Making fast multiplication of polynomials numerically stable*

Joris van der Hoeven

CNRS, Dépt. de Mathématiques (Bât. 425)
Université Paris-Sud
91405 Orsay Cedex
France
Email: joris@texmacs.org

December 9, 2008

Consider two polynomials P and Q with multiple precision floating point coefficients. Although the product $P\,Q$ can in principle be computed efficiently using FFT multiplication, this algorithm is numerically stable only if the coefficients of P are of the same order of magnitude and similarly for Q. In this paper, we present a new asymptotically fast multiplication algorithm which has a "better" numerically stability. We also provide some theoretical support for what we feel to be "better", by introducing the concept of "relative Newton error".

Keywords: polynomial, floating point number, multiplication, algorithm, FFT

A.M.S. subject classification: 65T50, 42-04, 68W30

1. Introduction

Let $\mathbb{F}_{n,s}$ be the set of floating point numbers with an $(n-1)$-bit mantissa and an s-bit exponent. Each non-zero number $x\in\mathbb{F}_{n,s}$ of this kind can be written as $x=\pm m_x 2^{e_x}$, with

$$\begin{aligned} m_x &\in \left\{\frac{2^{n-1}}{2^n},\ldots,\frac{2^n-1}{2^n}\right\}\\ e_x &\in \{-2^{s-1},\ldots,-1,0,1,\ldots,2^{s-1}\}. \end{aligned}$$

In what follows, s is assumed to be fixed once and for all. For the sake of simplicity, we will not study overflows and underflows in detail and instead make the assumption that s has been chosen large enough with respect to the input data. Consider two polynomials

$$\begin{aligned} P &= P_0+\cdots+P_{p-1}z^{p-1}\\ Q &= Q_0+\cdots+Q_{q-1}z^{q-1} \end{aligned}$$

with coefficients in $\mathbb{F}_{n,s}$. For definiteness, we will assume $P_0\neq 0$, $P_{p-1}\neq 0$, $Q_0\neq 0$ and $Q_{q-1}\neq 0$. In this paper, we study the problem of multiplying P and Q.

Using FFT-multiplication [CT65], the polynomials P and Q can in principle be multiplied in time $O(d\log d\,\mathsf{M}(n))$. Here $\mathsf{M}(n)=O(n\log n\log\log n)$ is the complexity of n-bit integer multiplication [SS71] and it can be assumed that $\mathsf{M}(n)/n$ increases with n. Alternatively, one may rewrite P and Q as "floating point polynomials", with an s-bit integer exponent and polynomials with n-bit integer coefficients as mantissas. Using Kronecker's method, we may then multiply P and Q in time $O(\mathsf{M}(n\,d))$, where $d=\max\{p,q\}$. If $p\gg q$, then this complexity can be improved a bit further into $O(\mathsf{M}(n\,q)\;p/q)$ by using $O(p/q)$ multiplications of degrees q.

*. This work has partially been supported by the ANR Gecko project.

In the case when the coefficients of P are all of the same order of magnitude, and similarly for Q, the above algorithms are very efficient and numerically stable. In section 2, we will give a more detailed uniform error analysis. Unfortunately, the algorithms are very unstable if the coefficients of P or Q are not of the same order of magnitude. For instance, taking $P = Q = z + \varepsilon$ with $\varepsilon < 2^{-n}$, the transformation of P and Q into floating point polynomials $\tilde{P} = z \approx P$ and $\tilde{Q} = z \approx Q$ amounts to neglecting the coefficients P_0 and Q_0. In particular, when approximating $P Q \approx \tilde{P} \tilde{Q}$, only the leading coefficient $(P Q)_2$ of the product will be computed with a satisfactory accuracy.

A situation which arises often in practice is that the above fast multiplication algorithms can be made numerically stable using a suitable scaling

$$\begin{array}{rcl} P(z) & \mapsto & P_{\times \lambda}(z) = P(\lambda z) \\ Q(z) & \mapsto & Q_{\times \lambda}(z) = Q(\lambda z). \end{array}$$

For instance, in the above example when $P = Q = z + \varepsilon$, we may multiply P and Q using

$$PQ = (P_{\times \varepsilon} Q_{\times \varepsilon})_{\times \varepsilon^{-1}}.$$

This scaling method is particularly useful in the case when P and Q are truncations of formal power series. In section 3, we will briefly explain why and we refer to [vdH02a, Section 6.2] for a more detailed analysis.

However, fast multiplication with scaling only works in the case when the coefficients of $P_{\times \lambda}$ have similar orders of magnitude for a suitable λ. Geometrically speaking, this means that the roots of P all lie in a "not so large" annulus around the circle of radius λ. In practice, this condition is not always satisfied, which leaves us with the question of designing a more general multiplication algorithm which is both asymptotically fast and numerically stable.

In its full generality, this problem is ill-posed. For instance, in cases when cancellation occurs, it is impossible to design an algorithm which computes all coefficients $(P\ Q)_k$ with a relative error bounded by $2^{o(n)+o(d)-n}$. A simple example of this situation is the computation of $(PQ)_1 = 0$ for $P = z + 1$, $Q = z - 1$. Nevertheless, such cancellations can only occur "below" the "numerical Newton polygon" of PQ: when using the naive $O(d^2\, \mathsf{M}(n))$ multiplication algorithm, based on the convolution sums

$$(PQ)_k = \sum_{i+j=k} P_i Q_j,$$

the coefficients $(PQ)_k$ which correspond to vertices of this Newton polygon are computed accurately, with a relative error bounded by $2^{\log d - n}$. Numerical Newton polygons will be defined and discussed in section 4. They are the natural numeric counterpart of Newton polygons for polynomials with power series coefficients, based on the analogy between floating point numbers and (Laurent) series, which goes back to Newton himself.

A reasonable problem is therefore to compute PQ with a similar kind of accuracy as the naive $O(d^2\, \mathsf{M}(n))$ multiplication algorithm does, but faster. This also leads to the question of formalizing what can be meant by "similar kind of accuracy". In section 5, we define the notion of "relative Newton error" ρ_R^{new} for the computation of $R \approx PQ$. We regard this notion as the natural generalization of the notion of "relative error", when numbers are replaced by numerical polynomials. In section 9, we will state a few interesting properties of this concept, which provide further evidence for its usefulness.

The central part of the paper is devoted to the description of an asymptotically fast multiplication algorithm which has a good numerical stability in terms of the relative Newton error. The main ideas behind the algorithm are the following:

1. The product PQ is modeled by a rectangle $0...p\times 0...q=\{0,...,p-1\}\times\{0,...q-1\}$ in which each pair (i,j) corresponds to the computation of $P_i Q_j$.
2. Only a subregion $\mathcal{S}\subseteq 0...p\times 0...q$, which can be determined precisely (section 5), contributes substantially to the product PQ. The complement can be neglected.
3. The region $\mathcal{S}$ can be covered by a suitable disjoint union of rectangles $i...i'\times j...j'$ (see sections 6 and 7).
4. The contribution of each such rectangle $i...i'\times j...j'$ to $P\,Q$ can be computed efficiently using a fast multiplication algorithm with scaling (section 8).

The main result of the paper is the following:

THEOREM 1. *There exists an algorithm to compute a floating point approximation* $R\approx PQ$ *with relative Newton error* $\rho_R^{\text{new}}\leqslant 2^{-n}$ *in time* $O(\mathsf{M}(n\,d))$*, provided that* $\log d=O(n)$.

2. Multiplication of pre-conditioned polynomials

Let us describe in more detail the second classical multiplication algorithm mentioned in the introduction, when the coefficients of P are of the same order of magnitude, and similarly for Q. We start with the determination of the maximum exponents

$$\begin{aligned} e_P &= \max\{e_{P_0},...,e_{P_{p-1}}\}\\ e_Q &= \max\{e_{Q_0},...,e_{Q_{q-1}}\}\end{aligned}$$

and the approximation of $P\,2^{n-e_P}$ and $Q\,2^{n-e_Q}$ by integer polynomials $\tilde{P}$ and $\tilde{Q}$ whose coefficients have bit-sizes bounded by n. At the second step, we use Kronecker's method for the multiplication of $\tilde{P}$ and $\tilde{Q}$. Taking $k\geqslant 2\,(n+2)+\log_2 d$, the product $\tilde{P}\tilde{Q}$ can be reconstructed [Kro82, Sch82] from the product of the integers $\tilde{P}(2^k)$ and $\tilde{Q}(2^k)$. Assuming that $\log_2 d=O(n)$, these integers have sizes $O(n\,p)$ and $O(n\,q)$, whence can be multiplied in time $O(\mathsf{M}(n\,d))$. We finally obtain an approximation R of the product $P\,Q$ by fitting the coefficients of $\tilde{P}\tilde{Q}\,2^{e_P+e_Q-2n}$ into the nearest available floating point numbers in $\mathbb{F}_{p,s}$.

Let us now analyze the errors induced by the above scheme, where we recall our assumption that no overflows or underflows occur during our computations. When computing $\tilde{P}$ and $\tilde{Q}$ by rounding to the nearest, we obtain the uniform absolute error bounds

$$\begin{aligned} \|\tilde{P}-P\,2^{n-e_P}\| &\leqslant \tfrac{1}{2}\\ \|\tilde{Q}-Q\,2^{n-e_Q}\| &\leqslant \tfrac{1}{2},\end{aligned}$$

where

$$\|P\|=\max\{|P_0|,...,|P_{p-1}|\}$$

denotes the sup-norm of P if we consider P as a vector of coefficients. Consequently,

$$\begin{aligned} \|\tilde{P}\tilde{Q}-PQ\,2^{2n-e_P-e_Q}\| &\leqslant \|\tilde{P}\tilde{Q}-P\tilde{Q}\,2^{n-e_P}\|+\|P\tilde{Q}\,2^{n-e_P}-PQ\,2^{2n-e_P-e_Q}\|\\ &\leqslant \frac{1}{2}\|\tilde{Q}\|+\frac{1}{2}\|P\|\,2^{n-e_P}\\ &\leqslant 2^n. \end{aligned} \tag{1}$$

On the other hand, we also have the bounds

$$\begin{aligned} \|PQ\| &\leqslant d\,\|P\|\,\|Q\| && (2)\\ \|PQ\| &\geqslant \tfrac{1}{2\,d}\,\|P\|\,\|Q\|. && (3) \end{aligned}$$

The second bound follows by considering the FFT transforms of P, Q and PQ with respect to a $2\,d$-th root of unity ω. Indeed, the sup-norms of these transforms satisfy

$$\begin{aligned} \|\mathrm{FFT}_\omega(P)\| &\leqslant 2\,d\,\|P\|\\ \|P\| = \frac{1}{2\,d}\,\|\mathrm{FFT}_{\omega^{-1}}(\mathrm{FFT}_\omega(P))\| &\leqslant \|\mathrm{FFT}_\omega(P)\|, \end{aligned}$$

whence

$$\|P\|\,\|Q\| \leqslant \|\mathrm{FFT}_\omega(P)\|\,\|\mathrm{FFT}_\omega(Q)\| = \|\mathrm{FFT}_\omega(PQ)\| \leqslant 2\,d\,\|PQ\|.$$

When $e_{PQ} \geqslant e_P + e_Q$, the combination of (1) and (2) yields the (rough) bound

$$\|R - PQ\| \leqslant 4\,d\,2^{-n}\,\|PQ\|.$$

In the case when $e_{PQ} \leqslant e_P + e_Q$, the combination of (1) and (3) leads to the same bound. In all cases, we thus obtain a *uniform relative error bound*

$$\rho_R = \frac{\|R - PQ\|}{\|PQ\|} \leqslant 2^{\log_2 d + 2 - n}. \tag{4}$$

Remark 2. We have chosen Kronecker's method for the multiplication of integer polynomials, because it is relatively straightforward to implement on top of a fast library for integer arithmetic such as Gmp [Gra91]. Internally, such integer libraries rewrite numbers as polynomials, say in u, and rely on FFT multiplication of these polynomials. At the price of improving Gmp, better performance can be achieved by using a bi-variate FFT on both u and z. In general, this yields only a constant speed-up, but in the somewhat exotic case when $\log_2 d \gg n$, this method still admits a complexity $O(\mathsf{M}(n\,d))$ instead of $O(\mathsf{M}(d \log d))$.

3. Multiplication with scaling

In the case when the polynomials P and Q are truncations of formal power series f and g, the asymptotic behaviour of the coefficients is usually of the form

$$\log_2 |P_k| = \alpha\,k + O(\varphi(k)), \tag{5}$$

for some suitable slowly increasing function $\varphi(k) = o(k)$. It is natural to rescale P by considering the polynomial $P_{\times\lambda}$

$$P_{\times\lambda}(z) = P(\lambda\,z),$$

with $\lambda = 2^{-\alpha}$, since its coefficients $P_{\times\lambda,k} = P_k\,2^{-\alpha k}$ are comparable in magnitude:

$$\log_2 |P_{\times\lambda,k}| = O(\varphi(k)).$$

If Q has a similar asymptotic behaviour

$$\log_2 |Q_k| = \alpha\,k + O(\varphi(k)), \tag{6}$$

we may then compute $R \approx PQ$ using the formula

$$R \approx (P_{\times\lambda}\,Q_{\times\lambda})_{\times\lambda^{-1}}$$

and the algorithm from the previous section for the multiplication of $P_{\times\lambda}$ and $Q_{\times\lambda}$.

Now assume that we also have an asymptotic formula

$$\log_2 |R_k| = \alpha\, k + O(\varphi(k)).$$

Then it can be shown that the relative error of each *individual* coefficient R_k is bounded by

$$\rho_{R_k} = \frac{|R_k - (PQ)_k|}{(PQ)_k} = \mathrm{e}^{O(\varphi(k)) + \log_2 d - n} \tag{7}$$

for all sufficiently large k. In other words, the multiplication method with scaling amounts to a loss of $O(\varphi(k)) + \log_2 d$ bits of precision. Under the assumption that $n \geqslant C\, \varphi(d) + 2 \log_2 d$ for a sufficiently large constant C, we are thus left with at least $n/2$ correct leading bits in the mantissa of each coefficient. The asymptotic behaviour of $\varphi(k)$ is closely related to the dominant singularities of f and g and for many interesting applications in combinatorics or physics, one has $\varphi(k) = O(\log k)$. For more details, and the application of this technique to the relaxed computation of formal power series, we refer to [vdH02a, Section 6.2].

It sometimes happens that only "limsup versions"

$$\begin{aligned} \max_{l \leqslant k} \log_2 |P_l| &= \alpha\, k + O(\varphi(k)) \\ \max_{l \leqslant k} \log_2 |Q_l| &= \alpha\, k + O(\varphi(k)) \end{aligned}$$

of (5) and (6) are verified. In that case, the individual relative error estimates (7) cease to hold, but we still have the scaled uniform bound

$$\rho_{R_{\times\lambda}} = \frac{\|R_{\times\lambda} - (PQ)_{\times\lambda}\|}{\|(PQ)_{\times\lambda}\|} \leqslant 2^{\log_2 d + 2 - n}, \tag{8}$$

which is a direct consequence of the corresponding bound (4) in the previous section.

It also frequently occurs that g does not have the same radius of convergence as f, in which case we rather have an asymptotic form

$$\log_2 |Q_k| = \beta\, k + O(\psi(k)).$$

Without loss of generality, we may assume that $\alpha > \beta$. When considering the convolution product

$$(PQ)_k = \sum_{i=0}^{\min(k, p-1)} P_i\, Q_{k-i}, \tag{9}$$

we observe that the summands $P_i\, Q_{k-i}$ approximately form a geometric progression. In first approximation, only the last $w = n/(\alpha - \beta)$ terms therefore contribute to the leading n bits of the result (see figure 1). The computation of $P\, Q$ can thus be reduced to the computation of the products

$$P Q_{0\ldots w}, P_{p-w\ldots p}\, Q_{w\ldots 2w}, P_{p-w\ldots p}\, Q_{2w\ldots 3w}, \ldots, P_{p-w\ldots p}\, Q_{\lfloor q/w \rfloor w\ldots q},$$

using the notation

$$\begin{aligned} i\ldots j &= \{i, \ldots, j-1\} \\ P_{i\ldots j} &= P_i\, z^i + \cdots + P_j\, z^{j-1}. \end{aligned}$$

When computing each of these products using the scaling $\times \lambda$ and a doubled precision $2\, d$, the individual coefficients R_k of $R \approx PQ$ are again obtained with a relative precision of the form $O(2^{o(n) + \log d - n})$. The total cost of this algorithm is $O(\mathsf{M}(w)\, d/w)$, which reduces to $O(\mathsf{M}(n)\, d)$ for large d and fixed n. A generalization of this idea will be analyzed in more detail in section 7.

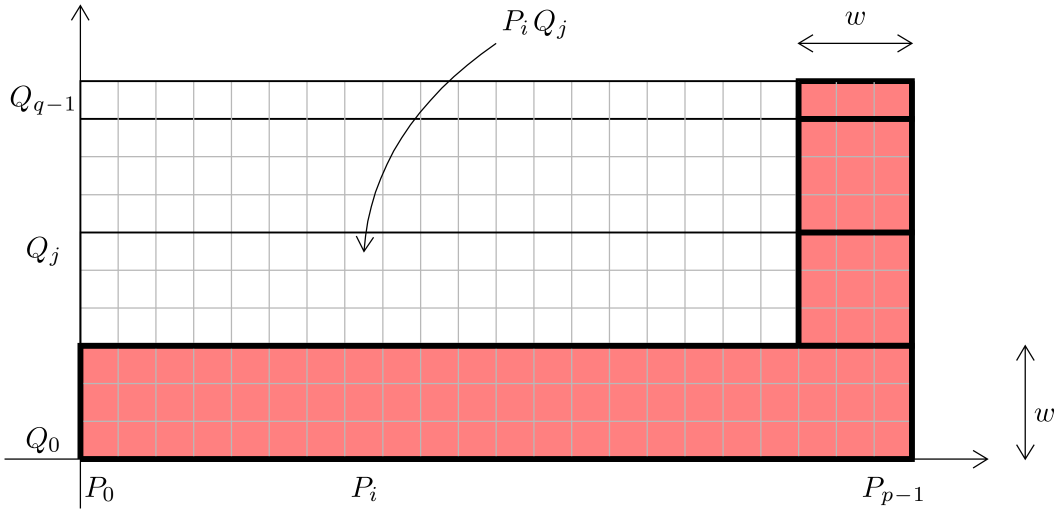

Figure 1. Illustration of the computation of PQ for two polynomials P and Q whose coefficients satisfy $\log_2|P_k| = \alpha k + o(k)$ and $\log_2|Q_k| \approx \beta k + o(k)$ with $\alpha > \beta$. Each square corresponds to the contribution of the product $P_i Q_j$ to $(PQ)_{i+j}$. Setting $w = n/(\alpha - \beta)$, only the terms $P_i Q_j$ in the shaded area contribute to the n leading bits of $(PQ)_{i+j}$, in first approximation.

4. Numeric Newton polygons

Roughly speaking, the algorithms from sections 2 and 3 are numerically stable when the roots of P and Q lie in an annulus around the unit circle resp. the circle of radius λ. When this is no longer the case, one may still try to apply the algorithm from section 3 for partial products $P_{i\ldots i'} Q_{j\ldots j'}$, using scales λ which vary as a function of $i\ldots i'$ and $j\ldots j'$.

A central concept for doing this is the *numeric Newton polygon* N_P of P, which is the convex hull of the half-lines $(i, \log_2|P_i| - \mathbb{R}^{\geqslant})$ for all $0 \leqslant i < p$, with the convention that $\log_2|0| = -\infty$ (see figure 2 for two examples). The numeric Newton polygon N_P is characterized by its *exponent polynomial*

$$E_P = E_{N_P} = E_{P,0} + \cdots + E_{P,p-1} z^{p-1}$$

with coefficients

$$E_{P,i} = \max\{y: (i, y) \in N_P\}.$$

Equivalently, we may characterize N_P by the first exponent $E_{P,0}$ and the slopes $\sigma_{P,i} = E_{P,i} - E_{P,i-1}$ with $0 < i < p$. Notice that $\sigma_{P,1} \geqslant \cdots \geqslant \sigma_{P,p-1}$.

It is easy to compute the vertices (i_0, y_0), ..., (i_j, y_j), $i_0 < \cdots < i_j$ of the Newton polynomial of $P_{0\ldots k}$ by induction over k: at each step, we add $(k, \log_2|P_k|)$ at the right of the sequence and repeatedly remove the before last pair (i_{j-1}, y_{j-1}) as long as it lies below the edge from (i_{j-2}, y_{j-2}) to (i_j, y_j). Using this method, the vertices of N_P, as well as the numbers $E_{P,i}$ and $\sigma_{P,i}$, can be computed in linear time.

The scaling operation $P \mapsto P_{\times\lambda}$ has a simple impact on exponent polynomials:

$$E_{P_{\times\lambda}}(z) = E_{P,\boxplus\log_2\lambda}(z) = \sum_{0 \leqslant i < p} (E_{P,i} + i \log_2 \lambda) z^i.$$

The norms of P and $P_{\times\lambda}$ and their exponent polynomials are related by

$$\log_2\|P\| = \|E_P\| \quad (10)$$
$$\log_2\|P_{\times\lambda}\| = \|E_{P,\boxplus\log_2\lambda}\|, \quad (11)$$

since the maximum y-value of a convex hull of a set of pairs $(i,\ y)$ coincides with the maximum y-values of the points themselves.

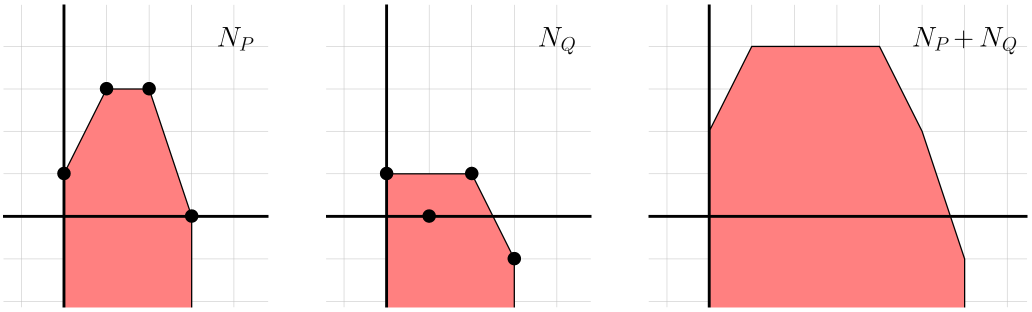

Figure 2. Illustrations of the numeric Newton polygons of $P = 2 + 8\,z - 8\,z^2 + 1$ and $Q = 2 - z - 2\,z^2 + \frac{1}{2}$ and the Minkowski sum of these polygons.

Recall that the *Minkowski sum* of the convex sets N_P and N_Q is defined by

$$N_P + N_Q = \{(i + i', j + j') : (i, j) \in N_P, (i', j') \in N_Q\}.$$

An example of a Minkowski sum is shown in figure 2. It is convenient to regard E_P and E_Q as "max-plus polynomials" over the semi-algebra $\mathbb{R} \cup \{-\infty\}$ in which multiplication and addition are replaced by addition and the operator max. These polynomials are convex in the sense that its sequences of coefficients are convex. The exponent polynomial of $N_P + N_Q$ may then be reinterpreted as a "product" of polynomials:

$$\begin{aligned} E_{N_P + N_Q} &= E_P \boxplus E_Q \\ (E_P \boxplus E_Q)_k &= \max_{i+j=k} E_{P,i} + E_{Q,j}. \end{aligned}$$

Clearly, $(E_P \boxplus E_Q)_0 = E_{P,0} + E_{Q,0}$, and it is easily shown that the slopes of $N_P + N_Q$ are obtained by merging the ordered lists $\sigma_{P,1} \geqslant \cdots \geqslant \sigma_{P,p-1}$ and $\sigma_{Q,1} \geqslant \cdots \geqslant \sigma_{Q,q-1}$. This provides us with a linear-time algorithm for the computation of $E_P \boxplus E_Q$ as a function of E_P and E_Q. The crucial property of $N_P + N_Q$ is that it approximates N_{PQ} well.

Proposition 3. *We have* $\|E_{PQ} - E_P \boxplus E_Q\| \leqslant \log_2 d + 1$.

Proof. Let $\lambda \in \mathbb{R}^{>}$. Applying (2) and (3), we have

$$|\log_2 \|(PQ)_{\times\lambda}\| - \log_2 \|P_{\times\lambda}\| - \log_2 \|Q_{\times\lambda}\|| \leqslant \log_2 d + 1.$$

Using (11) and the fact that $\|E_{P \boxplus Q}\| = \|E_P\| + \|E_Q\|$, we obtain

$$|\|E_{PQ,\boxplus \log_2 \lambda}\| - \|(E_P \boxplus E_Q)_{\boxplus \log_2 \lambda}\|| \leqslant \log_2 d + 1.$$

Given arbitrary convex max-plus polynomials E and F with $\deg E = \deg F$, it therefore suffices to show that

$$\|E - F\| \leqslant m = \max_{\sigma \in \mathbb{R}} |\|E_{\boxplus\sigma}\| - \|F_{\boxplus\sigma}\||.$$

Indeed, for $0 \leqslant i \leqslant \deg E$, let σ be such that $\|E_{\boxplus\sigma}\| = E_i + i\,\sigma$. Then

$$F_i \leqslant \|F_{\boxplus\sigma}\| - i\,\sigma \leqslant \|E_{\boxplus\sigma}\| + m - i\,\sigma = E_i + m.$$

The relation $E_i \leqslant F_i + m$ is proved similarly. □

5. RELATIVE NEWTON ERROR AND TRUNCATION

It is classical that the numeric properties of a polynomial P can be greatly effected by slightly modifying one of its coefficients. For instance, if $P = 2^{-n} z - 1$, then its root can move from 2^n to $+\infty$, when modifying its coefficient P_1 by an amount as small as $2^{-n} \|P\|$. Nevertheless, as we will detail in section 9, one may expect that many of the numeric properties of P do not sensibly change if we modify a coefficient P_i by less than $2^{E_{P,i}-n}$.

If the coefficients P_i of P are known with absolute errors ε_{P_i}, this motivates us to introduce the *relative Newton error* of P by

$$\rho_P^{\text{new}} = \max \left\{ \frac{\varepsilon_{P_0}}{2^{E_{P,0}}}, \ldots, \frac{\varepsilon_{P_{p-1}}}{2^{E_{P,p-1}}} \right\}.$$

For instance, if R is an approximation of the product PQ, then its relative Newton error would typically be given by

$$\rho_R^{\text{new}} = \max \left\{ \frac{|R_0 - (PQ)_0|}{2^{E_{(PQ)_0}}}, \ldots, \frac{|R_{p+q-1} - (PQ)_{p+q-1}|}{2^{E_{(PQ)_{p+q-1}}}} \right\}.$$

Alternatively, we may regard ρ_R^{new} as the minimal uniform error of the approximations $R_{\times\lambda} \approx P_{\times\lambda} Q_{\times\lambda}$ for all possible scales $\lambda \in \mathbb{R}^>$:

$$\rho_R^{\text{new}} = \min_{\lambda \in \mathbb{R}^>} \rho_{R_{\times\lambda}}.$$

In this section and the subsequent sections, we propose an efficient algorithm for computing an approximation $R \approx PQ$, which is numerically stable in the sense that ρ_R^{new} is small.

The first main idea behind Newton multiplication is that only part of the products $P_i Q_j$ substantially contribute to PQ: the pair (i, j) and the corresponding product $P_i Q_j$ are said to be *negligible* if

$$E_{P,i} + E_{Q,j} \leqslant (E_P \boxplus E_Q)_{i+j} - n.$$

Indeed, when neglecting the sum of all such products in the computation of $P Q$, the corresponding increase of the relative Newton error is bounded by $2^{2\log_2 d + 1 - n}$. This follows from proposition 3 and the fact that there are at most d pairs (i, j) which contribute to a fixed coefficient $(PQ)_k$. The left-hand picture in figure 3 illustrates a typical zone of non-negligible contributions $P_i Q_j$. The lower and upper limits $j_{\text{lo}}(i)$ and $j_{\text{up}}(i)$ of this zone are increasing functions in i and can therefore be determined in linear time as a function of E_P and E_Q.

A set $\mathcal{P}$ of pairs (i, j) is said to be negligible if each of its elements is negligible. A partial product $P_{i\ldots i'} Q_{j\ldots j'}$ is said to be *negligible* if $i\ldots i' \times j\ldots j'$ is negligible and *admissible* if there exists no proper subrectangle $\tilde{\imath}\ldots\tilde{\imath}' \times \tilde{\jmath}\ldots\tilde{\jmath}'$ of $i\ldots i' \times j\ldots j'$ such that $i\ldots i' \times j\ldots j' \setminus \tilde{\imath}\ldots\tilde{\imath}' \times \tilde{\jmath}\ldots\tilde{\jmath}'$ is negligible. Clearly, any non-negligible partial product $P_{i\ldots i'} Q_{j\ldots j'}$ gives rise to a unique admissible partial product $P_{\tilde{\imath}\ldots\tilde{\imath}'} Q_{\tilde{\jmath}\ldots\tilde{\jmath}'}$ by taking $\tilde{\imath}\ldots\tilde{\imath}' \times \tilde{\jmath}\ldots\tilde{\jmath}'$ to be the smallest subrectangle whose complement in $i\ldots i' \times j\ldots j'$ is negligible. Any subdivision

$$0\ldots p \times 0\ldots q = \coprod_r i_r\ldots i_r' \times j_r\ldots j_r' \tag{12}$$

therefore gives rise to a new disjoint union

$$\coprod_r \tilde{\imath}_r\ldots\tilde{\imath}_r' \times \tilde{\jmath}_r\ldots\tilde{\jmath}_r' \tag{13}$$

of admissible subrectangles, whose complement in $0\ldots p \times 0\ldots q$ is negligible. The second disjoint union will be called the *admissible part* of the first one. The right-hand side of figure 3 illustrates the admissible part of a so-called product subdivision (see section 7 below).

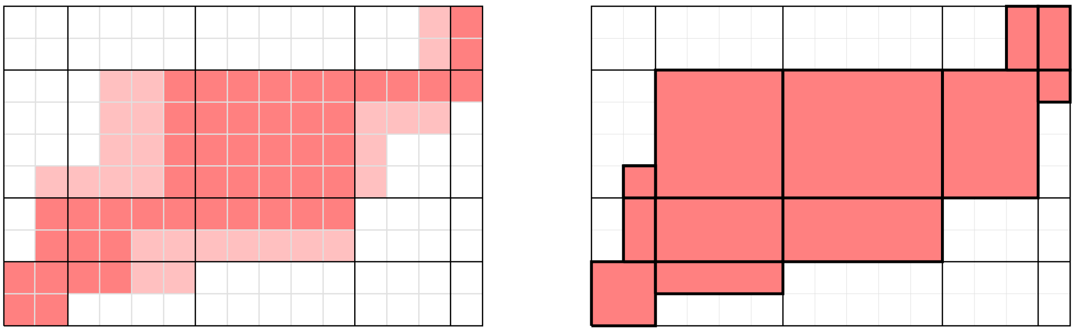

Figure 3. On the left hand side we displayed the non-negligible contributions $P_i Q_j$ to a typical product PQ. The dark squares correspond to products $P_i Q_j$ for which $E_{P_i} + E_{Q_j} = (E_P \boxplus E_Q)_{i+j}$ and the light squares to products where we only have $E_{P_i} + E_{Q_j} > (E_P \boxplus E_Q)_{i+j} - n$. At the right hand side, we have shown the admissible part of a product subdivision of $0\ldots p \times 0\ldots q$ formed by the thin black rectangles.

6. Fitting the Newton polygon using fewer edges

Our next task is to find a suitable subdivision (12) with admissible part (13), such that $\sum_r P_{\tilde{\imath}_r\ldots\tilde{\imath}_r'} Q_{\tilde{\jmath}_r\ldots\tilde{\jmath}_r'}$ can be computed both efficiently and in a numerically stable way. The second main idea behind Newton multiplication is to search for a subdivision such that the products $P_{\tilde{\imath}_r\ldots\tilde{\imath}_r'} Q_{\tilde{\jmath}_r\ldots\tilde{\jmath}_r'}$ can be performed using suitable scalings *and* such that the rectangles $\tilde{\imath}_r\ldots\tilde{\imath}_r' \times \tilde{\jmath}_r\ldots\tilde{\jmath}_r'$ are yet reasonably large. As a first step in this construction, we will approximate the Newton polygons of P and Q by simpler ones, merging those edges whose slopes differ only by a small amount.

Given a slope $\sigma \in \mathbb{R}$ and a convex max-plus polynomial E, we define the σ-discrepancy $h_{E,\sigma}$ of E by

$$h_{E,\sigma} = \|E_{\boxplus\sigma}\| + \|-E_{\boxplus\sigma}\|.$$

Then the individual exponents $E_{\boxplus\sigma,i}$ of $E_{\boxplus\sigma}$ lie in a strip of height $h_{E,\sigma}$:

$$\|E_{\boxplus\sigma}\| - h_{E,\sigma} \leqslant E_{\boxplus\sigma,i} \leqslant \|E_{\boxplus\sigma}\|. \tag{14}$$

In the particular case when σ equals the main slope of E,

$$\sigma_E = \frac{E_{\deg E} - E_0}{\deg E},$$

we call h_{E,σ_E} the *discrepancy* of E and also denote it by h_E. For any $\sigma \in \mathbb{R}$ and any second convex max-plus polynomial F, the convexity of E and F implies

$$h_{E,\sigma} \;\geqslant\; h_E \tag{15}$$

$$h_{E+F} \;\geqslant\; \max(h_E, h_F) \tag{16}$$

More generally, one defines the σ-discrepancy $h_{E,\sigma,i\ldots i'}$ of E on a range $i\ldots i' \subseteq 0\ldots(\deg E + 1)$ by $h_{E,\sigma,i\ldots i'} = h_{z^{-i}E_{i\ldots i'},\sigma}$ and similarly $h_{E,i\ldots i'} = h_{z^{-i}E_{i\ldots i'}}$. We have

$$h_{E,i\ldots i''} \;\geqslant\; h_{E,i\ldots i'} + h_{E,i'\ldots i''} \tag{17}$$

for any subdivision of the range $i...i''$ into $i...i'$ and $i'...i''$.

Let us now return to the approximation of the Newton polygon E_P by a new one with fewer edges. Let $\kappa > 0$ be an arbitrary but fixed constant, such as $\kappa = 1$, and denote $\delta = \lceil \kappa\, n \rceil$. We now construct the sequence $0 = i_0 < \cdots < i_k = p$ by taking each i_r maximal such that $h_{E_P, i_r...i_{r+1}} \leqslant \delta$ (see figure 4 below). Using the convexity of E_P, it is not hard to check that this construction can be done in linear time using a single traversal of $0...p$. The edges of E_P between i_r and $i_{r+1} - 1$ are well-approximated by a single edge from (i_r, E_{P,i_r}) to $(i_{r+1} - 1, E_{P,i_{r+1}-1})$. A similar sequence $0 = j_0 < \cdots < j_l = q$ is constructed for Q.

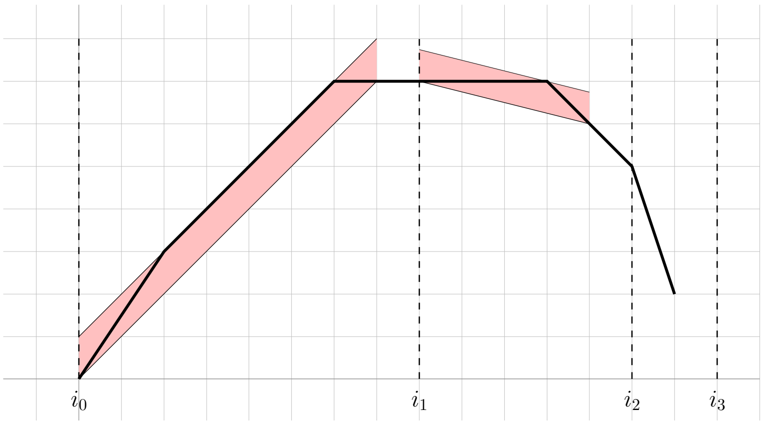

Figure 4. Illustration of the sequence $0 = i_0 < \cdots < i_k = p$ for a given Newton polygon and a maximal discrepancy $\delta = 1$. The border of the Newton polygon corresponds to the thick black line. The parts corresponding to the segments $i_0...i_1$ and $i_1...i_2$ fit into the indicated strips of heights 1 and $\frac{3}{4}$. The last part, which corresponds to the segment $i_2...i_3$ is a straight segment which fits into a strip of height 0.

7. The product and Newton subdivisions

A first candidate for the subdivision (12) is the *product subdivision*:

$$0...p \times 0...q = \coprod_{r,s} i_r...i_{r+1} \times j_s...j_{s+1}, \tag{18}$$

where the i_r and j_s are as in the previous section. Figure 3 shows an example of a product subdivision.

LEMMA 4. *Given k, there exist at most $2\,(1/\kappa + 2)$ rectangles in the product subdivision, for which the admissible part contains a pair (i, j) with $i + j = k$.*

PROOF. Assume the contrary and let $i_r...i_{r+1} \times j_s...j_{s+1}$ and $i_{r'-1}...i_{r'} \times j_{s'-1}...j_{s'}$ be the top left and bottom right rectangles in the product subdivision, for which the admissible parts contribute to $(PQ)_k$. We have $r' - r > 1/\kappa + 2$ or $s' - s > 1/\kappa + 2$. By symmetry, we may assume $r' - r > 1/\kappa + 2$. Now consider the convex max-plus polynomials

$$\begin{aligned} E &= E_{P,\quad i_{r+1}} + \cdots + E_{P,\quad i_{r'-1}-1}\, z^{i_{r'-1}-i_{r+1}-1} \\ F &= E_{Q,k-i_{r+1}} + \cdots + E_{Q,k-i_{r'-1}+1}\, z^{i_{r'-1}-i_{r+1}-1} \end{aligned}$$

Using (17), we have $h_E = h_{E_P, i_{r+1}...i_{r'-1}} > (1/\kappa)\, \delta \geqslant n$. From (15) and (16), we thus get

$$h_{E+F,0} \geqslant h_{E+F} > n.$$

Now $\|E+F\|$ coincides with $(E_P \boxplus E_Q)_k$, whence there exists an index $i \in i_{r+1} \ldots i_{r'-1}$ with

$$E_{P,i} + E_{Q,k-i} = (E+F)_i \leqslant \|E+F\| - h_{E+F,0} < (E_P \boxplus E_Q)_k - n,$$

which means that the pair $(i, k-i)$ is negligible. But this is only possible if either $(i_{r+1}, k - i_{r+1})$ or $(i_{r'-1} - 1, k - i_{r'-1} + 1)$ is also negligible. This, in its turn, implies that either the rectangle $i_r \ldots i_{r+1} \times j_s \ldots j_{s+1}$ or the rectangle $i_{r'-1} \ldots i_{r'} \times j_{s'-1} \ldots j_{s'}$ is negligible, in contradiction with our hypothesis. □

COROLLARY 5. *There are at most $O(d)$ non-negligible rectangles in the product subdivision, and they can be computed in linear time.*

PROOF. The first assertion follows from the facts that each diagonal $\mathcal{D}_k = \{(i,j): i+j=k\}$ with $k \in 0 \ldots p + 0 \ldots q$ intersects at most $2\,(1/\kappa + 2)$ rectangles and that there are less than $2\,d$ such diagonals. The rectangles can be computed in time $O(d)$, by following the lower and upper limits j_{lo} and j_{up} of the zone of non-negligible pairs. □

The subdivision we are really after refines (18). More precisely, consider a non-negligible rectangle $i \ldots i' \times j \ldots j'$ of the form $i_r \ldots i_{r+1} \times j_s \ldots j_{s+1}$. Let

$$\sigma = \sigma_{P,i\ldots i'} = \frac{E_{P,i'-1} - E_{P,i}}{i' - 1 - i}$$

be the slope of P on $i \ldots i'$. Swapping P and Q if necessary, we may assume without loss of generality that $\sigma \geqslant \sigma_{P,j\ldots j'}$. We now construct the sequence $j = \hat{\jmath}_0 < \cdots < \hat{\jmath}_{\hat{s}} = j'$, by choosing each $\hat{\jmath}_{\hat{s}}$ minimal such that $h_{E_Q,\sigma} \leqslant \delta$, and subdivide

$$i \ldots i' \times j \ldots j' = \coprod_{\hat{s}} i \ldots i' \times \hat{\jmath}_{\hat{s}} \ldots \hat{\jmath}_{\hat{s}+1}. \tag{19}$$

We may re-interpret figure 1 as an illustration of the non-negligible pairs inside the rectangle $i \ldots i' \times j \ldots j'$, together with the subdivision (19) and its corresponding admissible part. Performing the replacement (19) for all non-negligible rectangles $i_r \ldots i_{r+1} \times j_s \ldots j_{s+1}$ in (18), while leaving the negligible rectangles untouched, we obtain a finer subdivision of $0 \ldots p \times 0 \ldots q$, which we call the *Newton subdivision*.

LEMMA 6. *Given k, there exist at most $2\,(1/\kappa + 3)^2$ rectangles in the Newton subdivision, for which the admissible part contains a pair (i,j) with $i + j = k$.*

PROOF. Consider a non-negligible rectangle $i \ldots i' \times j \ldots j' = i_r \ldots i_{r+1} \times j_s \ldots j_{s+1}$ in the product subdivision. In view of lemma (4), it suffices to show that there exists at most $1/\kappa + 3$ rectangles in the admissible part of its subdivision (19) which contribute to $(PQ)_k$.

Now assume that $\hat{\jmath} \in \hat{\jmath}_{\hat{s}} \ldots \hat{\jmath}_{\hat{s}+1}$ and $\hat{\jmath}' \in \hat{\jmath}_{\hat{s}'} \ldots \hat{\jmath}_{\hat{s}'+1}$ satisfy $\hat{s}' - \hat{s} \geqslant 1/\kappa + 3$ and assume that $\hat{\imath} < \hat{\imath}' \in i \ldots i'$ satisfy $\hat{\imath} + \hat{\jmath} = \hat{\imath}' + \hat{\jmath}' = k$. Setting $\lambda = 2^{\sigma}$, we then have

$$\begin{array}{rcl} E_{P_{\times \lambda},\hat{\imath}} & \leqslant & E_{P_{\times \lambda},\hat{\imath}'} + \delta \\ E_{Q_{\times \lambda},\hat{\jmath}} & \leqslant & E_{P_{\times \lambda},\hat{\jmath}'} - (1/\kappa + 1)\,\delta \leqslant E_{P_{\times \lambda},\hat{\jmath}'} - n - \delta, \end{array}$$

whence

$$E_{P,\hat{\imath}} + E_{Q,\hat{\jmath}} \leqslant E_{P,\hat{\imath}'} + E_{Q,\hat{\jmath}'} - n \leqslant (E_P \boxplus E_Q)_k - n.$$

This shows that the pair $(\hat{\imath}, \hat{\jmath})$ is negligible. We conclude that the diagonal $i + j = k$ intersects at most $1/\kappa + 3$ admissible rectangles. □

Corollary 7. *There are at most $O(d)$ non-negligible rectangles in the Newton subdivision, and they can be computed in linear time.*

8. Newton multiplication

We can now describe our algorithm for computing the product $R \approx P\,Q$. We start by the computation of the admissible part of the Newton subdivision. For each rectangle $i...i' \times j...j'$ in this admissible part, we compute its contribution to $P\,Q$ as follows. By construction, we have $h_{E_P,\sigma} \leqslant \delta$ or $h_{E_Q,\sigma} \leqslant \delta$ for $\sigma = \sigma_{P,i...i'}$ or $\sigma = \sigma_{Q,j...j'}$. Setting $\lambda = 2^{\sigma}$, we compute the contribution $R_{i...i' \times j...j'}$ of $P_{i...i'}\,Q_{j...j'}$ to $P\,Q$ using the multiplication method with scaling from section 3,

$$R_{i...i' \times j...j'} \approx (P_{i...i', \times\lambda}\,Q_{j...j', \times\lambda})_{\times\lambda^{-1}}, \tag{20}$$

using a precision of $n + 2\,\delta$ instead of n bits. The final result $R \approx P\,Q$ is obtained by summing all partial products $R_{i...i' \times j...j'}$ using straightforward floating point arithmetic. This algorithm leads to the following refinement of theorem 1.

Theorem 8. *There exists an algorithm of time complexity $O(\mathsf{M}(n\ d))$, which computes $R \approx PQ$ with relative Newton error*

$$\rho_R^{\text{new}} \leqslant 2^{3 \log d + O(1) - n}. \tag{21}$$

Proof. We claim that the computation of the product (20) using a precision of $n + 2\,\delta$ bits contributes to the relative Newton error of the global product PQ by an amount which is bounded by $2^{3 \log d + O(1) - n}$. This clearly implies (21), since for a fixed coefficient $(PQ)_k$ in the product, at most $O(1)$ partial products of the form $R_{i...i' \times j...j'}$ contribute to $(PQ)_k$.

Setting $\hat{P} = P_{i...i'}$, $\hat{Q} = Q_{j...j'}$ and $\hat{R} = R_{i...i' \times j...j'}$, the bound (8) yields

$$\frac{\|\hat{R}_{\times\lambda} - (\hat{P}\hat{Q})_{\times\lambda}\|}{\|(\hat{P}\hat{Q})_{\times\lambda}\|} \leqslant 2^{\log_2 d + O(1) - n - 2\delta}.$$

For any $k = i'' + j'' \in i...i' + j...j'$, the bounds (2), (14) and proposition 3 also imply

$$\begin{aligned}
\log_2 \|(\hat{P}\hat{Q})_{\times\lambda}\| - \log_2 d &\leqslant \log_2 \|\hat{P}_{\times\lambda}\| + \log_2 \|\hat{Q}_{\times\lambda}\| \\
&\leqslant \|E_{P_{\times\lambda}, i...i'}\| + \|E_{Q_{\times\lambda}, j...j'}\| \\
&\leqslant E_{P_{\times\lambda}, i''} + h_{E_P, \log_2 \lambda, i...i'} + E_{Q_{\times\lambda}, j''} + h_{E_Q, \log_2 \lambda, j...j'} \\
&\leqslant E_{P_{\times\lambda} \boxplus Q_{\times\lambda}, k} + 2\,\delta \\
&\leqslant E_{(PQ)_{\times\lambda}, k} + 2\,\delta + \log_2 d + 1.
\end{aligned}$$

Combining both estimates, we obtain

$$\begin{aligned}
\frac{|\hat{R}_k - (\hat{P}\hat{Q})_k|}{2^{E_{(PQ)_k}}} &= \frac{|\hat{R}_{\times\lambda, k} - (\hat{P}\hat{Q})_{\times\lambda, k}|}{2^{E_{(PQ)_{\times\lambda, k}}}} \\
&\leqslant \frac{\|\hat{R}_{\times\lambda} - (\hat{P}\hat{Q})_{\times\lambda}\|}{2^{E_{(PQ)_{\times\lambda, k}}}} \\
&\leqslant 2^{\log_2 d + O(1) - n - 2\delta} \frac{\|(\hat{P}\hat{Q})_{\times\lambda}\|}{2^{E_{(PQ)_{\times\lambda, k}}}} \\
&\leqslant 2^{3 \log_2 d + O(1) - n}.
\end{aligned}$$

This completes the proof of our claim.

The proof of the complexity bound relies on lemma 6. Let

$$\coprod_r \tilde{\imath}_r \ldots \tilde{\imath}_r' \times \tilde{\jmath}_r \ldots \tilde{\jmath}_r'$$

be the admissible part of the Newton subdivision and let $K = 2\,(1/\kappa + 3)^2$. Consider the total length

$$L = \sum_r \tilde{\imath}_r' - \tilde{\imath}_r + \tilde{\jmath}_r' - \tilde{\jmath}_r$$

of the left borders and the lower borders of all rectangles $\tilde{\imath}_r \ldots \tilde{\imath}_r' \times \tilde{\jmath}_r \ldots \tilde{\jmath}_r'$. Since every diagonal hits the union of these borders in at most $2\,K$ points, we have $L \leqslant 4\,d\,K$. The total cost of all multiplications $P_{\tilde{\imath}_r \ldots \tilde{\imath}_r'} Q_{\tilde{\jmath}_r \ldots \tilde{\jmath}_r'}$ is therefore bounded by

$$O\left(\sum_r \mathsf{M}((\tilde{\imath}_r' - \tilde{\imath}_r + \tilde{\jmath}_r' - \tilde{\jmath}_r)\,n) \right) \leqslant O(\mathsf{M}(L\,n)) \leqslant O(\mathsf{M}(d\,n)).$$

This completes the proof of our theorem. □

Remark 9. Theorem 1 follows from theorem 8, when applying it for the increased precision $n + 3 \log d + O(1) = O(n)$.

9. Properties of the relative Newton error

Now that we have an efficient multiplication which is numerically stable in terms of the relative Newton error, it is interesting to investigate this concept a bit closer. First of all, its name is justified by the property

$$\rho_{PQ}^{\text{new}} \leqslant 2\,d\,(\rho_P^{\text{new}} + \rho_Q^{\text{new}} + O(\rho_P^{\text{new}}\,\rho_Q^{\text{new}})),$$

which follows from proposition 3. Some natural numeric quantities associated to a numeric polynomial P are its coefficients, its roots and its evaluations at points. For our notion of relative Newton error to be satisfactory, a small relative Newton error should therefore imply small relative errors for the coefficients, the roots and evaluations at points.

Now the example $(z+1)(z-1)$ from the introduction shows that it is impossible to obtain good relative errors for all coefficients. Nevertheless, given a polynomial P with a small relative Newton error ρ_P^{new}, we can at least guarantee small relative errors $\rho_{P_i} \leqslant \rho_P^{\text{new}}$ for those coefficients P_i which correspond to vertices $(i, E_{P,i})$ of the Newton polygon. For most practical purposes, we expect this to be sufficient. However, theoretically speaking, the naive multiplication algorithm of polynomials sometimes provides better relative errors for other coefficients. This is for instance the case when squaring a polynomial of the form $P(z) = A(z^2) + B(z^2)\,z$, where B is very small compared to A.

For what follows, it will be convenient to redefine $d := \deg P$. Let $U = \{u_1, \ldots, u_d\} \in \mathbb{C}$ denote the set of roots of P, ordered by increasing magnitudes $|u_1| \leqslant \cdots \leqslant |u_d|$. By analogy with the power series setting, the norms $|u_i|$ roughly correspond to the slopes $\sigma_{P,i}$ of the Newton polygon. We expect the existence of a bound

$$|\log_2 |u_i| + \sigma_{P,i}| \leqslant O(\log_2 d),$$

although we have not yet proved this. The worst case arises when P admits a single root of multiplicity d.

We define the *relative distance* $\delta_{z,z'}$ between $z, z' \in \mathbb{C}$ (not both zero) by

$$\delta_{v,w} = \frac{|z - z'|}{|z| + |z'|} \leqslant 1.$$

and the relative distance $\delta_{z,U}$ between z and U, counted with multiplicities, by

$$\delta_{z,U} = \delta_{z,u_1} \cdots \delta_{z,u_d}.$$

Clearly, the relative error of the evaluation of P at z increases when z approaches U. The following proposition shows that the relative error of $P(z)$ can be kept small as long as $\delta_{z,U}$ is sufficiently large.

Proposition 10. *The relative error of the evaluation of P at $z \in \mathbb{C}$ is bounded by*

$$\rho_{P(z)} \leqslant \frac{d\,\rho_P^{\text{new}}}{\delta_{z,U}}.$$

Proof. Modulo a suitable scaling and multiplication of P by a constant, we may assume without loss of generality that $|z| = 1$ and $\|P\| = 1$. We claim that $|P(z)| \geqslant \delta_{z,U}$. Indeed,

$$P(z) = P_d\,(z - u_1) \cdots (z - u_d),$$

so, considering FFT transforms with respect to a d-th root of unity ω, we get

$$1 = \|P\| \leqslant \|\mathrm{FFT}_\omega(P)\| \leqslant |P_d|\,(1 + |u_1|) \cdots (1 + |u_d|) = \frac{|P(z)|}{\delta_{z,U}}.$$

Since $\|P\| = 1$, its coefficients are known with absolute errors $\varepsilon_{P_i} \leqslant \rho_P^{\text{new}}$. We can therefore evaluate $P(z)$ with absolute error $\varepsilon_{P(z)} \leqslant d\,\rho_P^{\text{new}}$ and relative error $\rho_{P(z)} \leqslant \varepsilon_{P(z)} / |P(z)|$. □

Let us finally consider the computation of the roots u_1, ..., u_d in the case when P is known with relative Newton error ρ_P^{new}. The worst case again arises when $P = P_d\,(z - u)^d$ has a single root of multiplicity d. Indeed, a relative error ρ inside P_0 gives rise to a relative error $\sqrt[d]{\rho}$ for u, so that we cannot expect anything better than $\rho_{u_i} \leqslant \sqrt[d]{\rho_P^{\text{new}}}$ for each i. Nevertheless, given an isolated root u_i and $Q = P/(z - u_i)$, we have $P'(u_i) = Q(u_i)$ and

$$\varepsilon_{P(u_i)} \approx |Q(u_i)|\,\varepsilon_{u_i}.$$

Using a similar argument as in the proof of proposition 10, it follows that

$$\rho_{u_i} \leqslant \frac{d\,\rho_P^{\text{new}}}{\delta_{u_i, U \setminus \{u_i\}}}.$$

Indeed, after reduction to the case when $|u_i| = 1$ and $\|P\| = 1$, we combine the facts that $\varepsilon_{P(u_i)} \leqslant d\,\rho_P^{\text{new}}$, $|Q(u_i)| \geqslant \delta_{u_i, U \setminus \{u_i\}}$ and $\rho_{u_i} = \varepsilon_{u_i}$.

10. Conclusion

In this paper, we have presented a fast multiplication algorithm for polynomials over $\mathbb{F}_{n,s}$, which is numerically stable in a wide variety of cases. In order to capture the increased stability in a precise theorem, we have shown the usefulness of numeric Newton polygons and the relative Newton error. Even though our algorithm was presented in the case of real coefficients, it is straightforward to generalize it to the complex case.

Multiplication being the most fundamental operation on polynomials, it can be hoped that fast and numerically stable algorithms for other operations (division, g.c.d., multipoint evaluation, etc.) can be developed along similar lines. This might for instance have applications to polynomial root finding [Pan96]. Indeed, an operation such as the Graeffe transform can be done both efficiently and accurately using our algorithm.

The multiplication algorithm with scaling from [vdH02a, Section 6.2] and section 3 has been implemented inside the Mathemagix system [vdH02b]. We have experienced it to be very efficient for the expansion of power series. For instance, the accurate computation of 100000 terms of $\mathrm{e}^{z/(1-z)}$ with a 256-bit precision typically takes about one minute. One reason behind this efficiency stems from the reduction to Kronecker multiplication. Indeed, multiple precision libraries for floating point numbers, such as Mpfr [HLRZ00], involve a lot of overhead for every single operation. We rather reduce the whole problem to one big integer multiplication.

Only a preliminary version of Newton multiplication has been implemented so far inside Mathemagix. Although this implementation uses a simpler, but not always efficient subdivision, it already produces encouraging results. Work is in progress to improve the implementation and reduce its combinatorial overhead. It remains worth investigating whether there exist other, simpler and/or more efficient subdivision schemes.

Bibliography

[CT65] J.W. Cooley and J.W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Computat.*, 19:297–301, 1965.

[Gra91] T. Granlund et al. GMP, the GNU multiple precision arithmetic library. `http://www.swox.com/gmp`, 1991.

[HLRZ00] G. Hanrot, V. Lefèvre, K. Ryde, and P. Zimmermann. MPFR, a C library for multiple-precision floating-point computations with exact rounding. `http://www.mpfr.org`, 2000.

[Kro82] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen. *Jour. für die reine und ang. Math.*, 92:1–122, 1882.

[Pan96] Victor Y. Pan. On approximating complex polynomial zeros: Modified quadtree (Weyl's) construction and improved Newton's iteration. Technical Report RR-2894, INRIA Sophia, 1996.

[Sch82] A. Schönhage. Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients. In J. Calmet, editor, *EUROCAM '82: European Computer Algebra Conference*, volume 144 of *Lect. Notes Comp. Sci.*, pages 3–15, Marseille, France, April 1982. Springer.

[SS71] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971.

[vdH02a] J. van der Hoeven. Relax, but don't be too lazy. *JSC*, 34:479–542, 2002.

[vdH02b] J. van der Hoeven et al. Mathemagix, 2002. `http://www.mathemagix.org`.