omega| < \eta^2$. Then ξ is a spiroid and we have $\varepsilon \leqslant n\,\bar{\delta}\,2^4 |\xi - \omega|$.*

Proof. From Lemmas 11 and 12, and using

$$|\xi - \omega| < \eta^2 = \frac{2^{-\kappa}\eta}{2^{2n}\Delta} \leqslant \frac{\eta}{\bar{\delta}+1},$$

we obtain

$$\begin{aligned} |F_\epsilon(\xi)| &= |F_\epsilon(\xi) - F_\epsilon(\omega)| \\ &\leqslant n\,(\bar{\delta}+1)\,2\,(1+|\xi-\omega|)^{n(\bar{\delta}+1)}\,|\xi-\omega| \\ &\leqslant n\,(\bar{\delta}+1)\,2\left(1+\frac{\eta}{\bar{\delta}+1}\right)^{n(\bar{\delta}+1)}|\xi-\omega| \\ &\leqslant n\,\bar{\delta}\,2^3|\xi-\omega|. \end{aligned}$$

Thanks to Lemma 13, there exists $E_\epsilon \in \mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that $E_\epsilon(\xi) = -F_\epsilon(\xi)$ and

$$|E_\epsilon| \leqslant 2|F_\epsilon(\xi)| \leqslant n\,\bar{\delta}\,2^4|\xi-\omega|,$$

whence $\varepsilon \leqslant n\,\bar{\delta}\,2^4|\xi-\omega| < n\,\bar{\delta}\,2^4\eta^2 \leqslant \frac{1}{2^{2n}\Delta}$. $\square$

3.4. Reduction by spiroids

In the rest of this section, we fix $P \in \mathbb{C}[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$ and assume that $\xi \in (\mathbb{C}^n)^\Delta$ is a spiroid. A family

$$(A_\epsilon)_{\epsilon\in\mathbb{E}_n} \in \mathbb{C}[z_1,\ldots,z_n]^{\mathbb{E}_n}_{\delta_1,\ldots,\delta_n}.$$

is called an *extended reduction* of P by the spiroid ξ if

$$P = \sum_{\epsilon\in\mathbb{E}_n} A_\epsilon G_\epsilon.$$

Since $G_0 = 1$, we have $P \operatorname{rem} I_\xi = A_0$.

PROPOSITION 15. *Assume that $\varepsilon < \eta$. Given $P \in \mathbb{C}[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$, there exists an extended reduction $(A_\epsilon)_{\epsilon\in\mathbb{E}_n}$ of P by ξ with $\|A\| \leqslant |P| \frac{2^n}{1-\varepsilon/\eta}$.*

Proof. If $P = 0$ then we take $A_\epsilon := 0$ for all $\epsilon \in \mathbb{E}_n$. Otherwise, without loss of generality, we may assume that $|P| = 1$. For $i \geqslant -1$, we construct sequences $P^{[i]}$, $(A^{[i]}_\epsilon)_{\epsilon\in\mathbb{E}_n}$ of polynomials as follows. For $i = -1$, we set $P^{[-1]} := P$ and $A^{[-1]}_\epsilon := 0$ for $\epsilon \in \mathbb{E}_n$. By induction, we assume that these sequences are defined up to some index $i-1$ for some $i \geqslant 0$. We let

$$P^{[i]} := P^{[i-1]} - \sum_{\epsilon\in\mathbb{E}_n} A^{[i-1]}_\epsilon G_\epsilon,$$

and apply Proposition 9 to $P^{[i]}$ without any rounding, and let $(A^{[i]}_\epsilon)_{\epsilon\in\mathbb{E}_n}$ be the corresponding extended reduction with $\|A\| \leqslant 2^n |P^{[i]}|$. We have

$$P^{[i]} - \sum_{\epsilon\in\mathbb{E}_n} A^{[i]}_\epsilon F_\epsilon = 0.$$

It follows that

$$\begin{aligned} |P^{[i+1]}| &= \left| P^{[i]} - \sum_{\epsilon \in \mathbb{E}_n} A_\epsilon^{[i]} G_\epsilon \right| \\ &= \left| \sum_{\epsilon \in \mathbb{E}_n} A_\epsilon^{[i]} (F_\epsilon - G_\epsilon) \right| \\ &\leqslant \Delta 2^{2n} |P^{[i]}| \varepsilon \\ &\leqslant |P^{[i]}| \frac{\varepsilon}{\eta}. \end{aligned}$$

Hence,

$$|P^{[i]}| \leqslant \left(\frac{\varepsilon}{\eta}\right)^i, \text{ for } i \geqslant 0.$$

In particular $\sum_{i \geqslant 0} A_\epsilon^{[i]}$ converges to a limit written A_ϵ. For $\epsilon \in \mathbb{E}_n$ we have

$$|A_\epsilon| \leqslant \sum_{i \geqslant 0} |A_\epsilon^{[i]}| \leqslant \sum_{i \geqslant 0} |P^{[i]}| 2^n \leqslant 2^n \sum_{i \geqslant 0} \left(\frac{\varepsilon}{\eta}\right)^i = \frac{2^n}{1 - \varepsilon/\eta}. \qquad \square$$

The proof of Proposition 15 can be turned into an algorithm to compute approximations of extended reductions. For efficiency reasons, ε/η is required to be sufficiently small.

LEMMA 16. *Assume $\kappa \geqslant 4$ and $\varepsilon \leqslant \eta^2$. Let $p \in \mathbb{N}$ and*

$$q := p + n - \log_2 \eta + 1 = O(p + n + \log \Delta).$$

Given $P \in \mathbb{C}_p[z_1, \ldots, z_n]_{2\delta_1, \ldots, 2\delta_n}$, we can compute

$$(A_\epsilon)_{\epsilon \in \mathbb{E}_n} \in (\mathbb{C}_q 2^{n+1})[z_1, \ldots, z_n]^{\mathbb{E}_n}_{\delta_1, \ldots, \delta_n}$$

such that

$$\left| P - \sum_{\epsilon \in \mathbb{E}_n} A_\epsilon G_\epsilon \right| \leqslant 2^{-p},$$

in time $O(\mathsf{I}(4^n \Delta (p + n + \log \Delta))\, p)$.

Proof. We adapt the proof of Proposition 15. For $i \geqslant -1$, we construct sequences $P^{[i]} \in \mathbb{C}[z_1, \ldots, z_n]_{2\delta_1, \ldots, 2\delta_n}$ with $|P^{[i]}| \leqslant 1$ and $(A_\epsilon^{[i]})_{\epsilon \in \mathbb{E}_n} \in (\mathbb{C}_q 2^n)[z_1, \ldots, z_n]^{\mathbb{E}_n}_{\delta_1, \ldots, \delta_n}$ as follows.

For $i = -1$, we set $P^{[-1]} := P$ and $A_\epsilon^{[-1]} := 0$ for $\epsilon \in \mathbb{E}_n$. By induction, we assume that these sequences have been defined up to $i-1$ for some $i \geqslant 0$. We compute

$$P^{[i]} := P^{[i-1]} - \sum_{\epsilon \in \mathbb{E}_n} A_\epsilon^{[i-1]} G_{\epsilon; p}$$

exactly. Note that $P^{[0]} = P$. We further assume by induction that $|P^{[i]}| \leqslant 1$. Applying Proposition 9 to $P^{[i]}$, let $(A'_\epsilon)_{\epsilon \in \mathbb{E}_n}$ be the obtained extended reduction. Then

$$P^{[i]} - \sum_{\epsilon \in \mathbb{E}_n} A'_\epsilon F_\epsilon = 0.$$

Using Lemma 3, we compute a truncation $A_{\boldsymbol{\epsilon}}^{[i]}$ of $A'_{\boldsymbol{\epsilon}}$ in $(\mathbb{C}_q 2^n)[z_1,\ldots,z_n]$ with error $\leqslant 2^{n-q+1/2} \leqslant \eta 2^{-p}$. We obtain

$$\begin{aligned}
|P^{[i+1]}| &= \left| P^{[i]} - \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}}^{[i]} G_{\boldsymbol{\epsilon};p} \right| \\
&= \left| \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} F_{\boldsymbol{\epsilon}} - \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}}^{[i]} G_{\boldsymbol{\epsilon};p} \right| \\
&\leqslant \left| \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} (F_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p}) \right| + \left| \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} (A_{\boldsymbol{\epsilon}}^{[i]} - A'_{\boldsymbol{\epsilon}}) G_{\boldsymbol{\epsilon};p} \right| \\
&\leqslant \Delta 2^{2n} (\varepsilon + \eta 2^{-p}) |P^{[i]}| + 2^n \Delta (1 + \varepsilon + \eta 2^{-p}) \eta 2^{-p} \\
&\leqslant 2^{2n} \Delta \eta (\varepsilon/\eta + 2^{-p}) |P^{[i]}| + 2^n \Delta (1 + \varepsilon + \eta 2^{-p}) \eta 2^{-p} \\
&\leqslant 2^{-\kappa+1} |P^{[i]}| + 2^{-p-\kappa}. && \text{(since } \eta \leqslant 2^{-6}\text{)}
\end{aligned}$$

Iterating the latter inequality yields

$$\begin{aligned}
|P^{[i+1]}| &\leqslant (2^{-\kappa+1})^{i+1} + 2^{-p-\kappa} \sum_{0\leqslant j\leqslant i} (2^{-\kappa+1})^j \\
&\leqslant (2^{-\kappa+1})^{i+1} + \frac{2^{-p-\kappa}}{1 - 2^{-\kappa+1}} \\
&\leqslant (2^{-\kappa+1})^{i+1} + 2^{-p-2}. && (3)
\end{aligned}$$

In particular, $|P^{[i+1]}| \leqslant 1$. We are done with the construction of the sequences $P^{[i]}$ and $(A_{\boldsymbol{\epsilon}}^{[i]})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$. Let $I \geqslant 0$ be the smallest integer such that

$$(2^{-\kappa+1})^I + 2^{-p-2} \leqslant 2^{-p-1}, \qquad (4)$$

that is

$$I := \left\lceil \frac{p+2}{\kappa-1} \right\rceil \leqslant \frac{p+5}{3},$$

since $\kappa \geqslant 4$. Note that

$$I 2^{-p-2} \leqslant \frac{1}{2}. \qquad (5)$$

We take $A_{\boldsymbol{\epsilon}} := \sum_{0\leqslant j<I} A_{\boldsymbol{\epsilon}}^{[j]}$ and verify that

$$\begin{aligned}
|A_{\boldsymbol{\epsilon}}| &\leqslant \sum_{0\leqslant j<I} |P^{[j]}| 2^n \\
&\leqslant \left(\frac{1}{1 - 2^{-\kappa+1}} + I 2^{-p-2} \right) 2^n && \text{(using (3))} \\
&\leqslant \left(\frac{8}{7} + \frac{1}{2} \right) 2^n && \text{(using } \kappa \geqslant 4 \text{ and (5))} \\
&\leqslant 2^{n+1}. && (6)
\end{aligned}$$

Finally, we obtain

$$\begin{aligned}
\left| P - \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon}} \right| &\leqslant \left| P - \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon};p} \right| + \left| \sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}} (G_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p}) \right| \\
&\leqslant (2^{-\kappa+1})^I + 2^{-p-2} + \Delta 2^{2n+1} \eta 2^{-p} && \text{(using (3) and (6))} \\
&\leqslant (2^{-1} + 2^{-\kappa+1}) 2^{-p} && \text{(using (4))} \\
&\leqslant 2^{-p}. && \text{(since } \kappa \geqslant 4\text{)}
\end{aligned}$$

As for the complexity, one product $A_\epsilon^{[i-1]} G_{\epsilon;p}$ takes time $O(\mathsf{I}(2^n \Delta\,(p+n+\log\Delta)))$, and $|P^{[i]}|$ is computed with error $2^{-O(q)}$. Consequently, the computation of $P^{[i]}$ from $P^{[i-1]}$ runs in time

$$O(2^n \mathsf{I}(2^n \Delta\,(p+n+\log\Delta))) \;=\; O(\mathsf{I}(4^n \Delta\,(p+n+\log\Delta))).$$

Each use of Proposition 9 contributes $O(2^n \Delta\,(p+n+\log\Delta))$. Since $I = O(p)$, the total cost for computing $P^{[i]}$ and $(A_\epsilon^{[i]})_{\epsilon\in\mathbb{E}_n}$ up to index $I-1$ is $O(\mathsf{I}(4^n \Delta\,(p+n+\log\Delta))\,p)$. □

The numerical convergence of the method behind Lemma 16 is linear, which is sufficient for small precisions. For higher precisions, our next faster algorithm benefits from the "divide and conquer" paradigm. In order to upper bound the norms of the extended reductions, we introduce the auxiliary notation

$$\pi_p := \begin{cases} 1 \text{ if } p < 10 \\ (1+2^{-6})\,(1+2^{-7})\cdots(1+2^{-\lfloor p/2\rfloor-1}) \text{ if } p \geqslant 10. \end{cases}$$

Note that π_p is nondecreasing in p and that $\pi_p \leqslant 2$ for all $p \in \mathbb{N}$.

Algorithm 2

Input. $P \in \mathbb{C}_p[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$.

Output. $(A_\epsilon)_{\epsilon\in\mathbb{E}_n} \in (\mathbb{C}_q 2^{n+2})[z_1,\ldots,z_n]^{\mathbb{E}_n}_{\delta_1,\ldots,\delta_n}$, where $q := p+n-\log_2\eta+2$,

$$\left| P - \sum_{\epsilon\in\mathbb{E}_n} A_\epsilon G_\epsilon \right| \;\leqslant\; 2^{-p} \quad\text{and}\quad \|A\| \;\leqslant\; 2^{n+1}\pi_p.$$

Assumption. $\kappa \geqslant 4$ and $\varepsilon \leqslant \eta^2$.

1. If $p \leqslant m := 9\,(n+\log_2\Delta)$ then return the extended reduction from Lemma 16.
2. Compute a truncation $P' \in \mathbb{C}_{\lfloor p/2\rfloor+3}[z_1,\ldots,z_n]$ of P with error $\leqslant 2^{-\lfloor p/2\rfloor-5/2}$ using Lemma 3. Apply Algorithm 2 recursively to P' and let $(A'_\epsilon)_{\epsilon\in\mathbb{E}_n}$ denote the result.
3. Compute a truncation $P'' \in \mathbb{C}_{p-\lfloor p/2\rfloor+3}[z_1,\ldots,z_n]$ of
$$2^{\lfloor p/2\rfloor+1}\left(P - \sum_{\epsilon\in\mathbb{E}_n} A'_\epsilon G_{\epsilon;p}\right)$$
with error $\leqslant 2^{-p+\lfloor p/2\rfloor-5/2}$. Apply Algorithm 2 recursively to P'' and let $(A''_\epsilon)_{\epsilon\in\mathbb{E}_n}$ denote the result.
4. Return a truncation of $(A'_\epsilon + 2^{-\lfloor p/2\rfloor-1} A''_\epsilon)_{\epsilon\in\mathbb{E}_n}$ with error $\leqslant 2^{-q+n+5/2}$, by using Lemma 3.

PROPOSITION 17. *Algorithm 2 is correct and runs in time*

$$O(\mathsf{I}(4^n \Delta\,(p+n+\log\Delta))\,(\log p+n+\log\Delta)).$$

Proof. If $p \leqslant m$ then the output is correct in step 1 by Lemma 16. Otherwise $p \geqslant 10$, so that $\lfloor p/2\rfloor + 3 < p$ and $p - \lfloor p/2\rfloor + 3 < p$. Consequently, the algorithm always terminates. In step 2, by induction, we have

$$\left| P' - \sum_{\epsilon\in\mathbb{E}_n} A'_\epsilon G_\epsilon \right| \;\leqslant\; 2^{-\lfloor p/2\rfloor-3} \quad\text{and}\quad \|A'\| \;\leqslant\; 2^{n+2}.$$

We verify that

$$\begin{aligned}\left|P-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon};p}\right| &\leqslant |P-P'| + \left|P'-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon}}\right| + \left|\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} (G_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p})\right| \\ &\leqslant 2^{-\lfloor p/2\rfloor-5/2} + 2^{-\lfloor p/2\rfloor-3} + \Delta 2^{2n+2}\eta 2^{-p} \\ &\leqslant 2^{-\lfloor p/2\rfloor-3}(2^{1/2}+1+\Delta 2^{2n}\eta 2^{-p+\lfloor p/2\rfloor+5}) \\ &\leqslant 2^{-\lfloor p/2\rfloor-3}(2^{1/2}+1+2^{-\kappa}) && \text{(since } p\geqslant 10\text{)} \\ &\leqslant 2^{-\lfloor p/2\rfloor-1}, && \text{(since } \kappa\geqslant 4\text{)}\end{aligned}$$

so the polynomial P'' is well defined in step 3, and we have

$$\left|P''-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A''_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon}}\right| \leqslant 2^{-p+\lfloor p/2\rfloor-3}.$$

Let $A_{\boldsymbol{\epsilon}}$ be a truncation of $\tilde{A}_{\boldsymbol{\epsilon}} := A'_{\boldsymbol{\epsilon}} + 2^{-\lfloor p/2\rfloor-1} A''_{\boldsymbol{\epsilon}}$ with error $\leqslant 2^{-q+n+5/2}$. On the one hand,

$$\begin{aligned}&\left|P-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} \tilde{A}_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon};p}\right| \\ \leqslant\ &\left|P-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A'_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon};p} - 2^{-\lfloor p/2\rfloor-1} P''\right| \\ &\quad + 2^{-\lfloor p/2\rfloor-1}\left|P''-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A''_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon}}\right| + 2^{-\lfloor p/2\rfloor-1}\left|\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A''_{\boldsymbol{\epsilon}} (G_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p})\right| \\ \leqslant\ &2^{-\lfloor p/2\rfloor-1}(2^{-p+\lfloor p/2\rfloor-5/2} + 2^{-p+\lfloor p/2\rfloor-3} + \Delta 2^{2n+2}\eta 2^{-p}) \\ \leqslant\ &2^{-p-3}(2^{-1/2}+2^{-1}+2^{-\kappa-1}) && \text{(since } p\geqslant 10\text{)} \\ \leqslant\ &2^{-p-2}. && (7)\end{aligned}$$

On the other hand, we obtain

$$\begin{aligned}|A_{\boldsymbol{\epsilon}}| \leqslant |\tilde{A}_{\boldsymbol{\epsilon}}| &\leqslant |A'_{\boldsymbol{\epsilon}}| + 2^{-\lfloor p/2\rfloor-1}|A''_{\boldsymbol{\epsilon}}| \\ &\leqslant 2^{n+1}(\pi_{\lfloor p/2\rfloor+3} + 2^{-\lfloor p/2\rfloor-1}\pi_{p-\lfloor p/2\rfloor+3}) \\ &\leqslant 2^{n+1}\pi_{p-\lfloor p/2\rfloor+3}(1+2^{-\lfloor p/2\rfloor-1}) \\ &\leqslant 2^{n+1}\pi_p, && \text{(since } p\geqslant 10\text{)}\end{aligned}$$

which yields

$$\left|\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} \tilde{A}_{\boldsymbol{\epsilon}} (G_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p})\right| \leqslant \Delta 2^{2n+2}\eta 2^{-p} \leqslant 2^{-p-\kappa+2}. \qquad (8)$$

Combining (7) and (8) we deduce

$$\begin{aligned}\left|P-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} A_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon}}\right| &\leqslant \left|P-\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} \tilde{A}_{\boldsymbol{\epsilon}} G_{\boldsymbol{\epsilon};p}\right| + \left|\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} \tilde{A}_{\boldsymbol{\epsilon}} (G_{\boldsymbol{\epsilon}} - G_{\boldsymbol{\epsilon};p})\right| + \left|\sum_{\boldsymbol{\epsilon}\in\mathbb{E}_n} (A_{\boldsymbol{\epsilon}} - \tilde{A}_{\boldsymbol{\epsilon}}) G_{\boldsymbol{\epsilon}}\right| \\ &\leqslant 2^{-p-2} + 2^{-p-\kappa+2} + 2^n \Delta 2^{-q+n+5/2}(1+\varepsilon) \\ &\leqslant 2^{-p-2} + 2^{-p-\kappa+2} + 2^n \Delta \eta 2^{-p+3/2} \\ &\leqslant 2^{-p}(2^{-2} + 2^{-\kappa+2} + 2^{-n-\kappa+3/2}) \\ &\leqslant 2^{-p}.\end{aligned}$$

This concludes the proof of the correctness of the algorithm. Let us now analyze the complexity. By Lemma 16, step 1 contributes

$$O(\mathsf{I}(4^n \Delta (n+\log\Delta))\, p).$$

Each product $A'_\epsilon G_{\epsilon;p}$ can be computed in time

$$O(\mathsf{I}(2^n \Delta\,(p+n-\log\eta+\log\Delta))) \;=\; O(\mathsf{I}(2^n\Delta p)).$$

The total cost to obtain P' is therefore bounded by

$$O(2^n \mathsf{I}(2^n\Delta p)) \;=\; O(\mathsf{I}(4^n\Delta p)).$$

Let $\mathsf{T}(p)$ be the computation time as a function of p. So far, we have shown that

$$\mathsf{T}(p) \;\leqslant\; \mathsf{T}(\lfloor p/2\rfloor+3)+\mathsf{T}(p-\lfloor p/2\rfloor+3)+O(\mathsf{I}(4^n\Delta p)),$$

for $p \geqslant m$. Unrolling this inequality leads to

$$\begin{aligned}\mathsf{T}(p) \;&=\; O\Big(\mathsf{I}(4^n\Delta p)\log p+\Big\lceil\frac{p}{m}\Big\rceil \mathsf{I}(4^n\Delta m)\,m\Big)\\ &=\; O(\mathsf{I}(4^n\Delta p)\log p+\mathsf{I}(4^n\Delta p)\,m+\mathsf{I}(4^n\Delta m)\,m)\\ &=\; O(\mathsf{I}(4^n\Delta\,(p+n+\log\Delta))\,(\log p+n+\log\Delta)).\end{aligned}$$

□

The following corollary is the main result of this section.

COROLLARY 18. *Assume $\kappa \geqslant 4$ and $\varepsilon \leqslant \eta^2$. Given $P \in \mathbb{C}_p[z_1,\ldots,z_n]_{2\delta_1,\ldots,2\delta_n}$, an approximation $\tilde{R} \in (\mathbb{C}_p 2^{n+2})[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ of $R := P \operatorname{rem} I_\xi$ with error $\leqslant 2^{-p+n+2}$ can be computed in time*

$$O(\mathsf{I}(4^n\Delta\,(p+n+\log\Delta))\,(\log p+n+\log\Delta)).$$

Proof. Proposition 17 allows us to compute

$$\tilde{R} \in (\mathbb{C}_{p+n-\log_2\eta+2}\,2^{n+2})[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$$

such that $|P-(H+\tilde{R})| \leqslant 2^{-p}$ for some $H \in I_\xi$. Using Proposition 15, we deduce that

$$|\tilde{R}-P \operatorname{rem} I_\xi| \;=\; |(P-(H+\tilde{R})) \operatorname{rem} I_\xi| \;\leqslant\; \frac{2^{-p+n}}{1-\varepsilon/\eta}.$$

Using Lemma 2, we finally compute an approximation of $\tilde{R}$ in $(\mathbb{C}_p 2^{n+2})[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ with error $\leqslant 2^{-p+n+3/2}$ and verify that

$$2^{-p+n+3/2}+\frac{2^{-p+n}}{1-\varepsilon/\eta} \;\leqslant\; 2^{-p+n+2}\left(2^{-1/2}+\frac{2^{-2}}{1-2^{-6}}\right) \;\leqslant\; 2^{-p+n+2}.$$

□

4. MULTI-POINT EVALUATION AND INTERPOLATION

This section is devoted to multi-point evaluation and interpolation at a spiroid. Evaluation will follow the classical "divide and conquer" paradigm with respect to the number of evaluation points. Interpolation will reduce to evaluation.

4.1. Evaluation

Let $\boldsymbol{\xi} \in (\mathbb{C}^n)^\Delta$ be as in the previous section. If $\delta_{k+1}=\cdots=\delta_n=1$ for some $k \geqslant 0$, then

$$\mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n} \;=\; \mathbb{C}[z_1,\ldots,z_k]_{\delta_1,\ldots,\delta_k}.$$

Without loss of generality, this allows us to replace $\boldsymbol{\xi}$ by $\boldsymbol{\xi}^*$ of dimension $n^* := k$ with $\xi_i^* := ((\xi_i)_1,\ldots,(\xi_i)_k)$ for each i. In what follows, we will only work with spiroids $\boldsymbol{\xi}$ for which $\delta_n \neq 1$ whenever $n \geqslant 2$.

Our algorithm for multi-point evaluation of a polynomial $P \in \mathbb{C}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ at $\boldsymbol{\xi}$ is based on a generalization of the classical remainder tree technique used for the univariate case [5, Chapter 10]. If $\Delta = 1$, then an *evaluation tree* for $\boldsymbol{\xi}$ consists of a single leaf labeled by $(G_{\boldsymbol{\epsilon};p})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$. If $\Delta \geqslant 2$ and $\delta_n \geqslant 2$, then $\boldsymbol{\xi}$ induces two sequences of points $\boldsymbol{\xi}^{[0]}, \boldsymbol{\xi}^{[1]}$ which are defined by

$$\begin{aligned} \boldsymbol{\xi}^{[0]} &:= (\xi_0, \xi_2, \ldots, \xi_{\Delta-2})^* \\ \boldsymbol{\xi}^{[1]} &:= \left(\frac{\xi_1}{\omega_1}, \frac{\xi_3}{\omega_1}, \ldots, \frac{\xi_{\Delta-1}}{\omega_1}\right)^*. \end{aligned}$$

For $i = 0, 1$, we write $n^{[i]}$ for the dimension of $\boldsymbol{\xi}^{[i]}$, $\delta_1^{[i]}, \ldots, \delta_{n^{[i]}}^{[i]}$ for its partial degrees, and $\Delta^{[i]}$ for its degree. The perturbations of the defining polynomials of $\boldsymbol{\xi}^{[i]}$ are written $E^{[i]} := (E_{\boldsymbol{\epsilon}}^{[i]})_{\boldsymbol{\epsilon}\in\mathbb{E}_{n^{[i]}}}$, so we have

$$G_{\boldsymbol{\epsilon}}^{[i]} := F_{\boldsymbol{\epsilon}} + E_{\boldsymbol{\epsilon}}^{[i]} \in I_{\boldsymbol{\xi}^{[i]}}$$

for all $\boldsymbol{\epsilon} \in \mathbb{E}^*_{n^{[i]}}$. We further define

$$\begin{aligned} \varepsilon^{[i]} &:= \max\left(|\boldsymbol{\xi}^{[i]} - \boldsymbol{\omega}^{[i]}|, \|E^{[i]}\|\right), \\ \bar{\delta}^{[i]} &:= \max_{j=1,\ldots,n^{[i]}} (\delta_j^{[i]} - 1), \\ \eta^{[i]} &:= \frac{2^{-\kappa}}{2^{2n^{[i]}}\Delta^{[i]}}. \end{aligned}$$

Note that $\Delta^{[i]} := \Delta/2$, $\eta^{[i]} \geqslant 2\eta$, and $\bar{\delta}^{[i]} \leqslant \bar{\delta}$, for $i = 0, 1$.

An *evaluation tree* for the spiroid $\boldsymbol{\xi}$ consists of a root that is labeled by the family $(G_{\boldsymbol{\epsilon};p})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$ and two children which are recursively the evaluation trees for $\boldsymbol{\xi}^{[0]}$ and $\boldsymbol{\xi}^{[1]}$. Since $|\boldsymbol{\xi}^{[i]} - \boldsymbol{\omega}^{[i]}| \leqslant |\boldsymbol{\xi} - \boldsymbol{\omega}| \leqslant \varepsilon$, if $|\boldsymbol{\xi} - \boldsymbol{\omega}| < \eta^2$ then $|\boldsymbol{\xi}^{[i]} - \boldsymbol{\omega}^{[i]}| < (\eta^{[i]})^2 2^{-2}$. So, Proposition 14 ensures that the $E_{\boldsymbol{\epsilon}}^{[i]}$ exists with $|E_{\boldsymbol{\epsilon}}^{[i]}| \leqslant n^{[i]}\,\bar{\delta}^{[i]}\,2^4\,|\boldsymbol{\xi}^{[i]} - \boldsymbol{\omega}^{[i]}|$, whenever $\kappa \geqslant 1$. This shows that evaluation trees exist whenever $|\boldsymbol{\xi} - \boldsymbol{\omega}|$ is sufficiently small. The construction of these trees is postponed to the end of this section.

An evaluation tree for $\boldsymbol{\xi}$ is said to be known *with error* $\leqslant \eta 2^{-p}$ if $(G_{\boldsymbol{\epsilon}})_{\boldsymbol{\epsilon}\in\mathbb{E}_n}$ is given with error $\leqslant \eta 2^{-p}$, and if the evaluation tree for $\boldsymbol{\xi}^{[i]}$ is known with error $\leqslant \eta^{[i]} 2^{-p}$ for $i = 0, 1$. We will see in section 4.3 that for such an evaluation tree with error $\leqslant \eta 2^{-p}$, the $G_{\boldsymbol{\epsilon};p}$ will be elements of $\mathbb{C}_r 2[z_1,\ldots,z_n]$ with $r = p - \log_2 \eta + O(1)$. Our next algorithm performs fast evaluation at $\boldsymbol{\xi}$ by taking advantage of evaluation trees.

Algorithm 3

Input. $P \in \mathbb{C}_p[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ and evaluation trees for $\boldsymbol{\xi}^{[i]}$ with error $\leqslant \eta^{[i]} 2^{-p}$ for $i = 0, 1$.

Output. An approximation of $P(\boldsymbol{\xi})$ in $(\mathbb{C}_q 2\Delta)^\Delta$ with error $\leqslant (2\Delta)\,2^{-q} = \Delta^{n+4} 2^{-p}$, where $q := p - (n+3)\log_2 \Delta + 1$.

Assumption. $\kappa \geqslant 4$ and $\varepsilon < \eta^2$.

1. If $\Delta = 1$ then return an approximation of $P(0)$ in $(\mathbb{C}_q 2\Delta)^\Delta$ with error $\leqslant (2\Delta)\,2^{-q}$, by using Lemma 2.
2. Compute a truncation $P'(z) \in \mathbb{C}_{p+1}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ of $P(\omega_1 z)$ with error $\leqslant 2^{-p}$.
3. Compute truncations $P^{[0]}$ and $P^{[1]}$ of P rem $I_{\boldsymbol{\xi}^{[0]}}$ and P' rem $I_{\boldsymbol{\xi}^{[1]}}$ in $\mathbb{C}_p 2^{n+2}[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ with error $\leqslant 2^{-p+n+2}$ via Corollary 18.
4. Apply Algorithm 3 recursively to $2^{-n-2}P^{[0]}$ and the evaluation sub-trees for $\boldsymbol{\xi}^{[0]}$ in order to obtain approximations $(v_0,\ldots,v_{\Delta/2-1}) \in (\mathbb{C}_{q^{[0]}}\Delta)^{\Delta/2}$ of $2^{-n-2}P^{[0]}(\boldsymbol{\xi}^{[0]})$ with error $\leqslant \Delta 2^{-q^{[0]}}$, where $q^{[0]} := p - (n^{[0]}+3)\log_2(\Delta/2) + 1$.

5. Apply Algorithm 3 recursively to $2^{-n-2}P^{[1]}$ and the evaluation sub-trees for $\boldsymbol{\xi}^{[1]}$ in order to obtain an approximation $(w_0,\ldots,w_{\Delta/2-1}) \in (\mathbb{C}_{q^{[1]}}\Delta)^{\Delta/2}$ of $2^{-n-2}P^{[1]}(\boldsymbol{\xi}^{[1]})$ with error $\leqslant \Delta 2^{-q^{[1]}}$, where $q^{[1]} := p - (n^{[1]}+3)\log_2(\Delta/2)+1$.
6. Return an approximation of $2^{n+2}(v_0, w_0, \ldots, v_{\Delta/2-1}, w_{\Delta/2-1})$ in $(\mathbb{C}_q 2\Delta)^\Delta$ with error $\leqslant (2\Delta)2^{-q}$.

PROPOSITION 19. *Algorithm 3 is correct and takes time*

$$O(\mathsf{I}(4^n\Delta p)(\log p + n + \log\Delta)\log\Delta),$$

whenever $n + \log\Delta = O(p)$.

Proof. If $\Delta = 1$ then the algorithm behaves as expected. Let us now assume that $\Delta \geqslant 2$. The correctness mostly relies on the identities

$$\begin{aligned} P(\xi_{2i}) &= (P \operatorname{rem} I_{\boldsymbol{\xi}^{[0]}})(\xi_{2i}) \\ P(\xi_{2i+1}) &= (P(\omega_1 z) \operatorname{rem} I_{\boldsymbol{\xi}^{[1]}})(\xi_{2i+1}/\omega_1), \end{aligned}$$

for $i = 0,\ldots,\Delta/2$. Let us verify that computations are done with sufficiently large precisions. By construction, we have $|P' - P(\omega_1 z)| \leqslant 2^{-p}$ and $|P'| \leqslant |P(\omega_1 z)| \leqslant 1$. Corollary 18 yields $|P^{[0]}| \leqslant 2^{n+2}$, $|P^{[1]}| \leqslant 2^{n+2}$ and

$$\begin{aligned} |P^{[0]} - (P \operatorname{rem} I_{\boldsymbol{\xi}^{[0]}})| &\leqslant 2^{-p+n+2} \\ |P^{[1]} - (P' \operatorname{rem} I_{\boldsymbol{\xi}^{[1]}})| &\leqslant |P'|2^{-p+n+2} \leqslant 2^{-p+n+2}. \end{aligned}$$

Since $\varepsilon < \eta^2 \leqslant \eta/\bar{\delta}$, Lemma 12 ensures that

$$(1+\varepsilon)^{n\bar{\delta}} \leqslant 2. \tag{9}$$

In particular, it follows that $|P(\boldsymbol{\xi})| \leqslant 2\Delta$. Assuming by induction that the claimed properties hold for the recursive call in step 4, we obtain

$$\begin{aligned} |2^{n+2}v_i - P(\xi_{2i})| &\leqslant 2^{n+2}|v_i - 2^{-n-2}P^{[0]}(\xi_{2i})| + |P^{[0]}(\xi_{2i}) - (P \operatorname{rem} I_{\boldsymbol{\xi}^{[0]}})(\xi_{2i})| \\ &\leqslant 2^{n+2}(\Delta/2)^{n+4}2^{-p} + (\Delta/2)2^{-p+n+2}(1+\varepsilon)^{n\bar{\delta}} \\ &\leqslant \Delta^{n+4}2^{-p-2} + \Delta 2^{-p+n+2} && \text{(using (9))} \\ &\leqslant \Delta^{n+4}2^{-p}(2^{-2}+2^{-1}) && \text{(since } \Delta \geqslant 2\text{)} \\ &\leqslant \Delta^{n+4}2^{-p}. \end{aligned}$$

Similarly, for w_i we obtain

$$\begin{aligned} |2^{n+2}w_i - P(\xi_{2i+1})| &\leqslant 2^{n+2}|w_i - 2^{-n-2}P^{[1]}(\xi_{2i+1}/\omega_1)| \\ &\quad + |P^{[1]}(\xi_{2i+1}/\omega_1) - (P' \operatorname{rem} I_{\boldsymbol{\xi}^{[1]}})(\xi_{2i+1}/\omega_1)| \\ &\quad + |P'(\xi_{2i+1}/\omega_1) - P(\xi_{2i+1})| \\ &\leqslant 2^{n+2}(\Delta/2)^{n+4}2^{-p} + (\Delta/2)2^{-p+n+2}(1+\varepsilon)^{n\bar{\delta}} + \Delta 2^{-p}(1+\varepsilon)^{n\bar{\delta}} \\ &\leqslant \Delta^{n+4}2^{-p-2} + \Delta 2^{-p+n+2} + \Delta 2^{-p+1} && \text{(using (9))} \\ &\leqslant \Delta^{n+4}2^{-p}(2^{-2}+2^{-1}+2^{-n-2}) && \text{(since } \Delta \geqslant 2\text{)} \\ &\leqslant \Delta^{n+4}2^{-p}. \end{aligned}$$

We are done with the correctness. As for the complexity analysis, the computation of $(\omega^i)_{i=0,\ldots,\Delta-1}$ and then of ω_1, with $O(p)$ bits of precision in step 2 can be done in time $O(\Delta\mathsf{I}(p))$ by Lemma 4, under the assumption $n + \log_2\Delta = O(p)$. Therefore, step 2 runs in time $O(\Delta\mathsf{I}(p))$. In view of Corollary 18, the complexity $\mathsf{T}(\delta_1,\ldots,\delta_n)$ of Algorithm 3 satisfies

$$\mathsf{T}(\delta_1,\ldots,\delta_n) \leqslant C\mathsf{I}(4^n\Delta(p+n+\log\Delta))(\log p + n + \log\Delta) + 2\mathsf{T}\Big(\delta_1,\ldots,\delta_{n-1},\frac{\delta_n}{2}\Big),$$

for some universal constant $C>0$. Unrolling this inequality, we obtain

$$\mathsf{T}(\delta_1,\ldots,\delta_n) \leqslant C\,\mathsf{I}(4^n\Delta p)\,(\log p+n+\log\Delta)\log\delta_n+\delta_n\,\mathsf{T}(\delta_1,\ldots,\delta_{n-1}).$$

Unrolling this new inequality, while using the geometric increase of 4^n, we obtain the desired complexity bound. □

4.2. Interpolation

Our interpolation algorithms rely on the evaluation one. We begin with a method dedicated to small precisions.

LEMMA 20. *Given $\boldsymbol{v}\in\mathbb{C}_p^{\Delta}$, we can compute $P\in\mathbb{C}_p[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that $|P|\leqslant|\boldsymbol{v}|$ and $|P(\boldsymbol{\omega})-\boldsymbol{v}|\leqslant\Delta 2^{-p+1}$ in time*

$$O(\mathsf{I}(\Delta\,(p+\log\Delta))).$$

Proof. Using Lemma 5, we compute $\boldsymbol{u}\in(\mathbb{C}_p\Delta)^{\Delta}$ such that

$$|\boldsymbol{u}-\mathrm{FFT}_{\omega^{-1}}(\boldsymbol{v})| \leqslant \Delta 2^{-p+1} \tag{10}$$

and $|\boldsymbol{u}|\leqslant\Delta|\boldsymbol{v}|$ in time $O(\mathsf{I}(\Delta\,(p+\log\Delta)))$. For P we take the unique polynomial of $\mathbb{C}_p[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that

$$P(x^{\Delta/\delta_1},x^{\Delta/(\delta_1\delta_2)},\ldots,x) = \Delta^{-1}\sum_{i=0,\ldots,\Delta-1}u_{i+1}x^i,$$

so we have $P(\boldsymbol{\omega})=\Delta^{-1}\mathrm{FFT}_{\omega}(\boldsymbol{u})$. Finally (10) yields

$$\begin{aligned}|P(\boldsymbol{\omega})-\boldsymbol{v}| &= |\Delta^{-1}\mathrm{FFT}_{\omega}(\boldsymbol{u})-\boldsymbol{v}|\\ &\leqslant \Delta 2^{-p+1}.\end{aligned}$$

□

LEMMA 21. *Assume $\kappa\geqslant 4$, $\varepsilon<\eta^2$, and $\Delta^{n+4}2^{-p+4}\leqslant 1$. Given an evaluation tree for $\boldsymbol{\xi}^{[i]}$ with error $\leqslant\eta^{[i]}2^{-p}$ for $i=0,1$, and $\boldsymbol{v}\in\mathbb{C}_p^{\Delta}$, we can compute $P\in(\mathbb{C}_p 2)[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ such that*

$$|P(\boldsymbol{\xi})-\boldsymbol{v}| \leqslant \Delta^{n+4}2^{-p+4}$$

in time $O(\mathsf{I}(4^n\Delta p^2)\,(\log p+n+\log\Delta)\log\Delta)$.

Proof. Let $q:=p-(n+3)\log_2\Delta+1$. By induction, we construct sequences $(P_i)_{\geqslant 0}$ and $(\boldsymbol{v}_i)_{i\geqslant 0}$ such that $P_i\in(\mathbb{C}_{p+i+1}2)[z_1,\ldots,z_n]$ and $\boldsymbol{v}_i\in(\mathbb{C}_p 2^{-i})^{\Delta}$. We set $P_0:=0$ and $\boldsymbol{v}_0:=\boldsymbol{v}$. For $i\geqslant 0$, we first compute $Q_i\in(\mathbb{C}_p 2^{-i})[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ as the polynomial interpolated from $\boldsymbol{v}_i$ by inverse FFT as in Lemma 20, so we have $|Q_i|\leqslant\boldsymbol{v}_i$ and $|Q_i(\boldsymbol{\omega})-\boldsymbol{v}_i|\leqslant\Delta 2^{-i-p+1}$. Using Algorithm 3, we then compute an approximation $\boldsymbol{w}_i\in(\mathbb{C}_q 2^{-i+1}\Delta)^{\Delta}$ of $Q_i(\boldsymbol{\xi})$ with error

$$\leqslant (2\Delta)\,2^{-i-q} = \Delta^{n+4}2^{-i-p}.$$

Using Lemma 3, we next compute a truncation $\boldsymbol{v}_{i+1}$ of $\boldsymbol{v}_i-\boldsymbol{w}_i$ with error $\leqslant 2^{-i-p+1/2}$. Then

$$\begin{aligned}|Q_i(\boldsymbol{\xi})-\boldsymbol{v}_i+\boldsymbol{v}_{i+1}| &\leqslant |Q_i(\boldsymbol{\xi})-\boldsymbol{w}_i|+|\boldsymbol{w}_i-\boldsymbol{v}_i+\boldsymbol{v}_{i+1}|\\ &\leqslant 2^{-i}\,(\Delta^{n+4}2^{-p}+2^{-p+1/2})\\ &\leqslant 2^{-i}\Delta^{n+4}2^{-p+1}.\end{aligned} \tag{11}$$

Using Lemmas 11 and 12, we obtain

$$\begin{aligned}
|Q_i(\boldsymbol{\omega}) - Q_i(\boldsymbol{\xi})| &\leqslant n\,\bar{\delta}\,\Delta\,|Q_i|\,(1+\varepsilon)^{n\bar{\delta}-1}\,\varepsilon \\
&\leqslant 2^{-i-2\kappa-3n+1} && \text{(using (9))} \\
&\leqslant 2^{-i-3n-7}. && (12)
\end{aligned}$$

It follows that

$$\begin{aligned}
|\boldsymbol{v}_{i+1}| &\leqslant |\boldsymbol{v}_i - \boldsymbol{w}_i| \\
&\leqslant |\boldsymbol{v}_i - Q_i(\boldsymbol{\omega})| + |Q_i(\boldsymbol{\omega}) - Q_i(\boldsymbol{\xi})| + |Q_i(\boldsymbol{\xi}) - \boldsymbol{v}_i + \boldsymbol{v}_{i+1}| + |\boldsymbol{v}_{i+1} - (\boldsymbol{v}_i - \boldsymbol{w}_i)| \\
&\leqslant 2^{-i}\,(\Delta 2^{-p+1} + 2^{-3n-7} + \Delta^{n+4}2^{-p+1} + 2^{-p+1}) && \text{(using (11) and (12))} \\
&\leqslant 2^{-i}\,(2^{-3} + 2^{-10} + 2^{-3} + 2^{-3}) && \text{(since } \Delta^{n+4}2^{-p+4} \leqslant 1) \\
&\leqslant 2^{-i-1},
\end{aligned}$$

so the sequence $(\boldsymbol{v}_i)_{i\geqslant 0}$ converges to 0. We define $P_{i+1} := P_i + Q_i$ and verify that

$$\begin{aligned}
|P_{i+1}(\boldsymbol{\xi}) - \boldsymbol{v}_0 + \boldsymbol{v}_{i+1}| &= \left| \sum_{0\leqslant j\leqslant i} (Q_j(\boldsymbol{\xi}) - \boldsymbol{v}_j + \boldsymbol{v}_{j+1}) \right| \\
&\leqslant \sum_{0\leqslant j\leqslant i} |Q_j(\boldsymbol{\xi}) - \boldsymbol{v}_j + \boldsymbol{v}_{j+1}| \\
&\leqslant \sum_{0\leqslant j\leqslant i} 2^{-j}\Delta^{n+4}2^{-p+1} && \text{(using (11))} \\
&= \Delta^{n+4}2^{-p+1} \sum_{0\leqslant j\leqslant i} 2^{-j} \\
&\leqslant \Delta^{n+4}2^{-p+2}. && (13)
\end{aligned}$$

On the other hand, we have $|P_{i+1}| \leqslant \sum_{0\leqslant j\leqslant i} |Q_j| \leqslant 2$. We take I minimal such that

$$2^{-I} \leqslant \Delta^{n+4}2^{-p},$$

so we have $I = O(p)$. Let P be a truncation of P_I with error $\leqslant 2^{-p+1}$. Then

$$\begin{aligned}
|P(\boldsymbol{\xi}) - \boldsymbol{v}| &\leqslant |P(\boldsymbol{\xi}) - P_I(\boldsymbol{\xi})| + |P_I(\boldsymbol{\xi}) - \boldsymbol{v}_0 + \boldsymbol{v}_I| + |\boldsymbol{v}_I| \\
&\leqslant (1+\varepsilon)^{n\bar{\delta}}\Delta 2^{-p+1} + \Delta^{n+4}2^{-p+2} + 2^{-I} && \text{(using (13))} \\
&\leqslant \Delta 2^{-p+2} + \Delta^{n+4}2^{-p+2} + \Delta^{n+4}2^{-p} && \text{(using (9))} \\
&\leqslant \Delta^{n+4}2^{-p+4}.
\end{aligned}$$

Finally, Lemma 20 asserts that each Q_i can be computed in time $O(\mathsf{I}(\Delta p))$. By Proposition 19, each $\boldsymbol{w}_i$ is obtained in time $O(\mathsf{I}(4^n \Delta p)\,(\log p + n + \log\Delta)\log\Delta)$. □

The next faster algorithm exploits the usual "divide and conquer" paradigm for large precisions p.

Algorithm 4

Input. $\boldsymbol{v} \in \mathbb{C}_p^{\Delta}$ with $\Delta \geqslant 2$ and evaluation trees for $\boldsymbol{\xi}^{[i]}$ with error $\leqslant \eta^{[i]}2^{-p}$ for $i = 0, 1$.

Output. $P \in (\mathbb{C}_p 2^2)[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n}$ such that $|P(\boldsymbol{\xi}) - \boldsymbol{v}| \leqslant \Delta^{n+4}2^{-p+4}$.

Assumption. $\kappa \geqslant 4$ and $\varepsilon < \eta^2$.

1. If $p < m := (n+4)\log_2\Delta + 11$ then compute $P' \in (\mathbb{C}_m 2)[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n}$ such that $|P'(\boldsymbol{\xi}) - \boldsymbol{v}| \leqslant \Delta^{n+4}2^{-m+4}$ via Lemma 21. Return an approximation P of P' in $(\mathbb{C}_p 2^2)[z_1, \ldots, z_n]_{\delta_1,\ldots,\delta_n}$ with error $\leqslant 2^{-p+2}$, by using Lemma 2.

2. Let $h := \lfloor p/2 \rfloor$ and $l := \lceil ((n+4)\log_2 \Delta + 7)/2 \rceil$ and let v' be a truncation of v in $\mathbb{C}^{\Delta}_{h+l}$ with error $\leqslant 2^{-h-l+1/2}$. Apply Algorithm 4 recursively to v' and let $P' \in (\mathbb{C}_{h+l}2^2)[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ denote the output.
3. Compute an approximation $w \in (\mathbb{C}_q 2^3\Delta)^{\Delta}$ of $P'(\xi)$ with error $\leqslant \Delta^{n+4}2^{-p+2}$, by applying Algorithm 3 to $2^{-2}P'$, where $q := p-(n+3)\log_2\Delta+1$.
4. Let $e := (n+4)\log_2\Delta - h - l + 5$ and compute a truncation $v'' \in (\mathbb{C}_{p-h+l}2^e)^{\Delta}$ of $v-w$ with error $\leqslant 2^{e-p+h-l+1/2}$. Apply Algorithm 4 recursively to $2^{-e}v''$ and let $P'' \in (\mathbb{C}_{p-h+l}2^2)[z_1,\ldots,z_n]_{\delta_1,\ldots,\delta_n}$ denote the output.
5. Return an approximation of $P' + 2^e P''$ with error $\leqslant 2^{-p+2}$.

PROPOSITION 22. *Algorithm 4 is correct and runs in time*

$$O(\mathsf{I}(4^n \Delta p\,(n\log\Delta)^4)\log^2 p).$$

Proof. If $p \geqslant m$, then $h > p/2 - 1 \geqslant l$, whence $p - h + l < p$, which implies the termination of the algorithm. In step 1 we obtain

$$\begin{aligned}
|P(\xi) - v| &\leqslant |P(\xi) - P'(\xi)| + |P'(\xi) - v| \\
&\leqslant \Delta 2^{-p+2}(1+\varepsilon)^{n\bar{\delta}} + \Delta^{n+4}2^{-m+4} \\
&\leqslant \Delta 2^{-p+3} + \Delta^{n+4}2^{-m+4} && \text{(using (9))} \\
&\leqslant \Delta^{n+4}2^{-p+3}(2^{-n-3}+1) && \text{(since } p<m \text{ and } \Delta\geqslant 2\text{)} \\
&\leqslant \Delta^{n+4}2^{-p+4},
\end{aligned}$$

so the output is correct when the algorithm exits at step 1. After step 3 we have

$$|P'(\xi) - v'| \leqslant \Delta^{n+4}2^{-h-l+4} \tag{14}$$
$$|P'(\xi) - w| \leqslant \Delta^{n+4}2^{-p+2}. \tag{15}$$

In step 4, we note that

$$h + l < \frac{p}{2} + \frac{(n+4)\log_2\Delta + 7}{2} + 1 \leqslant \frac{p}{2} + \frac{p-4}{2} + 1 = p - 1 \tag{16}$$

and deduce

$$\begin{aligned}
|v''| &\leqslant |v - w| \\
&= |P'(\xi) - v'| + |P'(\xi) - w| + |v' - v| \\
&\leqslant \Delta^{n+4}(2^{-h-l+4} + 2^{-p+2}) + 2^{-h-l+1/2} && \text{(using (14) and (15))} \\
&\leqslant \Delta^{n+4}(2^{-h-l+4} + 2^{-h-l} + 2^{-h-l-n-7/2}) && \text{(using (16) and } \Delta\geqslant 2\text{)} \\
&\leqslant \Delta^{n+4}2^{-h-l+5} = 2^e.
\end{aligned}$$

Therefore, the recursive call in step 4 is valid and gives

$$|P''(\xi) - 2^{-e}v''| \leqslant \Delta^{n+4}2^{-p+h-l+4}. \tag{17}$$

Combining (15) and (17) leads to

$$\begin{aligned}
|P'(\xi) + 2^e P''(\xi) - v| &\leqslant |P'(\xi) - w| + 2^e|P''(\xi) - 2^{-e}v''| + |v'' - v + w| \\
&\leqslant \Delta^{n+4}2^{-p+2} + 2^e\Delta^{n+4}2^{-p+h-l+4} + 2^{e-p+h-l+1/2} \\
&\leqslant \Delta^{n+4}2^{-p+4}(2^{-2} + 2^{(n+4)\log_2\Delta - 2l + 5} + 2^{-2l+3/2}) \\
&\leqslant \Delta^{n+4}2^{-p+4}(2^{-1} + 2^{-5.5}).
\end{aligned}$$

Consequently,

$$\begin{aligned}
|P(\xi)-v| &\leqslant |P'(\xi)+2^e P''(\xi)-v|+|P(\xi)-P'(\xi)-2^e P''(\xi)| \\
&\leqslant \Delta^{n+4}2^{-p+4}(2^{-1}+2^{-5.5})+\Delta 2^{-p+2}(1+\varepsilon)^{n\bar{\delta}} \\
&\leqslant \Delta^{n+4}2^{-p+4}(2^{-1}+2^{-5.5})+\Delta 2^{-p+3} && \text{(using (9))} \\
&\leqslant \Delta^{n+4}2^{-p+4}(2^{-1}+2^{-5.5}+2^{-n-4}) && \text{(since } \Delta\geqslant 2\text{)} \\
&\leqslant \Delta^{n+4}2^{-p+4}.
\end{aligned}$$

Step 1 takes time

$$O(\mathsf{I}(4^n\Delta m^2)\,(\log m+n+\log\Delta)\log\Delta)=O(\mathsf{I}(4^n\Delta m^4)),$$

by Lemma 21. Steps 3 runs in time

$$O(\mathsf{I}(4^n\Delta p)\,(\log p+n+\log\Delta)\log\Delta)=O(\mathsf{I}(4^n\Delta p\,m^2)\log p),$$

by Proposition 19. When $p\geqslant m$, the complexity $\mathsf{T}(p)$ of our algorithm therefore satisfies

$$\mathsf{T}(p)\ \leqslant\ C\,\mathsf{I}(4^n\Delta p\,m^2)\log p+\mathsf{T}(h+l)+\mathsf{T}(p-h+l),$$

for some universal constant $C>0$. Unrolling this inequality, we obtain

$$\begin{aligned}
\mathsf{T}(p) &= O\Big(\mathsf{I}(4^n\Delta p\,m^2)\log^2 p+\left\lceil\frac{p}{m}\right\rceil\mathsf{I}(4^n\Delta m^4)\Big) \\
&= O(\mathsf{I}(4^n\Delta p\,m^4)\log^2 p),
\end{aligned}$$

which concludes the proof. □

4.3. Computation of the evaluation tree

The evaluation tree for ξ is obtained by induction: we recursively compute evaluation trees for $\xi^{[0]}$ and $\xi^{[1]}$, and deduce the one for ξ as follows.

Algorithm 5

Input. An approximation $\tilde{\xi}\in(\mathbb{C}_q^n 2)^\Delta$ of ξ with error $\leqslant 2^{-q}$ where $q:=p-3\log_2\eta+(n+4)\log_2^2\Delta$.

Output. An evaluation tree for ξ with error $\leqslant\eta 2^{-p}$.

Assumption. $\kappa\geqslant 4$, $|\xi-\omega|\leqslant\eta^3$.

1. If $\Delta=1$ then return $G_{(0);p}:=1$ and an approximation $G_{(1);p}$ of $z_1-\tilde{\xi}_{1,1}$ with error $\leqslant\eta 2^{-p}$.
2. Let $\tilde{p}:=p+(n+4)\log_2\Delta$. For $i=0,1$, let $q^{[i]}:=\tilde{p}-3\log_2\eta^{[i]}+(n+4)\log_2^2\Delta^{[i]}$ and compute $\xi^{[i]}\in(\mathbb{C}_{q^{[i]}}^n 2)^\Delta$ with error $\leqslant 2^{-q^{[i]}}$. Recursively compute the evaluation trees $T^{[i]}$ for $\xi^{[i]}$ with error $\leqslant\eta^{[i]}2^{-\tilde{p}}$.
3. For each $\varepsilon\in\mathbb{E}_n^*$ do:
   a. Compute an approximation $\tilde{v}\in(\mathbb{C}_{\tilde{p}}\eta^2)^\Delta$ of $-F_\varepsilon(\tilde{\xi})$ with error $\leqslant\eta^2 2^{-\tilde{p}}$.
   b. Apply Algorithm 4 to $\eta^{-2}\tilde{v}$, $T^{[0]}$, $T^{[1]}$; let $\tilde{E}_\varepsilon\in(\mathbb{C}_{\tilde{p}}2^2)[z_1,\ldots,z_n]$ be the returned polynomial, such that $|\tilde{E}_\varepsilon(\xi)-\eta^{-2}\tilde{v}|\leqslant\Delta^{n+4}2^{-\tilde{p}+4}$.
   c. Compute an approximation $G_{\varepsilon;p}\in(\mathbb{C}_r 2)[z_1,\ldots,z_n]$ of $F_\varepsilon+\eta^2\tilde{E}_\varepsilon$ with error $\leqslant 2^{-r+1/2}$, where $r:=p-\log_2\eta+1$.
4. Set $G_{\mathbf{0};p}:=1$ and return $(G_{\varepsilon;p})_{\varepsilon\in\mathbb{E}_n}\in(\mathbb{C}_r 2)[z_1,\ldots,z_n]^{\mathbb{E}_n}$, and approximations of $T^{[0]}$, $T^{[1]}$ with errors $\leqslant\eta^{[0]}2^{-p}$ and $\leqslant\eta^{[1]}2^{-p}$.

PROPOSITION 23. *Algorithm 5 is correct and runs in time*

$$O(2^n \mathsf{I}(4^n \Delta p\,(n \log \Delta)^5) \log^2 p),$$

provided that $n \log^2 \Delta = O(p)$.

Proof. If the algorithm exits in step 1, then we have $G_{(1)} = z_1 - \xi_{1,1}$, so the output is correct. Now, let us assume that $\Delta \geqslant 2$ and note that

$$\begin{aligned} q^{[i]} &\leqslant p - 3\log_2 \eta + (n+4)\log_2^2 \Delta - (n+4)\log_2 \Delta + n + 1 \\ &\leqslant q - 3 \end{aligned} \tag{18}$$

and

$$|\tilde{\xi}_i| \leqslant |\omega_i| + |\xi_i - \omega_i| + |\xi_i - \tilde{\xi}_i| \leqslant 1 + \eta^3 + 2^{-q} \leqslant 1 + \eta^3 + \eta^3 2^{-4} \leqslant 1.1. \tag{19}$$

For $i = 0, \ldots, \Delta/2$, we compute $\tilde{\xi}_i^{[0]}$ as the truncation of $\tilde{\xi}_{2i}$ with error $\leqslant 2^{-q^{[0]}-1/2}$ using Lemma 2. Hence,

$$\begin{aligned} |\tilde{\xi}_i^{[0]} - \xi_i^{[0]}| &\leqslant |\tilde{\xi}_i^{[0]} - \tilde{\xi}_{2i}| + |\tilde{\xi}_{2i} - \xi_i^{[0]}| \\ &\leqslant 2^{-q^{[0]}-1/2} + 2^{-q} \\ &\leqslant 2^{-q^{[0]}}(2^{-1/2} + 2^{-3}) && \text{(using (18))} \\ &\leqslant 2^{-q^{[0]}}. \end{aligned}$$

From Lemma 4, we may compute a truncation $w \in \mathbb{C}_{q+3}$ of ω^{-1} with error $\leqslant 2^{-q-2}$ in time $O(\Delta \mathsf{I}(p))$, provided that $n \log^2 \Delta = O(p)$. We compute an approximation χ_i of $w\tilde{\xi}_{2i+1}$ with error $\leqslant 2^{-q-3}$. It follows that

$$\begin{aligned} |\chi_i - \xi_i^{[1]}| &\leqslant |\chi_i - w\tilde{\xi}_{2i+1}| + |w\tilde{\xi}_{2i+1} - \omega^{-1}\tilde{\xi}_{2i+1}| + |\omega^{-1}\tilde{\xi}_{2i+1} - \omega^{-1}\xi_{2i+1}| \\ &\leqslant 2^{-q-3} + 1.1 \times 2^{-q-2} + 2^{-q} && \text{(using (19))} \\ &\leqslant 2^{-q}(2^{-3} + 1.1 \times 2^{-2} + 1) \\ &\leqslant 2^{-q+1/2} \\ &\leqslant 2^{-q^{[1]}-5/2}. && \text{(using (18))} \end{aligned}$$

Using Lemma 2, we obtain $\tilde{\xi}_i^{[1]}$ as the truncation of χ_i with error $\leqslant 2^{-q^{[1]}-1/2}$. Hence,

$$|\tilde{\xi}_i^{[1]} - \xi_i^{[1]}| \leqslant |\tilde{\xi}_i^{[1]} - \chi_i| + |\chi_i - \xi_i^{[1]}| \leqslant 2^{-q^{[1]}-5/2} + 2^{-q^{[1]}-1/2} \leqslant 2^{-q^{[1]}}.$$

For step 3.a, we need to verify that

$$\begin{aligned} |\tilde{v}| &\leqslant |F_\epsilon(\tilde{\boldsymbol{\xi}})| + \eta^2 2^{-\tilde{p}} \\ &= |F_\epsilon(\tilde{\boldsymbol{\xi}}) - F_\epsilon(\boldsymbol{\omega})| + \eta^2 2^{-\tilde{p}} \\ &\leqslant n\,(\bar{\delta}+1)\,2\,(1+\eta^3+\eta^3 2^{-4})^{n(\bar{\delta}+1)}\,|\tilde{\boldsymbol{\xi}} - \boldsymbol{\omega}| + \eta^2 2^{-\tilde{p}} && \text{(by Lemma 11 and (19))} \\ &\leqslant n\,\bar{\delta}\,2^2\,(1+\eta/(\bar{\delta}+1))^{n(\bar{\delta}+1)}\,|\tilde{\boldsymbol{\xi}} - \boldsymbol{\omega}| + \eta^2 2^{-\tilde{p}} \\ &\leqslant n\,\bar{\delta}\,2^3 |\tilde{\boldsymbol{\xi}} - \boldsymbol{\omega}| + \eta^2 2^{-\tilde{p}} && \text{(by Lemma 12)} \\ &\leqslant n\,\bar{\delta}\,2^3\,(|\tilde{\boldsymbol{\xi}} - \boldsymbol{\xi}| + |\boldsymbol{\xi} - \boldsymbol{\omega}|) + \eta^2 2^{-\tilde{p}} \\ &\leqslant n\,\bar{\delta}\,2^3\,(2^{-q} + \eta^3) + \eta^2 2^{-\tilde{p}} \\ &\leqslant 2^n \Delta 2^3\,(\eta^3 + \eta^3) + \eta^2 2^{-\tilde{p}} \\ &\leqslant 2^n \Delta 2^4 \eta^3 + \eta^2 2^{-\tilde{p}} \\ &\leqslant \eta^2, && \text{(since } \kappa \geqslant 4 \text{ and } \tilde{p} \geqslant 5) \end{aligned}$$

so the use of Algorithm 4 is valid in step 3.b. Thanks to Proposition 22 and using that $E_\epsilon(\xi) = -F_\epsilon(\xi)$, we obtain

$$\begin{aligned}
&|\eta^2 \tilde{E}_\epsilon(\xi) - E_\epsilon(\xi)| \\
&\leqslant \eta^2 |\tilde{E}_\epsilon(\xi) - \eta^{-2}\tilde{v}| + |\tilde{v} + F_\epsilon(\tilde{\xi})| + |F_\epsilon(\tilde{\xi}) - F_\epsilon(\xi)| \\
&\leqslant \eta^2 \Delta^{n+4} 2^{-\tilde{p}+4} + \eta^2 2^{-\tilde{p}} \\
&\qquad + n\,(\bar{\delta}+1)\,2\,(1+\eta^3+\eta^3 2^{-4})^{n(\bar{\delta}+1)} |\tilde{\xi} - \xi| \qquad \text{(by Lemma 11 and (19))} \\
&\leqslant \eta^2 2^{-p+4} + \eta^2 2^{-p-5} + n\,\bar{\delta} 2^2 (1 + \eta/(\bar{\delta}+1))^{n(\bar{\delta}+1)} |\tilde{\xi} - \xi| \\
&\leqslant \eta^2 2^{-p+4} + \eta^2 2^{-p-5} + n\,\bar{\delta} 2^3 |\tilde{\xi} - \xi| \qquad \text{(by Lemma 12)} \\
&\leqslant \eta^2 2^{-p+4} + \eta^2 2^{-p-5} + n\,\bar{\delta}\eta^3 2^{-p-2} \\
&\leqslant \eta 2^{-p} (2^{-3} + 2^{-12} + 2^{-7}) \\
&\leqslant \eta 2^{-p-2.9}.
\end{aligned}$$

Lemma 13 yields $|\eta^2 \tilde{E}_\epsilon - E_\epsilon| \leqslant \eta 2^{-p-1.9}$. It follows that

$$\begin{aligned}
|G_{\epsilon;p} - (F_\epsilon + E_\epsilon)| &\leqslant |G_{\epsilon;p} - (F_\epsilon + \eta^2 \tilde{E}_\epsilon)| + |\eta^2 \tilde{E}_\epsilon - E_\epsilon| \\
&\leqslant 2^{-r+1/2} + \eta 2^{-p-1} \\
&\leqslant \eta 2^{-p} (2^{-1/2} + 2^{-1.9}) \\
&\leqslant \eta 2^{-p}.
\end{aligned}$$

This concludes the correctness proof. As for the complexity we note that $\tilde{p} = O(p)$ whenever $n \log^2 \Delta = O(p)$. In step 3, $F_\epsilon(\tilde{\xi})$ can be computed in time $O(\mathsf{I}(\Delta p\, n) \log \Delta)$, using binary powering. In step 3.b, the interpolation of $\tilde{E}_\epsilon$ takes

$$O(\mathsf{I}(4^n \Delta p\, (n \log \Delta)^4) \log^2 p)$$

operations, by Proposition 22. Overall the complexity $\mathsf{T}(\Delta)$ of our algorithm satisfies

$$\mathsf{T}(\Delta) \leqslant 2\,\mathsf{T}(\Delta/2) + C 2^n \mathsf{I}(4^n \Delta p\, (n \log \Delta)^4) \log^2 p,$$

for some universal constant $C > 0$. Unrolling this inequality yields the claimed complexity bound. □

5. MODULAR COMPOSITION

Consider $P, Q \in (\mathbb{Z}/r\mathbb{Z})[x]_D$ and $R \in x^D + (\mathbb{Z}/r\mathbb{Z})[x]_D$, where r is a positive integer. Our goal is to compute $(P \circ Q) \operatorname{rem} R$. We first follow Kedlaya and Umans [14], by transforming the problem into a multivariate one. For some $n > 1$ and powers of two $\tilde{\delta}_1, \delta_2, \ldots, \delta_n$ with $\tilde{\Delta} := \tilde{\delta}_1 \delta_2 \cdots \delta_n \geqslant D$ and $\delta_2 \cdots \delta_n < D$, there exists a unique lift

$$\check{P} \in (\mathbb{Z}/r\mathbb{Z})[z_1, \ldots, z_n]_{\tilde{\delta}_1, \delta_2, \ldots, \delta_n}$$

such that

$$P(x) := \check{P}(x^{\delta_2 \cdots \delta_n}, x^{\delta_3 \cdots \delta_n}, \ldots, x).$$

The first degree bound $\tilde{\delta}_1$ will be adjusted below to a larger value δ_1 that will match the setting of the previous section. We define

$$Q_i := Q^{\delta_{i+1} \cdots \delta_n} \operatorname{rem} R, \qquad i = 1, \ldots, n,$$

so we have

$$(P \circ Q) \operatorname{rem} R = \check{P}(Q_1, \ldots, Q_n) \operatorname{rem} R. \tag{20}$$

In order to compute the right-hand side of (20), we further lift the problem into another one with integer coefficients: let $\breve{P} \in \mathbb{Z}[z_1, \ldots, z_n]_{\tilde{\delta}_1, \delta_2, \ldots, \delta_n}$ and $\breve{Q}_1, \ldots, \breve{Q}_n \in \mathbb{Z}[x]_D$ be the canonical lifts of $\breve{P}, Q_1, \ldots, Q_n$ with coefficients in $\{0, \ldots, r-1\}$ and set

$$\begin{aligned} \tilde{P} &:= \breve{P} \\ \tilde{Q}_i &:= \breve{Q}_i + rMx^{\delta_{i+1} \cdots \delta_n}, \qquad i = 1, \ldots, n, \end{aligned}$$

where M is a positive integer that will be specified below. Note that $\tilde{P}, \tilde{Q}_1, \ldots, \tilde{Q}_n$ are still lifts of $\breve{P}, Q_1, \ldots, Q_n$, so

$$\breve{P}(Q_1, \ldots, Q_n) \operatorname{rem} R = (\tilde{P}(\tilde{Q}_1, \ldots, \tilde{Q}_n) \operatorname{rem} r) \operatorname{rem} R.$$

We finally rescale the polynomials, so that we can compute $\tilde{P}(\tilde{Q}_1, \ldots, \tilde{Q}_n)$ using a spiroid:

$$\begin{aligned} \hat{P}(z_1, \ldots, z_n) &:= \tilde{P}(rMz_1, \ldots, rMz_n) \\ \hat{Q}_i(x) &:= \frac{1}{rM} \tilde{Q}_i(x), \qquad i = 1, \ldots, n. \end{aligned}$$

Note that

$$\tilde{P}(\tilde{Q}_1, \ldots, \tilde{Q}_n) = \hat{P}(\hat{Q}_1, \ldots, \hat{Q}_n)$$

and

$$|\hat{Q}_i - x^{\delta_{i+1} \cdots \delta_n}| < \frac{1}{M}, \qquad i = 1, \ldots, n.$$

We will evaluate

$$\hat{P}(\hat{Q}_1, \ldots, \hat{Q}_n)$$

numerically using fixed point arithmetic. Setting $d := \deg \hat{P}$, we note that

$$\begin{aligned} d &\leqslant \tilde{\delta}_1 + \delta_2 + \cdots + \delta_n - n \\ \deg(\tilde{P}(\tilde{Q}_1, \ldots, \tilde{Q}_n)) &< d D. \end{aligned}$$

Let Δ be the smallest power of two satisfying $\Delta \geqslant d\tilde{\Delta}$, let $\delta_1 := \Delta / (\delta_2 \cdots \delta_n) > d\tilde{\delta}_1$, and consider the standard primitive Δ-th root of unity $\omega := \mathrm{e}^{2\pi \mathrm{i}/\Delta} \in \mathbb{C}$. We wish to determine $\hat{P}(\hat{Q}_1, \ldots, \hat{Q}_n)$ from its values at $1, \omega, \ldots, \omega^{\Delta-1}$. For $i = 0, \ldots, \Delta - 1$, we let

$$\xi_i := (\hat{Q}_1(\omega^i), \ldots, \hat{Q}_n(\omega^i)),$$

so we have

$$|\boldsymbol{\xi} - \boldsymbol{\omega}| \leqslant \frac{D}{M}. \tag{21}$$

and

$$\hat{P}(\hat{Q}_1, \ldots, \hat{Q}_n)(\omega^i) = \hat{P}(\xi_i).$$

Choosing M sufficiently large, we will be able to apply Algorithm 3 in order to evaluate $\hat{P}$ at the spiroid $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_{\Delta-1})$. We are now ready to prove the main theorem of this paper, from which Theorem 1 follows.

THEOREM 24. *The cost of degree D modular composition over $\mathbb{Z}/r\mathbb{Z}$ is bounded by*

$$O\Big(D^{1+5/\sqrt{\log_2 D}} \log^{13.5} D \log r \, (\log\log r)^3\Big).$$

Proof. Using binary powering, the computation of $Q_1, \ldots, Q_n$ takes time

$$O(\mathsf{M}(D)\, \mathsf{I}(\log r) \log D) = O(D \log^3 D \log r \log\log r).$$

Now we take

$$n := \lceil \sqrt{\log_2 D} \rceil - 1,$$

whence $\log D \asymp n^2$. More precisely, we have

$$\sqrt{\log_2 D} - 1 \leqslant n < \sqrt{\log_2 D},$$

$2^{n^2} < D$, and $2^{n^2+2n+1} \geqslant D$. We distinguish the following cases in order to construct the sequence $\tilde{\delta}_1, \delta_2, \ldots, \delta_n$:

- If $2^{n^2+n} \geqslant D$ then we take $k \geqslant 0$ minimal such that $2^{n^2+k} \geqslant D$, and then $\delta_n = \cdots = \delta_{n-k+1} = 2^{n+1}$ and $\delta_{n-k} = \cdots = \delta_2 = \tilde{\delta}_1 = 2^n$. Note that $k \leqslant n$ does exist.
- Otherwise, $2^{n^2+n} < D$.
  - If $2^{n^2+2n} \geqslant D$ then we take $k \geqslant 0$ minimal such that $2^{n^2+n+k} \geqslant D$, and then $\delta_n = \cdots = \delta_{n-k+1} = 2^{n+2}$ and $\delta_{n-k} = \cdots = \delta_2 = \tilde{\delta}_1 = 2^{n+1}$.
  - Otherwise we take $\delta_n = 2^{n+3}$, $\delta_{n-1} = \cdots = \delta_2 = \tilde{\delta}_1 = 2^{n+2}$.

In this way we achieve $D \leqslant \tilde{\Delta} < 2D$ and

$$d = O(n 2^n), \tag{22}$$

whence

$$\Delta = O(n 2^n D) = O(2^n D \log^{1/2} D) \tag{23}$$

and

$$\log \Delta = O(\log D). \tag{24}$$

Recall that $\eta := \frac{2^{-\kappa}}{2^{2n}\Delta}$. Now we take $\kappa := 4$ and $M \in \mathbb{N}$ minimal such that

$$\frac{D}{M} \leqslant \eta^3,$$

so, using (24), we have

$$\log M = O(n + \log \Delta + \log D) = O(\log D). \tag{25}$$

With this choice for M, the inequality (21) implies

$$|\xi - \omega| \leqslant \eta^3.$$

Let 2^m be the smallest power of two that satisfies $2^m \geqslant r\,(rM)^d$, so we have $|\hat{P}| \leqslant 2^m$. We set

$$p := m - \log_2 \eta + (n+4) \log_2^2 \Delta.$$

From (22) and (25) we obtain that

$$\begin{aligned} p &= O(d\,(\log r + \log M) + n \log^2 \Delta) \\ &= O(2^n\,(\log r + \log D) \log^{5/2} D) \\ &= O(2^n \log r \log^{7/2} D) \end{aligned} \tag{26}$$

and that

$$\log p = O(\log \log r + \log^{1/2} D). \tag{27}$$

Since $|\hat{Q}_1| = \cdots = |\hat{Q}_n| = 1$, using Lemma 5 with $q := p - 3\log_2 \eta + (n+4)\log_2^2 \Delta = O(p)$, we compute a truncation $\tilde{\xi}$ of ξ with error $\leqslant \Delta 2^{-q+1}$ in time

$$\begin{aligned} & O(\mathsf{I}(\Delta p)) \\ &= O(\mathsf{I}(2^{2n} D \log r \log^4 D)) && \text{(using (23) and (26))} \\ &= O(2^{2n} D \log r \log^4 D\,(\log \log r + \log D)) && \text{(using (24) and (27))} \\ &= O(2^{2n} D \log r \log^5 D \log \log r). && (28) \end{aligned}$$

Since $n\log^2\Delta = O(p)$, Proposition 23 allows us to compute an evaluation tree for $\boldsymbol{\xi}$ with error $\leqslant \eta 2^{-p}$ in time

$$\begin{aligned}
& O(2^n \mathsf{I}(4^n \Delta p\,(n\log\Delta)^5)\log^2 p) \\
=\ & O(2^{3n}\Delta p\,(n\log\Delta)^5 \log(4^n\Delta p\, n\log\Delta)\log^2 p) \\
=\ & O(2^{5n} D\log^4 D\log r && \text{(using (23) and (26))} \\
& \quad \times (n\log\Delta)^5\log(4^n\Delta p)\log^2 p) \\
=\ & O(2^{5n} D\log r\log^{13.5} D\,(\log\log r)^3). && \text{(using (24) and (27))} \qquad (29)
\end{aligned}$$

Following Proposition 19, this bound dominates the cost for obtaining an approximation $\boldsymbol{v} \in (\mathbb{C}_{q'}\Delta 2^{m+1})^\Delta$ of $\hat{P}(\boldsymbol{\xi})$ with error $\leqslant \Delta^{n+4}2^{m-p}$, where $q' := p - (n+3)\log_2\Delta + 1$, so we have

$$|\hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)(\boldsymbol{\omega}) - \boldsymbol{v}| \ \leqslant\ \Delta^{n+4}2^{m-p}$$

hence

$$|\hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n) - \mathrm{FFT}_\omega^{-1}(\boldsymbol{v})| \ \leqslant\ \Delta^{n+4}2^{m-p}, \qquad (30)$$

thanks to Lemma 10. Via Lemma 5, we compute $A(x) \in (\mathbb{C}_p\Delta 2^{m+1})[x]_\Delta$ such that

$$|A - \mathrm{FFT}_\omega^{-1}(\boldsymbol{v})| \ \leqslant\ \Delta 2^{m-p+2} \qquad (31)$$

in time $O(\mathsf{I}(\Delta p)) = O(2^{2n} D\log r\log^5 D\log\log r)$, using (28). Finally, we get

$$\begin{aligned}
|A - \hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)| \ &\leqslant\ |A - \mathrm{FFT}_\omega^{-1}(\boldsymbol{v})| + |\mathrm{FFT}_\omega^{-1}(\boldsymbol{v}) - \hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)| \\
&\leqslant\ \Delta 2^{m-p+2} + \Delta^{n+4}2^{m-p} && \text{(by (31) and (30))} \\
&\leqslant\ \eta\Delta 2^{-(n+4)\log_2^2\Delta+2} + \eta 2^{(n+4)\log_2\Delta - (n+4)\log_2^2\Delta} \\
&<\ 2^{-1},
\end{aligned}$$

so rounding the coefficients of A to the nearest integers yields $\hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)$. Since

$$\begin{aligned}
\log|A| \ &=\ O(\log D + m) \\
&=\ O(\log D + d\log(rM)) \\
&=\ O(n2^n(\log r + \log D)) && \text{(using (22) and (25))} \\
&=\ O(2^n \log r\log^{3/2} D),
\end{aligned}$$

we compute $\hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)$ rem r in time

$$\begin{aligned}
O(\Delta\,\mathsf{I}(\log|A|)) \ &=\ O(2^n\Delta\log r\log^{3/2} D\log(2^n\log r\log D)) \\
&=\ O(2^n\Delta\log r\log^2 D\log\log r) \\
&=\ O(2^{2n} D\log r\log^{5/2} D\log\log r). && \text{(using (23))}
\end{aligned}$$

The final division of $\hat{P}(\hat{Q}_1,\ldots,\hat{Q}_n)$ by R over $\mathbb{Z}/r\mathbb{Z}$ has cost

$$\begin{aligned}
O(\mathsf{M}(\Delta)\,\mathsf{I}(r)) \ &=\ O(\Delta\log^2\Delta\log r\log\log r) \\
&=\ O(2^n D\log r\log^{5/2} D\log\log r). && \text{(using (23) and (24))}
\end{aligned}$$

The total cost of the method is dominated by (29). □

BIBLIOGRAPHY

[1] V. Bhargava, S. Ghosh, Z. Guo, M. Kumar, and C. Umans. Fast multivariate multipoint evaluation over all finite fields. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 221–232. New York, NY, USA, 2022. IEEE.

[2] V. Bhargava, S. Ghosh, M. Kumar, and C. K. Mohapatra. Fast, algebraic multivariate multipoint evaluation in small characteristic and applications. *J. ACM*, 70(6):1–46, 2023. Article No. 42.

[3] R. P. Brent and H. T. Kung. Fast algorithms for manipulating formal power series. *J. ACM*, 25(4):581–595, 1978.

[4] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1997.

[5] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, 3rd edition, 2013.

[6] D. Harvey and J. van der Hoeven. Integer multiplication in time $O(n \log n)$. *Ann. Math.*, 193(2):563–617, 2021.

[7] D. Harvey, J. van der Hoeven, and G. Lecerf. Even faster integer multiplication. *J. Complexity*, 36:1–30, 2016.

[8] J. van der Hoeven. *The Jolly Writer. Your Guide to GNU TeXmacs*. Scypress, 2020.

[9] J. van der Hoeven and G. Lecerf. Composition modulo powers of polynomials. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, ISSAC '17, pages 445–452. New York, NY, USA, 2017. ACM.

[10] J. van der Hoeven and G. Lecerf. Modular composition via factorization. *J. Complexity*, 48:36–68, 2018.

[11] J. van der Hoeven and G. Lecerf. Fast multivariate multi-point evaluation revisited. *J. Complexity*, 56:101405, 2020.

[12] J. van der Hoeven and G. Lecerf. Univariate polynomial factorization over finite fields with large extension degree. *Appl. Algebra Eng. Commun. Comput.*, 35:121–149, 2024.

[13] E. Kaltofen and V. Shoup. Fast polynomial factorization over high algebraic extensions of finite fields. In *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, ISSAC '97, pages 184–188. New York, NY, USA, 1997. ACM.

[14] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Comput.*, 40(6):1767–1802, 2011.

[15] Y. Kinoshita and B. Li. Power series composition in near-linear time. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2180–2185. IEEE, 2024.

[16] V. Neiger, B. Salvy, É. Schost, and G. Villard. Faster modular composition. *J. ACM*, 71(2):1–79, 2023. Article No. 11.

[17] M. S. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2(1):60–66, 1973.

[18] A. Schönhage. Asymptotically fast algorithms for the numerical muitiplication and division of polynomials with complex coefficients. In J. Calmet, editor, *Computer Algebra. EUROCAM '82, European Computer Algebra Conference, Marseilles, France, April 5-7, 1982*, volume 144 of *Lect. Notes Comput. Sci.*, pages 3–15. Berlin, Heidelberg, 1982. Springer-Verlag.

[19] V. V. Williams, Y. Xu, Z. Xu, and R. Zhou. New bounds for matrix multiplication: from alpha to omega. In D. P. Woodruff, editor, *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835. Philadelphia, PA 19104 USA, 2024. SIAM.