A FAST AND STABLE ALGORITHM FOR SPLITTING POLYNOMIALS

GREGORIO MALAJOVICH AND JORGE P. ZUBELLI

ABSTRACT. A new algorithm for splitting polynomials is presented. This algorithm requires $O(d \log \epsilon^{-1})^{1+\delta}$ floating point operations, with $O(\log \epsilon^{-1})^{1+\delta}$ bits of precision. As far as complexity is concerned, this is the fastest algorithm known by the authors for that problem.

An important application of the method is factorizing polynomials or polynomial root-finding.

1. Introduction

The main motivation of this paper is to develop fast algorithms for solving univariate polynomials. Although this is a rather old subject, important progress has taken place in the last years (See the review by Pan [1]).

The problem of solving univariate polynomials may be reduced to factorization. Once a factorization $f = gh$ is known, factors g and h may be factorized recursively, until one obtains degree 1 or 2 factors.

Fast algorithms for factorizing a degree d polynomial may proceed by performing a change of coordinates and a small number (say $O(\log \log d)$) iterations of Graeffe's transformation. Graeffe's transformation (also

Date: January 22, 1996.

Key words and phrases. Polynomial equations, Factorization, Splitting. Sponsored by CNPq (Brasil).

Typeset using $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX.

studied by Dandelin and by Lobatchevski) was developed and extensively used before the advent of digital computers. See Ostrowski [2] and Uspensky [3, p.318-331]. For more recent applications, see Schönhage [4], Kirrinis [5] and Pan [6, 7].

Graeffe's transformation replaces a polynomial $f(x)$ by the polynomial $Gf(x) = (-1)^d f(\sqrt{x}) f(-\sqrt{x})$. The roots of the new polynomial are the square of the roots of the old polynomial. The effect of Graeffe's iteration is to "pack" the roots into clusters near zero and infinity, leaving a wide enough root-free annulus.

The next (and crucial) step is to factorize this new polynomial into factors having all the roots close to zero (resp. infinity). This is called *splitting.*

Finally, it is necessary to return from Graeffe's iteration. Explicitly, if $Gf(x) = Gg(x)Gh(x)$, one may set $g(x) = \gcd(Gg(x^2), f(x))$ and $h(x) = f(x)/g(x)$.

Known splitting algorithms require $O(d^{1+\delta})$ arithmetic operations, with $O((d(d - \log \epsilon))^{1+\delta})$ bits of precision. They produce approximate factors g and h such that $\|f - gh\|_2 < \epsilon$. We will prove instead:

Main Theorem . *Let $R > 64(a+b)^3$, $a, b \in \mathbb{N}$. Let f be a polynomial of degree $d = a + b$, such that a roots are contained inside the disk $|\zeta| < R^{-1}$ and b roots are contained outside the disk $|\zeta| > R$.*

Then, for $\epsilon = o(1/d)$, an ϵ-approximation of the factors g and h of f may be computed within

$$O(d \log \frac{1}{\epsilon} \log \log \frac{1}{\epsilon})$$

floating-point arithmetic operations, performed with:

$$O(\log \frac{1}{\epsilon})$$

bits of precision.

The algorithm we propose in this paper requires a more moderate precision than classical ones. Therefore, its bit complexity is significantly lower. Moreover, we define the ϵ-approximation in order to guarantee the stronger result

$$\begin{array}{rcl} \sqrt{\sum \left(2^{a-i}|g_i - {g^*}_i|\right)^2} & \leq & \epsilon \\ \sqrt{\sum \left(2^{i}|h_i - {h^*}_i|\right)^2} & \leq & \epsilon \end{array}$$

where $f = g^* h^*$ is the exact factorization. The price to pay is a larger splitting radius (meaning $O(\log \log d)$ extra Graeffe's iterations).

It is assumed that f is given as a vector of floating point numbers. The output is also given as a vector of floating point numbers, and we assume as a model of computation *correctly rounded floating-point arithmetic.*

2. Sketch of the proof

2.1. **General Idea.** We will show that factoring a polynomial f is equivalent to solving a certain system of polynomial equations. Under the hypotheses of the Main Theorem, this system will be solvable by Newton iteration.

It turns out that, with a proper choice of the initial guess, one has quadratic convergence since the first iteration.

2.2. **Background of α-theory.** Let $\varphi : \mathbb{C}^n \rightarrow \mathbb{C}^n$ be a system of polynomials. The Newton operator N_φ is defined by

$$\begin{array}{rrcl} N_\varphi : & \mathbb{C}^n & \rightarrow & \mathbb{C}^n \\ & x & \mapsto & x - D\varphi(x)^{-1}\varphi(x) \end{array} .$$

The following *invariants* were introduced by Smale in [8]:

$$\begin{aligned}
\beta(\varphi, x) &= \left\| D\varphi(x)^{-1}\varphi(x) \right\|_2 \\
\gamma(\varphi, x) &= \max_{k \geq 2} \left(\frac{\left\| D\varphi(x)^{-1} D^k\varphi(x) \right\|_2}{k!} \right)^{\frac{1}{k-1}} \\
\alpha(\varphi, x) &= \beta(\varphi, x)\gamma(\varphi, x)
\end{aligned}$$

It was proven in [8] that if $\alpha(\varphi, x) < \alpha_0 < \frac{1}{7}$, then the sequence (x_i) converges *quadratically* to a root of φ. See also [9, 10, 11, 12, 13]. This theorem can be generalized to an *approximation* of the Newton operator. We shall need that generalization in the sequel.

The space $\mathcal{H}_d$ is the space of all systems of homogeneous polynomials of degree $d = (d_1, \ldots, d_n)$. In order to work with non-homogeneous polynomials, we may set one coordinate of the variable $z = (z_0, \ldots, z_n)$ to be equal to 1 (Say $z_0 = 1$). Then α^{aff}, β^{aff} and γ^{aff} below are precisely α, β and γ defined above.

Under this notation, we have:

Theorem 1 (Malajovich [14], Theorem 2)**.** *Let $f \in \mathcal{H}_d$, $z^{(0)} \in \mathbb{C}^{n+1}$, the first coordinate of $z^{(0)}$ be non-zero, and $\delta \geq 0$ be such that: $(\beta^{\text{aff}}(f, z^{(0)}) + \delta)\gamma^{\text{aff}}(f, z^{(0)}) < 1/16$, and $\gamma^{\text{aff}}(f, z^{(0)})\delta < 1/384$. Suppose that the sequence $(z^{(i)})$, where the first coordinates of $z^{(i)}$ and $z^{(0)}$ are equal, satisfies*

$$\frac{\left\| z^{(i+1)} - N^{\text{aff}}(f, z^{(i)}) \right\|_2}{\left\| z^{(i)} \right\|_2} \leq \delta \ .$$

Then, there is a zero ζ of f such that

$$d_{\text{proj}}(z^{(i)}, \zeta) \leq \max\left(2^{-2^i - 1}, 6\delta\right) \ .$$

Notice that above we have $\left\|z^{(i)}\right\|_2^2 = |z_0^{(i)}|^2 + \cdots + |z_n^{(i)}|^2 = 1 + |z_1^{(i)}|^2 + \cdots + |z_n^{(i)}|^2$.

2.3. **Polynomial systems associated to splitting.** Let a and b be fixed. We want to *split* a polynomial f of degree $d = a + b$ into factors g and h of degree a and b, respectively. This means that we want to factor $f = gh$, so that the roots of g are inside the disk $D(R^{-1})$ and the roots of h are outside the disk $D(R)$.

For convenience, we choose $f_a = 1$. Polynomials with this property shall be called *hemimonic.*

We want to solve the system

$$\varphi_f(g,h) \overset{\text{def}}{=} gh - f = 0$$

where g is monic of degree a. In vector notation,

$$\varphi_f(g,h) = \begin{pmatrix} g_0h_0 & - & f_0 \\ g_1h_0 + g_0h_1 & - & f_1 \\ & & \\ \vdots & & \\ & & \\ h_{b-1} + g_{a-1}h_b & - & f_{d-1} \\ h_b & - & f_d \end{pmatrix} .$$

The system $\varphi_f(g,h)$ is a system of $d+1 = a+b+1$ non-homogeneous polynomial equations in $d+1$ variables.

In order to simplify the exposition, we shall assume $a \geq b$. The case $b < a$ is similar, *mutatis mutandis.*

The derivative of φ_f is given by

$$D\varphi_f(g,h) = \begin{pmatrix}
h_0 & & & & & g_0 & & & \\
h_1 & h_0 & & & & g_1 & g_0 & & \\
\vdots & & & & & & \ddots & \ddots & \\
h_b & & \ddots & \ddots & & \vdots & & & g_0 \\
 & & & & & & & & g_1 \\
 & & & & h_0 & g_{a-1} & & & \\
 & & & \ddots & h_1 & 1 & & & \vdots \\
 & & & & \vdots & & \ddots & \ddots & \\
 & & & & h_b & & & & g_{a-1} \\
 & & & & 0 & & & & 1
\end{pmatrix} .$$

The second derivative $D^2\varphi_f(g,h)$ is a bilinear operator from $\mathbb{C}^{d+1} \times \mathbb{C}^{d+1}$ into $\mathbb{C}^{d+1}$. The i-th coordinate of $D^2\varphi_f(g,h)$ can be represented by its Hessian matrix, with ones concentrated along one anti-diagonal. For instance, if $a = b+1$ and $i = b-1$,

$$\left(D^2\varphi_f(g,h)\right)_i = \begin{bmatrix}
 & & & & & & 1 & 0 \\
 & & & & & \iddots & & \\
 & & & & 1 & & & \\
 & & & & 0 & & & \\
 & & 1 & 0 & & & & \\
 & \iddots & & & & & & \\
1 & & & & & & & \\
0 & & & & & & &
\end{bmatrix} .$$

2.4. **A good metric.** In this section we shall introduce a few norms that will play an important role in the sequel. In some sense, those seem to be the *good* norms to use for the splitting problem. Those will be the norms referred to in the definition of α, β, γ and in Theorem 1. They correspond to a *scaling* of the system φ_f .

Before doing that we start with some intuitive motivation for such norms. First, let's consider the case of a polynomial with roots close to 0, say for example

$$p(x) = x^n + t_{n-1}x^{n-1} + t_0 \ ,$$

with sufficiently small t_{n-1} and t_0. Then, changes in t_0 will affect much more the roots than changes in t_{n-1}. This lead us to consider a norm that for $i > j$ emphasizes the contribution of the coefficient of x^j more than that of x^i. A similar reasoning holds for polynomials whose roots are close to ∞ except that more emphasis has to be put on the higher degree coefficients.

The problem we have at hand is that of splitting a polynomial f of degree $a + b$ into two factors g and h of degree a and b, respectively, so that the roots of g are close to 0 and the roots of h are close to ∞. Therefore, it is natural to consider a norm that captures the relative weight of the different coefficients as far as such coefficients affect the roots. Compare with definition (69.2) in Ostrowski [2], page 209.

To make the above heuristics more precise we introduce the following concepts:

The polynomial

$$h(x) = \sum_{n=0}^{b} h_n x^n$$

will be called *antimonic* if $h_0 = 1$.

The polynomial

$$f(x) = \sum_{n=0}^{a+b} f_n x^n$$

will be called *hemimonic* (w.r.t. the splitting into factors of degrees a and b) if

$$f_a = 1 \ .$$

Definition 1. For g a degree a polynomial we set the monic norm

$$\|g\|_{\mathfrak{m}} = \sqrt{\sum_{0 \le i \le a} \left(2^{a-i} |g_i|\right)^2} \ .$$

For h a degree b polynomial, the *antimonic* norm is defined by

$$\|h\|_{\mathfrak{a}} = \sqrt{\sum_{0 \le i \le b} \left(2^{i} |h_i|\right)^2} \ .$$

For φ a polynomial of degree $a+b$, we set the *hemimonic norm* with respect to the splitting (a, b) as

$$\|\varphi\|_{\mathfrak{h}} = \sqrt{\sum_{0 \le i \le a+b} \left(2^{|a-i|} |\varphi_i|\right)^2} \ .$$

Notice that if g is a monic polynomial with $\|g - x^a\|_{\mathfrak{m}}$ sufficiently small, then all its roots are close to 0. Similarly, if h is antimonic with $\left\|h - x^b\right\|_{\mathfrak{a}}$ small, then all its roots are next to ∞. As to the concept of (a, b)-hemimonic, the point is that if φ belongs to this class, then $\|\varphi - x^a\|_{\mathfrak{h}}$ small implies that a roots of φ are close to 0 and b roots are close to ∞.

Although the preceding remarks are true for all norms, the definitions above give sharper estimates than the usual 2-norm.

In order to make the distinction of the norm under consideration more apparent we will try to use the letters $\left\{ \begin{array}{c} g \\ h \\ \varphi \end{array} \right\}$ to denote $\left\{ \begin{array}{c} \text{monic} \\ \text{antimonic} \\ \text{hemimonic} \end{array} \right\}$ polynomials of degree $\left\{ \begin{array}{c} a \\ b \\ a+b \end{array} \right\}$, respectively.

We may estimate the norm of the operator

$$D^2\varphi_f(g,h) : (\mathbb{C}^{a+b+1}, \|\cdot\|_{\mathfrak{ma}}) \times (\mathbb{C}^{a+b+1}, \|\cdot\|_{\mathfrak{ma}}) \rightarrow (\mathbb{C}^{a+b+1}, \|\cdot\|_{\mathfrak{h}}) \ , \tag{1}$$

where the norm

$$\|(g,h)\|_{\mathfrak{ma}} \stackrel{\text{def}}{=} \sqrt{{\|g\|_{\mathfrak{m}}}^2 + {\|h\|_{\mathfrak{a}}}^2} \ ,$$

by the following

Lemma 1.

$$\left\|D^2\varphi_f(g,h)\right\|_{\mathfrak{ma} \to \mathfrak{h}} \leq \frac{\sqrt{d+1}}{4} \ .$$

The proof of this Lemma is postponed to Section 8.

We consider the Newton operator applied to the system $\varphi_f(g,h)$

$$N(f;g,h) = \begin{pmatrix} g \\ h \end{pmatrix} - D\varphi_f(g,h)^{-1}\varphi_f(g,h) \ .$$

Lemma 1 provides the following estimate for Smale's invariants:

$$\begin{aligned}
\gamma(\varphi_f;g,h) &= \frac{\|D\varphi_f(g,h)^{-1}D^2\varphi_f(g,h)\|_{\mathfrak{ma} \to \mathfrak{h}}}{2} \leq \left\|D\varphi_f(g,h)^{-1}\right\|_{\mathfrak{ma} \to \mathfrak{h}} \frac{\sqrt{d+1}}{8} \\
\beta(\varphi_f;g,h) &\leq \left\|D\varphi_f(g,h)^{-1}\right\|_{\mathfrak{ma} \to \mathfrak{h}} \|\varphi_f(g,h)\|_{\mathfrak{h}} \\
\alpha(\varphi_f;g,h) &\leq \frac{\sqrt{d+1}}{8} {\left\|D\varphi_f(g,h)^{-1}\right\|_{\mathfrak{ma} \to \mathfrak{h}}}^2 \ \|\varphi_f(g,h)\|_{\mathfrak{h}}
\end{aligned}$$

2.5. **A good starting point.** In this section, we shall prove that a good starting point for the Newton iteration is

$$g(x) = x^a \ ,$$

$$h(x) = 1 \ .$$

This choice makes the matrix $D\varphi_f(g,h)$ equal to the identity. This implies

Lemma 2. *Assume that $g = x^d$ and $h = 1$. Then,*

$$\left\| D\varphi_f(x^a, 1)^{-1} \right\|_{\mathfrak{ma} \to \mathfrak{h}} = 1 \ .$$

As we are using our non-standard norms, the proof will be postponed to Section 8.

We have:

$$\begin{aligned}
\alpha(\varphi_f; g, h) &\leq \frac{\sqrt{a+b+1}}{8} \|\varphi_f(g,h)\|_{\mathfrak{h}} \\
&= \frac{\sqrt{a+b+1}}{8} \|f - x^a\|_{\mathfrak{h}} \\
\beta(\varphi_f; g, h) &\leq \|f - x^a\|_{\mathfrak{h}} \\
\gamma(\varphi_f; g, h) &\leq \frac{\sqrt{a+b+1}}{4} \ .
\end{aligned}$$

Lemma 3. *Let's assume that f and $\hat{\epsilon}$ satisfy*

$$\hat{\epsilon} < \frac{1}{16\sqrt{a+b+1}}$$

and

$$\|f - x^a\|_{\mathfrak{h}} < \frac{4}{17\sqrt{a+b+1}} \ .$$

Then, if the sequences $g^{(k)}$ and $h^{(k)}$ satisfy

$$(g^{(0)}, h^{(0)}) = (x^d, 1) \ ,$$

with

$$\frac{\left\| (g^{(i+1)}, h^{(i+1)}) - N(\varphi_f; g^{(i)}, h^{(i)}) \right\|_{\mathfrak{ma}}}{\|(g^{(i)}, h^{(i)})\|_{\mathfrak{ma}}} \leq \frac{\hat{\epsilon}}{6}$$

there exist $\tilde{g}$ and $\tilde{h}$ such that $f = \tilde{g}\tilde{h}$, and for

$$k > \log_2 \log_2 \frac{1}{\hat{\epsilon}} \ ,$$

we have

$$\min_{\lambda \neq 0} \frac{\left\| (g^{(k)}, h^{(k)}) - \lambda(\tilde{g}, \tilde{h}) \right\|_{\mathfrak{ma}}}{\|(g^{(k)}, h^{(k)})\|_{\mathfrak{ma}}} \leq \hat{\epsilon} \ .$$

Proof of Lemma 3: We are under the hypotheses of the Theorem 1, where we set $\delta = \widehat{\epsilon}/6$. Indeed,

$$\begin{aligned}
(\beta(\varphi_f, (g^{(0)}, h^{(0)})) + \delta)\gamma &\leq (\|f - x^a\|_{\mathfrak{h}} + \delta)\frac{\sqrt{a+b+1}}{4} \\
&\leq (\frac{4}{17\sqrt{a+b+1}} + \frac{\widehat{\epsilon}}{6})\frac{\sqrt{a+b+1}}{4} \\
&\leq \frac{1}{17} + \frac{\widehat{\epsilon}\sqrt{a+b+1}}{24} \\
&\leq \frac{1}{16} \ .
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\gamma(f; (g^{(0)}, h^{(0)}))\delta &< \frac{\sqrt{a+b+1}}{4}\frac{1}{6\ 16\sqrt{a+b+1}} \\
&= \frac{1}{384} \ .
\end{aligned}$$

Therefore, if $k > \log_2 \log_2 (1/\widehat{\epsilon})$, the final $(g^{(k)}, h^{(k)})$ are within (projective) distance $\widehat{\epsilon}$ from the true factors $(\tilde{g}, \tilde{h})$.

□

Moreover, it can be seen that the usual distance between $(g^{(k)}, h^{(k)})$ and $(\tilde{g}, \tilde{h})$ is bounded by $4\widehat{\epsilon}$. Indeed, $\left\|(g^{(k)}, h^{(k)})\right\|_{\mathfrak{ma}} \geq |g_a| \geq 1$. Therefore,

$$\left\|(g^{(k)}, h^{(k)}) - \lambda(\tilde{g}, \tilde{h})\right\|_{\mathfrak{ma}} \leq \widehat{\epsilon} \ .$$

Since $g^{(k)}$ and $\tilde{g}$ are monic, we should have

$$1 - \widehat{\epsilon} \leq \lambda \leq 1 + \widehat{\epsilon} \ .$$

So,

$$\left\|(g^{(k)}, h^{(k)}) - (\tilde{g}, \tilde{h})\right\|_{\mathfrak{ma}} \leq \widehat{\epsilon}(1 + \left\|(\tilde{g}, \tilde{h})\right\|_{\mathfrak{ma}}) \ .$$

We will show later the following bound:

Lemma 4. *Let $R > 4(a+b)^2$. Let g be monic of degree a, with all roots in $D(R^{-1})$. Let h be of degree b, with all roots outside $D(R)$. Let $\varphi = gh$ be hemimonic.*

Then $\|(g,h)\|_{\mathfrak{ma}} \leq 3$.

Hence, assuming the conditions of Lemmas 3 and 4, we obtain

$$\left\|(g^{(k)}, h^{(k)}) - (\tilde{g}, \tilde{h})\right\|_{\mathfrak{ma}} \leq 4\hat{\epsilon} \ .$$

2.6. **End of the proof.** In order to prove the Main Theorem, we have to show that hemimonic polynomials with a large enough splitting annulus have small hemimonic norm. More precisely,

Theorem 2. *Let φ be an (a,b)-hemimonic polynomial, with a roots in the disk $D(R^{-1})$, and b roots outside the disk $D(R)$. If $R > 2\max(a,b)$ then:*

$$\|\varphi - x^a\|_{\mathfrak{h},\infty} < \frac{4}{3} \ \frac{2\max(a,b)}{R} \ \frac{1}{1 - \left(\frac{2\max(a,b)}{R}\right)^2}$$

In particular, let $R > 64(a+b)^3$. Then,

$$\|\varphi - x^a\|_{\mathfrak{h},\infty} < \frac{3}{64(a+b)^2} \ .$$

The proof will be given in section 3.

Theorem 2 shows that the conditions of the Main Theorem imply those of Lemma 3, and it remains to produce the sequence $(g^{(i)}, h^{(i)})$.

Theorem 3. *Assume the hypotheses of the Main Theorem. There is an algorithm to compute $(g^{(i+1)}, h^{(i+1)})$ of Lemma 3 out of f, $g^{(i)}$ $h^{(i)}$, within $O(d \log \hat{\epsilon}^{-1})$ floating point operations, performed with precision $O(\log \hat{\epsilon}^{-1})$, where $\hat{\epsilon} < o(1/d)$.*

Therefore, the ϵ-approximation of $\tilde{g}$ and $\tilde{h}$, $\epsilon = 4\hat{\epsilon}$, may be computed in a total of

$$O(d \log \frac{1}{\epsilon})$$

floating point operations, with precision

$$O(\log \frac{1}{\epsilon})$$

□

3. Splitting and hemimonicity

We will first prove a version of Theorem 2 for monic (resp. antimonic) polynomials:

Lemma 5. *Let g be monic, of degree a, with roots inside the disk $D(R^{-1})$, where $R > 2a$. Then,*

$$\|g - x^a\|_{\mathfrak{m},\infty} < \frac{2a}{R} \ .$$

Clearly, this lemma implies the analogous result for antimonic polynomials.

Proof of Lemma 5: Let g be monic, with all roots ζ_i inside $D(R^{-1})$. Now,

$$g_{a-i} = (-1)^i \sum_{j_1 < \cdots < j_i} \prod_{j_k} \zeta_{j_k} \ .$$

So,

$$2^i |g_{a-i}| \leq 2^i \left(\begin{array}{c} a \\ i \end{array} \right) R^{-i} \ .$$

Let $L_i = 2^i \begin{pmatrix} a \\ i \end{pmatrix} R^{-i}$. Then $L_0 = 1$, and

$$\frac{L_{i+1}}{L_i} = 2 \frac{\begin{pmatrix} a \\ i+1 \end{pmatrix}}{\begin{pmatrix} a \\ i \end{pmatrix}} R^{-1} = 2 \frac{a-i}{i+1} R^{-1} \ .$$

Therefore $L_1 = \frac{2a}{R}$. Since in general $\frac{a-i}{i+1} < a$ and $\frac{2a}{R} < 1$, one gets

$$2^i |g_{a-i}| \leq \frac{2a}{R}$$

□

We may prove now Theorem 2. Assume that g is monic of degree a, with roots inside $D(R^{-1})$. Assume that h is antimonic of degree b, with roots outside $D(R)$. Let $\varphi = gh$. We want to bound

$$2^{|a-i|} |\varphi_i| \ .$$

If $i \neq a$, we may distinguish two cases. For instance, assume $i > a$. (The case $i < a$ is analogous). We have

$$\varphi_i = g_a h_{i-a} + \sum_{\substack{0 \leq k < a \\ 0 < i-k \leq b}} g_k h_{i-k} \ .$$

Hence,

$$\begin{aligned}
2^{|a-i|} |\varphi_i| &\leq 2^{i-a} |h_{i-a}| + 2^{i-a} \sum_{\substack{0 \leq k < a \\ 0 < i-k \leq b}} |g_k| |h_{i-k}| \\
&\leq 2^{i-a-i+a} \|h-1\|_{\mathfrak{a},\infty} + \|g - x^a\|_{\mathfrak{m},\infty} \|h-1\|_{\mathfrak{a},\infty} \sum_{\substack{0 \leq k < a \\ 0 < i-k \leq b}} 2^{i-a-a+k-i+k} \\
&< \|h-1\|_{\mathfrak{a},\infty} + \frac{1}{3} \|g - x^a\|_{\mathfrak{m},\infty} \|h-1\|_{\mathfrak{a},\infty} \\
&< \frac{4}{3} \|h-1\|_{\mathfrak{a},\infty} \ .
\end{aligned}$$

On the other hand, we may write φ_a as

$$\varphi_a = g_a h_0 + \sum_{a-b \le k < a} g_k h_{a-k} \ .$$

Therefore,

$$|\varphi_a| \ge 1 - \left(\sum_{a-b \le k < a} 2^{-2a+2k} \right) \|g\|_{\mathfrak{m},\infty} \|h\|_{\mathfrak{a},\infty} \ge 1 - \|g - x^a\|_{\mathfrak{m},\infty} \|h - 1\|_{\mathfrak{a},\infty}$$

Hence,

$$\left\| \frac{\varphi}{\varphi_a} - x^a \right\|_{\mathfrak{h},\infty} \le \frac{4}{3} \, \frac{\max(\|g - x^a\|_{\mathfrak{m},\infty}, \|h-1\|_{\mathfrak{a},\infty})}{1 - \|g - x^a\|_{\mathfrak{m},\infty} \|h - 1\|_{\mathfrak{a},\infty}}$$

The bound of Lemma 5 may be inserted in the formula above

$$\left\| \frac{\varphi}{\varphi_a} - x^a \right\|_{\mathfrak{h},\infty} \le \frac{4}{3} \, \frac{2\max(a,b)}{R} \, \frac{1}{1 - \left(\frac{2\max(a,b)}{R} \right)^2}$$

□

We prove Lemma 4 now. Lemma 5 implies

$$\|g - x^a\|_{\mathfrak{m},\infty} \le \frac{2a}{R} \ .$$

Also,

$$\left\| \frac{h}{h_0} - 1 \right\|_{\mathfrak{a},\infty} \le \frac{2b}{R} \ .$$

In order to bound $\|h\|_{\mathfrak{a}}$ we first bound $|h_0|$. Since $\varphi = gh$ is hemimonic,

$$1 = g_a h_0 + \sum_{k \ge 1} g_{a-k} h_k \ .$$

Moreover,

$$|\sum_{k \ge 1} g_{a-k} h_k| \le \sum_{k \ge 1} 2^{-2k} \|g - x^a\|_{\mathfrak{m}} \|h - 1\|_{\mathfrak{a}} \le \frac{4ab|h_0|}{3R^2} \ .$$

It follows that

$$|h_0| < \frac{1}{1 - \frac{4ab}{3R^2}} < \frac{12}{11} \ .$$

Therefore,

$$\begin{aligned}
\|g,h\|_{\mathfrak{ma}} &\leq \frac{12}{11}\left(1+\frac{2a^2}{R}+1+\frac{2b^2}{R}\right)\\
&\leq \frac{12}{11}\left(2+\frac{2(a^2+b^2)}{R}\right)\\
&\leq \frac{12}{11}\,\frac{5}{2}=\frac{30}{11}\\
&\leq 3
\end{aligned}$$

□

4. Algorithm for solving the linear system

We will construct a first version of the algorithm of Theorem 3. The operation count will be $O(d^2)$. Error analysis will be dealt with in Section 5. A fast version will be constructed in Section 6. The proof of Theorem 3 finishes in Section 7.

Given a, b, g, h and φ, we will design an algorithm to solve

$$\begin{pmatrix}
h_0 & & & & & g_0 & & & \\
h_1 & h_0 & & & & g_1 & g_0 & & \\
\vdots & & & & & & \ddots & \ddots & \\
h_b & & \ddots & \ddots & & \vdots & & & g_0 \\
 & & & & & & & & g_1 \\
 & & & & h_0 & g_{a-1} & & & \\
 & & & \ddots & h_1 & 1 & & & \vdots \\
 & & & & \vdots & & \ddots & \ddots & \\
 & & & & h_b & & & & g_{a-1} \\
 & & & & 0 & & & & 1
\end{pmatrix}
\begin{pmatrix}
\\ \delta g \\ \\ \delta h \\ \\
\end{pmatrix}
=
\begin{pmatrix}
\\ \\ \varphi \\ \\ \\
\end{pmatrix}.$$

We assume at input that $g_a = h_b = 1$. We will write the algorithm recursively, but it can be made recursion-free by usual techniques. The main feature of this algorithm will be to preserve (in some sense) the monic, antimonic and hemimonic structure of the problem. This means that when $\|g\|_{\mathfrak{m}}$, $\|h\|_{\mathfrak{a}}$ and $\|\varphi\|_{\mathfrak{h}}$ are small, $\|\delta g\|_{\mathfrak{m}}$ and $\|\delta h\|_{\mathfrak{a}}$ should be small also.

It will be convenient to shift to the corresponding max norm. Those will be denoted by $\|.\|_{\mathfrak{m},\infty}$, $\|.\|_{\mathfrak{a},\infty}$ and $\|.\|_{\mathfrak{h},\infty}$.

Essentially, the algorithm is based on column operations, elimination of one variable, and a special pivoting operation. The algorithm below is written in a pseudo-code, and some lines are numbered for convenience.

The input data are a and b, positive integers; and polynomials g, h and φ, of degree respectively a, b and $a+b$. Those polynomials are represented by the vector of their coefficients. Index range between i and j is written $i:j$.

Therefore, the notation

$$g_{1:a} \leftarrow g_{1:a} - g_0 h_{1:b}$$

stands for $g_i \leftarrow g_i - g_0 h_i$, where $1 \le i \le a, b$.

The output of the algorithms are polynomials δg and δh, of degree a and b, respectively.

```
Algorithm Solve $(\delta g, \delta h) \leftarrow (a, b, g, h, \varphi)$
                If $a \geq b$
10                  If $a = 1$ and $b = 0$, Return $(0, \varphi_0)$ ;
20                  $g_{1:a} \leftarrow g_{1:a} - g_0 h_{1:b}$ ;
30                  $g' \leftarrow g_a$ ; $g_{1:a-1} \leftarrow \frac{g_{1:a-1}}{g'}$ ; $g_a \leftarrow 1$ ;
40                  $\delta g_0 \leftarrow \varphi_0$ ;
50                  $\varphi_{1:b} \leftarrow \varphi_{1:b} - \delta g_0 h_{1:b}$ ;
60                  $(\delta g_{1:a}, \delta h_{0:b}) \leftarrow$ Solve$(a-1, b, g_{1:a}, h_{0:b}, \varphi_{1:a+b})$ ;
70                  $\delta h_{0:b} \leftarrow \frac{\delta h_{0:b}}{g'}$ ;
```

```
80        $\delta g_{0:a} \leftarrow \delta g_{0:a} - g_0 \delta h_{0:b}$ ;
90        Return $(\delta g, \delta h)$ ;
      Else
100       $(\delta h_{b:0}, \delta g_{a:0}) \leftarrow$ Solve$(b, a, h_{b:0}, g_{a:0}, \varphi_{a+b:0})$ ;
110       Return$(\delta g, \delta h)$;
```

Lines 100 and 110 of the algorithm refer to the following pivoting procedure:

One replaces g and h by $x^b h(x^{-1})$ and $x^a g(x^{-1})$, respectively. φ is replaced by $x^{a+b}\varphi(x^{-1})$. The algorithm `Solve` is called again, this time ensuring that $a \geq b$. Then, the results are pivoted back, so that we obtain a solution of the original problem.

Line 10 deals with the trivial case $a = 1$, $b = 0$. This line is necessary to avoid infinite recursion.

Line 20 performs a column operation. Namely, we add g_0 times column 0 to b to columns a through $a + b + 1$ (recall $b \leq a$).

Line 30 ensures that g is still monic.

It is easy to find δg_0 (Line 40), and then to eliminate that variable of the problem (Line 50).

Then the algorithm is called recursively (Line 50), and the column operations of lines 20 and 30 are undone at lines 70 and 80.

This algorithm exploits the particular structure of the problem (g is almost x^a, h is almost 1). It has some remarkable stability properties. For instance, we can bound the norm of δg and δh in terms of the norm of the input. For clarity, we do that first without considering numerical error. Let's define

$$N = \max \left\{ \|g - x^a\|_{\mathfrak{m},\infty}, \|h - 1\|_{\mathfrak{a},\infty}, \|\varphi\|_{\mathfrak{h},\infty} \right\} \ .$$

At line 20 we have

$$|g_0| \leq \|g - x^a\|_{\mathfrak{m},\infty} 2^{-a} \leq N 2^{-a} \ .$$

Hence,

$$\|g_0 h_{1:b}\|_{\mathfrak{a},\infty} \leq \|g_0 (h-1)\|_{\mathfrak{a},\infty} \leq N^2 2^{-a} \ .$$

We estimate $\|g_0 h_{1:b}\|_{\mathfrak{m},\infty}$ as follows. Assume that

$$\|g_0 h_{1:b} - g_0 h_a x^a\|_{\mathfrak{m},\infty} = \max_{0 \leq i < a} |g_0 h_i| 2^{a-i}$$

was attained at i, $i \neq 0$:

$$\|g_0 h_{1:b} - g_0 h_a x^a\|_{\mathfrak{m},\infty} \leq N^2 2^{-2i} \leq N^2 2^{-2} \ .$$

Therefore,

$$\left\| g_{1:a}^{(20)} - {g_a}^{(20)} x^a \right\|_{\mathfrak{m},\infty} \leq N(1 + N 2^{-2}) \ .$$

The norm above is taken as if $g_{1:a}$ was a degree a polynomial, with lowest coefficient 0. In the sequel, $g_{1:a}$ will be treated as a degree $a-1$ polynomial. This does not increase its norm.

At line 30, we first perform the operation $g' \leftarrow g_a$. The value of g_a was possibly modified at line 20 (provided $a = b$) and

$$|g'^{(30)}| = |g_a^{20}| \geq 1 - |g_0||h_a| \geq 1 - N^2 2^{-2a} \ .$$

Hence,

$$|\frac{g_i}{g'}| \leq N \frac{1 + N 2^{-2}}{1 - N^2 2^{-2a}} 2^{i-a} \ .$$

Therefore,

$$\begin{aligned} \left\| g_{1:a}^{(30)} - x^{a-1} \right\|_{\mathfrak{m},\infty} &\leq N \frac{1 + N 2^{-2}}{1 - N^2 2^{-2a}} 2^{i-a+a-1-i+1} \leq N \frac{1 + N 2^{-2}}{1 - N^2 2^{-2a}} \leq \\ &\leq N \frac{1 + N 2^{-2}}{1 - (N 2^{-2})^2} = N \frac{1}{1 - 2^{-2} N} \ . \end{aligned}$$

At line 40, we have

$$|\delta g_0| = |\varphi_0| \leq \|\varphi\|_{\mathfrak{h},\infty} 2^{-a} \leq N 2^{-a} \ .$$

At line 50,

$$|h_i| \leq N 2^{-i} \ .$$

So $|\delta g_0 h_i| \leq 2^{-a-i} N^2$. On the other hand, $|\varphi_i| \leq 2^{-|a-i|} N$. Since we are choosing $i \leq b \leq a$, then $|a-i| = a-i$. Therefore,

$$|\varphi_i - \delta g_0 h_i| \leq 2^{i-a} N (1 + 2^{-2i} N) \ .$$

If $\varphi_{1:a+b}$ is considered as a polynomial of degree $(a-1)+b$, one gets

$$\left\| \varphi^{(50)}_{1:a+b} - \varphi^{(50)}_a x^{a-1} \right\|_{\mathfrak{h},\infty} \leq N(1 + 2^{-2} N) \ .$$

Line 60 performs a recursive call to the algorithm solve, where the norm N was replaced by $\frac{N}{1-N}$. Upon return, let's assume that δg and δh have norm bounded by some $N' > \frac{N}{1-N}$.

At line 70, one obtains

$$\left\| \delta h^{(70)}_{0:b} \right\|_{\mathfrak{a},\infty} \leq \| \delta h_{0:b} \|_{\mathfrak{a},\infty} \frac{1}{1 - 2^{-2a} N^2} \leq \frac{N'}{1 - 2^{-2a} N^2} \ .$$

Before execution of line 80

$$\| \delta g \|_{\mathfrak{m},\infty} \leq \max\{N', |\delta g_0| 2^a\} \leq \max\{N', N\} \leq N'.$$

So,

$$\| \delta h_{0:b} \|_{\mathfrak{m},\infty} \leq 2^a \| \delta h_{0:b} \|_{\mathfrak{a},\infty}$$
$$\| g_0 \delta h_{0:b} \|_{\mathfrak{m},\infty} \leq \frac{N N'}{1 - 2^{-a} N^2} \ .$$

Hence,

$$\delta g^{(80)}_{0:a} \leq N' \left(1 + \frac{N}{1 - 2^{-2a} N^2} \right) \leq \frac{N'}{1 - N'} \ .$$

In order to bound the values of N and N' throughout the execution of the recursive algorithm, one may define the recurrence

$$N_{i+1} = \frac{N_i}{1 - N_i}$$

where N_i bounds N at recurrence step $i \leq a+b$ and bounds the norm N' at step $2a+2b-i$, for $i > a+b$.

The general term of this recurrence is, for $k < \frac{1}{N_0}$,

$$N_k = \frac{1}{\frac{1}{N_0} - k} \ .$$

If one fixes, for instance, $N_0 < \frac{1}{4(a+b)}$, then $N_{2a+2b} < \frac{1}{2(a+b)}$.

5. Error analysis of the algorithm `Solve`

In this section we develop the rigorous error analysis of the Algorithm `Solve`. We assume that this algorithm is executed in finite precision floating point arithmetic. The machine arithmetic is is supposed to satisfy the "$1+\epsilon$" property:

> If $\Box$ is one of the operations $+$, $-$, $*$, $/$, and a and b are floating point numbers, the machine computes $\mathrm{fl}(a \Box b)$, where fl is a rounding-off operator such that
>
> $$\mathrm{fl}(a \Box b) = (a \Box b)(1+\epsilon)$$
>
> and
>
> $$\epsilon = \epsilon(a, b, \Box) < \epsilon_m \ ,$$
>
> where ϵ_m is a small constant called the "machine epsilon."

Let N_0 be the max-norm of the input $(g - g_a x^a, h-1, \varphi)$ to the Algorithm `Solve`. N_{2a+2b} is the max norm of δg and δh at output. E_{2a+2b} is the norm of the error at output. Norms are taken as $\|.\|_{\mathfrak{m},\infty}$,

$\|.\|_{\mathfrak{a},\infty}$ or $\|.\|_{\mathfrak{b},\infty}$, according to convenience. Also, we will rather look at $\|g - g_a x^a\|_{\mathfrak{m}}$ (resp. to $\|h-1\|_{\mathfrak{a}}$) than to $\|g\|_{\mathfrak{m}}$ (resp. $\|h\|_{\mathfrak{a}}$).

Using the above notation, we have the following:

Theorem 4. *Suppose that*

$$N_0 \le \frac{1}{4(a+b)} \quad \textit{and} \quad N_0 \le \frac{1}{10} \ ,$$

and that

$$\epsilon_m < \frac{1}{24(a+b)} \ .$$

Let $k \le 2a+2b$. Then,

$$N_k \le \frac{1}{a+b} \ ,$$

and

$$E_k \le 12000 \epsilon_m \ .$$

Theorem 4 implies that the algorithm `solve` will stay within the error bound of Theorem 3, provided its input satisfies the condition in N_0. To see that, just set $\epsilon_m = \frac{1}{12000}\hat{\epsilon}$.

Proof of Theorem 4: Consider first the following procedure:

$$w \leftarrow \lambda u + v \ ,$$

where w, λ, u and v are complex numbers. Assume we are in the presence of numerical error arising from two sources: the input λ, u and v; and rounding-off. So, the actual computation becomes

$$w + \delta w = \left((\lambda + \delta\lambda)(u + \delta u)(1+\epsilon_1) + (v + \delta v) \right)(1 + \epsilon_2) \ , \qquad (2)$$

where $|\epsilon_1|$ and $|\epsilon_2|$ are both smaller than ϵ_m. We subtract the equality $w = \lambda u + v$ from equation (2), and get

$$\begin{array}{rcl} \delta w & = & (\lambda + \delta\lambda)\delta u (1+\epsilon_1)(1+\epsilon_2) \\ & + & (v + \delta v)\epsilon_2 + \delta v \\ & + & \delta\lambda \ u(1+\epsilon_1)(1+\epsilon_2) \\ & + & \lambda u(\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2) \ . \end{array}$$

Furthermore,

$$|w + \delta w| \leq (|\lambda + \delta\lambda||u + \delta u|(1+\epsilon_m) + |v + \delta v|)(1 + \epsilon_m) \ .$$

So,

$$\left\{ \begin{array}{rcl} |w + \delta w| & \leq & (|\lambda + \delta\lambda||u + \delta u| + |v + \delta v|)(1+\epsilon_m)^2 \\ |\delta w| & \leq & (|\lambda + \delta\lambda||\delta u| + |u||\delta\lambda|)(1+\epsilon_m)^2 \\ & & +|v + \delta v|\epsilon_m + |\delta v| + |\lambda||u|(2\epsilon_m + {\epsilon_m}^2) \end{array} \right.$$

Hence,

$$\left\{ \begin{array}{rcl} 2^{a-i}|w + \delta w| & \leq & [(2^a|\lambda + \delta\lambda|)(2^i|u + \delta u|)2^{-2i} + 2^{a-i}|v + \delta v|)](1+\epsilon_m)^2 \\ 2^{a-i}|\delta w| & \leq & (2^a|\lambda + \delta\lambda| \ 2^i \ |\delta u| 2^{-2i} + 2^i|u| \ 2^a \ |\delta\lambda| 2^{-2i})(1+\epsilon_m)^2 \\ & & +2^{a-i}|v + \delta v|\epsilon_m + 2^{a-i}|\delta v| + 2^a|\lambda|2^i|u|2^{-2i}(2\epsilon_m + {\epsilon_m}^2) \end{array} \right.$$

Now, assume that v and w (resp. u) are endowed with the monic (resp. antimonic) norm. Now we consider the following line:

$$w_{0:a} \leftarrow \lambda u_{0:a} + v_{0:a}$$

This corresponds to line 80. It also bounds what happens in lines 20 and 50 of the algorithm. Summing up, we showed that

Lemma 6. *Assume that:*

i. $\|u\|_{\mathfrak{a},\infty}, \|u + \delta u\|_{\mathfrak{a},\infty} \leq N$

ii. $\|v\|_{\mathfrak{m},\infty}, \|v + \delta v\|_{\mathfrak{m},\infty} \leq N$

iii. $|\lambda|, |\lambda + \delta\lambda| \leq 2^{-a}N$

iv. $\|\delta u\|_{\mathfrak{a},\infty} \leq E$

v. $\|\delta v\|_{\mathfrak{m},\infty} \leq E$

vi. $|\delta \lambda| \leq 2^{-a} E$

Then,

$$\|w\|_{\mathfrak{m},\infty}, \|w+\delta w\|_{\mathfrak{m},\infty} \leq (N+N^2)(1+\epsilon_m)^2$$

and

$$\|\delta w\|_{\mathfrak{m},\infty} \leq 2NE(1+\epsilon_m)^2 + N\epsilon_m + E + N^2(2\epsilon_m + {\epsilon_m}^2) \ .$$

We consider now the double recurrence

$$\left\{ \begin{array}{lcl} N_{i+1} & = & \frac{N_i}{1-N_i}\frac{1}{1-3\epsilon_m} \\ E_{i+1} & = & E_i(1+4N_{i+1})\frac{1}{1-2\epsilon_m} + 8N_{i+1}\epsilon_m \end{array} \right. .$$

Lemma 7. *The sequences $\{N_i\}_{i=0}^{2a+2b}$ and $\{E_i\}_{i=0}^{2a+2b}$ bound the norm and the forward error, respectively, of the values of g, h, φ, δg and δh (in the appropriate norm) computed by the algorithm* `Solve`.

Proof. Lemma 6 models what happens in lines 20, 50 and 80. We are left with two divisions (lines 30 and 70) that may at most multiply N by $\frac{1}{g'}(1+\epsilon_m)$.

Let E be the "forward error" at the beginning of line 20 (resp. 50, 80). We denote by E' a bound of the error after the operation of line 20 (resp. 50, 80). We want a bound E'' of the error after lines 30 (resp. 50, or the combination of lines 70 and 80).

E' will be replaced by $E'' \geq \frac{1}{g'}E' + \frac{N}{g'}\epsilon_m$. Therefore, after lines 20 and 30, we have new values N' and E'' bounded by

$$N' \leq \frac{(N+N^2)\,(1+\epsilon_m)^2}{|g'|}(1+\epsilon_m) \ .$$

As $|g'| \geq 1 - N^2 2^{-2a}$ we get that

$$\begin{aligned} N' &\leq N \frac{1+N}{1-N^2}(1+\epsilon_m)^3 \\ &\leq N \frac{(1+\epsilon_m)^3}{1-N} \\ &\leq N \frac{1}{1-N} \, \frac{1}{1-3\epsilon_m} \ . \end{aligned}$$

Furthermore,

$$\begin{aligned} E'' &\leq \frac{1}{|g'|}E' + \frac{N'}{|g'|}\epsilon_m \\ &\leq \frac{1}{1-\frac{N^2}{4}} \Big(E(1+2N)(1+\epsilon_m)^2 + 4N\epsilon_m + N'\epsilon_m \Big) \\ &\leq \frac{1}{1-\frac{N^2}{4}} \Big(E(1+2N)(1+\epsilon_m)^2 + 4N\epsilon_m + N'\epsilon_m \Big) \ . \end{aligned}$$

But,

$$\frac{1}{1-\frac{N^2}{4}} = 1 + \frac{N^2}{4} + \frac{N^4}{16} + \cdots \leq 1 + \frac{N^2}{2} \ .$$

So,

$$\begin{aligned} E'' &\leq E(1+4N')(1+\epsilon_m)^2 + 8N'\epsilon_m \\ &\leq E(1+4N')\frac{1}{1-2\epsilon_m} + 8N'\epsilon_m \ . \end{aligned}$$

Composition of lines 70 and 80 is similar, since dividing δh by g' is certainly less problematic that dividing the final result by g'. So, for norm-increase and error-bound estimates, we can invert the order of those operators and use the previous bound.

□

The general term for N_i is given by the following: Set

$$\rho = \frac{1}{1-3\epsilon_m} \ .$$

Then,

$$N_i = \frac{\rho^i}{\frac{1}{N_0} - (1 + \rho + \rho^2 + \cdots + \rho^{i-1})} \ .$$

We will show the following bounds:

Lemma 8. *Let $k \in \mathbb{N}$ be fixed, and assume that $0 \leq i \leq k$, $N_0 \leq \min(\frac{1}{2k}, 1/10)$, and $\epsilon_m < \frac{1}{12k}$. Then,*

1. *N_i is well defined.*
2. $N_i < \frac{2}{k}$
3. $\sum_{i<k} N_i < 2$
4. $\prod_{i \leq k}(1 + 4N_i) < 5^4 = 625$.

Lemma 8 may be used to bound E_i, as follows:

$$E_i = E_0 \prod_{0 \leq p \leq i} \left(\frac{1 + 4N_p}{1 - 2\epsilon_m} \right) + 8 \sum_{0 \leq j \leq i} \epsilon_m N_j \prod_{j+1 \leq p \leq i} \left(\frac{1 + 4N_p}{1 - 2\epsilon_m} \right)$$

Since $E_0 = 0$ we bound

$$E_i < 8\epsilon_m \sum_{0 \leq j \leq i} N_j \prod_{0 \leq p \leq i} (1 + 4N_p) \left(\frac{1}{1 - 2\epsilon_m} \right)^i \ .$$

Using the results of Lemma 8 one obtains

$$E_i < 10000\epsilon_m \left(\frac{1}{1 - 2\epsilon_m} \right)^i \ .$$

Moreover, we have

$$\left(\frac{1}{1 - 2\epsilon_m} \right)^i \leq \frac{1}{1 - 2i\epsilon_m} \leq \frac{1}{1 - \frac{1}{6}} \leq \frac{6}{5} \ .$$

This follows from the fact that $\frac{1}{1-x}\frac{1}{1-y} \leq \frac{1}{1-(x+y)}$, x and y positive and small, and from induction on i. So, $E_i < 12000\epsilon_m$.

Since there are $a+b$ recursion calls, we set $k = 2a+2b$, and Theorem 4 is proven.

□

Proof of Lemma 8

Formula

$$\rho^i < \frac{1}{1 - 3i\epsilon_m} \ . \tag{3}$$

can be used to estimate

$$\begin{aligned} \frac{\rho^i - 1}{\rho - 1} &\leq \frac{\frac{1}{1-3i\epsilon_m} - 1}{\frac{1}{1-3\epsilon_m} - 1} \\ &\leq \frac{1 - 3\epsilon_m}{1 - 3i\epsilon_m} \frac{3i\epsilon_m}{3\epsilon_m} \\ &\leq \frac{1 - 3\epsilon_m}{1 - 3i\epsilon_m} i \\ &\leq \frac{1}{1 - 3i\epsilon_m} i \ . \end{aligned}$$

Therefore, using the hypothesis $\epsilon_m < \frac{1}{12k}$, we have $3i\epsilon_m \leq 3k\epsilon_m < \frac{1}{4}$ and

$$\frac{\rho^i - 1}{\rho - 1} \leq \frac{1}{1 - \frac{1}{4}} i = \frac{4}{3} i \ . \tag{4}$$

Using equation (4), we may bound the general term N_i by

$$N_i \leq \frac{\rho^i}{\frac{1}{N_0} - \frac{4}{3} i} \ .$$

This implies item 1.

Formula 3, together with hypothesis $\epsilon_m < \frac{1}{12k}$ implies: $\rho^i < \frac{4}{3}$. Thus,

$$N_i \leq \frac{4}{3} \frac{1}{\frac{1}{N_0} - \frac{4}{3} i} \ .$$

Using $\frac{1}{N_0} > 2k$, we obtain

$$N_i \leq \frac{2}{k} \ .$$

This proves item 2. Item 3 is now trivial. To prove item 4, notice that

$$1 + 4N_i \leq 1 + \frac{16}{3} \frac{1}{\frac{1}{N_0} - \frac{4}{3} i} = \frac{\frac{1}{N_0} - \frac{4}{3} i + \frac{16}{3}}{\frac{1}{N_0} - \frac{4}{3} i} \ .$$

This can be rewritten

$$1+4N_i \leq \frac{\frac{1}{N_0}-\frac{4}{3}(i-4)}{\frac{1}{N_0}-\frac{4}{3}i} \ .$$

Therefore, for $k \geq 4$, the product of $(1+4N_i)$ may be bounded by:

$$\begin{aligned}
\prod_{i\leq k}(1+4N_i) &< \frac{\left(\frac{1}{N_0}+\frac{16}{3}\right)\dots\left(\frac{1}{N_0}-\frac{4}{3}(k-4)\right)}{\left(\frac{1}{N_0}\right)\dots\left(\frac{1}{N_0}-\frac{4}{3}(k)\right)} \\
&< \frac{\left(\frac{1}{N_0}+\frac{16}{3}\right)\left(\frac{1}{N_0}+\frac{12}{3}\right)\left(\frac{1}{N_0}+\frac{8}{3}\right)\left(\frac{1}{N_0}+\frac{4}{3}\right)}{\left(\frac{1}{N_0}-\frac{4}{3}(k-3)\right)\left(\frac{1}{N_0}-\frac{4}{3}(k-2)\right)\left(\frac{1}{N_0}-\frac{4}{3}(k-1)\right)\left(\frac{1}{N_0}-\frac{4}{3}(k)\right)} \\
&< \left(\frac{\frac{1}{N_0}+\frac{16}{3}}{\frac{1}{N_0}-\frac{4}{3}k}\right)^4 \\
&< \left(\frac{2k+\frac{16}{3}}{\frac{2}{3}k}\right)^4 \\
&< \left(3+\frac{8}{k}\right)^4 \\
&< 5^4 = 625
\end{aligned}$$

□

6. Wrong algorithms run faster

The algorithm `Solve` is based on operations of the form

$$g_{1:a} \leftarrow g_{1:a} - g_0 h_{1:b} \ .$$

At a first glance, the operation count would be around $8(d+(d-1)+\dots)$. This can be estimated to $4d^2$.

However, a more careful study proves that not all those arithmetic operations need to be performed. One can save computer work at a cost of introducing some “extra” truncation error. Indeed, if the norms of g and h above are bounded by N, we have just seen that

$$|g_0| \leq 2^{-a}N$$

and

$$|g_0 h_i| \le 2^{-a-i} N^2 \ .$$

What about skipping the operation $g_i \leftarrow g_i - g_0 h_i$? The resulting truncation error (in the appropriate norm) will be bounded by

$$2^{a-i} |g_0 h_i| \le 2^{-2i} N^2 \ .$$

If this is less than $N\epsilon_m$, the error analysis of Lemma 6 and of Section 5 will still hold. Therefore, we will obtain a result within the error bound in Theorem 4.

The same is true for divisions in lines 30 and 70 of the algorithm.

Thence, we may replace lines like

$$g_{1:a} \leftarrow g_{1:a} - g_0 h_{1:b}$$

by

$$g_{1:j} \leftarrow g_{1:j} - g_0 h_{1:j} \ .$$

where $j = \log_2 \frac{1}{\epsilon}$ and ϵ is our error bound.

The operation count is now $4d \log \frac{1}{\epsilon}$ floating point operations, where it is assumed that $\epsilon < \frac{1}{24d}$.

7. Proof of Theorem 3

We can now finish off the proof of Theorem 3. All that remains is to check that, for each Newton iteration, the conditions of Theorem 4 are satisfied. Namely, we need

$$N_0 \le \frac{1}{4d} \text{ and } \ N_0 \le \frac{1}{10} \ .$$

Under the conditions of the Main Theorem and according to Theorem 2

$$\|f - x^a\|_{\mathfrak{h},\infty} < \frac{3}{64d^2} \ .$$

Also, if $f = g^* h^*$ is the exact factorization, Lemma 5 implies that

$$\begin{aligned} \|g^* - x^a\|_{\mathfrak{m},\infty} &< \frac{2a}{R} < \frac{a}{32d^3} \\ \|h^* - 1\|_{\mathfrak{a},\infty} &< \frac{2b}{R} < \frac{b}{32d^3} \ . \end{aligned}$$

Moreover, it is known that the distance of an approximate zero to the exact zero is bounded by 2α. This bound can be further sharpened, see [9]. So,

$$d((g,h),(g^*,h^*)) \le 2\alpha \le 2\frac{\sqrt{d+1}}{8} \ \frac{3\sqrt{d+1}}{64d^2} \ .$$

Using $\frac{d+1}{d^2} < \frac{2}{d}$, one obtains

$$2\alpha \le \frac{3}{128d} \ .$$

Hence,

$$\begin{aligned} \|g - x^a\|_{\mathfrak{m}} &\le \|g - g^*\|_{\mathfrak{m}} + \|g^* - x^a\|_{\mathfrak{m}} \le \frac{3}{128d} + \frac{1}{32d^2} \le \frac{1}{16d} \\ \|h - 1\|_{\mathfrak{a}} &< \|h - h^*\|_{\mathfrak{a}} + \|h^* - 1\|_{\mathfrak{a}} \le \frac{1}{16d} \ . \end{aligned}$$

We still need an estimate of $\|f - gh\|_{\mathfrak{h}}$. We write

$$\begin{aligned} f - gh &= (f - x^a) - (gh - x^a) \\ &= (f - x^a) - (g - x^a)(h-1) - (g - x^a) - x^a(h-1) \ . \end{aligned}$$

Therefore, the norms satisfy

$$\|f - gh\|_{\mathfrak{h}} \le \|f - x^a\|_{\mathfrak{h}} + \|(g - x^a)(h-1)\|_{\mathfrak{h}} + \|g - x^a\|_{\mathfrak{m}} + \|h - 1\|_{\mathfrak{a}} \ .$$

We will need the

Lemma 9. *Let g be monic and h antimonic. Then,*

$$\|(g - x^a)(h-1)\|_{\mathfrak{h}} \le \frac{d}{3}\|g - x^a\|_{\mathfrak{m}}\|h - 1\|_{\mathfrak{a}} \ .$$

Whence we derive that

$$\|f - gh\|_{\mathfrak{h}} \leq \frac{1}{32d^2} + \frac{1}{32^2 d} + \frac{1}{16d} + \frac{1}{16d} \leq \frac{1}{4d} \ .$$

If $d \geq 2$, then $1/4d < 1/10$. In the particular case $d = 2$, we also have

$$\|f - gh\|_{\mathfrak{h}} \leq \frac{1}{128} + \frac{1}{128} + \frac{1}{32} + \frac{1}{32} < \frac{1}{10} \ ,$$

almost finishing the proof of Theorem 3. It remains to prove Lemma 9

Proof of Lemma 9: Set,

$$g^\star = g - x^a \ ,$$

$$h^\star = h - 1 \ ,$$

and $\varphi = g^\star h^\star$. We note that for $0 \leq i < d$

$$\varphi_i = \sum_{k=\max(0,i-b)}^{\min(a,i)} g^\star_k h^\star_{i-k} \ .$$

Now we estimate,

$$\begin{aligned} \|\varphi\|_{\mathfrak{h},1} &\leq \sum_i \sum_{k=\max(0,i-b)}^{\min(a,i)} 2^{|a-i|} |g^\star_k h^\star_{i-k}| \\ &= \sum_i \sum_{k=\max(0,i-b)}^{\min(a,i)} 2^{|a-i|} 2^{-a-i+2k} 2^{a-k} |g^\star_k| 2^{i-k} |h^\star_{i-k}| \ . \end{aligned}$$

Using the fact $k < \min(a, i)$, one obtains

$$|a - i| - a - i + 2k \leq -2\min(a, i) + 2k$$

and hence

$$\begin{aligned}
\|\varphi\|_{\mathfrak{h},\mathtt{1}} &\leq \sum_{i} \sum_{k=\max(0,i-b)}^{\min(a,i)} 2^{-2\min(a,i)+2k} \|g^{\star}\|_{\mathfrak{m}} \|h^{\star}\|_{\mathfrak{a}} \\
&\leq \|g^{\star}\|_{\mathfrak{m}} \|h^{\star}\|_{\mathfrak{a}} \sum_{i} \left(\frac{1}{4} + \frac{1}{4^2} + \frac{1}{4^3} + \dots \right) \\
&\leq \|g^{\star}\|_{\mathfrak{m}} \|h^{\star}\|_{\mathfrak{a}} \sum_{i<d} \frac{1}{3} \\
&\leq \frac{d}{3} \|g^{\star}\|_{\mathfrak{m}} \|h^{\star}\|_{\mathfrak{a}} \ ,
\end{aligned}$$

where in the chain of inequalities we have used the Cauchy-Schwartz inequality.

So,

$$\|\varphi\|_{\mathfrak{h}} \leq \|\varphi\|_{\mathfrak{h},1} \leq \frac{d}{3}\|g^{\star}\|_{\mathfrak{m}}\|h^{\star}\|_{\mathfrak{a}}$$

□

8. Proof of remaining Lemmata

8.1. **Proof of Lemma 1.** Notice that for each i with $0 \leq i \leq a+b+1$ we have that the i-th component of $D^2\varphi_f(g,h)$ is given by

$$\Big(D^2\varphi_f(g,h)\Big)_i : (\widehat{g}, \widehat{h}, \widetilde{g}, \widetilde{h}) \mapsto \sum(\widehat{g}_j\widetilde{h}_{i-j}) + \sum(\widetilde{g}_j\widehat{h}_{i-j}) \ , \tag{5}$$

where the sum is taken over a range of subindices j such that the corresponding coefficients make sense and $\widehat{g}_j$ and $\widetilde{g}_j$ (resp. $\widehat{h}_j$ and $\widetilde{h}_j$) are elements of the tangent space to the affine linear manifold of monic (resp. antimonic) polynomials. To estimate the operator norm mentioned above we bound

$$\left\|D^2\varphi_f(g,h)(\widehat{g}, \widehat{h}, \widetilde{g}, \widetilde{h})\right\|_{\mathfrak{h}}$$

with $\left\|(\widehat{g}, \widehat{h})\right\|_{\mathfrak{ma}} = 1$ and $\left\|(\widetilde{g}, \widetilde{h})\right\|_{\mathfrak{ma}} = 1$. Now, we estimate each of the sums in equation (5). Recall that

$$\begin{aligned} 2^{|a-i|} \sum_{\substack{0\leq j<a \\ 0<i-j\leq b}} |\widehat{g}_j\widetilde{h}_{i-j}| &\leq \sum |\widehat{g}_j| 2^{a-j} 2^{i-j} |\widetilde{h}_{i-j}| 2^{|a-i|-a+2j-i} \\ &\leq \left(\max_{\substack{0\leq j<a \\ 0<i-j\leq b}} 2^{|a-i|-a-i+2j} \right) \|\widehat{g}\|_{\mathfrak{m}} \left\|\widetilde{h}\right\|_{\mathfrak{a}} . \end{aligned}$$

To estimate

$$M_{a,b} \stackrel{\text{def}}{=} \left(\max_{\substack{0\leq j<a \\ 0<i-j\leq b}} 2^{|a-i|-a-i+2j} \right) ,$$

we consider the cases $i \leq a$ and $i > a$. In the former, we get

$$2^{|a-i|-a-i+2j} = 2^{2j-2i} \leq \frac{1}{4} ,$$

since $0 < i - j$. In the latter,

$$2^{|a-i|-a-i+2j} = 2^{2(j-a)} \leq \frac{1}{4} \ ,$$

since $j < a$.

A similar reasoning gives,

$$2^{|a-i|} \Big| \sum_{\substack{0 \leq j < a \\ 0 < i-j \leq b}} \tilde{g}_j \hat{h}_{i-j} \Big| \ \leq \frac{1}{4} \|\tilde{g}\|_{\mathfrak{m}} \left\| \hat{h} \right\|_{\mathfrak{a}} \ .$$

Now we recall that

$$\|\hat{g}\|_{\mathfrak{m}}{}^2 + \left\| \hat{h} \right\|_{\mathfrak{a}}{}^2 = 1 \ ,$$

and

$$\|\tilde{g}\|_{\mathfrak{m}}{}^2 + \left\| \tilde{h} \right\|_{\mathfrak{a}}{}^2 = 1 \ .$$

Hence,

$$\begin{aligned} \sum_{0 \leq i \leq a+b} \left| 2^{|a-i|} \left(D^2 \varphi_f(g,h)(\hat{g}, \hat{h}, \tilde{g}, \tilde{h}) \right)_i \right|^2 &\leq \sum_{0 \leq i \leq a+b} \frac{1}{16} (\|\hat{g}\|_{\mathfrak{m}} \left\| \tilde{h} \right\|_{\mathfrak{a}} + \left\| \hat{h} \right\|_{\mathfrak{a}} \|\tilde{g}\|_{\mathfrak{m}})^2 \\ &\leq \frac{(a+b+1)}{16} \left\| (\hat{g}, \hat{h}) \right\|_{\mathfrak{ma}}{}^2 \left\| (\tilde{g}, \tilde{h}) \right\|_{\mathfrak{ma}}{}^2 \\ &\leq \frac{(a+b+1)}{16} \ . \end{aligned}$$

Therefore,

$$\left\| D^2 \varphi_f(g,h) \right\|_{\mathfrak{ma} \to \mathfrak{h}} \leq \frac{\sqrt{d+1}}{4} \ .$$

□

8.2. **Proof of Lemma 2.** We have to estimate the norm of the operator $D\varphi_f(g,h)^{-1}$ in the case $g = x^a$ and $h = 1$. Notice that in this case

$$D\varphi_f(x^a, 1)^{-1} = \begin{bmatrix} I_{a+1} & 0 \\ 0 & I_b \end{bmatrix} \ ,$$

where I_k denotes the $k \times k$ identity matrix.

Now, if

$$\widehat{\varphi} = \widehat{\varphi}_0 x^0 + \cdots + \widehat{\varphi}_{a+b} x^{a+b} \ ,$$

with $\|\widehat{\varphi}\|_{\mathfrak{h}} = 1$, then

$$\sum_{0 \le i \le a+b+1} 2^{2|a-i|} |\widehat{\varphi}_i|^2 = 1 \ ,$$

and

$$\left\| D\varphi_f(x^a, 1)^{-1} \cdot \widehat{\varphi} \right\|_{\mathfrak{ma}}^2 = \|\widehat{\varphi}\|_{\mathfrak{h}}^2 = 1 \ .$$

Hence,

$$\left\| D\varphi_f(x^a, 1)^{-1} \right\|_{\mathfrak{ma} \to \mathfrak{h}} = 1 \ .$$

9. Acknowledgments

We thank very helpful and stimulating conversations with D. Bini, J-P. Cardinal B. Fux Svaiter P. Kirrinis, V. Pan and .

References

[1] Victor Y. Pan, *Solving a Polynomial Equation: Some History and Recent Progress.* Preprint, CUNY (1995).

[2] Alexandre Ostrowski, Recherches sur la Méthode de Graeffe et les Zéros des Polynomes et des Séries de Laurent. *Acta Mathematica* **72**, 99-257 (1940).

[3] J. V. Uspensky *Theory of equations.* Mc Graw Hill, New York (1948).

[4] Arnold Schönhage, Equation Solving in Terms of Computational Complexity. *Proceedings of the International Congress of Mathematicians, Berkeley, 1986*, 131-153. American Mathematical Society (1987).

[5] Peter Kirrinis, *Partial Fraction Decomposition in* $\mathbb{C}(F)$ *and simultaneous Newton Iteration for Factorization in* $\mathbb{C}[F]$. Preprint, Bonn, (1995).

[6] Victor Y. Pan, Optimal and Nearly Optimal Algorithms for Approximating Polynomial Zeros. *Computers Math. Applic.)*, to appear.

[7] Victor Y. Pan, Deterministic Improvement of Complex Polynomial Factorization Based on the Properties of the Associated Resultant. *Computers Math. Applic.*, **30** (2), 71-94 (1995).

[8] Steve Smale, Newton method estimates from data at one point, in R. Erwing, K. Gross and C. Martin (editors). *The merging of disciplines: New directions in Pure, Applied and Computational Mathematics.* Springer, New York (1986)

[9] Michael Shub and Steve Smale, On the Complexity of Bezout's Theorem I - Geometric aspects. *Journal of the AMS*, **6**(2), (1993).

[10] Michael Shub and Steve Smale, On the complexity of Bezout's Theorem II - Volumes and Probabilities. in: F. Eysette and A. Galligo, eds: *Computational Algebraic Geometry.* Progress in Mathematics **109** 267-285, Birkhauser (1993).

[11] Michael Shub and Steve Smale, Complexity of Bezout's Theorem III ; Condition number and packing. *Journal of Complexity* **9**, 4-14 (1993).

[12] Michael Shub and Steve Smale, Complexity of Bezout's Theorem IV ; Probability of success ; Extensions. Preprint, Berkeley (1993).

[13] Michael Shub and Steve Smale, Complexity of Bezout's Theorem V: Polynomial time. *Theoretical Computer Science* **133**(1), 141-164 (1994)

[14] Gregorio Malajovich, On generalized Newton algorithms: quadratic convergence, path-following and error analysis. Theoretical Computer Science **133**, 65-84 (1994).

Departamento de Matematica Aplicada da UFRJ, Caixa Postal 68530, CEP 21945, Rio, RJ, Brasil

E-mail address: `gregorio@lyric.labma.ufrj.br`

Instituto de Matematica Pura e Aplicada – CNPq, Estrada Dona Castorina 110, Jardim Botanico, CEP 22460-320, Rio de Janeiro, RJ, Brasil

E-mail address: `zubelli@impa.br`