Polynomial-Time Computation of Taylor Series

N. Th. Müller
Abteilung Informatik Universität Trier
Postfach 3825 D-5500 Trier

Abstract. The connection between analytic complex functions and their Taylor series is studied from the viewpoint of computational complexity. Central result: If an analytic function f is computable in time $t(n)$, then the coefficients $(a_k)_{k\in\mathbb{N}}$ of its Taylor series are uniformly computable in time $(k+1)\cdot t(n+k)$.

Given a computable real (or complex) function of known computational complexity, it is natural to ask about the complexity of the roots, the derivatives or the integral. The underlying numerical transformations on real functions have recently been studied in a variety of papers. The results can usually be divided into two classes: Sometimes, problems from discrete complexity theory can be coded into real functions. Some examples are

- Root-finding [KoFr82]: Polynomial time computable functions may have roots of arbitrary complexity, this result even holds for functions computable in linear time [Mu86].
- Differentiation [Ko83]: Polynomial time computable functions may have noncomputable derivatives.
- Integration [Fr84]: The integral of a polynomial-time computable function is not necessarily polynomial time computable unless $P = \#P$.

Usually the real functions that are constructed in the corresponding proofs are easy to compute but they suffer from poor analytic properties. On the other hand, sufficient analytic properties usually imply that those transformations lead from polynomial time functions to polynomial time results [KoFr88, Mu87, Sc82].

A central analytic property is the expansibility of a function into a power series. In [Mu87] the author presented a proof showing a very effective transformation betweens analytic functions on the real numbers and their power series, with the following central result: If an analytic function f is computable in time $t(n)$, then the coefficients $(a_k)_{k\in\mathbb{N}}$ of its Taylor series are uniformly computable in time $(k+1)\cdot t(n+k)$. Applications were shown for integration and analytic continuation. Later, a similar result has been published by Ko and Friedman [KoFr88] showing the uniform polynomial time computabilty of the series if the function is polynomial time computable (but lacking a more detailed analysis of the complexity).

In this paper the proof from [Mu87] is extended to complex analytic functions and to functions of more than one variable. The author thanks Arnold Schönhage for pointing out to him the idea of Hermite's representation of interpolating polynomials.

1 Basic Notions

Our definition of computability for real functions uses *dyadic decimals* in an extension to the definition in [Fr84]: A dyadic decimal consists of the symbol + or the symbol −, followed by a first possibly empty string of 0's and 1's, followed by a decimal point '•', followed by a second possibly empty (and possibly infinite) string of 0's and 1's. If the first string is not empty, it must begin with 1. We let $\mathcal{D}$ denote the set of all these dyadic decimals. So any element w of $\mathcal{D}$ is of the form $w = \pm w_m \dots w_0 \bullet v_1 \dots v_n$, which is a finite string, or $w = \pm w_m \dots w_0 \bullet v_1 \dots$, which is an infinite string, with $n, m \in \mathbb{N}$ and $w_i, v_j \in \{0,1\}$. Such a dyadic decimal w denotes the following real number $\varrho(w)$:

$$\varrho(w) := (\pm 1)\cdot(\sum w_i 2^i + \sum v_j 2^{-j})$$

Usually we will make no difference between the dyadic decimal w and the corresponding real number $\varrho(w)$ (although there may be infinitely many dyadic decimals corresponding to the same real number). In fact, we will always write w instead of $\varrho(w)$ to increase the readability. The right interpretation can always be chosen from the context.

Our notion of computability and complexity of real functions is based on Turing machine computability on $\mathcal{D}$ and $\mathbb{N}$, interpreted as sets of strings. Here natural numbers are supposed to be in binary form (but this is not essential to the definition). Inputs are read from the left, and the Turing machines must stop after a finite number of steps to produce a welldefined output. So although the inputs might be infinite, the output is always finite and only depends on finite prefixes of the inputs.

1.1 Definition. A function $f\colon \mathbb{R}^i \dashrightarrow \mathbb{R}^j$ is called computable on a set $G \subseteq \mathbb{R}^i$, iff there is a computable function $\Gamma\colon \mathcal{D}^i \times \mathbb{N} \dashrightarrow \mathcal{D}^j$ such that $\Gamma(\bar{x}, n)$ exists for any $\bar{x} \in \mathcal{D}^i \cap G$ and for any $n \in \mathbb{N}$, and in addition:

$$|\Gamma(\bar{x}, n) - f(\bar{x})| \leq 2^{-n}$$

Here $|\cdot|$ denotes the maximum norm on $\mathbb{R}^j$.

f is called computable in time t (where $t\colon \mathbb{N} \to \mathbb{N}$) on G, iff there is a multitape Turing machine M computing the function Γ such that for any $\bar{x} \in \mathcal{D}^i \cap G$ and for any $n \in \mathbb{N}$ the number of steps M needs to compute $\Gamma(\bar{x}, n)$ is bounded by $t(n)$. □

The definition of computability is equivalent to the definitions used in [Kr84, We87, Mu86], but there are slight differences to [Fr84, KoFr82, Lo89] according to a different treatment of domains.

Several remarks to the definition should be made: Using $i = 0$ and the set $G = \mathbb{R}^0 = \{()\}$ consisting only of the empty tuple, the above definition includes a notion of computability for real tuples. For example, a real number x is computable in time t, iff there is a Turing machine M computing a function $\Gamma\colon \mathbb{N} \to \mathcal{D}$, such that $t(n)$ steps of M are sufficient to compute $\Gamma(n)$ and $|\Gamma(n) - x| \leq 2^{-n}$ holds for any n .

Furthermore in the definition $\bar{x}$ as well as $\Gamma(\bar{x}, n)$ are used in two different interpretations: as tuples of strings from $\mathcal{D}$ and also as tuples of real numbers. As noted above, the interpretation to choose in each case should be obvious from the context.

Since the inputs might be infinite, the multitape Turing machines should use separate input tapes (or separate tracks on a single input tape) for each of its inputs.

As examples we consider the elementary arithmetic operations. Let $\mathcal{M}(n) := n \operatorname{ld} n \operatorname{ld} \operatorname{ld} n$ be the complexity bound for fast integer multiplication due to the algorithm by Schönhage and Strassen [ScSt71]. (Here and in the rest of the paper ld denotes a binary logarithm with natural values: $\operatorname{ld} n := \min\{i \in \mathbb{N} \mid 2^i \geq n\}$)

1.2 Lemma. Let $m \in \mathbb{N}$ be given and let $G_1 := [-2^m, 2^m] \times [-2^m, 2^m] \subset \mathbb{R}^2$, $G_2 := [-2^m, 2^m] \times ([-2^m, -2^{-m}] \cup [2^{-m}, 2^m]) \subset \mathbb{R}^2$. Then addition and subtraction are computable in time $\mathcal{O}(\,n{+}m\,)$ on G_1, multiplication is computable in time $\mathcal{O}(\,\mathcal{M}(n+m)\,)$ on G_1 and division is computable in time $\mathcal{O}(\,\mathcal{M}(n+m)\,)$ on G_2. The Turing machines used to show the complexity bounds can be chosen *independent* from the sets G_1 resp. G_2. □

Proof.

Suppose $z_1, z_2 \in \mathcal{D}$ and $n \in \mathbb{N}$ are given. Let k_1 and k_2 be the numbers of digits to the left of the decimal points in z_1 resp. z_2, and let $k := \max\{k_1, k_2\}$, so $|z_1|, |z_2| \leq 2^k$. If $(z_1, z_2) \in G_1$, then $k \leq m{+}1$ (since in $\mathcal{D}$ a nonempty string on the left of the decimal point must begin with 1).

First we consider addition of reals: If z_1 has less than $n{+}1$ digits to the right of the decimal point, then extend z_1 with 0's to this length. Otherwise truncate z_1 after the $n{+}1$st digit. Let

$\bar{z}_1 \in \mathcal{D}$ be the result of this manipulation and let $\bar{z}_2$ be the corresponding result treating z_2. Add the finite strings $\bar{z}_1$ and $\bar{z}_2$ in the standard way to yield a result $\bar{z} \in \mathcal{D}$. Then $|\bar{z} - (z_1+z_2)| = |(\bar{z}_1+\bar{z}_2) - (z_1+z_2)| \leq |\bar{z}_1-z_1| + |\bar{z}_2-z_2| \leq 2^{-n-1} + 2^{-n-1} = 2^{-n}$. Since both $\bar{z}_1$ and $\bar{z}_2$ consist of at most $k+n+1$ digits, $\bar{z}$ can be computed in $c\cdot(k+n+1)$ steps for some constant c. So if $(z_1, z_2) \in G_1$, then $c\cdot(m+n+2)$ steps are sufficient. The same technique can be used for the subtraction of reals.

For the multiplication of reals extend or truncate z_1 and z_2 after the $n+k+2$nd digit to the right of the decimal point yielding $\hat{z}_1$ and $\hat{z}_2$. Discard the decimal points in $\hat{z}_1$ and $\hat{z}_2$ (which indeed is a multiplication of both numbers with 2^{n+k+2} and transforms them into integers). Then multiply these resulting integers using the fast integer multiplication algorithm. Into the integer product insert a decimal point on the left of the $2(n+k+2)$th digit from the right (which indeed is a division by $2^{2(n+k+2)}$) yielding a result $\hat{z} \in \mathcal{D}$ with $\hat{z}_1 \cdot \hat{z}_2 = \hat{z}$. Then $|\hat{z} - (z_1\cdot z_2)| = |(\hat{z}_1\cdot\hat{z}_2) - (z_1\cdot z_2)| \leq |\hat{z}_1|\cdot|\hat{z}_2 - z_2| + |z_2|\cdot|\hat{z}_1 - z_1| \leq (|z_1|+1)\cdot 2^{-(n+k+2)} + |z_2|\cdot 2^{-(n+k+2)} \leq 2^{k+1}\cdot 2^{-(n+k+2)} + 2^k\cdot 2^{-(n+k+2)} \leq 2^{k+2-(n+k+2)} = 2^{-n}$. If $(z_1, z_2) \in G_1$, then both $\hat{z}_1$ and $\hat{z}_2$ consist of at most $2m+n+4$ digits, so $\hat{z}$ can be computed in $\mathcal{O}(\mathcal{M}(2m+n+4)) = \mathcal{O}(\mathcal{M}(n+m))$ steps.

For the division of real numbers we use the known result that on any *fixed* compact set $[a, b] \times [c, d]$, where $0 \notin [c, d]$, we are able to divide in time $\mathcal{O}(\mathcal{M})$ [Br75]. Suppose $z_1 \in G_1$, $z_2 \in G_2$ and $n \in \mathbb{N}$ are given, again let k_1 be the number of digits to the left of the decimal point in z_1. W.l.o.g. let both z_1 and z_2 be positive. In order to divide z_1 by z_2 we transform z_1 into the interval $[0, 1]$ and z_2 into the interval $[1, 2]$, divide the transforms with the necessary precision and finally retransform the result: First we compute $\tilde{z}_1 = z_1 \cdot 2^{-k_1}$ by moving the decimal point to the beginning of z_1, so $0 \leq \tilde{z}_1 \leq 1$. Then we move the decimal point in z_2 to the right of the first '1' (erasing any resulting leading zeroes) yielding a result $1 \leq \tilde{z}_2 \leq 2$. By counting the number of necessary movement steps we determine $l \in \mathbb{Z}$ with $\tilde{z}_2 = z_2 \cdot 2^{-l}$. Next, using an algorithm for division on the set $[0, 1] \times [1, 2]$, we compute $\tilde{z}$ with $|\tilde{z} - \tilde{z}_1/\tilde{z}_2| \leq 2^{-k_1+l-n}$. Finally we determine $z := \tilde{z}\cdot 2^{k_1-l}$ by moving the decimal point in $\tilde{z}$ accordingly. So $|z - z_1/z_2| = 2^{k_1-l}\cdot|\tilde{z} - \tilde{z}_1/\tilde{z}_2| \leq 2^{k_1-l}\cdot 2^{-k_1+l-n} = 2^{-n}$. If $(z_1, z_2) \in G_2$, then $k_1 \leq m+1$ and $|l| \leq m+1$, so z can be computed in $\mathcal{O}(\mathcal{M}(n+2m+2)) = \mathcal{O}(\mathcal{M}(n+m))$ steps. □

Treating complex numbers as pairs of reals, corresponding results hold for the complex operations. Furthermore many elementary real functions like e^x, $\log x$, $\sin x$, $\cos x$ are computable in time $\mathcal{M}(n) \cdot \mathrm{ld}\, n$ on any *fixed* closed interval in their domains [Br76].

Later on we will consider relations between analytic functions and their Taylor series, so we will now also give a definition of computability and complexity for sequences of numbers or functions.

1.3 Definition. A m-dimensional sequence $(f_{\bar{k}})_{\bar{k}\in\mathbb{N}^m}$ of functions $f_{\bar{k}}\colon \mathbb{R}^i \dashrightarrow \mathbb{R}^j$ is called computable on a set $G \subseteq \mathbb{R}^i$, iff there is a computable function $\Gamma\colon \mathcal{D}^i \times \mathbb{N}^m \times \mathbb{N} \dashrightarrow \mathcal{D}^j$ such that $\Gamma(\bar{x}, \bar{k}, n)$ exists for any $\bar{x} \in \mathcal{D}^i \cap G$ and for any $\bar{k} \in \mathbb{N}^m, n \in \mathbb{N}$, and in addition:

$$\left|\Gamma(\bar{x}, \bar{k}, n) - f_{\bar{k}}(\bar{x})\right| \leq 2^{-n}$$

$(f_{\bar{k}})_{\bar{k}\in\mathbb{N}^m}$ is called computable in time t (where $t\colon \mathbb{N} \to \mathbb{N}$) on G, iff there is a multitape Turing machine M computing the function Γ such that for any $\bar{x} \in \mathcal{D}^i \cap G$ and for any $\bar{k} = (k_1, \ldots, k_m) \in \mathbb{N}^m, n \in \mathbb{N}$ the number of steps M needs to compute $\Gamma(\bar{x}, \bar{k}, n)$ is bounded by $t(k_1+\ldots+k_m + n)$. □

In the whole of the paper we restrict the class of complexity bounds $t\colon \mathbb{N} \to \mathbb{N}$ in order to allow simplifications; we will only use monotone bounds satisfying the following regularity condition for some constants c, m (depending on t) and for any $n \geq m$:

$$2 \cdot t(n) \overset{(1)}{\leq} t(2n) \overset{(2)}{\leq} c \cdot t(n)$$

Similar conditions have been used in a variety of papers, including [Al82, Br75, FiSt74]. Monotone functions t satisfying the inequalities will be called *regular*. Part (1) of the condition implies

$t(2^i m) \geq 2^i t(m)$, so t must grow at least linearly. By part (2), $t(2^i m) \leq c^i t(m) = t(m) \cdot (2^i)^{\operatorname{ld} c}$, so $t(n)$ is of order $\mathcal{O}(n^{\operatorname{ld} c})$ (where c is depending on t), and hence t may not grow exponentially. On the other hand, most of the complexity bounds of practical interest are, indeed, regular, e.g. all functions of the form $n^i \cdot \operatorname{ld}^j n$ with $i \geq 1$ and $j \geq 0$. Especially the cited complexity bound for integer multiplication, $\mathcal{M}(n) := n \operatorname{ld} n \operatorname{ld} \operatorname{ld} n$, is regular.

Part (2) of the regularity condition for a function t implies $t(\ell(n)) \in \mathcal{O}(t)$ for any *linear* function ℓ, i.e. a linear increase in the arguments of t has no influence on the asymptotic behaviour. This allows a simpler formulation of the asymptotic complexity bounds we are dealing with.

In order to simplify the formulation of algorithms, we will use the following notation: Suppose f is computable in time t on a set G, and suppose we know a machine M, which is a witness for the computability of f (in the sense of definition 1.1). Then for $\bar{x} \in G$ and $n \in \mathbb{N}$ let $\bar{y} :\approx_{(n)} f(\bar{x})$ be an abbreviation for $\bar{y} := \Gamma_M(\bar{x}, n)$. So we are able to compute $\bar{y}$ in at most $t(n)$ steps, where $|f(\bar{x}) - \bar{y}| \leq 2^{-n}$. (If f is a function with real numbers (not tuples) as values, we have $\bar{y} - 2^{-n} \leq f(\bar{x}) \leq \bar{y} + 2^{-n}$.)

As an example of the application of condition (2) for regular complexity bounds, we consider the differentiation of real functions: Given a function f with sufficient analytical properties and computable in time t for a regular t, we show that the derivative f' is also of complexity $\mathcal{O}(t)$.

To compute the derivative we use its definition as the limit of differential quotients, where the mean value theorem and a Lipschitz condition for f' allow the estimation of the speed of convergence.

1.4 Theorem. Let $f\colon [a,b] \to \mathbb{R}$ be given such that f' exists and such that f' is Lipschitz-continuous on $[a,b]$, where $a < b$. If f is computable in time t on $[a,b]$, where t is regular, then f' is computable in time $\mathcal{O}(t)$ on $[a,b]$. □

Proof.

Since f' is supposed to be Lipschitz-continuous, let $L \in \mathbb{N}$ be an appropriate constant with $|f'(y) - f'(z)| \leq 2^L \cdot |y - z|$ for any $y, z \in [a,b]$. W.l.o.g. let L be such that $b - a \geq 4 \cdot 2^{-L}$ and let a fixed $\zeta \in \mathcal{D}$ with a finite representation be given such that $a \leq \zeta - 2^{-L}$ and $b \geq \zeta + 2^{-L}$.

Let $x \in [a,b] \cap \mathcal{D}$ and $n \in \mathbb{N}$ be given. Compute $d :\approx_{(L)} x - \zeta$, so $\zeta + d - 2^{-L} \leq x \leq \zeta + d + 2^{-L}$. Then let $m := n + L + 2$. If $d \geq 0$, then let $x_1 := x - 2^{-m}$ and $x_2 := x$, otherwise let $x_2 := x + 2^{-m}$ and $x_1 := x$. By this construction, surely $x_1, x_2 \in [a,b]$ holds. Now compute approximations $f_1 :\approx_{(2m)} f(x_1)$ and $f_2 :\approx_{(2m)} f(x_2)$. Furthermore compute $\Delta f :\approx_{(2m)} f_2 - f_1$ and let $df := 2^m \cdot \Delta f$. Due to the mean value theorem there exists a ξ between x_1 and x_2 (so $|\xi - x| \leq 2^{-m}$) such that

$$
\begin{aligned}
|df - f'(x)| &= \left|\frac{\Delta f}{2^{-m}} - f'(x)\right| \\
&\leq \left|\frac{\Delta f}{2^{-m}} - \frac{f_2 - f_1}{2^{-m}}\right| + \left|\frac{f_2 - f_1}{2^{-m}} - \frac{f(x_2) - f(x_1)}{2^{-m}}\right| \\
&\quad + \left|\frac{f(x_2) - f(x_1)}{2^{-m}} - f'(x)\right| \\
&= 2^m \cdot |\Delta f - (f_2 - f_1)| + 2^m \cdot |(f_2 - f(x_2)) + (f(x_1) - f_1)| \\
&\quad + |f'(\xi) - f'(x)| \\
&\leq 2^m \cdot 2^{-2m} + 2^m \cdot (2^{-2m} + 2^{-2m}) + 2^L \cdot 2^{-m} \\
&\leq 2^{-m+L+2} \quad = 2^{-n}
\end{aligned}
$$

So df is the needed approximation to $f'(x)$. Obviously, there is a c_1 with $|x|, |\zeta| \leq 2^{c_1}$; since $\{f(y) \mid y \in [a,b]\}$ is compact, there also is a bound c_2 for $|f_1|$ and $|f_2|$. Finally the multiplication of Δf with 2^m is just a shift of the decimal point. So all operations except the evaluations of f

can be computed even in time linear in m. But to compute f_1 and f_2, $2t(2m) = 2t(2n + 2L + 4)$ steps are sufficient. Since t is supposed to be regular, this is in $\mathcal{O}(t)$. □

2 Taylor Coefficients

Applying Theorem 1.4 to a polynomial time computable function f, that is infinitely often differentiable, shows that any derivative of this function is asymptotically of the same complexity as f. Ko and Friedman [KoFr88] show that this result is inherently nonuniform: The sequence $\left(f^{(k)}(0)\right)$ might be not uniformly computable in polynomial time. On the other hand, for analytic real functions, a uniform result has been shown [Mu87, KoFr88]. We now extend this result to complex functions of one or more variables.

In the following let $f\colon \mathbb{C} \dashrightarrow \mathbb{C}$ be a function on the complex numbers that is analytic in $0 \in \mathbb{C}$ (of course any other easy to compute complex number could be used instead of 0). So the Taylor series

$$\sum_{k \in \mathbb{N}} \frac{1}{k!} f^{(k)}(0) \cdot z^k$$

has a radius of convergence R greater than zero. We will use the abbreviation $a_k := \dfrac{1}{k!} f^{(k)}(0)$ for the coefficients of the series.

Let γ be an arbitrary real number such that $0 < \gamma < R$. Then f is analytic in any z such that $|z| \le \gamma$. Let $\Gamma := \{t \in \mathbb{C} \mid |t| = \gamma\}$ be the circle with radius γ and let $M := \max\{|f(t)| \mid t \in \Gamma\}$ be the maximal value of $|f|$ on this circle.

Our aim is to develop complexity bounds for the Taylor coefficients from complexity bounds for the function f. In order to approximate a coefficient a_k with an error not exceeding 2^{-n}, for given k and n, we will use polynomials interpolating f. Although an interpolating polynomial is uniquely determined by f and the points of interpolation, there exist different representations of this unique polynomial. So we may use one representation to calculate interpolating errors and another one to actually compute the polynomial. We first introduce *Hermite's* representation:

Let $m + 1$ arbitrary distinct points $z_0, \ldots, z_m \in \mathbb{C}$ with $|z_i| < \gamma$ be given (for some arbitrary m). We define a polynomial ω by

$$\omega(z) := \prod_{0 \le i \le m} (z - z_i)$$

The coefficients of ω will be denoted by σ_j, so

$$\omega(z) = \sum_{0 \le j \le m+1} \sigma_j z^j.$$

Now consider the function P defined as

$$P(z) := \frac{1}{2\pi \mathbf{i}} \int_\Gamma \frac{f(t)}{\omega(t)} \cdot \frac{\omega(t) - \omega(z)}{t - z} dt$$

2.1 Lemma. P is the unique polynomial of degree $\le m$ interpolating f at the points $z_0, \ldots, z_m$. □

Although this result is well-known, we present a proof. Later on we will use parts of this proof to determine the interpolating errors.

Proof.

Since for any j

$$t^j - z^j = (t - z) \cdot (\sum_{0 \le i < j} z^i \cdot t^{j-1-i}),$$

the polynomial $\omega(t)-\omega(z)$ is a multiple of $t-z$. We achieve the following representation:

$$\begin{aligned}
\frac{\omega(t)-\omega(z)}{t-z} &= \frac{\sum_{1\le j\le m+1} \sigma_j\cdot(t^j-z^j)}{t-z}\\
&= \sum_{1\le j\le m+1} \sigma_j\cdot\frac{t^j-z^j}{t-z}\\
&= \sum_{1\le j\le m+1} \sigma_j\cdot(\sum_{0\le i<j} z^i\cdot t^{j-1-i})\\
&= \sum_{0\le i<j\le m+1} \sigma_j\cdot z^i\cdot t^{j-1-i}\\
&= \sum_{0\le i\le m} z^i\cdot(\sum_{i+1\le j\le m+1} \sigma_j\cdot t^{j-1-i})\\
&= \sum_{0\le i\le m} z^i\cdot(\sum_{i\le j\le m} \sigma_{j+1}\cdot t^{j-i})
\end{aligned}$$

Using this reformulation we get

$$P(z) = \sum_{0\le i\le m} z^i\cdot(\frac{1}{2\pi\mathbf{i}}\int_\Gamma \frac{f(t)}{\omega(t)}\cdot[\sum_{i\le j\le m}\sigma_{j+1}t^{j-i}]dt)$$

So in fact P is a polynomial of degree m. Since $\omega(z_i)=0$ holds by definition of ω, especially we get

$$\begin{aligned}
P(z_i) &= \frac{1}{2\pi\mathbf{i}}\int_\Gamma \frac{f(t)}{\omega(t)}\cdot\frac{\omega(t)-0}{t-z_i}dt\\
&= \frac{1}{2\pi\mathbf{i}}\int_\Gamma \frac{f(t)}{t-z_i}dt \quad = \quad f(z_i)
\end{aligned}$$

Thus P is the unique polynomial interpolating f at $z_0,\ldots,z_m$. □

As stated above, our aim is to approximate a_k with a precision of 2^{-n}. So we now examine the k-th derivative $P^{(k)}$ of P for arbitrary $k\le m$.

2.2 Lemma. If P is the unique polynomial interpolating f at $m+1$ distinct points $z_0,\ldots,z_m$ with $|z_i|\le\varepsilon$, where $\varepsilon\in\mathbb{R}$, $0<\varepsilon\le\gamma/2$, then for any $k\le m$

$$\left|a_k-\frac{1}{k!}\cdot P^{(k)}(0)\right| \le M\cdot\varepsilon^{m+1-k}\cdot\frac{4^{m+1}}{\gamma^{m+1}}$$

□

Proof.
In order to find a bound for $\left|\frac{1}{k!}P^{(k)}(0)-a_k\right|$ we consider $\left|P^{(k)}(0)-f^{(k)}(0)\right|$. Using Hermite's representation of P from above we immediately get

$$P^{(k)}(z) = \sum_{k\le i\le m}\frac{i!}{(i-k)!}\cdot z^{i-k}\cdot(\frac{1}{2\pi\mathbf{i}}\int_\Gamma \frac{f(t)}{\omega(t)}\cdot[\sum_{i\le j\le m}\sigma_{j+1}\cdot t^{j-i}]dt)$$

This implies

$$P^{(k)}(0) = \frac{k!}{2\pi\mathbf{i}}\int_\Gamma \frac{f(t)}{\omega(t)}\cdot[\sum_{k\le j\le m}\sigma_{j+1}\cdot t^{j-k}]dt$$

On the other hand

$$f^{(k)}(0) = \frac{k!}{2\pi\mathbf{i}} \int_\Gamma \frac{f(t)}{t^{k+1}} dt$$

So

$$\begin{aligned}
\left|P^{(k)}(0) - f^{(k)}(0)\right| &= \left| \frac{k!}{2\pi\mathbf{i}} \int_\Gamma \frac{f(t)}{\omega(t)} \cdot [\sum_{k\le j\le m} \sigma_{j+1} \cdot t^{j-k}] dt \quad - \quad \frac{k!}{2\pi\mathbf{i}} \int_\Gamma \frac{f(t)}{t^{k+1}} dt \right| \\
&= \frac{k!}{2\pi} \cdot \left| \int_\Gamma f(t) \cdot [\frac{\sum_{k\le j\le m} \sigma_{j+1} \cdot t^{j-k}}{\omega(t)} - \frac{1}{t^{k+1}} dt] \right| \\
&= \frac{k!}{2\pi} \cdot \left| \int_\Gamma f(t) \cdot \frac{[\sum_{k\le j\le m} \sigma_{j+1} \cdot t^{j+1}] - \omega(t)}{\omega(t) \cdot t^{k+1}} dt \right| \\
&= \frac{k!}{2\pi} \cdot \left| \int_\Gamma f(t) \cdot \frac{\sum_{0\le j\le k} \sigma_j \cdot t^j}{\omega(t) \cdot t^{k+1}} dt \right| \\
&\le \frac{k!}{2\pi} \cdot (2\pi\gamma) \cdot \max_{t\in\Gamma} \left| f(t) \cdot \frac{\sum_{0\le j\le k} \sigma_j \cdot t^j}{\omega(t) \cdot t^{k+1}} \right| \\
&\le k! \cdot \gamma \cdot \max_{t\in\Gamma} |f(t)| \cdot \frac{\max_{t\in\Gamma} \left|\sum_{0\le j\le k} \sigma_j \cdot t^j\right|}{\min_{t\in\Gamma} |\omega(t) \cdot t^{k+1}|}
\end{aligned}$$

Here $\max\{|f(t)| \mid t \in \Gamma\} = M$ is the constant defined at the beginning of this chapter. Furthermore for any $t \in \Gamma$: $|t - z_i| \ge |t| - |z_i| \ge \gamma - \varepsilon \ge \gamma/2$, so

$$\min_{t\in\Gamma} \left|\omega(t) \cdot t^{k+1}\right| \ge \min_{t\in\Gamma} |\omega(t)| \cdot \min_{t\in\Gamma} \left|t^{k+1}\right| \ge (\gamma/2)^{m+1} \cdot \gamma^{k+1}$$

By definition of $\omega(z) := \prod(z - z_i)$, each coefficient σ_j of ω is bounded by the corresponding coefficient of the polynomial

$$\prod_{0\le j\le m} (z+\varepsilon) = \sum_{0\le j\le m+1} \binom{m+1}{m+1-j} \cdot z^j \cdot \varepsilon^{m+1-j}$$

This implies

$$|\sigma_j| \le \binom{m+1}{m+1-j} \cdot \varepsilon^{m+1-j}$$

Using $\varepsilon < \gamma$, i.e. $\gamma/\varepsilon > 1$, we conclude

$$\begin{aligned}
\max_{t\in\Gamma} \left| \sum_{0\le j\le k} \sigma_j \cdot t^j \right| &\le \sum_{0\le j\le k} |\sigma_j| \cdot \gamma^j \\
&\le \sum_{0\le j\le k} \binom{m+1}{m+1-j} \cdot \varepsilon^{m+1-j} \cdot \gamma^j \\
&= \varepsilon^{m+1} \cdot \sum_{0\le j\le k} \binom{m+1}{m+1-j} \cdot \left(\frac{\gamma}{\varepsilon}\right)^j \\
&\le \varepsilon^{m+1} \cdot \frac{\gamma^k}{\varepsilon^k} \cdot \sum_{0\le j\le k} \binom{m+1}{m+1-j} \\
&\le \varepsilon^{m+1-k} \cdot \gamma^k \cdot 2^{m+1}
\end{aligned}$$

This leads to

$$\left|\frac{1}{k!}P^{(k)}(0) - a_k\right| = \frac{1}{k!}\left|P^{(k)}(0) - f^{(k)}(0)\right| \;\leq\; \frac{1}{k!}\cdot k!\cdot\gamma\cdot M\cdot\frac{\varepsilon^{m+1-k}\cdot\gamma^k\cdot 2^{m+1}}{(\gamma/2)^{m+1}\cdot\gamma^{k+1}}$$
$$= M\cdot\varepsilon^{m+1-k}\cdot\frac{4^{m+1}}{\gamma^{m+1}}$$

□

In order to allow a fast evaluation of $P^{(k)}(0)$, we now choose special points z_i. Since we want results that also hold for functions f that are given only on the real line, we will only use *real* points z_i. Furthermore we choose $m = 2k$. In fact we use the Lagrangian representation of interpolating polynomials: For any $h \in \mathbb{R}$, $h>0$, and any $i \in \mathbb{Z}$, $0 \leq i \leq 2k$, let $z_i := (i-k)\cdot h$. Later we will choose h such that $k\cdot h \leq \gamma/2$ corresponding to the variable ε in Lemma 2.2. We define

$$L_{k,i}(z) := \prod_{0\leq j\leq 2k\,,\,j\neq i}\frac{z - z_j}{z_i - z_j}$$

so the interpolating polynomial P can now be written as

$$P(z) = \sum_{i=0}^{2k} f(z_i)\cdot L_{k,i}(z).$$

Please note that from now on P implicitly depends on the parameters h and k. The value $P^{(k)}(0)$ obviously is a linear combination of the $2k{+}1$ values of f at the abscissae z_i:

$$P^{(k)}(0) = \sum_{i=0}^{2k} f(z_i)\cdot L_{k,i}^{(k)}(0)$$

In the following lemma we analyse this formula.

2.3 Lemma. For any $k > 0$ there are integers $l_{k,i}$ $(0 \leq i \leq 2k)$ such that

(a)

$$\frac{1}{k!}\cdot P^{(k)}(0) = \sum_{i=0}^{2k} f(z_i)\cdot\frac{h^{-k}\cdot l_{k,i}}{i!\cdot(2k-i)!}$$

(b)

$$\frac{|l_{k,i}|}{i!\cdot(2k-i)!} \leq 2^{7k-k\,\mathrm{ld}\,k}$$

(c) The binary representations of *all* the coefficients $l_{k,i}$ for $0\leq i\leq 2k$ can be computed in $c\cdot k^2\cdot\mathcal{M}(k\,\mathrm{ld}\,k)$ steps using an ordinary multitape Turing machine (for some constant $c \in \mathbb{N}$).

□

Proof.
For any $k, i \in \mathbb{N}$, $i \leq 2k$, define a polynomial $B_{k,i}$ of degree $2k$ and the corresponding coefficients $b_{k,i,\nu} \in \mathbb{Z}$ through the following identity:

$$B_{k,i}(x) = \sum_{\nu=0}^{2k} b_{k,i,\nu}\cdot x^\nu := \prod_{0\leq j\leq 2k\,,\,j\neq i}(x + (k-j))$$

(a) $B_{k,i}$ and $L_{k,i}$ are related as follows:

$$\begin{aligned}
L_{k,i}(z) &= \prod_{0\le j\le 2k\,,\,j\ne i} \frac{z - z_j}{z_i - z_j} \\
&= \prod_{0\le j\le 2k\,,\,j\ne i} \frac{z - (j-k)\cdot h}{(i-j)\cdot h} \\
&= \prod_{0\le j\le 2k\,,\,j\ne i} \frac{z/h + (k-j)}{(i-j)} \\
&= \frac{(-1)^i}{i!\cdot(2k-i)!} \cdot \prod_{0\le j\le 2k\,,\,j\ne i} (z/h + (k-j)) \\
&= \frac{(-1)^i}{i!\cdot(2k-i)!} \cdot B_{k,i}(z/h)
\end{aligned}$$

Hence

$$L_{k,i}^{(k)}(z) = \frac{(-1)^i}{i!\cdot(2k-i)!} \cdot h^{-k} \cdot B_{k,i}^{(k)}(z/h)$$

Obviously

$$B_{k,i}^{(k)}(x) = \sum_{\nu=0}^{k} \frac{(\nu+k)!}{\nu!} b_{k,i,k+\nu} \cdot x^{\nu}$$

and so

$$B_{k,i}^{(k)}(0) = k! \cdot b_{k,i,k}$$

Combining these results we get

$$\begin{aligned}
L_{k,i}^{(k)}(0) &= \frac{(-1)^i}{i!\cdot(2k-i)!} \cdot h^{-k} B_{k,i}^{(k)}(0) \\
&= \frac{(-1)^i}{i!\cdot(2k-i)!} \cdot h^{-k} \cdot k! \cdot b_{k,i,k}
\end{aligned}$$

So part (a) holds with $l_{k,i} := (-1)^i \cdot b_{k,i,k}$ and $P^{(k)}(0) = \sum_{i=0}^{2k} f(z_i) \cdot L_{k,i}^{(k)}(0)$.

(b) The polynomial $B_{k,i}$ is a product of $2k$ terms of the form $x + c$, where always $|c| \le k$ holds. So the absolute values of the coefficients of $B_{k,i}$ are bounded by the corresponding coefficients of the polynomial

$$\prod_{j=1}^{2k} (x + k) = (x + k)^{2k} = \sum_{\nu=0}^{2k} \binom{2k}{\nu} \cdot k^{\nu} \cdot x^{2k-\nu},$$

Especially, using $x = 1$, $\sum_{\nu=0}^{2k} |b_{k,i,\nu}| \le (1+k)^{2k}$ holds. For $\nu = k$ and $x = 1$ we get $|l_{k,i}| = |b_{k,i,k}| \le \binom{2k}{k} \cdot k^k$. Using the lower bound $k! \ge 3^{1-k}k^k$ (e.g. [He81], page 65) we conclude

$$\begin{aligned}
\frac{|l_{k,i}|}{i!\cdot(2k-i)!} &\le \binom{2k}{k} \cdot \frac{k^k}{i!\cdot(2k-i)!} = \binom{2k}{i} \cdot \frac{k^k}{k!\cdot k!} \\
&\le 2^{2k} \cdot \frac{k^k \cdot 3^{2k-2}}{k^{2k}} = \frac{2^{2k}\cdot 3^{2k}}{9k^k} \le 2^{7k} \cdot 2^{-k\,\mathrm{ld}\,k}
\end{aligned}$$

Better estimates for $k!$ like Stirling's formula would only improve the factor 2^{7k} but not the essential factor k^{-k}.

(c) For $\mu \in \mathbb{N}$ define the polynomial $\bar{B}_\mu$ of degree $2\mu+1$ and its coefficients $\bar{b}_{\mu,\nu}$ through the identity

$$\bar{B}_\mu(x) = \sum_{\nu=0}^{2\mu+1} \bar{b}_{\mu,\nu} \cdot x^\nu \quad := \quad \prod_{0\le j\le 2\mu} (x + (\mu - j))$$
$$= \prod_{-\mu\le j\le \mu} (x+j) = x \cdot \prod_{1\le j\le \mu} (x^2 - j^2)$$

Then obviously $\bar{b}_{\mu,\nu} = 0$ for any even ν. Since $\bar{B}_\mu(x) = \bar{B}_{\mu-1}(x)\cdot(x^2-\mu^2)$, we are able to determine the coefficients $\bar{b}_{\mu,\nu}$ for odd ν using the following recursion (where $\bar{b}_{\mu,-1} := 0$).

$$\bar{b}_{\mu,2\mu+1} = 1$$

$$\bar{b}_{\mu,\nu} = \bar{b}_{\mu-1,\nu-2} - \mu^2 \cdot \bar{b}_{\mu-1,\nu} \text{ for} \nu = 1, 3, 5, \ldots, 2\mu+1$$

Since $\bar{B}_k(x) = (x + (k-i)) \cdot B_{k,i}(x)$, the coefficients $b_{k,i,\nu}$ are determined by the following recursion:

$$b_{k,i,2k} = \bar{b}_{k,2k+1} = 1$$

$$b_{k,i,\nu} = \bar{b}_{k,\nu+1} - (k-i) \cdot b_{k,i,\nu+1} \text{ for } \nu = 2k-1, 2k-2, 2k-3, \ldots, 0$$

As in (b) we are able to compare the coefficients of $\bar{B}_\mu$ with the corresponding coefficients of $(x+k)^{2k+1}$, which shows that $\bar{b}_{\mu,\nu}$ is bounded by $(1+k)^{2k+1}$ for any $\mu \le k$, $\nu \le 2\mu+1$. So the binary representations of $\bar{b}_{\mu,\nu}$ and $b_{k,i,\nu}$ each consist of at most $(2k+1)\,\mathrm{ld}\,(k+1)$ digits.

Using the recursions to compute the $2k+2$ values $\bar{b}_{k,\nu}$ at most $c_1k^2 + c_1$ additions and multiplications are sufficient. To compute *all* the $b_{k,i,k}$ from the $\bar{b}_{k,\nu}$ another $c_2k^2 + c_2$ additions and multiplications suffice (for appropriate constants c_1, c_2). Due to the bound on the length of the binary representations, each of these operations can be carried out in at most $\mathcal{O}(\mathcal{M}((2k+1)\cdot \mathrm{ld}\,(k+1)))$ steps by a multitape Turing machine. Since $\mathcal{M}$ is regular, we are able to simplify this result yielding $\mathcal{O}(k^2 \cdot \mathcal{M}(k\,\mathrm{ld}\,k))$ steps as an upper bound for the computation of *all* the $2k+1$ integers $l_{k,i} = (-1)^i \cdot b_{k,i,k}$ from k. □

With these prerequisites we are able to show the main result of this chapter.

2.4 Theorem.
Let $f\colon \mathbb{C} \dashrightarrow \mathbb{C}$ be analytic in 0, let $2^{-\tau}$ be less than the radius of convergence of the corresponding power series and let $2^\sigma \ge \max\{|f(t)| \mid |t| = 2^{-\tau}\}$ for two natural numbers $\sigma, \tau > 0$. Then any coefficient a_k of the Taylor series can be approximated with an error not exceeding 2^{-n} by evaluating f at $2k+1$ points from the real interval $\{x \in \mathbb{R} \mid |x| \le 2^{-\tau}\}$ with a precision of $2n+2k\tau + 13k + \sigma + 6$ digits. The computational complexity of approximating a_k is bounded by $\mathcal{O}(\,(k+1)\cdot\mathcal{M}(n+k\tau+\sigma) + k^2\cdot\mathcal{M}(k\,\mathrm{ld}\,k)\,)$ *plus* the time necessary for the $2k+1$ evaluations of f. □

Proof.
To approximate $a_0 = f(0)$ with precision 2^{-n} we simply compute f with corresponding precision. To approximate a_k for a given $k > 0$ and with an error not exceeding 2^{-n}, we compute an approximation $\bar{a}_{k,n}$ to $1/k!\cdot P^{(k)}(0)$ as follows: As in Lemma 2.3, we use $2k+1$ points $z_i := h(i-k)$, $0\le i\le 2k$, to interpolate f, where h is such that $\log_2 h = -\,\mathrm{ld}\,k - \lceil\frac{n+\sigma+1}{k+1}\rceil - 2\tau - 4$. This implies $hk \le 2^{-\tau-1}$, so each z_i lies in the set $\{x \in \mathbb{R} \mid |x| \le 2^{-\tau}\}$. Choosing $\gamma := 2^{-\tau}$, $M := \max\{|f(t)| \mid |t| = 2^{-\tau}\} \le 2^\sigma$, $m := 2k$, and $\varepsilon := hk$, we are able to use Lemma 2.2 to control the interpolating error. We compute $\bar{a}_{k,n}$ as follows:

1. Compute the following natural numbers:
$\eta := \text{ld}\, k + \lceil \frac{n+\sigma+1}{k+1} \rceil + 2\tau + 4$,
$\nu_1 := n + 2$,
$\nu_2 := \nu_1 + 1 + \text{ ld}\,(2k{+}1)$,
$\nu_3 := \nu_2 + 2 + n + 2k\tau + 12k + \sigma$,
$\nu_4 := \nu_2 + \sigma + 2$.

2. For any i, $0{\le}i{\le}2k$, compute the integers $l_{k,i}$, and $i! \cdot (2k{-}i)!$ (according to Lemma 2.3).

3. For any i, $0{\le}i{\le}2k$, compute $\bar{q}_i :\approx_{(\nu_4)} \dfrac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!}$.

4. For any i, $0{\le}i{\le}2k$, compute $\bar{f}_i :\approx_{(\nu_3)} f(z_i)$, where $z_i := (i{-}k) \cdot 2^{-\eta}$.

5. For any i, $0{\le}i{\le}2k$, compute $\bar{p}_i :\approx_{(\nu_2)} \bar{f}_i \cdot \bar{q}_i$.

6. Compute $\bar{a}_{k,n} :\approx_{(\nu_1)} \sum_{i=0}^{2k} \bar{p}_i$ and output $\bar{a}_{k,n}$.

We begin the analysis of the algorithm considering the values $\bar{p}_i$. First note that $|\bar{f}_i| \le |f(z_i)| + 2^{-\nu_3} \le M + 1 \le 2^{\sigma+1}$, since the maximum value of an analytic function on a circle in its domain is always found on the border of the circle.

$$\begin{aligned}
\left|\bar{p}_i - f(z_i) \cdot \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!}\right| &\le |\bar{p}_i - \bar{f}_i \cdot \bar{q}_i| + \left|\bar{f}_i \cdot \bar{q}_i - f(z_i) \cdot \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!}\right| \\
&\le 2^{-\nu_2} + \left|\bar{f}_i \cdot (\bar{q}_i - \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!})\right| \\
&\quad + \left|(f(z_i) - \bar{f}_i) \cdot \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!}\right| \\
&\le 2^{-\nu_2} + 2^{\sigma+1} \cdot 2^{-\nu_4} + 2^{-\nu_3} \cdot 2^{k\eta} \cdot 2^{7k - k\,\text{ld}\,k} \\
&= 2^{-\nu_2} + 2^{\sigma+1} \cdot 2^{-(\nu_2+\sigma+2)} \\
&\quad + 2^{-(\nu_2+2+n+2k\tau+12k+\sigma)} \cdot 2^{k\eta} \cdot 2^{7k - k\,\text{ld}\,k} \\
&= 2^{-\nu_2} + 2^{-(\nu_2+1)} + 2^{-(\nu_2+2+n+k+\sigma)+k\lceil \frac{n+\sigma+1}{k+1} \rceil} \\
&\le 2^{1-\nu_2}
\end{aligned}$$

Then

$$\begin{aligned}
\left|\bar{a}_{k,n} - \sum_{i=0}^{2k} f(z_i) \cdot \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k{-}i)!}\right| &\le \left|\bar{a}_{k,n} - \sum_{i=0}^{2k} \bar{p}_i\right| \\
&\quad + \left|\sum_{i=0}^{2k} \bar{p}_i - \sum_{i=0}^{2k} f(z_i) \cdot \frac{2^{k\eta} \cdot l_k, i}{i! \cdot (2k{-}i)!}\right| \\
&\le 2^{-\nu_1} + (2k{+}1) \cdot 2^{1-\nu_2} \\
&= 2^{-\nu_1} + (2k{+}1) \cdot 2^{1-(\nu_1+1+\text{ld}\,(2k+1))} \\
&\le 2^{1-\nu_1}
\end{aligned}$$

With the choice of $h = 2^{-\eta}$, $\gamma := 2^{-\tau}$, $M := \max\{|f(t)| \mid |t| = 2^{-\tau}\} \le 2^{\sigma}$, $m := 2k$, and $\varepsilon := hk$, we conclude using Lemmata 2.2 and 2.3(a):

$$\begin{aligned}
|a_k - \bar{a}_{k,n}| &\leq \left|a_k - 1/k! \cdot P^{(k)}(0)\right| + \left|\bar{a}_{k,n} - 1/k! \cdot P^{(k)}(0)\right| \\
&\leq M \cdot \varepsilon^{m+1-k} \cdot \frac{4^{m+1}}{\gamma^{m+1}} + \left|\bar{a}_{k,n} - \sum_{i=0}^{2k} f(z_i) \cdot \frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k-i)!}\right| \\
&\leq 2^{\sigma} \cdot (hk)^{k+1} \cdot \frac{4^{2k+1}}{2^{-\tau(2k+1)}} + 2^{1-\nu_1} \\
&\leq 2^{\sigma} \cdot 2^{(\operatorname{ld} k - \eta)(k+1)} \cdot 2^{4k+2} \cdot 2^{\tau(2k+1)} + 2^{1-(n+2)} \\
&\leq 2^{-(n+1)} + 2^{-(n+1)} \\
&= 2^{-n}
\end{aligned}$$

To determine the complexity of the algorithm we study its six steps in detail: Please note that $k, \tau, \sigma > 0$, so we are able to hide any additive constants, e.g. $k\tau + c \leq (c+1) \cdot k\tau$. This allows significant easier formulations of the complexity bounds.

- The complexity of step 1. is at most $c_1 \cdot (n+k+\sigma+\tau)$.
- The complexity of step 2. is $c_2 \cdot k^2 \mathcal{M}(k \operatorname{ld} k)$ due to Lemma 2.3.
- Let $d_i := \operatorname{ld} l_{k,i}$ and $e_i = \operatorname{ld}(i! \cdot (2k-i)!)$. As in the proof of Lemma 1.2 we normalize $l_{k,i}$ and $i! \cdot (2k-i)!$ prior to the computation of $\bar{q}_i$: Instead of dividing $2^{k\eta} \cdot l_{k,i}$ and $i! \cdot (2k-i)!$ directly, we divide $2^{-d_i} \cdot l_{k,i}$ and $2^{-e_i} \cdot i! \cdot (2k-i)!$ with a precision of $\nu_4 - d_i + e_i - k\eta$ digits and then multiply the result by $2^{d_i - e_i + k\eta}$ (which is done by a mere shift of the decimal point). By Lemma 2.3(b), $d_i - e_i \leq 7k - k \operatorname{ld} k + 2$, so $d_i - e_i + k\eta \leq 7k - k \operatorname{ld} k + 2 + k(\operatorname{ld} k + \lceil \frac{n+\sigma+1}{k+1} \rceil + 2\tau + 4) \leq n + 2k\tau + 12k + \sigma + 3$. So the $2k+1$ divisions in step 3. can be completed in time $(2k+1) \cdot \mathcal{M}(n+2k\tau+12k+\sigma+3+\nu_4)$. Since $\mathcal{M}$ is regular, this complexity is bounded by $c_3 \cdot (k+1) \cdot \mathcal{M}(n + k\tau + \sigma)$.
- By definition, $\nu_3 = 2n + 2k\tau + 12k + \sigma + \operatorname{ld}(2k+1) + 5$, so $\nu_3 \leq 2n + 2k\tau + 13k + \sigma + 6$. The complexity of computing the z_i is bounded by $\mathcal{O}(\,(2k+1)\eta\,) \subseteq \mathcal{O}(\,(k+1)\mathcal{M}(n+k\tau+\sigma)\,)$.
- As shown above, $|\bar{f}_i| \leq 2^{\sigma+1}$. Moreover by Lemma 2.3(b) and similar to the treatment of step 3., $|\bar{q}_i| \leq \left|\frac{2^{k\eta} \cdot l_{k,i}}{i! \cdot (2k-i)!}\right| + 2^{-\nu_4} \leq 2^{k\eta + 7k - k \operatorname{ld} k} + 2^{-\nu_4} \leq 2^{n+1+2k\tau+12k+\sigma} + 2^{-\nu_4} \leq 2^{n+2+2k\tau+12k+\sigma}$. By Lemma 1.2, the complexity of one of the multiplications is bounded by $\mathcal{M}(n+2+2k\tau+12k+\sigma+\nu_2)$. So step 5. can be completed in time $c_5 \cdot (k+1) \cdot \mathcal{M}(n + k\tau + \sigma)$.
- From the preceding considerations we may conclude $|\bar{p}_i| \leq |\bar{f}_i \cdot \bar{q}_i| + 2^{-\nu_2} \leq 2^{\sigma+1} \cdot 2^{n+2+2k\tau+12k+\sigma} + 2^{-\nu_2}$. Again using Lemma 1.2, the complexity of the summation is bounded by $c_6 \cdot (k+1) \cdot (n+k\tau+\sigma)$.

Summing up the complexities of these steps proves the time bound given in the theorem. □

In the above result, the overhead $\mathcal{O}(\,(k+1) \cdot \mathcal{M}(n+k\tau+\sigma) + k^2 \cdot \mathcal{M}(k \operatorname{ld} k)\,)$ will often be dominated by the complexity of the evaluations of f. Moreover for a fixed f, σ and τ can be treated as constants depending on f, leading a simplified version of our main result.

2.5 Corollary.
Let $f\colon \mathbb{C} \dashrightarrow \mathbb{C}$ be analytic in 0. If f is computable in time $t\colon \mathbb{N} \to \mathbb{N}$ on an arbitrary real neighbourhood of 0, where t is a regular time bound satisfying $k \cdot \mathcal{M}(k \operatorname{ld} k) \in \mathcal{O}(\,t(k)\,)$, then the sequence $(a_k)_{k \in \mathbb{N}}$ can be computed in time $\mathcal{O}(\,(k+1) \cdot t(n+k)\,)$. □

Proof.
Using 2.4, the sequence $(a_k)_{k\in\mathbb{N}}$ is computable $\mathcal{O}(\,(k{+}1)\cdot\mathcal{M}(n{+}k\tau{+}\sigma){+}k^2\cdot\mathcal{M}(k\,\mathrm{ld}\,k){+}(2k{+}1)t(2n{+}2k\tau{+}13k+\sigma+6)\,)$. Treating τ and σ as constant, and using the regularity of t, this can be simplified yielding $\mathcal{O}(\,(k{+}1)\cdot\mathcal{M}(n{+}k)+k^2\cdot\mathcal{M}(k\,\mathrm{ld}\,k)+(k{+}1)t(n{+}k)\,)$. Since $k\cdot\mathcal{M}(k\,\mathrm{ld}\,k)\in\mathcal{O}(\,t(k)\,)$, both $(k{+}1)\mathcal{M}(n+k)$ and $k^2\mathcal{M}(k\,\mathrm{ld}\,k)$ are in $\mathcal{O}(\,(k{+}1)t(n{+}k)\,)$. □

This result can easily be generalized for multidimensional series, which are of interest for example in solving initial value problems [MuMo93].

2.6 Theorem.
Let $(a_{\bar{k}})_{\bar{k}\in\mathbb{N}^l}$ be an l-dimensional sequence of complex numbers satisfying

$$\sum_{k_1,\dots,k_l\in\mathbb{N}} |a_{k_1,\dots,k_l}|\cdot R^{k_1+\dots+k_l}\le\infty$$

for a real number $R>0$ and let $f\colon\mathbb{C}^l \dashrightarrow \mathbb{C}$ be the corresponding sum function, i.e. $f(x_1,\dots,x_l):=\sum_{k_1,\dots,k_l\in\mathbb{N}} a_{k_1,\dots,k_l}\cdot x_1^{k_1}\cdot\ldots\cdot x_l^{k_l}$. If f is computable in time t on an arbitrary neighbourhood of $(0,\dots,0)$, where t is a regular time bound satisfying $k\cdot\mathcal{M}(k\,\mathrm{ld}\,k)\in\mathcal{O}(\,t\,)$, then the sequence $(a_{\bar{k}})_{\bar{k}\in\mathbb{N}^l}$ can be computed in time

$$\mathcal{O}(\prod_{1\le i\le l}(k_i{+}1)\cdot t(n{+}\sum_{1\le i\le l}k_i)\,)$$

□

Proof.
We only consider the two-dimensional case, that can easily be generalized for arbitrary l.

Let $(a_{k_1,k_2})_{k_1,k_2\in\mathbb{N}}$, $R>0$ and positive, natural numbers σ, τ be given such that $R>2^{-\tau}$ and

$$\sum_{k_1,k_2\in\mathbb{N}}|a_{k_1,k_2}|\cdot R^{k_1+k_2}\le 2^\sigma$$

Consider $f\colon\mathbb{C}^2\dashrightarrow\mathbb{C}$ defined by $f(x_1,x_2):=\sum_{k_1,k_2\in\mathbb{N}} a_{k_1,k_2}\cdot x_1^{k_1}\cdot x_2^{k_2}$.

For any given x_2 with $|x_2|\le 2^{-\tau}$, we may rearrange the series yielding a power series in x_1:

$$f(x_1,x_2)=\sum_{k_1\in\mathbb{N}} x_1^{k_1}\cdot\left(\sum_{k_2\in\mathbb{N}} a_{k_1,k_2}x_2^{k_2}\right)$$

where

$$\sum_{k_1\in\mathbb{N}} R^{k_1}\cdot\left|\sum_{k_2\in\mathbb{N}} a_{k_1,k_2}x_2^{k_2}\right|\le\sum_{k_1\in\mathbb{N}}R^{k_1}\cdot\sum_{k_2\in\mathbb{N}}|a_{k_1,k_2}|\,R^{k_2}\le 2^\sigma$$

So for any fixed $\tilde{x_2}$, $|\tilde{x_2}|\le 2^{-\tau}$, the function $g_{\tilde{x_2}}(x_1):=f(x_1,\tilde{x_2})$ is analytic in 0, where the corresponding Taylor series has a radius of convergence of at least $R>2^{-\tau}$. In addition

$$\max\{|g_{\tilde{x_2}}(x_1)|\mid|x_1|\le R\}\le\max\{|f(x_1,x_2)|\mid|x_1|\le R\wedge|x_2|\le R\}\le 2^\sigma.$$

For any given $\tilde{k_1}$ the function $h_{\tilde{k_1}}(x_2) := \sum_{k_2 \in \mathbb{N}} a_{\tilde{k_1},k_2} x_2^{k_2}$ is a power series, too. The radius of convergence again is at least R, although now the maximum value of the sum may vary with $\tilde{k_1}$:

$$
\begin{aligned}
\max\{\left|h_{\tilde{k_1}}(x_2)\right| \mid |x_2| \leq R\} &\leq \sum_{k_2 \in \mathbb{N}} \left|a_{\tilde{k_1},k_2}\right| R^{k_2} \\
&\leq R^{-\tilde{k_1}} \cdot \sum_{k_2 \in \mathbb{N}} \left|a_{\tilde{k_1},k_2}\right| R^{\tilde{k_1}+k_2} \\
&\leq R^{-\tilde{k_1}} 2^{\sigma} \\
&\leq 2^{\sigma+\tilde{k_1}\tau}
\end{aligned}
$$

So in applying Theorem 2.4 on $h_{\tilde{k_1}}$, we must replace σ by $\sigma+\tilde{k_1}\tau$.

Now to approximate $a_{\tilde{k_1},\tilde{k_2}}$ for given $\tilde{k_1}$ and $\tilde{k_2}$ with a precision of n digits, we compute the $\tilde{k_2}$-th coefficient of the series $\sum_{k_2 \in \mathbb{N}} a_{\tilde{k_1},k_2} x_2^{k_2}$. By Theorem 2.4, it is sufficient to evaluate this series at $2\tilde{k_2}+1$ different points $x_{2,i}$ with a precision of $p(n, \tilde{k_1}, \tilde{k_2}) := 2n+2\tilde{k_2}\tau+13\tilde{k_2}+\sigma+\tilde{k_1}\tau+6$ digits. Each of these evaluations can be done by a computation of the $\tilde{k_1}$-th coefficient of the power series belonging to $g_{x_{2,i}}$. This can be done by evaluating $g_{x_{2,i}}$ at $2\tilde{k_1}+1$ points $x_{1,j}$ (i.e. we evaluate f at $(x_{1,j}, x_{2,i})$) with a precision of now $2p(n, \tilde{k_1}, \tilde{k_2})+2\tilde{k_1}\tau+13\tilde{k_1}+\sigma+6$ digits.

Now suppose that f is computable in time t on the set $\{(x_1, x_2) \in \mathbb{C}^2 \mid |x_1|, |x_2| \leq 2^{-\tau}\}$ for a regular t. All in all, $(2\tilde{k_1}+1)\cdot(2\tilde{k_2}+1)$ evaluations of f with a precision of $4n+4\tilde{k_2}\tau+26\tilde{k_2}+4\tilde{k_1}\tau+13\tilde{k_1}+3\sigma+18$ digits are sufficient. Since σ and τ are fixed and t is regular, this is $\mathcal{O}(\,(\tilde{k_1}+1)(\tilde{k_2}+1)t(n+\tilde{k_1}+\tilde{k_2})\,)$. Further suppose $k \cdot \mathcal{M}(k \,\mathrm{ld}\, k) \in \mathcal{O}(\,t\,)$. Then the overhead to the evaluations of f does not violate the asymptotic bound. □

3 Computing the sum function of a series

In [Mu87] we showed that the summation of a power series $\sum a_k \cdot (x-x_0)^k$ is possible for real arguments x with $|x-x_0| \leq R/2$, where R is a lower bound for the radius of convergence in x_0. We now generalize this result to complex numbers z (which is immediate) and especially consider the influence of the ratio $|x-x_0|/R$ on the complexity of the summation process. For simplicity we choose $x_0 = 0$.

3.1 Lemma. Let $(a_k)_{k\in\mathbb{N}}$ be a sequence of complex numbers, let $R, M \in \mathbb{R}$, $R, M > 0$, be given such that $\sum_{k\in\mathbb{N}} |a_k|\, R^k \leq M$. For any $Q \in \mathbb{R}$, $Q > 1$, and for any $n \in \mathbb{N}$ let

$$m(Q, n) := \lceil \frac{n + \log_2 M + 1}{\log_2 Q} \rceil.$$

Then for any $z \in \mathbb{C}$ with $|z| \leq R/Q$

$$\sum_{k\in\mathbb{N}, k\geq m(Q,n)} a_k z^k \leq 2^{-n-1}$$

□

Proof.
If $|z| \leq R/Q$ and $k \geq m(Q, n)$, then $|z|/R \leq 1/Q < 1$ and

$$|z|^k = R^k \cdot \left(\frac{|z|}{R}\right)^k \leq R^k \cdot \left(\frac{|z|}{R}\right)^{m(Q,n)}.$$

So

$$\begin{aligned}\left|\sum_{k\geq m(Q,n)} a_k z^k\right| &\leq \sum_{k\geq m(Q,n)} |a_k|\cdot\left|z^k\right| \\ &\leq \left(\frac{|z|}{R}\right)^{m(Q,n)}\cdot\sum_{k\geq m(Q,n)}|a_k|\cdot R^k \\ &\leq Q^{-m(Q,n)}\cdot M \\ &\leq 2^{-n-1}\end{aligned}$$

□

Using this result, we are able to compute the sum function of the power series, given its coefficients and the parameters R and M, for any z with $|z| < R$.

Since the disc $\{z \mid |z| < R\}$ is not a compact set, we can not expect to find an upper bound for the complexity of the computation of the sum that holds for any z from this disk. The lemma above indicates, that the computation gets arbitrarily hard as z approaches the border of the disk.

In consequence, we formulate the complexity of the summation not only depending on n, but also on M, R, and Q. Nevertheless, M, R, and Q will not be inputs to the algorithm below, but rather be something like a 'constant'. So it is necessary to use finite representations of the values.

R should obviously be choosen as large as possible, but R must be smaller than the radius of convergence of the series. For simplicity we choose $R := 2^{-\tau}$ for a dyadic decimal τ with a finite representation. On one hand, this should be sufficient to approximate the radius of convergence, while on the other hand the complexity of any necessary computations on τ are very easy to estimate. The same holds for Q, where we also choose $Q := 2^q$ for a dyadic decimal $q > 0$ with a finite representation. Here $q >> 0$ corresponds to a small spot in the center of the disk; $q \approx 0$ corresponds to the whole disk except a small ring at its border. M might be any upper bound for $\sum_{k\in\mathbb{N}} |a_k|\, R^k$, so it is sufficient to choose $M := 2^\sigma$ for even a natural number σ.

In the following $\tau + q$ might be positive (so $2^{-\tau-q} < 1$) or τ might be negative (corresponding to a large R). In consequence sometimes it would be close at hand to evaluate certain formulae with a precision of 2^{-n} for a *negative* n! Because our computational model introduced in section 1 is only defined for $n \geq 0$, we choose to introduce the symbol $[\![x]\!] := \min\{i \in \mathbb{N} \mid i \geq x\}$ to compensate this.

3.2 Theorem. Let $(a_k)_{k\in\mathbb{N}}$ be a sequence of complex numbers and let $f(z) := \sum a_k z^k$. Let a natural number σ and dyadic decimals τ, $q > 0$ with finite representations of length $\leq \ell$ be given such that $\sum_{k\in\mathbb{N}} |a_k|\, 2^{-k\tau} \leq 2^\sigma$. Then for any z such that $|z| \leq 2^{-\tau-q}$, $f(z)$ can be approximated with an error not exceeding 2^{-n} using approximations to the first m coefficients a_k with a precision of $[\![4 + n + \operatorname{ld} m - k(\tau + q)]\!]$ each. Here $m := \lceil \frac{1}{q}\cdot(n + \sigma + 1)\rceil$. The computational complexity of approximating $f(z)$ is bounded by $\mathcal{O}(\, m\cdot\mathcal{M}(n + m + \sigma + m[\![\tau + q]\!]) + m\cdot(\operatorname{ld} n + \operatorname{ld}\sigma + \ell)\,)$ *plus* the complexity of evaluating the coefficients a_k. □

Proof.
Consider the following Turing machine M computing a value $\bar{f}$ from inputs $z \in \mathbb{C}$ and $n \in \mathbb{N}$ (where z is supposed to be given as a pair of dyadic decimals). M consists of five parts executed in sequence:

(1) M computes the following natural numbers:

$$\begin{aligned} m &:= \lceil \frac{1}{q} \cdot (n+\sigma+1) \rceil \\ \nu_1 &:= n+2 \\ \nu_2 &:= \nu_1 + 1 + \operatorname{ld} m \\ \nu_3 &:= 1+\nu_2 \\ \nu_4 &:= \nu_2 + m + 3 \\ \nu_5 &:= [\![\nu_4 + \sigma + \tau + q + m + 1]\!] \end{aligned}$$

(2) M computes $\bar{z}_0 := 1$, $\bar{z}_1 :\approx_{(\nu_5)} z$ and

$$\bar{z}_k :\approx_{(\nu_{4,k})} \bar{z}_1 \cdot \bar{z}_{k-1}$$

for any k, $2 \le k < m$, where $\nu_{4,k} := [\![\nu_4 + k(\tau + q - 1) + \sigma]\!]$.

(3) For any k, $0 \le k < m$, M computes

$$\bar{a}_k :\approx_{(\nu_{3,k})} a_k,$$

where $\nu_{3,k} := [\![\nu_3 - k(\tau+q)]\!]$.

(4) For any k, $0 \le k < m$, M computes

$$\bar{p}_k :\approx_{(\nu_2)} \bar{a}_k \cdot \bar{z}_k.$$

(5) Finally M computes

$$\bar{f} :\approx_{(\nu_1)} \sum_{k=0}^{m-1} \bar{p}_k$$

as the output of the algorithm.

In the following suppose $z \in \mathbb{C}$ is given such that $|z| \le 2^{-\tau-q}$. A simple inductive argument shows $\left|\bar{z}_k - z^k\right| \le 2^{2-\nu_4-k(\tau+q-1)-\sigma}$ for any k, $0 \le k < m$: This inequality is obviously valid for $k \le 1$. For $k > 1$ (and $m>k>1$) suppose $\left|\bar{z}_{k-1} - z^{k-1}\right| \le 2^{2-\nu_4-(k-1)(\tau+q-1)-\sigma}$. Then:

$$\begin{aligned} \left|\bar{z}_k - z^k\right| &\le |\bar{z}_k - \bar{z}_1 \cdot \bar{z}_{k-1}| + \left|\bar{z}_1 \cdot \bar{z}_{k-1} - \bar{z}_1 \cdot z^{k-1}\right| \\ &\quad + \left|\bar{z}_1 \cdot z^{k-1} - z^k\right| \\ &\le 2^{-\nu_{4,k}} + (|z| + |\bar{z}_1 - z|) \cdot \left|\bar{z}_{k-1} - z^{k-1}\right| \\ &\quad + |\bar{z}_1 - z| \cdot \left|z^{k-1}\right| \\ &\le 2^{-\nu_{4,k}} + (2^{-\tau-q} + 2^{-\nu_5}) \cdot 2^{2-\nu_4-(k-1)(\tau+q-1)-\sigma}) \\ &\quad +2^{-\nu_5} \cdot 2^{-(k-1)(\tau+q)} \\ &\le 2^{-\nu_{4,k}} \\ &\quad +2^{1-\nu_4-k(\tau+q-1)-\sigma} + 2^{-\nu_4-\sigma-\tau-q-m-1} \cdot 2^{2-\nu_4-(k-1)(\tau+q-1)-\sigma}) \\ &\quad +2^{-\nu_4-\sigma-\tau-q-m-1} \cdot 2^{-(k-1)(\tau+q)} \\ &\le 2^{-[\![\nu_4 - k(\tau+q-1)-\sigma]\!]} + 2^{1-\nu_4-k(\tau+q-1)-\sigma} + 2^{-2\nu_4-m-k(\tau+q-1)-2\sigma-1} \\ &\quad +2^{-\nu_4-\sigma-m-k(\tau+q)-1} \\ &\le 2^{-\nu_4-k(\tau+q-1)-\sigma} + 2^{1-\nu_4-k(\tau+q-1)-\sigma} \\ &\quad +2^{-\nu_4-\sigma-k(\tau+q-1)-1} + 2^{-\nu_4-k(\tau+q-1)-\sigma-1} \\ &= 2^{2-\nu_4-k(\tau+q-1)-\sigma} \end{aligned}$$

Since $\sum |a_k| 2^{-k\tau} \leq 2^\sigma$, especially $|a_k| \leq 2^{\sigma-k\tau}$ holds. So $|\bar a_k| \leq |a_k| + 2^{-\nu_{3,k}} \leq 2^{\sigma+k\tau} + 2^{-\lceil \nu_3 - k(\tau+q)\rceil} \leq 2^{\sigma+k\tau} + 2^{-\nu_3+k(\tau+q)} \leq 2^{\sigma+k\tau} + 2^{k(\tau+q)} \leq 2^{1+\sigma+k(\tau+q)}$. The last inequation holds since σ, k and q are non-negative. We conclude using $k < m$

$$\begin{aligned}
\left|\bar p_k - a_k z^k\right| &\leq |\bar p_k - \bar a_k \bar z_k| + \left|\bar a_k \bar z_k - a_k z^k\right| \\
&\leq 2^{-\nu_2} + \left|\bar a_k \bar z_k - a_k z^k\right| \\
&\leq 2^{-\nu_2} + |\bar a_k| \cdot \left|\bar z_k - z^k\right| + |\bar a_k - a_k| \cdot \left|z^k\right| \\
&\leq 2^{-\nu_2} + 2^{1+\sigma+k(\tau+q)} \cdot 2^{2-\nu_4-k(\tau+q-1)-\sigma} + 2^{-\nu_{3,k}} \cdot 2^{-k(\tau+q)} \\
&\leq 2^{-\nu_2} + 2^{3-\nu_4+k} + 2^{-\nu_3+k(\tau+q)} \cdot 2^{-k(\tau+q)} \\
&\leq 2^{-\nu_2} + 2^{3-\nu_4+k} + 2^{-\nu_3} \\
&\leq 2^{-\nu_2} + 2^{-\nu_2+k-m} + 2^{-\nu_3} \\
&\leq 2^{1-\nu_2}
\end{aligned}$$

Using Lemma 2.7 (with $R := 2^{-\tau}$, $M := 2^\sigma$, and $Q := 2^q > 1$) we have

$$\begin{aligned}
|\bar f - f(z)| &\leq \left|\bar f - \sum_{k=0}^{m-1} \bar p_k\right| + \left|\sum_{k=0}^{m-1} (\bar p_k - a_k z^k)\right| + \left|f(z) - \sum_{k=0}^{m-1} a_k z^k\right| \\
&\leq 2^{-\nu_1} + m \cdot 2^{1-\nu_2} + 2^{-n-1} \\
&= 2^{-n-2} + m \cdot 2^{1-n-2-1-\operatorname{ld} m} + 2^{-n-1} \\
&\leq 2^{-n}
\end{aligned}$$

So our Turing machine M indeed computes f on the set $\{z \in \mathbb{C} \mid |z| \leq 2^{-\tau-q}\}$.

To analyze the complexity of M we first determine a bound for the size of the operands z, $\bar z_k$, $\bar a_k$ and $\bar p_k$ used in (2), (4) and (5). As shown above, we have $|\bar a_k| \leq 2^{1+\sigma+k(\tau+q)}$. Also $|\bar p_k| \leq \left|a_k z^k\right| + 2^{1-\nu_2} \leq 2^{\sigma+k\tau} \cdot 2^{-k\tau} + 2^0 \leq 2^{1+\sigma}$. Finally $|\bar z_k| \leq \left|\bar z_k - z^k\right| + \left|z^k\right| \leq 2^{2-\nu_4-k(\tau+q-1)-\sigma} + 2^{-k\tau} \leq 2^{2-\nu_2-m-3-k(\tau+q-1)-\sigma} + 2^{-k(\tau+q)} \leq 2^{-k(\tau+q)} + 2^{-k(\tau+q)} \leq 2^{1-k(\tau+q)}$.

In consequence, each involved operand is bounded by $2^{1+\sigma+m[\![\tau+q]\!]}$.

The length ℓ of the representations of τ and q has influence upon the computation of m, ν_i and the $\nu_{i,j}$. Surely $c_1 \cdot m \cdot (\operatorname{ld} n + \operatorname{ld} \sigma + \ell)$ can be used as an upper bound for the complexity of part (1) and of the computation of all the $\nu_{i,j}$ in parts (2) and (3) together.

In part (2) we have $m-2$ multiplications. Using Lemma 1.2, each of these multiplications has a complexity of at most $\mathcal{O}(\mathcal{M}(\nu_{(4,k)} + 1 + \sigma + m[\![\tau+q]\!]))$. Here $\nu_{(4,k)} + 1 + \sigma + m[\![\tau+q]\!] = [\![\nu_4 + k(\tau+q-1) + \sigma]\!] + 1 + \sigma + m[\![\tau+q]\!] \leq \nu_4 + k[\![\tau+q]\!] + \sigma + 1 + \sigma + m[\![\tau+q]\!] \leq \nu_4 + 2m[\![\tau+q]\!] + 2\sigma + 1$. So the complexity of part (2) is bounded by $c_2 \cdot m \cdot \mathcal{M}(n + 6 + m + \operatorname{ld} m + 2\sigma + 2m[\![\tau+q]\!])$

In part (4) we have m multiplications. Again using Lemma 1.2, this part has a complexity of $c_4 \cdot m \cdot \mathcal{M}(\nu_2 + 1 + \sigma + m[\![\tau+q]\!]) = c_4 \cdot m \cdot \mathcal{M}(n + 4 + \operatorname{ld} m + \sigma + m\,|\tau+q|)$

Part (5) can be executed in $c_5 \cdot m \cdot (\nu_2 + 1 + \sigma + m[\![\tau+q]\!]) = c_5 \cdot m \cdot (n + 4 + \operatorname{ld} m + \sigma + m[\![\tau+q]\!])$ steps.

So the complexity of the algorithm is bounded by $\mathcal{O}(m \cdot \mathcal{M}(n + m + \sigma + m[\![\tau+q]\!]))$ plus the necessary evalutations of the a_k in part (3).

In part (3) each coefficient (a_k) has to be computed with an error of at most $2^{-\nu_{3,k}}$. Here $\nu_{3,k} = [\![\nu_3 - k(\tau+q)]\!] = [\![4 + n + \operatorname{ld} m - k(\tau+q)]\!]$. □

For a fixed computable sequence (a_k) and fixed values σ, τ and q, we are able to simplify the result:

3.3 Corollary. If a sequence $(a_k)_{k\in\mathbb{N}}$ is computable in time t for a regular t with $\mathcal{M} \in \mathcal{O}(t)$, then its sum function $f = \sum a_k z^k$ is computable in time $\mathcal{O}(n \cdot t(n))$ on any (fixed) compact subset of the circle of convergence. □

Proof.
Obviously, for fixed σ, τ, and q, we have $m = \lceil \frac{1}{q} \cdot (n + \sigma + 1) \rceil = \Theta(n)$. So

$$\mathcal{O}(\, m \cdot \mathcal{M}(n + m + \sigma + m[\![\tau + q]\!]) + m \cdot (\, \mathrm{ld}\, n + \,\mathrm{ld}\, \sigma + \ell)\,) \; = \Theta(n \cdot \mathcal{M}(n)).$$

Finally $[\![4 + n + \,\mathrm{ld}\, m - k(\tau + q)]\!] = \mathcal{O}(\, n\,)$, for $0 \le k \le m$. As t is supposed to be regular, we have $\sum_0^m t([\![4 + n + \,\mathrm{ld}\, m - k(\tau + q)]\!]) = \mathcal{O}(\, n \cdot t(n)\,)$. □

4 Conclusions

As immediate consequences of the previous chapters we have

4.1 Theorem. The coefficients of a Taylor series $\sum a_k z^k$ are uniformly computable if and only if the sum function $f(z)$ is computable in a neighbourhood of zero. □

and

4.2 Theorem. The coefficients of a Taylor series $\sum a_k z^k$ are uniformly computable *in polynomial time* if and only if the sum function $f(z)$ is computable *in polynomial time* in a neighbourhood of zero. □

The transformations between the series and the sum were possible using two additional values: In section 2 we used a lower bound γ for the radius of convergence and an upper bound M_1 for the value of the sum function on the circle Γ with radius γ, while in section 3 we used M_2 and R such that $\sum |a_k|\, R^k \le M_2$.

The information given by (M_1, γ) or by (M_2, R) is of the same value: Given M_2 and R, we may obviously choose any $M_1 \ge M_2$ and any $\gamma < R$. On the other hand, from given M_1 and γ we get $|a_k| \le M_1 \gamma^{-k}$ and so for any $R < \gamma$ we may choose $M_2 := M_1 \cdot \frac{R}{\gamma - R} = \sum_k (M_1 \gamma^{-k} \cdot R^k) \ge \sum_k |a_k|\, R^k$

We suppose that this additional information is necessary for any *constructive* proof of the theorems above.

Bibliography

[Al82] Alt, H., Multiplication is the easiest nontrivial arithmetic function, *Theoret. Comput. Sci.* **36** (1985) 333-339

[Br75] Brent, R.P., The complexity of multiple precision arithmetic, Proc. Seminar on Complexity of Computational Problem Solving, Queensland U. Press, Brisbane, Australia (1975) 126-165

[Br76] Brent, R.P., Fast multiple precision evaluation of elementary functions, *J. ACM* **23** (1976) 242-251

[FiSt74] Fischer, M.J., Stockmeyer, L.J., Fast on-line integer multiplication, *J. Comput. System Scis.* **9** (1974) 317-331

[Fr84] Friedman, H., The Computational Complexity of Maximization and Integration, *Advances in Mathematics* **53** (1984) 80-98

[He81] Heuser, H., *Lehrbuch der Analysis 1/2*, (Teubner, Stuttgart, 1981)

[Ko83] Ko, K., On the computational complexity of differentiation, *Technical Report #UH-CS-83-2* (1983)

[KoFr82] Ko, K., Friedman, H., Computational complexity of real functions, *Theoret. Comput. Sci.* **20** (1982) 323-352

[KoFr88] Ko, K., Friedman, H., Computing Power Series in Polynomial Time, *Adv. in Appl. Mathematics* **9** (1988) 40-50

[Kr84] Kreitz, C., Theorie der Darstellungen und ihre Anwendung in der konstruktiven Analysis, Thesis, *Informatik Berichte* **50** FernUniversität Hagen (1984)

[Lo89] Lombardi, H., Nombres algébriques & approximations Thesis (3rd Part), Université de Franche Compte, Besancon (1989)

[MuMo93] Moiske, B., Müller, N.Th., Solving initial value problems in polynomial time, *in preparation*

[Mu86] Müller, N.Th., Subpolynomial complexity classes of real functions and real numbers, *Proc. 13th ICALP*, *Lecture notes in computer science* **226** (Springer, Berlin, 1986) 284-293

[Mu87] Müller, N.Th., Uniform computational complexity of Taylor series, *Proc. 14th ICALP*, *Lecture notes in computer science* **267** (Springer, Berlin, 1987) 435-444

[Sc82] Schönhage, A., The fundamental theorem of algebra in terms of computational complexity, Preliminary Report, Mathematisches Institut der Universität Tübingen, 1982

[ScSt71] Schönhage, A., Strassen, V., Schnelle Multiplikation großer Zahlen, *Computing* **7** (1971) 281-292

[We87] Weihrauch, K., *Computability*, (Springer, Berlin, 1987)