# Untersuchung von Methoden zur algorithmisierten Entwicklung komplexwertiger Funktionen in formale Laurentreihen

Wissenschaftliche Hausarbeit
zur Ersten (Wissenschaftlichen) Staatsprüfung
für das Amt des Studienrats

Vorgelegt von:


Axel Rennoch
Prager Platz 5
1000 Berlin 30


Berlin, den 16. Juli 1992

## Inhaltsverzeichnis

# 1 Einleitung

In der reellen und komplexen Analysis sind Potenzreihen wichtigste Objekte. Sie werden meist als spezielle Funktionenreihen eingeführt. Dabei ist eine Funktionenreihe eine Folge von Partialsummen

$$s_n(z) = \sum_{k=0}^{n} f_k(z)$$

im allgemeinen komplexwertiger Funktionen $f_k(z)$. Im Fall $f_k(z) = a_k(z - z_0)^k$ für alle $k \in \mathbb{N}_0$ wird von einer Potenzreihe gesprochen, $a_k \in \mathbb{C}$ heißen Koeffizienten und $z_0$ Entwicklungspunkt der Potenzreihe

$$F(z) = \sum_{k=0}^{\infty} a_k(z - z_0)^k .$$

Mit Potenzreihen lassen sich u. a. die elementar-transzendenten Funktionen und ihre Eigenschaften in einfacher Weise darstellen bzw. nachweisen. Vor allem in der komplexen Analysis zeigen die Potenzreihen ihren hohen Nutzen (z. B. beim Nachweis der Eulerschen Gleichung $e^{iz} = \cos(z) + i\sin(z)$ ) [W90, S. 178]. Die Entwicklung einer komplexwertigen Funktion $f(z)$ in die geschlossene Form einer Potenzreihe gewinnt damit eine große Bedeutung. Zugleich entsteht ein Interesse an Methoden zur Entwicklung der zahlreichen Funktionen, und insbesondere stellt sich die Frage, ob diese Entwicklungen algorithmisiert werden können, d. h. ob es universelle Verfahren dafür gibt. Mit dem Aufkommen der an der Mathematik orientierten symbolischen Programmiersprachen, sog. Computer Algebra Systeme, finden die Antworten auf die letzte Frage eine zusätzliche und wertvolle Anwendung.

Die vorliegende Arbeit untersucht verschiedene Methoden zur algorithmisierten Entwicklung komplexwertiger Funktionen in eine formale Potenzreihe $\sum_{k=0}^{\infty} a_k X^k$ bzw. - in einer leichten Verallgemeinerung - in eine formale Laurentreihe. Ein Merkmal einer "formalen" Reihe ist ein beliebig zugrunde gelegter Körper $K$, mit $a_k \in K$ und $X$ als Platzhalter. Die Frage nach dem Konvergenzradius der Reihe ist deshalb zunächst von geringerer Bedeutung [GKP88, S. 206 u. 317]. Die vorliegende Methodenuntersuchung beschränkt sich aber auf den Körper der komplexen Zahlen $K = \mathbb{C}$. Die Einbeziehung von formalen Laurentreihen neben den formalen Potenzreihen entspricht dabei nicht nur einer in der Funktionentheorie üblichen Betrachtungsweise [DR72, S. 87], sondern ermöglicht von Anfang an eine einheitliche Methodenanalyse. Den Schwerpunkt dieser Arbeit bildet die Transformation:

$$f(z) \to \sum_{k=k_0}^{\infty} a_k z^k, \qquad \text{mit } k_0 \in \mathbb{Z} \text{ und } a_k, z \in \mathbb{C}.$$

Die Beschäftigung mit Methoden zur algorithmisierten Entwicklung von Funktionen in eine Reihenentwicklung bezog sich in der Vergangenheit vor allem auf die Fehleranalyse bei der Verknüpfung von "abbrechenden", nur aus endlich vielen Gliedern bestehenden "Potenzreihen"

$\sum_{k=k_0}^{k_1} a_k z^k$ (truncated power series) [DST88, S. 88 - 93]. Unter Berücksichtigung der fortgeschrittenen Möglichkeiten symbolischer Programmierung geben diese Arbeiten einen Anstoß zu einer Erarbeitung von Algorithmen zur Erstellung von geschlossenen Reihendarstellungen, auf deren Grundlage auch Verknüpfungen zwischen formalen Reihen erfolgen können.

Neben dem erst vor kurzem erschienenen Artikel [K92], an dessen Grundlage sich die vorliegende Arbeit orientiert, gibt es nur wenige in ähnlicher Weise ausgerichtete Veröffentlichungen: Ein Beispiel ist die sog. “Norman-Methode” [N75], die wie in [K92] von einer gewöhnlichen Differentialgleichung der zu entwickelnden Funktion ausgeht und diese zur Gewinnung der Koeffizienten einer formalen Potenzreihe benutzt. Norman beschäftigt sich allerdings nicht mit einem vollständigen Entwicklungsalgorithmus, da die benötigte Differentialgleichung als bereits bekannt vorausgesetzt wird und wiederum nur endlich viele Koeffizienten der formalen Potenzreihe bestimmt werden. Interessant ist Norman’s Methode aber, weil formale Potenzreihen durch Differentialgleichungen definiert werden und diese auch die Grundlage für das Rechnen mit formalen Potenzreihen bilden. Dabei werden auch Funktionen mehrerer Variablen (z. B. $f(y, z) = y^z$) und nichtlineare Differentialgleichungen (z. B. Emdens Differentialgleichung $y'' + 2 \cdot \frac{y'}{x} + y^n = 0$) nicht ausgenommen [N75, S. 352 u. 355], für die eine Lösung in Form einer geschlossenen formalen Potenzreihe überhaupt nicht vorliegt [Ka77, S. 560 u. 561].

Formale Laurentreihen nehmen zwar als Lösung von Differentialgleichungen stets eine ausgezeichnete Position ein, ihre Rolle bezieht sich dann jedoch in der Regel nur auf Lösungsformeln mit $n$ linear unabhängigen Termen der Form $y_l = \sum_{j=j_0}^{\infty} a_{lj} x^j$, $0 \le l < n-1$, deren Koeffizienten $a_{lj} \in \mathbb{C}$ aber in rekursiven Verfahren einzeln bestimmt werden müssen [Si90, S. 59 - 62].

Abgesehen von der Benutzung einzelner bekannter Reihenentwicklungen (z. B. der Binominalreihe) oder der Taylor- bzw. Laurent-Entwicklung, sind kaum konstruktive Methoden bekannt, die eine Funktion mit geschlossenen Formen von unendlichen Reihen in direkten Bezug setzen. Einen Konstruktionsalgorithmus zwischen einer aus endlich vielen Summanden bestehenden Partialsumme $s_n(z) = \sum_{k=0}^{n} a_k z^k$ und einer geschlossenen Form für $s_n(z)$ gibt es mit dem sog. Gosper-Algorithmus [G78; GKP88, S. 224 - 226]. Gosper setzt dabei einen sog. “hypergeometrischen Term” für die Koeffizienten $a_k$ voraus, d. h. $a_k$ muß zurückgeführt werden auf die spezifische Form:

$$a_k = \frac{(p_1)_k \cdot \ldots \cdot (p_m)_k}{(q_1)_k \cdot \ldots \cdot (q_n)_k \cdot k!} ,$$

wobei $(x)_k = x(x+1)\ldots(x+k-1)$ das Pochhammersymbol ist. Ziel des Gosper-Algorithmus ist, zu gegebenen Werten $p_1, \ldots, p_m, q_1, \ldots, q_n$, über den Ansatz

$$a_k z^k = T(k+1) - T(k)$$

die Parameter $c, P_1, \ldots, P_M, Q_1, \ldots, Q_N, Z$ eines hypergeometrischen Terms

$$T(k) = c \cdot \frac{(P_1)_k \cdot \ldots \cdot (P_M)_k \cdot Z^k}{(Q_1)_k \cdot \ldots \cdot (Q_N)_k \cdot k!}$$

zu bestimmen, so daß

$$\sum_{k=0}^{n} a_k z^k = T(n+1) - T(0) .$$

Dies gelingt durch die Konstruktion einer rationalen Beziehung zwischen $a_k z^k$ und $T(k)$, die insbesondere auf der Rekursionsgleichung der $a_k$ aufbaut, schließlich gilt $a_{k+1} = R(k) a_k$, mit $R(k)$ rational. Der Gosper-Algorithmus entscheidet nun darüber, ob $\sum_{k=0}^{n} a_k z^k$ mit einem geschlossenen hypergeometrischen Term bestimmt werden kann. Damit betont Gosper bereits die - in dieser Methodenanalyse besonders berücksichtigte - Klasse der hypergeometrischen Funktionen.

Die Rekursionsgleichung der Koeffizienten $a_k$ ist für die Partialsumme $s_n(z)$ und die formale Potenzreihe $\sum_{k=0}^{\infty} a_k z^k$ von zentraler Bedeutung. Ihre Überführung in eine geschlossene Formel der $a_k$ ist ein Hauptproblem der algorithmisierten Entwicklung einer Funktion in eine formale Reihe.

Diese zentrale Aufgabe findet sich wieder in der Theorie der erzeugenden Funktionen (generating functions), die auch von einer Rekursionsgleichung der $a_k$ ausgeht und nach der Funktion fragt, die eine Potenzreihenentwicklung mit genau den Koeffizienten $a_k$ erzeugt. Allerdings wird dabei in der Regel die Potenzreihendarstellung nicht direkt, sondern auch wiederum über eine geschlossene komplexwertige erzeugende Funktion gewonnen [Wi90, S. 8]. Die hier untersuchten Methoden zur Entwicklung komplexwertiger Funktionen in eine formale Laurentreihe liefern in diesem Sinne eher einen Beitrag zur Theorie der erzeugenden Funktionen. Andererseits geben deren Ergebnisse auch einen Einblick und damit Anregungen für den Zusammenhang zwischen erzeugenden Funktionen und den entsprechenden Potenzreihenkoeffizienten. Die dabei aufgestellten Regeln werden meist in drei Gruppen zusammengefaßt. Z. B. gilt für “gewöhnliche” erzeugende Funktionen (formal ordinary power series generating functions: ogf) [Wi90, S. 32]:

$$\left(f(z) = \sum_{k=0}^{\infty} a_k z^k\right) \Leftrightarrow f \overset{\text{ogf}}{\leftrightarrow} \{a_k\}_0^{\infty} \qquad \Rightarrow \qquad P\left(z \frac{d}{dz}\right) f \overset{\text{ogf}}{\leftrightarrow} \{P(k) a_k\}_0^{\infty} ,$$

für beliebiges Polynom $P$.

Für entsprechende Regeln zu “exponentialen” erzeugenden Funktionen (formal exponential generating functions: egf) wird ein spezieller Ansatz eingeführt [Wi90, S. 36]:

$$\left(f(z) = \sum_{k=0}^{\infty} \frac{a_k}{k!} z^k\right) \Leftrightarrow f \overset{\text{egf}}{\leftrightarrow} \{a_k\}_0^{\infty} .$$

Ein häufig genannter dritter Ansatz mit sog. Dirichlet-Reihen ist für die Entwicklung in formale

Laurentreihen nicht zu berücksichtigen, denn es gilt dabei [Wi90, S. 52]:

$$\left(f(z) = \sum_{k=1}^{\infty} \frac{a_k}{k^z}\right) \Leftrightarrow f \overset{\text{Dir}}{\leftrightarrow} \{a_k\}_1^{\infty} .$$

Mit der Kenntnis und dem Verstehen eines Algorithmus folgen einerseits wiederholte Anwendungen mit verschiedensten Beispielen und andererseits wird der Algorithmus selbst - im Sinne eines mathematischen Satzes - zur Grundlage weiterer mathematischer Überlegungen. Die symbolische Programmierung von Algorithmen mit einem Computer Algebra System bildet dafür ein experimentelles Arbeitswerkzeug und entlastet zugleich von zeitintensiven Routinearbeiten. Algorithmisierte Methoden gewinnen damit neben ihrer mathematischen Aussage mit der Verbreitung von Computer Algebra Systemen einen zusätzlichen Stellenwert für die Mathematik in Forschung und Lehre. Nicht zuletzt werden manche Algorithmen, wie z. B. der Gosper-Algorithmus [G78, S. 42], überhaupt erst mit Hilfe von Computer Algebra Systemen gefunden.

Beispiele für die Berücksichtigung von Computer Algebra Systemen sind sehr zahlreich und finden sich insbesondere in Fachzeitschriften oder Diplomarbeiten [z. B. Si90, T89]. In zunehmenden Maße gilt dies auch für den Mathematikunterricht an Universitäten und Schulen: Der Unterrichtsstoff wird ergänzt durch “Experimente” mit einem Computer Algebra System und zugleich entlastet z. B. von routinemäßigem Rechnen oder Zeichnen von Funktionen [z. B. GM90, BGK92, S92]. Ein Zeichen für den Erfolg erster Unterrichtsexperimente kann darin gesehen werden, daß z. B. das österreichische Unterrichtsministerium im vergangenen Jahr ein Computer Algebra System als Standardwerkzeug für den Mathematikunterricht an allen Gymnasien angeschafft hat [Ku91, S. 21].

Die vorliegende Methodenuntersuchung befaßt sich zunächst im Rahmen einer Bedingungsanalyse (Kapitel 2) insbesondere mit der Frage nach den Funktionen, die sich für eine algorithmisierte Entwicklung in eine formale Reihe eignen. Es folgen dann die im Mittelpunkt stehenden verschiedenen Methoden mit ihren Algorithmen für die Entwicklung in eine formale Laurentreihe (Kapitel 3). In diesem Zusammenhang bilden die mathematischen Beweise der Algorithmen stets einen Schwerpunkt. In Kapitel 4 wird einerseits die Anwendbarkeit der vorgestellten Methoden durch einzelne Ansätze erweitert, als auch ihr Nutzen an einzelnen Beispielen diskutiert. Die Fragestellungen, die sich aus einer Verknüpfung der verschiedenen Ansätze zu einem einheitlichen Algorithmus ergeben, sind das Thema in Kapitel 5. Dabei wird in einem Exkurs auch auf die Fähigkeiten heute weit verbreiteter Computer Algebra Systeme zur Entwicklung von Funktionen in formale Reihen eingegangen. Abschließend werden die Fragen nach der Umkehrbarkeit der Methoden, d. h. der Transformation einer formalen Laurentreihe in eine - diese Reihe erzeugende - Funktion ($F \to f(z)$), und dem Konvergenzradius der formalen Reihen behandelt (Kapitel 6) sowie eine Schlußbetrachtung (Kapitel 7) gegeben. Anhang A enthält eine Liste von Funktionen, auf die der Algorithmus aus Kapitel 5 angewandt wurde.

# 2 Bedingungsanalyse für eine algorithmisierte Entwicklung

## 2.1 Formale Laurentreihen

Für eine algorithmisierte Betrachtung lassen sich Potenzreihen abstrakt als formale Potenzreihen definieren. Eine “formale Potenzreihe” ist dabei eine unendliche Folge von Elementen $a_k$ eines Körpers $K$ (in der Regel $K = \mathbb{C}$) [H74, S. 8]: $\{a_0, a_1, a_2, \dots\}$ . Meist wird die formale Potenzreihe mit großen Buchstaben bezeichnet und folgendermaßen geschrieben:

(2.1-1) $$F := \sum_{k=0}^{\infty} a_k z^k = a_0 + a_1 z + a_2 z^2 + \dots \quad .$$

Eine Verallgemeinerung des Begriffs der formalen Potenzreihe führt zur sog. “formalen Laurentreihe”, die sich durch einen beliebigen Anfangsindex $k_0 \in \mathbb{Z}$ von den formalen Potenzreihen unterscheidet [H74, S. 52]:

(2.1-2) $$F := \sum_{k=k_0}^{\infty} a_k z^k, \qquad \text{mit } a_{k_0} \neq 0.$$

Hier dient $z$ als Platzhalter, und durch Substitution $z \to (z - z_0)$ entstehen formale Laurentreihen mit dem Entwicklungspunkt $z_0$.

Im Gegensatz zur Definition der “Laurentreihe” $\sum_{k=-\infty}^{\infty} a_k z^k$, besitzt die formale Laurentreihe höchstens endlich viele nicht-verschwindende Summanden $a_k$ mit $k < 0$.

O. B. d. A. seien zwei beliebige formale Laurentreihen mit gleichem Anfangsindex $k_0$ gegeben:

$$F := \sum_{k=k_0}^{\infty} a_k z^k,\ G := \sum_{k=k_0}^{\infty} b_k z^k \in L.$$

Mit den Operationen (2.1-3) - (2.1-5) bilden die formalen Laurentreihen einen Integritätsbereich $L$ mit dem Einselement $I = 1 + 0 \cdot z + 0 \cdot z^2 + \dots$ bzw. einen $\mathbb{C}$-Vektorraum [DR72, S. 87 - 91].

(2.1-3) $$F + G := \sum_{k=k_0}^{\infty} (a_k + b_k) z^k,$$

(2.1-4) $$F \cdot G := \sum_{k=k_1}^{\infty} c_k z^k, \quad \text{Cauchy-Produkt mit } c_k := \sum_{j+l=k} a_j b_l,\ j, l \geq k_0 \text{ und } k_1 \geq 2k_0,$$

(2.1-5) $$c \cdot F := \sum_{k=k_0}^{\infty} c \cdot a_k z^k, \text{ mit } c \in \mathbb{C}.$$

Insbesondere gelten deshalb mit $H \in L$ folgende Eigenschaften der Multiplikation im Ring:

(2.1-6) $$F \cdot G = G \cdot F,$$

(2.1-7) $\quad F \cdot (G + H) = F \cdot G + F \cdot H$,

(2.1-8) $\quad F \cdot (G \cdot H) = (F \cdot G) \cdot H$ .

Darüberhinaus ergibt sich:

(2.1-9) $\quad \frac{1}{F} := z^{-k_0} \frac{1}{P}$ existiert, falls $a_{k_0} \neq 0$ [H74, S. 53],

wobei $F = z^{k_0} \cdot P = z^{k_0} \cdot \sum_{k = k_0}^{\infty} a_k z^{k - k_0} = z^{k_0} \cdot \sum_{k = 0}^{\infty} p_k z^k$, d. h. $p_k = a_{k_0 + k}$.

$\frac{1}{P} := \sum_{k = 0}^{\infty} c_k z^k$ ergibt sich mit der Wronski-Formel [H74, S. 17]:

$$c_k := \frac{(-1)^k}{p_0^{k+1}} \begin{vmatrix} p_1 & p_2 & \cdots & & p_k \\ p_0 & p_1 & \cdots & & p_{k-1} \\ 0 & p_0 & \cdots & & p_{k-2} \\ & & \cdots & & \\ 0 & 0 & \cdots & p_0 & p_1 \end{vmatrix} \text{ für } k = 1, 2, \ldots,$$

(2.1-10) $\quad F' := \sum_{k = k_0}^{\infty} k a_k z^{k-1} = \sum_{k = k_0 - 1}^{\infty} (k+1) a_{k+1} z^k$ ,

(2.1-11) $\quad \int F dz := a_{-1} \ln(z) + \sum_{\substack{k = k_0 \\ k \neq -1}}^{\infty} \frac{a_k}{k+1} z^{k+1} = a_{-1} \ln(z) + \sum_{\substack{k = k_0 + 1 \\ k \neq -1}}^{\infty} \frac{a_{k-1}}{k} z^k$ .

## 2.2 Meromorphe Funktionen

Die algorithmisierte Entwicklung von Funktionen in ihre formalen Laurentreihen stellt die Frage nach den Bedingungen an die zu entwickelnden Funktionen. Eine erste Antwort gibt der Taylorsche Entwicklungssatz [BS76, S. 131]:

> “Ist die Funktion $f(z)$ im endlichen Punkt $z_0$ holomorph, so läßt sie sich in eine Potenzreihe um $z_0$ entwickeln.”

Wird $f(z)$ in der Umgebung von $z_0$ durch $\sum_{n = 0}^{\infty} a_n (z - z_0)^n$ dargestellt, sind auch alle Koeffizienten eindeutig bestimmt [BS76, S. 133] durch:

$$a_n = \frac{1}{n!} f^{(n)}(z_0) .$$

Eine Reihenentwicklung um $z_0$ ist aber auch dann noch möglich, falls die Funktion $f(z)$ eine schlichte isolierte Singularität in $z_0$ besitzt, es also eine Umgebung von $z_0$ gibt, in der $f(z)$ überall außer in $z_0$ definiert und holomorph ist [BS76, S. 191]. Diese verallgemeinerte Taylor-Entwicklung führt zur Laurent-Entwicklung [BS76, S. 193]:

Ist $f$ holomorph in $G\backslash\{z_0\}$ , $G \subset \mathbb{C}$ Gebiet, $z_0 \in G$, $r_0$ Cauchy-Radius von $z_0$, so gilt:

$$\forall(z \in K(r_0,z_0)\backslash\{z_0\}): \qquad f(z) = \sum_{k=0}^{\infty} a_k(z-z_0)^k + \sum_{k=1}^{\infty} \frac{a_{-k}}{(z-z_0)^k} \quad .$$

Die erste Summe in dieser Darstellung heißt Nebenteil, die zweite Summe Hauptteil der Laurent-Entwicklung. Enthält der Hauptteil einer Laurent-Entwicklung gerade unendlich viele von Null verschiedene Glieder, so liegt eine wesentliche Singularität vor [BS76, S. 199], die für $f(z)$ keine Darstellung durch eine formale Laurentreihe gestattet.

Besitzt der Hauptteil der Laurent-Entwicklung jedoch nur endlich viele Glieder $\sum_{k=1}^{p} \frac{a_{-k}}{(z-z_0)^k}$ , $a_{-p} \neq 0$, so heißt $z_0$ außerwesentliche Singularität oder Pol der Funktion $f(z)$, und $p$ ist die Ordnung der Polstelle [BS76, S. 198]. Funktionen, die in ihrem Definitionsbereich $D$ außer in einer (von $f$ abhängigen) diskreten Teilmenge $P(f) \subset D$ holomorph sind und in jedem Punkt von $P(f)$ einen Pol haben, werden meromorph genannt. Beispiele meromorpher Funktionen sind alle rationalen Funktionen $f \neq 0$ [BS76, S. 201], also:

$$f(z) = \frac{a_0 + a_1 z + \ldots + a_m z^m}{b_0 + b_1 z + \ldots + b_n z^n}, \qquad \text{mit } b_n \neq 0,\, m, n \in \mathbb{N}_0.$$

In die Menge der in eine formale Laurentreihe zu entwickelnden Funktionen können deshalb alle rationalen Funktionen aufgenommen werden.

Daneben sollen möglichst viele elementar-transzendente Funktionen Berücksichtigung finden. Diese lassen sich in Abhängigkeit ihrer Koeffizientendarstellung in zwei Kategorien einteilen. Zunächst notieren wir die Funktionen mit direkter Koeffizientenformel:

(2.2-1) $$(1+z)^\alpha = \sum_{k=0}^{\infty} \binom{\alpha}{k} z^k ,$$

$$\text{mit } \binom{\alpha}{k} := \begin{cases} 1 & \text{, falls } k=0 \\ \alpha(\alpha-1)\ldots(\alpha-k+1) & \text{, falls } k>0 \end{cases} \qquad \text{und } \alpha \in \mathbb{R}, k \in \mathbb{N},$$

(2.2-2) $$\exp(z) = \sum_{k=0}^{\infty} \frac{1}{k!} z^k ,$$

(2.2-3) $$\ln(1+z) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k} z^k ,$$

(2.2-4) $$\sin(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} z^{2k+1} ,$$

(2.2-5) $$\cos(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k)!} z^{2k} \quad ,$$

(2.2-6) $$\sinh(z) = \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{2k+1} \quad ,$$

(2.2-7) $$\cosh(z) = \sum_{k=0}^{\infty} \frac{1}{(2k)!} z^{2k} \quad ,$$

(2.2-8) $$\arcsin(z) = z + \sum_{k=1}^{\infty} \frac{1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2k-1)}{(2k)!} z^{2k+1} \quad ,$$

(2.2-9) $$\arctan(z) = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} z^{2k+1},$$

(2.2-10) $$\operatorname{arcsinh}(z) = z + \sum_{k=1}^{\infty} \frac{(-1)^k 1 \cdot 3 \cdot 5 \cdot \ldots \cdot (2k-1)}{2^k k!(2k+1)} z^{2k+1} \quad ,$$

(2.2-11) $$\operatorname{arctanh}(z) = \sum_{k=0}^{\infty} \frac{1}{2k+1} z^{2k+1} \quad .$$

Demgegenüber gibt es elementar-transzendente Funktionen, deren Koeffizienten sich nur rekursiv darstellen lassen. Dazu zählen:

(2.2-12) $$\tan(z) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1} 4^k (4^k - 1) B_{2k}}{(2k)!} z^{2k-1} \quad ,$$

(2.2-13) $$\cot(z) = \sum_{k=0}^{\infty} \frac{(-1)^{k-1} 4^k B_{2k}}{(2k)!} z^{2k-1} \quad ,$$

(2.2-14) $$\tanh(z) = \sum_{k=1}^{\infty} \frac{4^k (4^k - 1) B_{2k}}{(2k)!} z^{2k} \quad ,$$

(2.2-15) $$\coth(z) = \sum_{k=0}^{\infty} \frac{4^k B_{2k}}{(2k)!} z^{2k-1} \quad ,$$

wobei $B_k$ die rekursiv definierten Bernoullizahlen sind:

$$B_n := \begin{cases} 1 & \text{, falls } n=0 \\ \left(-\frac{1}{n+1}\right) \sum_{k=0}^{n-1} \binom{n+1}{k} B_k & \text{, falls } n > 0 \end{cases} \quad .$$

Es ist offensichtlich, daß eine algorithmisierte Entwicklung der Rekursionsgleichungen für die Koeffizienten in den Fällen (2.2-12) bis (2.2-15) eine andere Aufgabenkategorie darstellt als bei den Funktionen (2.2-1) bis (2.2-11). Die Betrachtung der Funktionen der ersten Kategorie ist elementarer und soll deshalb im Vordergrund stehen.

Andererseits eignen sich viele spezielle Funktionen geradezu für eine algorithmisierte Entwicklung in ihre formale Laurentreihe. Es sind die Funktionen vom hypergeometrischen Typ (siehe Kapitel 2.4), denen auch die o. g. elementaren Funktionen der 1. Gruppe zugeordnet werden können und die sich auf direktem Weg mit hypergeometrischen Reihen in Verbindung bringen und damit in ihre formalen Laurentreihen entwickeln lassen.

## 2.3 Hypergeometrische Funktionen

Eine “hypergeometrische Reihe” ist definiert durch [H74, S. 27]:

$$F = \sum_{k=0}^{\infty} A_k z^k \text{ mit } A_k = \frac{(a_1)_k \dots (a_p)_k}{(b_1)_k \dots (b_q)_k k!} .$$

Die dabei verwendeten Pochhammersymbole $(a)_k$ sind für $k \in \mathbb{N}_0$ definiert als:

$$(a)_k := \begin{cases} 1 & \text{, falls } k=0 \\ a(a+1)\dots(a+k-1) & \text{, falls } k>0 \end{cases} .$$

Pochhammersymbole lassen sich auch direkt in Binomialkoeffizienten umwandeln:

$$\frac{(a)_k}{k!} = \frac{(a+k-1)([a+k-1]-1)\dots([a+k-1]-(k-1))}{k!} = \binom{a+k-1}{k} .$$

Für die Koeffizienten $A_k$ der hypergeometrischen Reihe läßt sich sofort eine Rekursionsbeziehung aufstellen:

<u>LEMMA 2.3-1:</u>

Die Koeffizienten $A_k$ einer unendlichen Reihe $F$ genügen genau dann der Rekursionsgleichung $A_{k+1} = R(k)A_k$ mit $R(k)$ rational, falls $F$ eine hypergeometrische Reihe ist.

<u>BEWEIS von Lemma 2.3-1:</u>

Für $k \in \mathbb{N}_0$ sei $A_{k+1} = \dfrac{(a_1)_{k+1} \dots (a_p)_{k+1}}{(b_1)_{k+1} \dots (b_q)_{k+1} (k+1)!}$ .

Wegen $(n)_{k+1} = n(n+1)\dots(n+k-1)(n+k) = (n)_k (n+k)$ ist:

$$\text{(2.3-1)} \qquad A_{k+1} = \frac{(a_1)_k \dots (a_p)_k (a_1+k)\dots(a_p+k)}{(b_1)_k \dots (b_q)_k (b_1+k)\dots(b_q+k)} \frac{1}{(k+1)!}$$

$$= \frac{(a_1+k)\dots(a_p+k)}{(b_1+k)\dots(b_q+k)(k+1)} A_k \text{ , wobei o. B. d. A. } A_0 = 1 .$$

Die Existenz der angegebenen Rekursionsgleichung für hypergeometrische Reihen ist damit offensichtlich. Andererseits läßt sich jede rationale Funktion $R(k)$ im Körper der komplexen Zahlen nach Berechnung ihrer Nullstellen in der Form

$$R(k) = \frac{(a_1+k)\dots(a_p+k)}{(b_1+k)\dots(b_q+k)(k+1)}$$

darstellen, und daraus folgt dann eine der hypergeometrischen Reihe entsprechende Koeffizientendarstellung. ❏

Die durch eine hypergeometrische Reihe $F$ dargestellte "hypergeometrische" Funktion $f$ wird geschrieben als:

$$f(z) = {}_pF_q\left(\begin{matrix} a_1 & \dots & a_p \\ b_1 & \dots & b_q \end{matrix}\middle| z\right).$$

Eine hypergeometrische Reihe $F$ zeichnet sich dadurch aus, daß sie einer homogenen linearen Differentialgleichung mit polynomialen Koeffizienten genügt. Im folgenden wird eine solche Gleichung auch "einfache Differentialgleichung" genannt.

Für $Q{:=}\max\{p, q\}+1$ gibt es $c_{jn} \in \mathbb{C}$, $j, n = 0, \dots, Q$, so daß

$$\text{(2.3-2)} \qquad \sum_{n=0}^{Q}\left(\sum_{j=0}^{Q} c_{jn}z^j\right)f^{(n)} = \sum_{n=0}^{Q}(c_{0n}+c_{1n}z+\dots+c_{Qn}z^Q)f^{(n)} = 0,$$

Diese Behauptung geht auf den folgenden Satz [H74, S. 30] zurück:

SATZ 2.3-2:

Sei $F := \sum_{k=0}^{\infty} B_k$ hypergeometrische Reihe mit $B_k = \frac{(a_1)_k \dots (a_p)_k}{(b_1)_k \dots (b_q)_k k!} z^k$ .

Dann gilt mit $\theta := z\frac{d}{dz}$:

a) $F$ ist Lösung der Differentialgleichung:

$$\text{(2.3-3)} \qquad \theta(\theta+b_1-1)\dots(\theta+b_q-1)Y = z(\theta+a_1)\dots(\theta+a_p)Y \quad .$$

b) Jede formale Potenzreihe, die Lösung dieser Differentialgleichung ist, kann durch $c \cdot F$, $c \in \mathbb{C}$, dargestellt werden.

BEWEIS von Satz 2.3-2:

O. B. d. A. sei $B_0 = 1$.

Wegen $(\theta+a)z^k = z\left(\frac{d}{dz}z^k\right)+az^k = z(kz^{k-1})+az^k = (k+a)z^k$ gilt

$$\begin{aligned}\text{(2.3-4)} \qquad & \theta(\theta+b_1-1)\dots(\theta+b_q-1)B_k \\ &= k(k+[b_1-1])\dots(k+[b_q-1])B_k \\ &= k(b_1+k-1)\dots(b_q+k-1)\frac{(a_1)_k\dots(a_p)_k}{(b_1)_k\dots(b_q)_k k!}z^k\end{aligned}$$

$$\begin{aligned}
&= \frac{(a_1)_k \dots (a_p)_k}{(b_1)_{k-1} \dots (b_q)_{k-1}(k-1)!} z^k \\
&= z(a_1 + k - 1)\dots(a_p + k - 1)B_{k-1} \\
&= z([k-1] + a_1)\dots([k-1] + a_p)B_{k-1} \\
&= z(\theta + a_1)\dots(\theta + a_p)B_{k-1}
\end{aligned}$$

zu a) $\theta(\theta + b_1 - 1)\dots(\theta + b_q - 1)B_0 = 0$ , da $B_0 = 1$ und wegen (2.3-4) ist

$\forall k \in \mathbb{N}$: $\theta(\theta + b_1 - 1)\dots(\theta + b_q - 1)B_k = z(\theta + a_1)\dots(\theta + a_p)B_{k-1}$ ,

so daß (2.3-3) für $Y = F$ durch Koeffizientenvergleich folgt.

zu b) Sei $P := \sum_{k=0}^{\infty} c_k z^k$ eine Lösung von (2.3-3).

Beweis durch vollständige Induktion:

Für $k = 0$ gilt o. B. d. A. $c_0 = 1$ (Induktionsanfang), sonst wähle $c := \frac{1}{c_0}$ .

Sei $c_k = \frac{(a_1)_k \dots (a_p)_k}{(b_1)_k \dots (b_q)_k k!}$ wahr für $k = k_0$ (Induktionsannahme).

Der $(k+1)$ -te Koeffizient von $z(\theta + a_1)\dots(\theta + a_p)P$ ist wegen (2.3-4):

$(a_1 + k)\dots(a_p + k)c_k$ .

Der ($k$+1)-te Koeffizient von $\theta(\theta + b_1 - 1)\dots(\theta + b_q - 1)P$ ist

$(k+1)(k+b_1)\dots(k+b_q)c_{k+1}$ .

Damit ergibt sich:

$$\begin{aligned}
c_{k+1} &= \frac{(a_1 + k)\dots(a_p + k)}{(b_1 + k)\dots(b_q + k)(k+1)} c_k \\
&= \frac{(a_1)_{k+1} \dots (a_p)_{k+1}}{(b_1)_{k+1} \dots (b_q)_{k+1}(k+1)!} ,
\end{aligned}$$

(Induktionsschluß), also genau die Koeffizienten der hypergeometrischen Reihe. ❑

Mit dem Beweis von Satz 2.3-2 ist offensichtlich, daß in der Differentialgleichung (2.3-3) der hypergeometrischen Reihe links höchstens die $(q+1)$ -te Ableitung von $f$ erscheint bzw. rechts die $p$-te Ableitung von $f$. Der polynomiale Koeffizient von $Y^{(n)}$ besitzt höchstens den Grad $Q = \max\{p, q\} + 1$ . Somit gilt (2.3-2).

## 2.4 Funktionen vom hypergeometrischen Typ

Satz 2.3-2 setzt hypergeometrische Reihen in Beziehung zu den Lösungen der Differentialgleichung (2.3-3) und zu (2.3-2). In einer Verallgemeinerung der hypergeometrischen Reihe können Funktionen vom hypergeometrischen Typ betrachtet werden [K92, S. 5]. Dabei wird eine Funktion $f$ als “Funktion vom hypergeometrischen Typ” bezeichnet, falls die zugehörige formale Laurentreihe Koeffizienten $a_k$ besitzt mit

(2.4-1) $\qquad a_{k+m} = R(k) a_k$, $\qquad$ für $k \geq k_0$, und $a_{k_0} = a$,
wobei $a \in \mathbb{C} \backslash \{0\}$, $R$ rationale Funktion ist und $m \in \mathbb{N}$ Symmetriezahl heißt.

Insbesondere heißt $f(z)$ "$m$-fach symmetrisch", falls (2.4-1) gilt mit

$$a_{k+1} = \ldots = a_{k+(m-1)} = 0 .$$

Z. B. sind die Funktionen (2.2-4) - (2.2-11) zweifach symmetrisch.

Der hypergeometrische Typ einer Funktion, der für die Entwicklung in Potenzreihen von zentraler Bedeutung ist, bleibt auch bei einfachen Transformationen erhalten:

<u>LEMMA 2.4-1:</u>

1) Sei $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ eine Funktion vom hypergeometrischen Typ, dann gilt dies auch für:

a) $zf(z)$, $\qquad$ b) $\dfrac{f(z)}{z}$,

c) $f(cz)$, $\quad$ mit $c \in \mathbb{C}$, $\qquad$ d) $f(z^n)$, $\quad$ mit $n \in \mathbb{N}$,

e) $\dfrac{f(z) + f(-z)}{2}$, $\qquad$ f) $\dfrac{f(z) - f(-z)}{2}$,

g) $f'(z)$, $\qquad$ h) $\int f(z) dz$, $\quad$ falls $k_0 \geq 0$.

2) Sei $m$ Symmetriezahl von $f(z)$, dann ist auch jedes Vielfache eine Symmetriezahl von $f(z)$.

<u>BEWEIS von Lemma 2.4-1:</u>

zu 1a) Sei $g(z) = zf(z) = \sum_{k=k_0+1}^{\infty} a_{k-1} z^k$ .

Für die Koeffizienten von $g(z) = \sum_{k=k_0+1}^{\infty} b_k z^k$ gilt wegen $b_k = a_{k-1}$ auch:

$$b_{k+m} = a_{(k-1)+m} = R(k-1) a_{k-1} = R(k-1) b_k \; .$$

Mit $R(k)$ ist auch $R(k-1)$ rational.

zu 1b) $g(z) = \dfrac{f(z)}{z} = \sum_{k=k_0-1}^{\infty} a_{k+1} z^k$ .

Analog zu Teil 1a: $b_{k+m} = a_{k+1+m} = R(k+1) a_{k+1} = R(k+1) b_k$

zu 1c) $g(z) = f(cz) = \sum_{k=k_0}^{\infty} (a_k c^k) z^k$ , also:

$$b_{k+m} = a_{k+m} c^{k+m} = R(k) a_k c^k c^m = R(k) c^m \cdot b_k ,$$

wobei $R(k) c^m$ rational bleibt.

zu 1d) $g(z) = f(z^n) = \sum_{k=k_0}^{\infty} a_k z^{kn} = \sum_{k=k_0}^{\infty} b_k z^k$,

mit $b_{n(k+m)} = a_{k+m} = R(k)a_k = R(k)b_{nk}$ ,

damit folgt $b_{k+nm} = R\left(\frac{k}{n}\right)b_k$ mit der Symmetriezahl $n \cdot m$.

zu 1e) $g_1(z) = \frac{f(z)+f(-z)}{2} = \sum_{k=k_0}^{\infty} b_k z^k$ ist der gerade Anteil von $f(z)$,

d. h. $b_{2k+1} = 0$ und $b_{2k} = a_{2k}$ für $k \in \mathbb{Z}$.

Für $b_k$ mit $k$ gerade gilt also

$$b_{k+2m} = a_{k+2m} = a_{(k+m)+m}$$
$$= R(k+m)a_{k+m} = R(k+m)R(m)a_k = R(k+m)R(m)b_k .$$

Wegen $b_{k+2m} = R(k+m)R(m)b_k = 0$ für $k$ ungerade gilt insgesamt:

$$b_{k+2m} = R_1^*(k)b_k \text{ mit } R_1^*(k) = R(k+m)R(m) \quad \text{rational}$$

und $g_1(z)$ hat die Symmetriezahl $2 \cdot m$.

zu 1f) Analog zu Teil 1e ergibt sich für den ungeraden Anteil von $f(z)$, $g_2(z) = \frac{f(z)-f(-z)}{2}$ ,

ebenfalls $R_2^*(k) = R(k+m)R(m)$ und die Symmetriezahl $2 \cdot m$.

zu 1g) $g(z) = f'(z) = \sum_{k=k_0-1}^{\infty} (k+1)a_{k+1}z^k$ ,

so daß $b_{k+m} = (k+1+m)a_{k+1+m}$

$$= (k+1+m)R(k+1)a_{k+1} = (k+1+m)R(k+1)\frac{b_k}{k+1}$$

und $\frac{k+1+m}{k+1}R(k+1)$ rational bleibt.

zu 1h) $g(z) = \int f(z)dz = \sum_{k=k_0+1}^{\infty} \frac{a_{k-1}}{k} z^k$ (siehe (2.1-8)) und

$$b_{k+m} = \frac{a_{k+m-1}}{k+m} = \frac{R(k-1)a_{k-1}}{k+m} = \frac{R(k-1)k \cdot b_k}{k+m} ,$$

mit $\frac{R(k-1)k}{k+m}$ rational.

zu 2) Mit $a_{k+m} = R(k)a_k$ folgt:

$$a_{k+jm} = a_{k+(j-1)m+m}$$
$$= R(k+(j-1)m)a_{k+(j-1)m} = \dots$$
$$= R(k+(j-1)m) \cdot R(k+(j-2)m) \cdot \dots \cdot R(k+m) \cdot R(k) \cdot a_m ,$$

wobei $R(k+(j-1)m) \cdot \ldots \cdot R(k)$ rational ist. ❑

Funktionen vom hypergeometrischen Typ haben wie die hypergeometrische Reihe Differentialgleichungen der Form (2.3-2):

SATZ 2.4-2:

Jede Funktion $f$ vom hypergeometrischen Typ erfüllt eine einfache Differentialgleichung der Form (2.3-2).

BEWEIS von Satz 2.4-2:

Sei $f$ dargestellt durch $F := \sum_{k=k_0}^{\infty} a_k z^k$ mit $a_{k_0} \neq 0$. Für $\theta = z\frac{d}{dz}$ ergibt sich $\theta F = \sum_{k=k_0}^{\infty} k a_k z^k$ sowie mit $n \in \mathbb{N}$: $\theta^n F = \sum_{k=k_0}^{\infty} k^n a_k z^k$.

Für ein beliebiges Polynom $T$ gilt also:

$$\text{(2.4-2)} \qquad T(\theta)F = \sum_{k=k_0}^{\infty} T(k) a_k z^k .$$

Die Koeffizienten von $F$ seien gekennzeichnet durch $a_{k+m} = R(k)a_k = \frac{P(k)}{Q(k)}a_k$, wobei $k \in \mathbb{N}$ sowie $P$ und $Q$ Polynome sind, also gilt:

$$\text{(2.4-3)} \qquad Q(k)a_{k+m} = P(k)a_k .$$

O.b.d.A. seien $P$ und $Q$ so gewählt, daß

$$Q(k_0-1) = Q(k_0-2) = \ldots = Q(k_0-m) = 0 ,$$

da für $j \in \{1, \ldots, m\}$ mit $Q(k_0-j) \neq 0$, die Polynome

$$Q^*(k) := Q(k)(k-k_0+j) \quad \text{und} \quad P^*(k) := P(k)(k-k_0+j)$$

die Gleichung $Q^*(k)a_{k+m} = P^*(k)a_k$ erfüllen und $Q^*(k_0-j) = 0$ ist.

Jetzt kann für $F$ eine Differentialgleichung aufgestellt werden:

$$\begin{aligned}
Q(\theta-m)F &= \sum_{k=k_0}^{\infty} Q(k-m)a_k z^k , && \text{mit (2.4-2)} \\
&= \sum_{k=k_0+m}^{\infty} Q(k-m)a_k z^k , && \text{da } Q(k_0-1) = \ldots = Q(k_0-m) = 0 \\
&= \sum_{k=k_0}^{\infty} Q(k)a_{k+m} z^{k+m} , && \text{nach Indexverschiebung}
\end{aligned}$$

$$= z^m \sum_{k=k_0}^{\infty} Q(k) a_{k+m} z^k$$

$$= z^m \sum_{k=k_0}^{\infty} P(k) a_k z^k, \qquad \text{mit (2.4-3)}$$

$$= z^m P(\theta) F, \qquad \text{mit (2.4-2).}$$

Es ergibt sich damit eine Differentialgleichung der Form (2.3-2), deren Ordnung maximal $\max\{\operatorname{Grad}(Q)+1, \operatorname{Grad}(P)\}$ beträgt. ❑

Der Beweis von Satz 2.4-2 bringt die Differentialgleichung einer Funktion vom hypergeometrischen Typ in Zusammenhang mit der Rekursionsgleichung ihrer Potenzreihenkoeffizienten. Diese Rekursionsgleichung ist jedoch nur eine Verallgemeinerung der Rekursionsgleichung bei hypergeometrischen Reihen.

<u>LEMMA 2.4-3:</u>

Eine Funktion vom hypergeometrischen Typ $f$ läßt sich auch als Summe von $m$ “$m$-fach symmetrischen” Reihen darstellen.

<u>BEWEIS von Lemma 2.4-3:</u>

O. B. d. A. sei $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ und $a_{k+m} = R(k) a_k$.

Durch $a_{k_0}^{(j)} := a_j$, $a_k^{(j)} := a_{mk+j}$ für $j = 0, \ldots, m-1$

wird $f(z)$ durch $m$ Reihen $f_{(j)}(z) = \sum_{k=k_0}^{\infty} a_k^{(j)} z^{mk+j}$ dargestellt:

$$\text{(2.4-4)} \qquad \sum_{k=k_0}^{\infty} a_k z^k = \sum_{j=0}^{m-1} \sum_{k=k_0}^{\infty} a_{mk+j} z^{mk+j}$$

$$= \sum_{k=k_0}^{\infty} a_{mk} z^{mk} + \sum_{k=k_0}^{\infty} a_{mk+1} z^{mk+1} + \ldots + \sum_{k=k_0}^{\infty} a_{mk+(m-1)} z^{mk+(m-1)}.$$

Nun gilt: $a_{k+1}^{(j)} = a_{m(k+1)+j} = a_{(mk+j)+m}$

$$= R(mk+j) a_{mk+j} = R(mk+j) a_k^{(j)}$$

mit $R(mk+j)$ rational, d. h. $f_{(j)}(z) = z^j \sum_{k=k_0}^{\infty} a_k^{(j)} z^{mk}$ sind hypergeometrische Reihen, die bei $z^m$ ausgewertet und deren Koeffizienten um $j$ Stellen verschoben wurden. ❑

Also wird $f$ auf $m$ hypergeometrische Reihen zurückgeführt und $f$ kann in eine Potenzreihe direkt entwickelt werden. Für die Entwicklung einer Funktion in eine Potenzreihe macht es damit Sinn, zu prüfen, ob die Funktion vom hypergeometrischen Typ ist. Der Satz 2.4-2 zeigt dabei, daß diese Untersuchung durch Einbeziehung von Differentialgleichungen unterstützt wird.

Die elementar-transzendenten Funktionen (2.2-1) bis (2.2-11) sind alle vom hypergeometrischen Typ, wie bereits die folgenden Beziehungen zeigen:

$$(1+z)^{\alpha} = {}_1F_0(-\alpha|z) \ ,$$

$$\exp(z) = {}_0F_0(z) \ , \qquad \ln(1+z) = -z \cdot {}_2F_1\left(\begin{matrix} 1 & & 1 \\ & 2 & \end{matrix}\middle| -z\right),$$

$$\sin(z) = z \cdot {}_0F_1\left(3/2 \ \middle| \ \frac{-z^2}{4}\right) \ , \qquad \cos(z) = {}_0F_1\left(1/2 \ \middle| \ \frac{-z^2}{4}\right) \ ,$$

$$\sinh(z) = z \cdot {}_0F_1\left(3/2 \ \middle| \ \frac{z^2}{4}\right) \ , \qquad \cosh(z) = {}_0F_1\left(1/2 \ \middle| \ \frac{z^2}{4}\right) \ ,$$

$$\arcsin(z) = z \cdot {}_2F_1\left(\begin{matrix} 1/2 & & 1/2 \\ & 3/2 & \end{matrix}\middle| z^2\right), \qquad \arctan(z) = z \cdot {}_2F_1\left(\begin{matrix} 1/2 & & 1 \\ & 3/2 & \end{matrix}\middle| -z^2\right),$$

$$\operatorname{arcsinh}(z) = z \cdot {}_2F_1\left(\begin{matrix} 1/2 & & 1/2 \\ & 3/2 & \end{matrix}\middle| -z^2\right), \qquad \operatorname{arctanh}(z) = z \cdot {}_2F_1\left(\begin{matrix} 1/2 & & 1 \\ & 3/2 & \end{matrix}\middle| z^2\right).$$

Ein Nachweis wird hier exemplarisch für $\sin(z)$ gegeben.

<u>BEISPIEL 2.4-4:</u>

$$\begin{aligned}
z \cdot {}_0F_1\left(3/2 \ \middle| \ \frac{-z^2}{4}\right) &= z \cdot \sum_{k=0}^{\infty} \frac{1}{\left(\frac{3}{2}\right)_k k!}\left(-\frac{z^2}{4}\right)^k \\
&= \sum_{k=0}^{\infty} \frac{(-1)^k}{\left(\frac{3}{2}\right)_k 4^k k!} z^{2k+1} \\
&= \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} z^{2k+1} \\
&= \sin(z) \ .
\end{aligned}$$

Die vorletzte Gleichung gilt wegen:

$$\left(\frac{3}{2}\right)_k = \frac{3}{2}\left(\frac{3}{2}+1\right)\left(\frac{3}{2}+2\right)\ldots\left(\frac{3}{2}+(k-1)\right) = \left(1+\frac{1}{2}\right)\left(2+\frac{1}{2}\right)\ldots\left(k+\frac{1}{2}\right) \qquad ,$$

also $\quad 2^k\left(\frac{3}{2}\right)_k = (2+1)(4+1)\ldots(2k+1) = 3\cdot 5\cdot\ldots\cdot(2k+1) = \prod_{j=1}^{k}(2j+1) \quad ,$

sowie

$$2^k k! = 2^k(1\cdot 2\cdot\ldots\cdot k) = 2\cdot 4\cdot\ldots\cdot(2k) = \prod_{j=1}^{k}(2j) \ .$$

Mit $\prod_{j=1}^{k}(2j+1)\cdot\prod_{j=1}^{k}(2j) = (2k+1)!$ folgt:

$$\left(2^k\left(\frac{3}{2}\right)_k\right)\cdot(2^k k!) = (2k+1)! \quad .$$

# 3 Methoden zur algorithmisierten Entwicklung von Funktionen in formale Laurentreihen

## 3.1 Überblick

In Kapitel 2 ist bei Polynomen, rationalen Funktionen und insbesondere Funktionen vom hypergeometrischen Typ erkannt worden, daß sie sich grundsätzlich in ihre Potenzreihen entwickeln lassen. Die methodischen Schritte dieser Entwicklung sind aber abhängig von der Ausgangsfunktion und teilweise sehr vielfältig.

Rationale Funktionen erlauben einen direkten Weg der Entwicklung in eine Potenzreihe. Die einfachsten Fälle werden unter Benutzung der geometrischen Reihe bzw. Binominalreihe gelöst:

$$\frac{1}{1-z} = \sum_{k=0}^{\infty} z^k ,$$

$$\frac{1}{(1-z)^j} = \sum_{k=0}^{\infty} \binom{j+k-1}{k} z^k .$$

Der allgemeine Fall einer rationalen Funktion $R(z) = \frac{P(z)}{Q(z)}$, $P$ und $Q$ Polynome, soll daher auf diese Grundformen zurückgeführt werden. Dafür muß $R(z)$ in ein Polynom und einen echt gebrochen-rationalen Anteil aufgeteilt werden. Polynome werden hier also als Spezialfall einer rationalen Funktion angesehen. Für echt gebrochen-rationale Funktionen $R(z) = \frac{r(z)}{Q(z)}$, $r$ und $Q$ Polynome, existiert dabei genau eine Darstellung [W90, S. 171]:

$$\frac{r(z)}{Q(z)} = \sum_{t=1}^{m} \left( \frac{c_{t1}}{z-z_t} + \frac{c_{t2}}{(z-z_t)^2} + \ldots + \frac{c_{tq_t}}{(z-z_t)^{q_t}} \right) = \sum_{t=1}^{m} \sum_{j=1}^{q_t} \frac{c_{tj}}{(z-z_t)^j} .$$

Diese Methode zur Entwicklung rationaler Funktionen in ihre Potenzreihe wird als “rationaler Ansatz” in Kapitel 3.2 ausgeführt.

Liegt keine rationale Funktion vor, wird die Entwicklung in eine formale Laurentreihe aufwendiger. Eine Lösung ist für Funktionen vom hypergeometrischen Typ prinzipiell möglich (vergleiche Lemma 2.4-3). Nach Satz 2.4-2 stellt die Existenz einer einfachen Differentialgleichung der Form (2.3-2) bereits ein erstes Kriterium für die Frage, ob eine Funktion vom hypergeometrischen Typ ist. Ein Algorithmus zur Konstruktion dieser Differentialgleichung wird in Kapitel 3.3 vorgestellt. Definitionsgemäß wird eine Funktion vom hypergeometrischen Typ an ihrer Reihenkoeffizientenformel $a_{k+m} = R(k)a_k$ erkannt und auch erst mit deren Hilfe in die gesuchte formale Laurentreihe entwickelt. Die Aufstellung der rekursiven Koeffizientenformel ist direkt möglich durch das Einsetzen von $(k+1-j)_n a_{k+n-j}$ für $z^j f^{(n)}$ in der Differentialgleichung (siehe “hypergeometrischer Ansatz” in Kapitel 3.4).

Nach Erstellung der Rekursionsformel für die Reihenkoeffizienten zeigt sich, daß die zu entwickelnde Funktion natürlich nicht immer vom hypergeometrischen Typ ist. Sofern die gefundene Differentialgleichung von $f$ aber die Form

$$\sum_{n=0}^{N} p_n f^{(n)} = 0, \text{ mit } p_n \in \mathbb{C},$$

hat, d. h. konstante Koeffizienten besitzt, kann für die Reihenkoeffizienten die Rekursionsgleichung

$$\sum_{n=0}^{N} p_n b_{k+n} = 0, \text{ mit } b_k = k! a_k,$$

aufgestellt werden. Eine solche Folge $b_0, b_1, \ldots$ heißt auch “rekurrente Reihe” der Ordnung $N$ [W90, S. 35]. Bei festem $k$, $N$ kann ein lineares Gleichungssystem aufgestellt werden, mit dessen Hilfe alle $b_k$ und damit auch alle $a_k$ mit $k \in \mathbb{N}_0$ durch die $p_n$ und die Anfangswerte $a_0, \ldots, a_{N-1}$ allgemein auf algebraischem Weg bestimmt sind. Ein entsprechender Beweis wird unter Einbeziehung gewöhnlicher Differentialgleichungen geführt (siehe “rekurrenter Ansatz” in Kapitel 3.5).

## 3.2 Rationaler Ansatz

Der rationale Ansatz geht bei der zu entwickelnden Funktion aus von der Form:

$$R(z) = \frac{P(z)}{Q(z)} \text{ mit } P(z) = \sum_{k=0}^{k_p} p_k z^k \text{ und } Q(z) = \sum_{k=0}^{k_q} q_k z^k .$$

Falls $R(z)$ keine echt gebrochen-rationale Funktion ist, $\mathrm{Grad}(P) \geq \mathrm{Grad}(Q)$, kann sie durch Polynomdivision auf einen polynomialen Teil $p(z)$ und einen echt gebrochen-rationalen Anteil $\frac{r(z)}{Q(z)}$ zurückgeführt werden. Mit Satz 3.2-1 (vgl. [W90, S. 171]) kann $\frac{r(z)}{Q(z)}$ in eindeutiger Weise als Summe seiner Partialbrüche geschrieben werden. Damit entsteht für $\frac{r(z)}{Q(z)}$ eine Darstellung:

$$\frac{r(z)}{Q(z)} = \sum_{t=1}^{m} \sum_{j=1}^{q_t} \frac{c_{tj}}{(z - z_t)^j} .$$

Die auftretenden Summanden werden unter Benutzung der Binominalreihe auf Potenzreihen zurückgeführt:

$$\text{(3.2-1)} \qquad \frac{c}{(z-z_t)^j} = \frac{(-1)^j c}{z_t^j} \frac{1}{\left(1 - \frac{z}{z_t}\right)^j}$$

$$= \frac{(-1)^j c}{z_t^j} \sum_{k=0}^{\infty} \binom{j+k-1}{k} \left(\frac{z}{z_t}\right)^k$$

$$= \sum_{k=0}^{\infty} \frac{(-1)^j c}{z_t^{j+k}} \binom{j+k-1}{k} z^k .$$

<u>SATZ 3.2-1:</u>

Jede echt gebrochen-rationale Funktion läßt sich genau auf eine Weise als Summe ihrer Partialbrüche schreiben.

<u>BEWEIS von Satz 3.2-1:</u>

Gegeben sei die echt gebrochen-rationale Funktion $\frac{P(z)}{Q(z)}$.

O. B. d. A. sei das komplexe Polynom $Q(z)$ so normiert, daß es dabei im Komplexen die Produktdarstellung $Q(z) = (z-\zeta_1)^{q_1}\dots(z-\zeta_m)^{q_m}$ habe, wobei $\zeta_j \in \mathbb{C}$ die $m$ verschiedenen Nullstellen von $Q(z)$ sind und $q_t \in \mathbb{N}$ deren Vielfachheit angeben. Die Existenz dieser Zerlegung wird durch den Fundamentalsatz der Algebra [Ko83, S. 232] gewährleistet.

Mit Beweis durch vollständige Induktion über $n = \mathrm{Grad}(Q)$ wird gezeigt, daß $a_{tj} \in \mathbb{C}$ existieren damit

$$\text{(3.2-2)} \qquad \frac{P(z)}{Q(z)} = \sum_{t=1}^{k}\left(\frac{a_{t1}}{z-\zeta_t} + \dots + \frac{a_{tq_t}}{(z-\zeta_t)^{q_t}}\right)$$

gilt. Für $n = 1$ (Induktionsanfang) ist $P(z)$ konstant wegen $\mathrm{Grad}(P) < \mathrm{Grad}(Q)$. Die Induktionsannahme geht davon aus, daß die Darstellung (3.2-2) möglich ist für $\mathrm{Grad}(Q) = n_0 - 1 \geq 1$.

Zu $\mathrm{Grad}(Q) = n_0$ sei $\zeta$ eine Nullstelle von $Q(z)$ mit der Ordnung $q$, also $Q(z) = (z-\zeta)^q S(z)$ mit $S(\zeta) \neq 0$ und $q \geq 1$. Zu dieser Nullstelle $\zeta$ kann ein Polynom $T(z)$ aufgestellt werden mit:

$$\begin{aligned}\frac{P(z)}{S(z)} - \frac{P(\zeta)}{S(\zeta)} &= \frac{P(z)S(\zeta) - P(\zeta)S(z)}{S(z)S(\zeta)} \\ &= \frac{(z-\zeta)T(z)}{S(z)}, \qquad \text{da } P(z)S(\zeta) - P(\zeta)S(z) = 0 \text{ für } z = \zeta.\end{aligned}$$

Wegen $\mathrm{Grad}(P) \leq \mathrm{Grad}(S) \leq n_0 - 1$ gilt insbesondere $\mathrm{Grad}(T) \leq n_0 - 2$.

Mit $$\frac{1}{(z-\zeta)^j}\left(\frac{P(z)}{S(z)} - \frac{P(\zeta)}{S(\zeta)}\right) = R(z) - \frac{P(\zeta)}{(z-\zeta)^j S(\zeta)}$$

und $$\frac{1}{(z-\zeta)^j}\frac{(z-\zeta)T(z)}{S(z)} = \frac{T(z)}{(z-\zeta)^{j-1}S(z)}$$

ist $$R(z) = \frac{T(z)}{(z-\zeta)^{j-1}S(z)} + \frac{P(\zeta)/S(\zeta)}{(z-\zeta)^j}.$$

Damit gibt es für $R(z)$ eine Darstellung (3.2-2), denn wegen $\mathrm{Grad}(T) \leq n_0 - 2$ und $\mathrm{Grad}((z-\zeta)^{j-1}S(z)) = n_0 - 1$ existiert nach Induktionsannahme die Darstellung (3.2-2) auch für $\frac{T(z)}{(z-\zeta)^{j-1}S(z)}$.

Zum Beweis der Eindeutigkeit von (3.2-2) werden zwei Zerlegungen angenommen:

$$\sum_{t=1}^{m}\left(\frac{a_{t1}}{z-\zeta_t}+\ldots+\frac{a_{tq_t}}{(z-\zeta_t)^{q_t}}\right) \quad \text{und} \quad \sum_{t=1}^{m}\left(\frac{b_{t1}}{z-\zeta_t}+\ldots+\frac{b_{tq_t}}{(z-\zeta_t)^{q_t}}\right) .$$

Für alle $l = 1, \ldots, m$ gilt: Durch Multiplikation mit $(z-\zeta_l)^{q_l}$ und anschließender Grenzwertbildung $z \to \zeta_l$ folgt $a_{lq_l} = b_{lq_l}$ und daraufhin analog und sukzessiv $a_{l,q_l-1} = b_{l,q_l-1}$, $\ldots$, $a_{l1} = b_{l1}$. ❑

Analog zum Beweis von Satz 3.2-1 vollzieht der folgende Algorithmus eine komplexe Partialbruchzerlegung.

<u>ALGORITHMUS 3.2-2:</u>

Gegeben ist die rationale Funktion $R(z) = \dfrac{P(z)}{Q(z)}$ .

(a) Zerlege $R(z)$ ggf. durch Polynomdivision in einen polynomialen und einen echt gebrochen-rationalen Teil, so daß:

$$R(z) = p(z) + \frac{r(z)}{Q(z)} ,$$

mit $p(z)$ Polynom, $\operatorname{Grad}(p) = \operatorname{Grad}(P) - \operatorname{Grad}(Q) + 1$ und $r(z) = P(z) - Q(z)p(z)$ , also $\operatorname{Grad}(r) < \operatorname{Grad}(Q)$ .

(b) Berechne alle Nullstellen $\zeta_t$ von $Q(z)$ und ihre Vielfachheiten $q_t$ und erstelle:

$$R(z) = p(z) + \sum_{t=1}^{m}\sum_{j=1}^{q_t}\frac{c_{tj}}{(z-\zeta_t)^j} .$$

(c) Berechne die unbekannten Koeffizienten von $\dfrac{r(z)}{Q(z)}$

(vgl. - insbesondere wegen der Reihenfolge der Koeffizientenberechnung - mit dem Eindeutigkeitsbeweis von Satz 3.2-1):

[redacted]:

[redacted].

<u>BEISPIEL 3.2-3:</u>

$$f(z) = \frac{1}{z^2+2z+2} = \frac{1}{(z-(-1+i))(z-(-1-i))}$$

$$= \frac{c_{1,1}}{(z-(-1+i))} + \frac{c_{2,1}}{(z-(-1-i))} ,$$

da $\zeta_1 = -1+i$ und $\zeta_2 = -1-i$ Nullstellen des Nenners von $f$ sind mit den Vielfachheiten $q_1 = q_2 = 1$.

$$c_{1,1} = \lim_{z \to (-1+i)} \frac{z-(-1+i)}{(z-(-1+i))(z-(-1-i))} = \frac{1}{(-1+i)-(-1-i)} = \frac{1}{2i} = -\frac{i}{2} .$$

$$c_{2,1} = \lim_{z \to (-1-i)} \frac{z - (-1-i)}{(z - (-1+i))(z - (-1-i))} = \frac{1}{(-1-i) - (-1+i)} = \frac{1}{-2i} = \frac{i}{2} \quad .$$

Somit wird $\quad f(z) = \dfrac{-i/2}{z + (1-i)} + \dfrac{i/2}{z + (1+i)} = \dfrac{i}{2}\left(-\dfrac{1}{1-i}\dfrac{1}{1+\dfrac{z}{1-i}} + \dfrac{1}{1+i}\dfrac{1}{1+\dfrac{z}{1+i}}\right) \quad .$

Damit gilt für die Reihenkoeffizienten von $f(z) = \sum\limits_{k=0}^{\infty} a_k z^k$:

$$a_k = \frac{i}{2}\left(\left(-\frac{1}{1-i}\right)^{k+1} - \left(-\frac{1}{1+i}\right)^{k+1}\right) \quad .$$

## 3.3 Konstruktion einfacher Differentialgleichungen

Ausgangspunkt für die Aufstellung einer Rekursionsbeziehung der Reihenkoeffizienten von nicht-rationalen Funktionen und damit für die Entwicklung in eine Potenzreihe ist die Konstruktion einer einfachen Differentialgleichung der Form (2.3-2).

Mit dem hier gegebenen Algorithmus und Satz aus [K92, S. 14 - 15] wird für eine Funktion $f$ und ein vorgegebenes Maximum $N_{max}$ festgestellt, ob zu $f$ eine einfache Differentialgleichung der Ordnung $N \le N_{max}$ existiert.

Am Beispiel der Exponentialfunktion $f(z) = e^z$ läßt sich schnell zeigen, daß eine einfache Differentialgleichung eine weitere einfache Differentialgleichung anderer Ordnung nicht ausschließt: $e^z = f(z) = f'(z) = f''(z)$, also $f' - f = 0$ und $f'' - f' = 0$, aber auch $f'' - f' + 2(f' - f) = f'' + f' - 2f = 0$. Die letzte Differentialgleichung unterscheidet sich aufgrund ihrer Rekursionsgleichung der Koeffizienten wesentlich von den vorherigen Differentialgleichungen.

<u>ALGORITHMUS 3.3-1:</u>

Gegeben ist $f \ne 0$.

(a) Setze $N := 1$. Untersuche, ob $f(z)$ einer einfachen Differentialgleichung 1. Ordnung genügt, d. h. prüfe, ob $A_0(z)$ rational ist in:

(3.3-1) $\qquad f'(z) + A_0 f(z) = 0$

Ist $A_0 = \dfrac{f'(z)}{f(z)}$ rational, also $A_0 = \dfrac{P_0}{Q_0}$ mit $P_0$, $Q_0$ Polynome der Variablen $z$, lautet die gesuchte einfache Differentialgleichung:

$$Q_0 f'(z) + P_0 f(z) = 0 \, .$$

Ansonsten:

(b) Erhöhe $N$ um 1 und zerlege

(3.3-2) $\qquad f^{(N)}(z) + A_{N-1} f^{(N-1)}(z) + \ldots + A_0 f(z)$

in elementare Summanden (durch Ausmultiplizieren und Vereinfachen):

$\sum_{j=0}^{t} s_j + \sum_{n=0}^{N} \sum_{j=0}^{t_n} A_n s_{jn}$ mit $s_j, s_{jn}$ unabhängig von $A_n$. Betrachte $A_n$ als Konstanten und prüfe, ob es höchstens $N$ rational unabhängige Terme unter den $s_j$ und $A_n s_{jn}$ gibt.

( $a(z), b(z)$ heißen dabei rational abhängig, falls Polynome $P(z), Q(z)$ existieren mit $\frac{a}{b} = \frac{P}{Q}$.)

Ist dies nicht der Fall, wiederhole (b).

(c) Ansonsten gibt es eine eindeutige Lösung:
Sortiere nach den rational unabhängigen Termen, setze diese Terme gleich 0 und löse das entstehende lineare Gleichungssystem nach $A_0, \ldots, A_{N-1}$ auf. Die gesuchte einfache Differentialgleichung lautet:

$$K \cdot f^{(N)}(z) + K \sum_{n=0}^{N-1} A_n f^{(n)}(z) = 0 \text{ mit } K = kgV(A_0, \ldots, A_{N-1}).$$

<u>SATZ 3.3-2:</u>

Wird durch den Algorithmus 3.3-1 bis zum Schritt $N = N_{max}$ keine einfache Differentialgleichung für $f \neq 0$, $f(z)$ nicht-rational, gefunden, so existiert zu $f$ keine einfache Differentialgleichung mit der Ordnung $N \leq N_{max}$.

<u>BEWEIS von Satz 3.3-2:</u>

Gezeigt wird mit Induktionsbeweis, daß im Schritt $N = N_0$ entweder

(i) der Ausdruck (3.3-2) in höchstens $N_0$ rational unabhängige Summanden zerlegt werden kann und damit eine eindeutige Lösung einer einfachen Differentialgleichung existiert, oder

(ii) die Anzahl der rational unabhängigen Summanden größer als $N_0$ ist und keine einfache Differentialgleichung bis zur Ordnung $N_0$ existiert.

Für $N = 1$ existiert wegen $f$ nicht-rational mindestens ein rational unabhängiger Summand. Damit nun (i) erfüllt ist, muß $f'$ rational abhängig von $f$ sein, d. h. es existiert $P, Q$ mit $\frac{P}{Q} = \frac{f}{f'}$ und damit ist eine Lösung für $A_0$ nach Teil (a) vom Algorithmus 3.3-1 gefunden. Andernfalls gilt (ii).

Für $N = N_0 \geq 2$ sei die Anzahl der rational unabhängigen Summanden höchstens $N_0$. Dann existiert für $A_0, \ldots, A_{N_0-1}$ eine Lösung, da das entsprechende Gleichungssystem nach (c) von Algorithmus 3.3-1 höchstens $N_0$ Gleichungen besitzt. Diese Lösung ist eindeutig, da für zwei Lösungen $A_0, \ldots, A_{N_0-1}$ und $B_0, \ldots, B_{N_0-1}$ gilt:

$$\left.\begin{aligned} f^{(N)}(z) + \sum_{n=0}^{N-1} A_n f^{(n)}(z) &= 0 \\ f^{(N)}(z) + \sum_{n=0}^{N-1} B_n f^{(n)}(z) &= 0 \end{aligned}\right\} \Rightarrow \left( \sum_{n=0}^{N-1} (A_n - B_n) f^{(n)}(z) = 0 \right)$$

Damit existiert aber bereits eine einfache Differentialgleichung der Ordnung $N_0 - 1$ und dies wäre ein Widerspruch zur Induktionsanahme. Also gilt $A_n - B_n = 0$ bzw. $A_n = B_n$ für $n = 0, \ldots, N_0 - 1$, d. h. die Lösung $A_0, \ldots, A_{N_0 - 1}$ ist eindeutig. ❑

BEISPIEL 3.3-3:

$f(z) = e^z \sin(z)$.

Es ist $f'(z) = e^z(\sin(z) + \cos(z))$ und $f''(z) = 2e^z\cos(z)$.

Da $A_0 = \frac{f'}{f} = -(1 + \cot(z))$ nicht rational ist, wird der Ansatz

$$f'' + A_1 f' + A_0 f = 2e^z\cos(z) + A_1 e^z(\sin(z) + \cos(z)) + A_0 e^z \sin(z) = 0 \text{ gemacht.}$$

Die elementaren Summanden sind:

$2e^z\cos(z)$, $A_1 e^z \sin(z)$, $A_1 e^z \cos(z)$ und $A_0 e^z \sin(z)$.

Unter der Annahme, $A_0$ und $A_1$ sind konstant, gibt es nur die beiden rational unabhängigen Terme $e^z\cos(z)$ und $e^z\sin(z)$. Nach Algorithmus 3.3-1 gibt es also bereits eine eindeutige Lösung. Aus

$$(2 + A_1)e^z\cos(z) + (A_1 + A_0)e^z\sin(z) = 0$$

entsteht das Gleichungssystem:

$$\begin{aligned} 2 + A_1 &= 0 \\ A_1 + A_0 &= 0 \end{aligned}$$

Mit der Lösung $A_1 = -2$ und $A_0 = 2$ kann die Differentialgleichung aufgestellt werden:

$$f'' - 2f' + 2f = 0.$$

## 3.4 Hypergeometrischer Ansatz

Für alle Funktionen vom hypergeometrischen Typ ist es wegen Satz 2.4-2 sinnvoll, mit dem Algorithmus 3.3-1 nach einer einfachen Differentialgleichung zu suchen. Im nächsten Schritt gilt es, die gefundene einfache Differentialgleichung in eine Rekursionsgleichung für die Koeffizienten von $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ zu transformieren [K92, S. 16]:

LEMMA 3.4-1:

Sei $\sum_{j=0}^{Q}\sum_{n=0}^{Q} c_{jn} z^j f^{(n)}(z) = 0$ eine einfache Differentialgleichung einer Funktion $f(z)$. Dann wird durch die Substitution $z^j f^{(n)}(z) \to (k+1-j)_n a_{k+n-j}$ eine Rekursionsgleichung für die Reihenkoeffizienten von $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ aufgestellt.

BEWEIS von Lemma 3.4-1:

Für $f(z)$ wird
$$\begin{aligned} f^{(n)}(z) &= \sum_{k=k_0}^{\infty} k(k-1)\dots(k-n+1) a_k z^{k-n} \\ &= \sum_{k=k_0-n}^{\infty} (k+n)(k+n-1)\dots(k+1) a_{k+n} z^k \\ &= \sum_{k=k_0-n}^{\infty} (k+1)_n a_{k+n} z^k . \end{aligned}$$

Dies wird in der einfachen Differentialgleichung benutzt:

$$\begin{aligned} 0 &= \sum_{j=0}^{Q}\sum_{n=0}^{Q} c_{jn} z^j \left( \sum_{k=k_0-n}^{\infty} (k+1)_n a_{k+n} z^k \right) \\ &= \sum_{j=0}^{Q}\sum_{n=0}^{Q} c_{jn} z^j \sum_{k=k_0+j-n}^{\infty} (k-j+1)_n a_{k-j+n} z^{k-j} \\ &= \sum_{j=0}^{Q}\sum_{n=0}^{Q} \sum_{k=k_0+(j-n)}^{\infty} c_{jn} (k-j+1)_n a_{k-j+n} z^k \\ &= \sum_{j=0}^{Q}\sum_{n=0}^{Q} \sum_{k=k_0+(j-n)}^{k_0+Q-1} c_{jn} (k-j+1)_n a_{k-j+n} z^k \\ &\quad + \sum_{k=k_0+Q}^{\infty} \left( \sum_{j=0}^{Q}\sum_{n=0}^{Q} c_{jn} (k-j+1)_n a_{k-j+n} \right) z^k , \end{aligned}$$

wobei die Potenzen von $z$ in der vorletzten Zeile wegen $k = k_0 + (j-n) \le k_0 + Q - 1$ kleiner sind als $k_0 + Q$. Für die Koeffizienten $a_k$ in der letzten Zeile, also $k \ge k_0 + Q$, gilt durch Koeffizientenvergleich:

$$0 = \sum_{j=0}^{Q}\sum_{n=0}^{Q} c_{jn} (k-j+1)_n a_{k-j+n},$$

d. h. eine Rekursionsgleichung, die sich aus der einfachen Differentialgleichung mit der angegebenen Substitution direkt aufstellen läßt. Bei festem *j* und *n* ist $(k-j+1)_n$ ein Polynom nach der Variablen *k* und die Rekursionsgleichung hat die Form:

$$a_{k+Q} = \sum_{l=-Q}^{Q-1} r_l(k)a_{k+l}, \qquad \text{wobei } r_l(k) \text{ rationale Funktion.} \;❏$$

Sofern mit Lemma 3.4-1 eine Rekursionsgleichung einer Funktion vom hypergeometrischen Typ entsteht, ist die zu bestimmende rationale Funktion $R(k)$ der Gleichung (2.4-1) festgelegt. Wegen Lemma 2.4-3 genügt es, sich nun auf eine Rekursionsformel für eine hypergeometrische Reihe, also $m = 1$, zu beschränken. Damit verbleibt für die Entwicklung von Funktionen vom hypergeometrischen Typ wegen Lemma 2.3-1 die Aufgabe, $R(k) = \frac{P(k)}{Q(k)}$, $P$ und $Q$ sind Polynome, in die Form $\frac{(a_1+k)\dots(a_p+k)}{(b_1+k)\dots(b_q+k)(k+1)}$ umzuwandeln (ggf. muß mit $(k+1)$ erweitert werden).

<u>BEISPIEL 3.4-2:</u>

$$f(z) = \arcsin(z) = \sum_{k=0}^{\infty} a_k z^k.$$

Es gilt $f'(z) = \frac{1}{\sqrt{1-z^2}}$ und $f''(z) = \frac{z}{\sqrt{1-z^2}^3}$. Mit dem Algorithmus 3.3-1 ergibt sich für *f* die Differentialgleichung: $(1-z^2)f''-zf' = 0$.

Also besitzt *f* eine einfache Differentialgleichung, die mit dem hypergeometrischen Ansatz (Lemma 3.4-1) in eine Rekursionsgleichung überführt werden kann:

$$\begin{aligned} 0 &= (k+1)_2 a_{k+2} - (k-1)_2 a_k - (k)_1 a_k \\ &= (k+2)(k+1)a_{k+2} - k(k-1)a_k - ka_k \\ &= (k+2)(k+1)a_{k+2} - k^2 a_k \;, \end{aligned}$$

also $R(k) = \frac{k^2}{(k+2)(k+1)}$ .

Hier ist die Symmetriezahl $m = 2$, und *f* wird nach Lemma 2.4-3 aufgeteilt in:

$$f(z) = f_{(0)}(z) + f_{(1)}(z) = \sum_{k=0}^{\infty} a_k^{(0)} z^{2k} + \sum_{k=0}^{\infty} a_k^{(1)} z^{2k+1}, \text{ mit } a_k^{(0)} = a_{2k} \text{ und } a_k^{(1)} = a_{2k+1}.$$

Wegen $a_0^{(0)} = a_0 = f(0) = 0$, ist $f_{(0)}(z) \equiv 0$ (*f* ist eine ungerade Funktion).

Die Berechnung von $f_{(1)}(z)$ erfolgt mit:

$$a_{k+1}^{(1)} = R(2k+1)a_k^{(1)}\;, \quad R(2k+1) = \frac{(2k+1)^2}{(2k+3)(2k+2)} = \frac{(k+1/2)}{(k+3/2)(k+1)} \quad .$$

Durch Lemma 2.3-1 und $a_0^{(1)} = a_1 = \frac{f'(0)}{1!} = 1$ wird $a_k^{(1)} = \frac{\left(\frac{1}{2}\right)_k \left(\frac{1}{2}\right)_k}{\left(\frac{3}{2}\right)_k k!}$ .

Insgesamt folgt $f(z) = \sum_{k=0}^{\infty} \frac{(1/2)_k (1/2)_k}{(3/2)_k \; k!} z^{2k+1}$ .

## 3.5 Rekurrenter Ansatz

Sofern die mit dem Algorithmus 3.3-1 konstruierte einfache Differentialgleichung $\sum_{n=0}^{N} c_n f^{(n)} = 0$ konstante Koeffizienten $c_n \in \mathbb{C}$ besitzt, kann eine Rekursionsgleichung für die Reihenkoeffizienten direkt abgelesen werden:

LEMMA 3.5-1:

Sei $\sum_{n=0}^{N} c_n f^{(n)} = 0$ Differentialgleichung von $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ und $c_n \in \mathbb{C}$ für $n = 0, \ldots, N$.

Dann gilt für die Reihenkoeffizienten von $g(z) = \sum_{k=k_0}^{\infty} (k! a_k) z^k = \sum_{k=k_0}^{\infty} b_k z^k$ die Rekursionsgleichung:

$$\text{(3.5-1)} \qquad \sum_{n=0}^{N} c_n b_{k+n} = 0.$$

BEWEIS von Lemma 3.5-1:

Gegeben sei $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$.

Der Beweis läuft zunächst wie bei Lemma 3.4-1, so daß $f^{(n)}(z) = \sum_{k=k_0-n}^{\infty} (k+1)_n a_{k+n} z^k$ für $n \in \mathbb{N}_0$ gilt und mit der Differentialgleichung von $f$ wird:

$$\begin{aligned} 0 &= \sum_{n=0}^{N} \sum_{k=k_0-n}^{\infty} c_n (k+1)_n a_{k+n} z^k \\ &= \sum_{n=0}^{N} \sum_{k=k_0-n}^{k_0-1} c_n (k+1)_n a_{k+n} z^k + \sum_{k=k_0}^{\infty} \left( \sum_{n=0}^{N} c_n (k+1)_n a_{k+n} \right) z^k . \end{aligned}$$

Die Potenzen von $z$ der ersten Doppelsumme der letzten Zeile sind kleiner als $k_0$, so daß durch Koeffizientenvergleich die Koeffizienten der zweiten Doppelsumme Null werden. Diese Rekursionsgleichung ergibt nach Multiplikation mit $k!$ bereits:

$$\sum_{n=0}^{N} c_n (k+n)! a_{k+n} = 0 .$$

Nun ist klar, daß mit $b_k = k! a_k$ die Rekursionsgleichung

$$\sum_{n=0}^{N} c_n b_{k+n} = 0$$

direkt aus der Differentialgleichung von $f(z)$ abgelesen werden kann. ❑

Da die erstellte Rekursionsgleichung im allgemeinen der Form (2.4-1) nicht entspricht, kann der hypergeometrische Ansatz nicht angewandt werden. Dafür sind die $c_n \in \mathbb{C}$ und keine Polynome von *n*, so daß bei festem $k^* \in \mathbb{N}$ für das lineare Gleichungssystem mit den $(k^* + 1)$ Gleichungen

$$\sum_{n=0}^{N} c_n b_{0+n} = 0, \ldots, \sum_{n=0}^{N} c_n b_{k^*+n} = 0$$

die $(k^* + N + 1)$ Koeffizienten $b_0, \ldots, b_{N-1}, b_N, \ldots, b_{k^*+N}$ unbestimmt sind. Unter Einbeziehung der Anfangswerte $b_0, \ldots, b_{N-1}$ bleiben $(k^* + 1)$ unbestimmte Koeffizienten, und das Gleichungssystem wird wegen Satz 3.5-3 mit dem Algorithmus 3.5-2 allgemein lösbar. Nach Aufstellung einer allgemeinen Formel für die $b_k$ ist $f(z) = \sum_{k=k_0}^{\infty} a_k z^k = \sum_{k=k_0}^{\infty} \frac{b_k}{k!} z^k$ eindeutig festgelegt.

<u>ALGORITHMUS 3.5-2:</u>

(a) Die Reihenglieder $b_k$ der rekurrenten Reihe werden in ihrer Rekursionsformel (3.5-1) mit $b_k := \lambda^k$ ersetzt, damit wird $\sum_{n=0}^{N} c_n \lambda^{k+n} = \lambda^k \left( \sum_{n=0}^{N} c_n \lambda^n \right) = 0$.

(b) Bestimme die (paarweise verschiedenen) Nullstellen $\lambda_1, \ldots, \lambda_l$, $l \le N$, des (charakteristischen) Polynoms

$$\text{(3.5-2)} \qquad P(z) = \sum_{n=0}^{N} c_n \lambda^n .$$

(c) Falls $l < N$:
(3.5-2) besitze genau die (paarweise verschiedenen) mehrfachen Nullstellen $\mu_1, \ldots, \mu_m \in \{\lambda_1, \ldots, \lambda_l\}$ mit ihren Vielfachheiten $q_1, \ldots, q_m \ge 2$. Dann seien $\lambda_1^*, \ldots, \lambda_{N-l}^*$ genau die Elemente der Menge $\left\{ (k+1-\nu_j)_{\nu_j} \cdot \mu_j^k \;\middle|\; j=1,\ldots m;\ \nu_j=1,\ldots,(q_j-1) \right\}$.

(d) Zu gegebenen Anfangswerten $b_0, \ldots, b_{N-1}$ werden im linearen Gleichungssystem

$$\text{(3.5-3)} \qquad b_k = z_1 \lambda_1^k + \ldots + z_l \lambda_l^k + z_{l+1} \lambda_1^* + \ldots + z_N \lambda_{N-l}^*,\ k = 0, \ldots, N-1$$

die unbekannten Koeffizienten $z_1, \ldots, z_N \in \mathbb{C}$ berechnet.

Auf der rechten Seite von (3.5-3) stehen nun keine Unbekannten mehr, und damit können alle $b_k$ mit (3.5-3) ermittelt werden.

SATZ 3.5-3:

Bei gegebenen Anfangswerten $b_0, \dots, b_{N-1}$ einer rekurrenten Reihe existiert eine eindeutige Darstellung für alle Reihenglieder $b_k \in \mathbb{N}_0$ mit der Form (3.5-3).

BEWEIS von Satz 3.5-3:

Der Beweis wird zunächst für den Fall $l = N$ mit algebraischen Mitteln geführt.

Eine rekurrente Reihe $\sum_{k=0}^{\infty} b_k z^k$ sei gegeben durch die Koeffizienten $c_k$ ihrer Rekursionsformel (3.5-1). Mit den (paarweise verschiedenen) Nullstellen $\lambda_1, \dots, \lambda_N$ des (charakteristischen) Polynoms (3.5-2) sei $b_k = \sum_{j=1}^{N} z_j \lambda_j^k$. Für $k = 0, \dots, N-1$ werden nun die Unbekannten $z_1, \dots, z_N$ bestimmt durch das lineare Gleichungssystem:

$$\left(\begin{array}{rcl} b_0 & = & \sum_{j=1}^{N} z_j \lambda_j^0 \\ & \dots & \\ b_{N-1} & = & \sum_{j=1}^{N} z_j \lambda_j^{N-1} \end{array}\right) \Leftrightarrow (b_0, \dots, b_{N-1}) = (z_1, \dots, z_N) \cdot V \text{, mit } V = \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{N-1} \\ 1 & \lambda_2 & \dots & \lambda_2^{N-1} \\ & & \dots & \\ 1 & \lambda_N & \dots & \lambda_N^{N-1} \end{pmatrix}.$$

$\mathrm{Det}(V)$ ist dabei die Vandermonde'sche Determinante, so daß nach [Ko83, S. 126]:

$$\mathrm{Det}(V) = \prod_{1 \le i < j \le N} (\lambda_j - \lambda_i) \neq 0 .$$

Damit ist $V$ regulär und wegen $(z_1, \dots, z_N) = (b_0, \dots, b_{N-1}) \cdot V^{-1}$, können $b_0, \dots, b_{N-1}$ mit (3.5-3) eindeutig berechnet werden.

Die Rekursionsgleichung (3.5-1) wird mit (3.5-3) durch Vorgabe von $b_0, \dots, b_{N-1}$ für $k = 0$ bzw. durch Induktion für alle $n \in \mathbb{N}_0$ erfüllt, da:

$$0 = \sum_{n=0}^{N} c_n b_{k+n}$$

$$= c_N b_{k+N} + \sum_{n=0}^{N-1} c_n \left( \sum_{j=1}^{N} z_j \lambda_j^{k+n} \right),$$

nach Induktionsvoraussetzung sind $b_0, \dots, b_{k+N-1}$ durch (3.5-3) dargestellt,

$$= c_N b_{k+N} + \sum_{j=1}^{N} z_j \left( \sum_{n=0}^{N-1} c_n \lambda_j^n \right) \lambda_j^k$$

$$= c_N b_{k+N} + \sum_{j=1}^{N} z_j(-c_N\lambda_j^N)\lambda_j^k, \qquad \text{da } \lambda_1, \ldots, \lambda_N \text{ Nullstellen sind von (3.5-2),}$$

$$= c_N b_{k+N} - c_N\left(\sum_{j=1}^{N} z_j\lambda_j^{k+N}\right)$$

$$= c_N\left(b_{k+N} - \sum_{j=1}^{N} z_j\lambda_j^{k+N}\right).$$

Nach Voraussetzung hat das charakteristische Polynom (3.5-2) $N$ paarweise verschiedene Nullstellen, d. h. es gilt $c_N \neq 0$, so daß (3.5-3) auch für $b_{k+N}$ gilt (Induktionsschluß) und damit allgemein für $b_k$ mit $k \geq N$, $k \in \mathbb{N}_0$.

Für $l < N$ ist es mit rein algebraischen Mitteln nicht leicht zu zeigen, daß für (3.5-3) eine eindeutige Lösung $(z_1, \ldots, z_N)$ existiert. Das liegt z. T. am Nachweis $\mathrm{Rg}(V^*) = N$ für eine analog zur o. g. Matrix $V$ aufzustellenden Matrix $V^*$. Zum allgemeinen Beweis wird das Problem daher in die Theorie der gewöhnlichen Differentialgleichungen übertragen und mit darin bekannten Ergebnissen gelöst.

Aus Lemma 3.5-1 folgt, daß die Berechnung einer allgemeinen Formel für die $b_k$ der Rekursionsgleichung (3.5-1) bei gegebenen Koeffizienten $b_0, \ldots, b_{N-1}$ der Lösung der homogenen linearen Differentialgleichung $N$-ter Ordnung mit konstanten Koeffizienten einer meromorphen Funktion $u(z)$ entspricht:

$$\text{(3.5-4)} \qquad \sum_{n=0}^{N} c_n u^{(n)}(z) = 0\text{, wobei } u(z) = \sum_{k=k_0}^{\infty} \frac{b_k}{k!} z^k$$

$$\text{mit den Anfangswerten } u^{(n)}(0) = b_{k_0+n}\text{, } n = 0, \ldots, N-1.$$

Zunächst halten wir fest, daß eine Lösung des Anfangswertproblems (3.5-4) stets existiert sowie eindeutig ist [W90a, S. 132] und jede Lösung sich als Linearkombination von $N$ linear unabhängigen Lösungen $u_1(z), \ldots, u_N(z)$ darstellen läßt [W90a, S. 144], also existieren $z_1, \ldots, z_N \in \mathbb{C}$ mit

$$\text{(3.5-5)} \qquad u(z) = z_1 u_1(z) + \ldots + z_N u_N(z) \;.$$

Die $N$ linear unabhängigen Lösungen werden ein "Hauptsystem" von Lösungen genannt. Die linear unabhängigen Lösungen von (3.5-4) können aus den Nullstellen des charakteristischen Polynoms (3.5-2) bestimmt werden [W90a, S. 137]:

> "Ist $\lambda$ eine $q$-fache Nullstelle des charakteristischen Polynoms, so entsprechen ihr $q$ Lösungen
>
> $$e^{\lambda z}, ze^{\lambda z}, \ldots, z^{q-1}e^{\lambda z}$$
>
> der Differentialgleichung. Aus den $N$ Nullstellen des charakteristischen Polynoms $P(\lambda)$ (jede mit ihrer Vielfachheit gezählt) ergeben sich auf diese

Weise $N$ linear unabhängige Lösungen, also ein Hauptsystem."

O. B. d. A. sei nun ein Hauptsystem mit genau einer $q$-fachen $(q > 1)$ Nullstelle $\lambda_1$, also $l{:=}N-(q-1)$ verschiedenen Nullstellen $\lambda_1, \ldots, \lambda_l$ vorausgesetzt. In (3.5-5) seien nun:

(3.5-6) $$u_j(z) = e^{\lambda_j z} = \sum_{k=0}^{\infty} \frac{\lambda_j^k}{k!} z^k, \text{ für } j = 1, \ldots, l \text{ und}$$

(3.5-7) $$u_j(z) = z^{j-l} e^{\lambda_1 z} = \sum_{k=0}^{\infty} \frac{\lambda_1^k}{k!} z^{k+(j-l)}, \text{ für } j = l+1, \ldots, N.$$

Mit $u_j(z) = \sum_{k=0}^{\infty} {}_{(j)}b_k \frac{z^k}{k!}$ ergibt sich für $j = 1, \ldots, l$: ${}_{(j)}b_k = \lambda_j^k$ und für $j = l+1, \ldots, N$ wird gezeigt, daß:

(3.5-8) $${}_{(j)}b_k = (k+1-(j-l))_{j-l} \cdot \lambda_1^{k-(j-l)}, \text{ für } k \geq j-l, \text{ sowie}$$
$${}_{(j)}b_0 = \ldots = {}_{(j)}b_{(j-l)-1} = 0 .$$

Es folgt der Beweis von (3.5-8) durch vollständige Induktion über $j = l+m$ mit $m \in \mathbb{N}_0$:

$$\begin{aligned} u_{l+1}(z) &= \sum_{k=0}^{\infty} \frac{\lambda_1^k}{k!} z^{k+1} \\ &= \sum_{k=1}^{\infty} \frac{\lambda_1^{k-1}}{(k-1)!} z^k \\ &= \sum_{k=1}^{\infty} k\lambda_1^{k-1} \frac{z^k}{k!} \\ &= \sum_{k=1}^{\infty} {}_{(l+1)}b_k \frac{z^k}{k!} \quad \text{(Induktionsanfang für } j = l+1\text{) und} \end{aligned}$$

$$\begin{aligned} u_{l+(m_0+1)}(z) &= z \cdot u_{l+m_0}(z) = z \cdot \sum_{k=m_0}^{\infty} {}_{(l+m_0)}b_k \cdot \frac{z^k}{k!} \\ &= \sum_{k=m_0}^{\infty} (k+1-m_0)_{m_0} \cdot \lambda_1^{k-m_0} \frac{z^{k+1}}{k!} \qquad \text{(Induktionsannahme für } j = l+m_0\text{)} \\ &= \sum_{k=m_0+1}^{\infty} (k-m_0)_{m_0} \cdot \lambda_1^{k-(m_0+1)} \frac{z^k}{(k-1)!} \\ &= \sum_{k=m_0+1}^{\infty} k(k-m_0)_{m_0} \cdot \lambda_1^{k-(m_0+1)} \frac{z^k}{k!} \end{aligned}$$

$$= \sum_{k = m_0 + 1}^{\infty} (k - m_0)_{m_0 + 1} \cdot \lambda_1^{k - (m_0 + 1)} \frac{z^k}{k!}$$

$$= \sum_{k = m_0 + 1}^{\infty} {}_{(l + m_0 + 1)}b_k \cdot \frac{z^k}{k!} \qquad \text{(Induktionsschluß für } j = l + (m_0 + 1)\text{).}$$

Mit (3.5-6) und (3.5-7) kann die gesuchte Lösung mit (3.5-5) berechnet werden:

$$u(z) = \sum_{j=1}^{N} z_j u_j(z) = \sum_{j=1}^{l} z_j u_j(z) + \sum_{j=l+1}^{N} z_j^* u_j^*(z), \text{ mit } z_j^* := \frac{z_j}{\lambda_1^{j-l}},$$

$$u_j^*(z) = \sum_{k=j-l}^{\infty} {}_{(j)}b_k^* \cdot \frac{z^k}{k!},$$

$${}_{(j)}b_k^* = \lambda_1^{j-l} \cdot {}_{(j)}b_k = (k + 1 - (j - l))_{j-l} \cdot \lambda_1^k, \text{ für } j = l+1, \ldots, N.$$

Für die gesuchte Funktion $u(z)$ gilt somit:

$$b_k = \sum_{j=1}^{N} {}_{(j)}b_k = \sum_{j=1}^{l} z_j \lambda_j^k + \sum_{j=l+1}^{N} z_j^* (k + 1 - (j - l))_{j-l} \cdot \lambda_1^k, \text{ für } k \geq N. \;\square$$

Alternativ zu der mit Satz 3.5-3 nachgewiesenen Lösungsformel (3.5-3) findet sich in der Literatur auch das äquivalente Ergebnis

$$\text{(3.5-9)} \qquad b_k = \sum_{j=1}^{l} \left( \lambda_j^k \sum_{\nu=0}^{q_j - 1} z_{j\nu} k^{\nu} \right),$$

wobei $\lambda_j$ die (paarweise verschiedenen) Nullstellen des charakteristischen Polynoms (3.5-2), $q_j$ ihre Vielfachheiten und $z_{j\nu}$ Konstanten sind [Lu80, S. 423]. Während (3.5-9) eine einfachere Darstellung hat, eignet sich (3.5-3) mehr im Beweisgang von Satz 3.5-3. Eine Beweisskizze für den Nachweis von (3.5-9) findet sich in [Li68, S. 64 - 65 u. 88 - 90].

<u>BEISPIEL 3.5-4:</u>

Weiterführung von Beispiel 3.3-3:

$f(z) = e^z \sin(z) = \sum_{k=0}^{\infty} a_k z^k$ besitzt die Differentialgleichung $f'' - 2f' + 2f = 0$. Hier treten nur konstante Koeffizienten auf, so daß mit Lemma 3.5-1 nach Substitution $b_k := k! a_k$ eine Rekursionsgleichung für $\sum_{k=0}^{\infty} \frac{b_k}{k!} z^k$ aufgestellt werden kann:

$$b_{k+2} - 2b_{k+1} + 2b_k = 0.$$

Das zugehörige (charakteristische) Polynom lautet

$$\lambda^2 - 2\lambda + 2\lambda = 0.$$

Die Nullstellen haben die Werte $\lambda_1 = 1 + i$ und $\lambda_2 = 1 - i$.
Das zu lösende Gleichungssystem von Teil (d) des Algorithmus 3.5-2 ist:

$$b_0 = z_1(1+i)^0 + z_2(1-i)^0 \ ,$$
$$b_1 = z_1(1+i)^1 + z_2(1-i)^1 \ .$$

Mit den Anfangswerten $b_0 = 0!a_0 = 0!\frac{f(0)}{0!} = 0$ und $b_1 = 1!a_1 = 1!\frac{f(0)}{1!} = 1$ ergibt sich:

$$0 = z_1 + z_2,$$
$$1 = z_1(1+i) + z_2(1-i) \ .$$

Die Lösung dieses Gleichungssystems ist $z_1 = \frac{1}{2i}$ und $z_2 = -\frac{1}{2i}$. Mit (3.5-3) folgt:

$$a_k = \frac{b_k}{k!} = \frac{1}{k!}\left(\frac{(1+i)^k - (1-i)^k}{2i}\right) \ .$$

Insgesamt ergibt sich $f(z) = \sum_{k=0}^{\infty} \frac{(1+i)^k - (1-i)^k}{2i \cdot k!} z^k = \sum_{k=1}^{\infty} -\frac{2^{k/2}\sin\left(-\frac{\pi \cdot k}{4}\right)}{k!} z^k$ .

# 4 Bewertung der Entwicklungsmethoden

## 4.1 Erweiterung der Anwendungsmöglichkeiten

In Kapitel 3 wurden prinzipielle Methoden für die Entwicklung von Funktionen in ihre Potenzreihen angegeben. Damit läßt sich eine Vielzahl elementarer und zusammengesetzter Funktionen entwickeln (vgl. Anhang A). Durch Substitution und Integration läßt sich die Vielfalt der entwickelbaren Funktionen noch erweitern.

<u>BEISPIEL 4.1-1:</u>

Sollte eine wesentliche Singularität bei $f(z)$ auftreten, so ist es möglich, daß der Nebenteil der Laurent-Entwicklung von $f(z)$ endlich viele Glieder hat und $f(z)$ mit Hilfe von $f(1/z)$ entwickelt werden kann. Dieser Fall soll hier auch als "wesentliche Singularität mit endlichem Nebenteil" bezeichnet werden. Im Beispiel $f(z) = e^{1/z}$ lautet die mit dem Algorithmus 3.3-1 gefundene Differentialgleichung $f(z) + z^2 f'(z) = 0$, und mit Lemma 3.4-1 wird die Rekursionsgleichung $a_{k+1} = -k a_k$, $k \in \mathbb{Z}$, abgeleitet. Also gilt wegen $a_1 = 0$ für $k \in \mathbb{N}$: $a_k = 0$. $f(1/z) = \sum_{k=0}^{\infty} a_k^* z^{-k}$ ist wieder Funktion vom hypergeometrischen Typ mit gleicher Symmetriezahl, und für die entsprechende Rekursionsgleichung $a_{k+m}^* = R^*(k) a_k^*$ von $f(1/z)$ ist $R^*(k) = \frac{1}{R(-k-1)} = \frac{1}{k+1}$. Mit dem hypergeometrischen Ansatz ergibt sich $f\left(\frac{1}{z}\right) = \sum_{k=0}^{\infty} \frac{z^k}{k!}$ und durch Rücksubstitution $z \to 1/z$ folgt die Laurent-Entwicklung:

$$f(z) = \sum_{k=0}^{\infty} \frac{1}{k!} z^{-k}.$$

<u>BEISPIEL 4.1-2:</u>

Eine formale Laurentreihe der Gestalt (2.1-2) ist für Funktionen wie z. B.

$$f(z) = e^{\sqrt{z}} = 1 + \sqrt{z} + \frac{1}{2}z + \frac{1}{6}z^{3/2} + \frac{1}{24}z^2 + \dots$$

natürlich auch nicht zu erwarten. Eine Reihe der Form $\sum_{k=k_0}^{\infty} a_k z^{k/r}$ mit $r \in \mathbb{N}$ wird auch Puiseux-Reihe genannt [DST88, S. 91]. Die mit dem Algorithmus 3.3-1 gefundene einfache Differentialgleichung $-f + 2f' + 4zf'' = 0$ führt mit dem Lemma 3.4-1 auf die Rekursionsgleichung $(1+2k)(2+2k)a_{k+1} = a_k$. Mit dem hypergeometrischen Ansatz kann aus dieser Rekursionsgleichung die formale Laurentreihe

$$F_1 = \sum_{k=0}^{\infty} \frac{1}{(2k)!} z^k = 1 + \frac{1}{2}z + \frac{1}{24}z^2 + \dots$$

hergeleitet werden. $F_1$ entspricht allerdings nicht $f(z)$, da der Anteil

$$F_2 = \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{k+1/2} = \sqrt{z} + \frac{1}{6} z^{3/2} + \ldots$$

mit den Summanden der nicht-ganzzahligen rationalen Potenzen von $z$ fehlt. Damit der hypergeometrische Ansatz die vollständige Koeffizientenfolge von $f(z) = F_1 + F_2 = \sum_{k=k_0}^{\infty} a_{k/2} z^{k/2}$ erfaßt, wird übergegangen zu:

$$f(z^2) = \sum_{k=k_0}^{\infty} a_{k/2} z^k .$$

Nach Lemma 2.4-1 (Teil d) folgt mit $k \to k/2$ die Rekursionsgleichung $(1+k)(2+k)a_{k/2+1} = a_{k/2}$ und nach Umbenennung $\overset{*}{a}_k = a_{k/2}$: $(1+k)(2+k)\overset{*}{a}_{k+2} = \overset{*}{a}_k$ zu $f(z^2) = \sum_{k=k_0}^{\infty} \overset{*}{a}_k z^k$.

Daraus entsteht mit dem hypergeometrischen Ansatz

$$f(z^2) = \sum_{k=0}^{\infty} \frac{1}{(2k)!} z^{2k} + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{2k+1}$$

und nach Rücksubstitution $z \to z^{1/2}$:

$$f(z) = \sum_{k=0}^{\infty} \frac{1}{(2k)!} z^k + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{k+1/2} .$$

<u>BEISPIEL 4.1-3:</u>

Die Entwicklung in eine formale Laurentreihe kann auch an der Unmöglichkeit scheitern, eine Initialisierung für die Rekursionsformel der Koeffizienten zu finden. Im Fall

$$f(z) = \mathrm{ei}(z) = -\int_{-z}^{\infty} \frac{e^{-t}}{t} dt$$

wird die einfache Differentialgleichung $(1-z)f' + zf'' = 0$ und die entsprechende Rekursionsformel $a_{k+1} = \frac{k}{(1+k)^2} a_k$ aufgestellt. Eine formale Laurentreihe der Form $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ ist wegen einer "logarithmischen" Singularität von $f(z)$ bei $z_0 = 0$ nicht zu erwarten, die Initialisierung der aufgestellten Rekursionsgleichung ist nicht möglich. Eine Funktion hat dabei genau dann eine logarithmische Singulatität, falls anhand einer ihrer Ableitungen erkannt wird, daß sie einen logarithmischen Term enthalten muß. Bei der gegebenen Funktion ist dies aufgrund der Reihenentwicklung von $f'(z) = \frac{e^z}{z}$ der Fall. Für $f'(z)$ besteht das Initialisierungsproblem der Rekursionsgleichung jedoch nicht und daher wird zunächst versucht, $f'(z)$ in eine formale Laurentreihe zu entwickeln. Für die entsprechende Rekursionsgleichung ergibt sich nach

Lemma 2.4-1 (Teil g) für $f'(z) = \sum_{k=k_0-1}^{\infty} a_k^* z^k$, wegen $a_{k+1} = \dfrac{a_k^*}{k+1}$, direkt:

$$a_{k+1}^* = (k+2)a_{k+2} = (k+2)\frac{(k+1)}{(k+2)^2}\ a_{k+1} = \frac{a_k^*}{k+2}\ .$$

Mit dem hypergeometrischen Ansatz folgt $f'(z) = \sum_{k=0}^{\infty} \dfrac{z^{k-1}}{k!}$ und nach Integration:

$$f(z) = \ln(z) + \sum_{k=1}^{\infty} \frac{z^k}{k \cdot k!}\ .$$

Im Fall einer mehrfachen Differenzierung treten nach entsprechender Integration auch mehrere Terme der Form $\dfrac{1}{n!}z^n \ln(z)$ auf.

In den o. g. drei Beispielen treten jeweils Rekursionsgleichungen der Form (2.4-1) auf, obwohl die Funktionen sich nicht durch eine formale Laurentreihe darstellen lassen. Dennoch können alle diese Funktionen mit Hilfe von Substitution oder Integration über den hypergeometrischen Ansatz in ihre Reihendarstellung entwickelt werden. Damit macht es Sinn, den Begriff “Funktion vom hypergeometrischen Typ” so zu erweitern, daß diese Funktionenklassen in die Definition mit einbezogen sind [K92, S. 18]: Eine Funktion $f$ heißt “Funktion vom hypergeometrischen Typ”, falls durch Anwendung des Algorithmus 3.3-1 und der Substitution aus Lemma 3.4-1 für $f$, abgesehen von logarithmischen Summanden, eine Darstellung möglich ist, durch Reihen der Form:

$$\sum_{k=k_0}^{\infty} a_k z^{v \cdot k/n},\ n \in \mathbb{N},\ v \in \{1, -1\}$$

und Koeffizienten $a_k \in \mathbb{C}$, die (2.4-1) erfüllen.

Die Operationen $f(z) \to f(1/z)$, $f(z) \to f(z^{1/n})$, mit $n \in \mathbb{N}$, und $f(z) \to \int f(z)dz$ sind wegen Lemma 4.1-1 mit dieser neuen Definition verträglich.

<u>Lemma 4.1-1:</u>

Sei $f(z)$ eine Funktion vom hypergeometrischen Typ, dann gilt dies auch für:

a) $f(1/z)$,

b) $f(z^{1/n})$ mit $n \in \mathbb{N}$,

c) $\int f(z)dz$.

<u>BEWEIS von Lemma 4.1-1:</u>

Für den Beweis der Beibehaltung des hypergeometrischen Typs beim Übergang von $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$ zu $f(1/z)$, $f(z^{1/n})$ und $\int f(z)\mathrm{d}z$ ist es wegen der Erweiterung der Definition von Funktionen des hypergeometrischen Typs nötig, die Existenz einer einfachen Differentialgleichung sowie einer Rekursionsgleichung der Form (2.4-1) zu zeigen. Die Symmetriezahl von $f$ sei dabei $m$.

zu a) Es sei $g(z) = f(1/z)$, also $g'(z) = f'(1/z) \cdot (-1/z^2)$ bzw. $f'(1/z) = (-z^2)g'(z)$. Damit wird offensichtlich $f^{(n)}(1/z)$ zurückgeführt auf $g^{(l)}(z), l \le n$, mit polynomialen Koeffizienten nach $z$. Eine einfache Differentialgleichung für $g(z)$ ergibt sich somit aus der einfachen Differentialgleichung von $f(z)$, wobei in $\sum_{n=0}^{Q} p_n(1/z) f^{(n)}(1/z) = 0$ nach Ersetzung von $f^{(n)}(1/z)$ nur noch mit $z^Q$ multipliziert werden muß.

Mit dem Übergang zu $g(z) = \sum_{k=k_0}^{\infty} a_k z^{-k} = \sum_{k=k_0}^{\infty} a^*_{-k} z^{-k}$ kann, wegen $a^*_k = a_{-k}, k \in \mathbb{Z}$, eine Rekursionsgleichung $a^*_{k+m} = R^*(k) a^*_k$ direkt abgeleitet werden. Dazu wird in $a_{k+m} = R(k)a_k$ die formale Substitution $k \to -k-m$ durchgeführt, damit ist:

$$a_{-k} = R(-k-m)a_{-k-m} \text{ und}$$

$$a^*_{k+m} = a_{-k-m} = \frac{1}{R(-k-m)} a_{-k} = \frac{1}{R(-k-m)} a^*_k ,$$

also $R^*(k) = \dfrac{1}{R(-k-m)}$ rational.

zu b) Mit $g(z) = f(z^{1/n}) = \sum_{k=k_0}^{\infty} a_k z^{k/n} = \sum_{k=(k_0+j)/n} a^*_k z^k$, $j \in \mathbb{N}_0$, besteht durch $k = \dfrac{k_0+j}{n}$ zwischen den Koeffizienten von $g(z)$ und $f(z)$ die Beziehung $a^*_k = a_{nk}$, z. B. $a^*_{k_0/n} = a_{k_0}$, $a^*_{(k_0+1)/n} = a_{k_0+1}$, (...).

Der Nachweis der Existenz einer einfachen Differentialgleichung erfolgt analog zum Beweis von Satz 2.4-2:

Für $\theta_n := nz\frac{d}{dz}$ ergibt sich $\theta_n \; g = \sum_{k=k_0}^{\infty} k \cdot a_k \cdot z^{k/n}$ und zu $p \in \mathbb{N}$:

$$\theta_n^p \; g = \sum_{k=k_0}^{\infty} k^p \cdot a_k \cdot z^{k/n}.$$

Mit einem beliebigen Polynom $T$ gilt:

(4.1-1) $$T(\theta_n)g = \sum_{k=k_0}^{\infty} T(k) \cdot a_k \cdot z^{k/n} .$$

Zu $f(z)$ gibt es wegen Lemma 2.4-1 (Teil 2) eine Rekursionsbeziehung:

(4.1-2) $$Q(k)a_{k+nm} = P(k)a_k$$

mit Polynomen $P$ und $Q$. O. B. d. A. seien $P$ und $Q$ so gewählt, daß

(4.1-3) $\forall t \in \{k_0, \ldots, (k_0 + nm - 1)\} : Q(t - nm) = 0$ .

Dies ist immer möglich, da zu jedem $t \in \{k_0, \ldots, (k_0 + nm - 1)\}$ mit $Q(t - nm) \neq 0$ übergegangen werden kann zu den Polynomen $Q^*(k) := Q(k) \quad (k - (t - nm))$ sowie $P^*(k) := P(k) \quad (k - (t - nm))$ und (4.1-2) entsprechend erhalten bleibt.

Die gesuchte Differentialgleichung existiert schließlich durch:

$$
\begin{aligned}
Q(\theta_n - nm)g &= \sum_{k = k_0}^{\infty} Q(k - nm)a_k z^{k/n} && \text{mit (4.1-1)}\\
&= \sum_{k = k_0 + nm}^{\infty} Q(k - nm)a_k z^{k/n} && \text{mit (4.1-3)}\\
&= \sum_{k = k_0}^{\infty} Q(k)a_{k + nm} z^{(k + nm)/n} && \text{nach Indexverschiebung}\\
&= z^m \sum_{k = k_0}^{\infty} Q(k)a_{k + nm} z^{k/n} &&\\
&= z^m \sum_{k = k_0}^{\infty} P(k)a_k z^{k/n} && \text{mit (4.1-2)}\\
&= z^m P(\theta_n)g && \text{mit (4.1-1).}
\end{aligned}
$$

Wegen $a^*_{k + m/n} = a_{n(k + m/n)} = a_{nk + m} = R(nk)a_{nk} = R(nk)a^*_k$ , folgt die Rekursionsgleichung $a^*_{k + m/n} = R^*(k)a^*_k$ mit $R^*(k) = R(nk)$ rational. Analog zu Lemma 2.4-1 (Teil 2) existiert dann $R^{**}(k)$ rational in $a^*_{k + m} = R^{**}(k)a^*_k$ .

zu c) Mit $g(z) = \int f(z)dz$ gilt $f^{(k)}(z) = g^{(k + 1)}(z)$ , und die einfache Differentialgleichung für $g(z)$ ergibt sich direkt aus der von $f(z)$:

$$\left(\sum_{n = 0}^{Q} p_n(z)f^{(n)}(z) = 0\right) \Rightarrow \left(\sum_{n = 0}^{Q} p_n(z)g^{(n + 1)}(z) = 0\right) \quad , \; p_n(z) \text{ Polynome.}$$

Für die Reihenkoeffizienten $a^*_k$ von $g(z)$ und $a_k$ von $f(z)$ gilt die Beziehung: $a^*_{k + 1} = \dfrac{a_k}{k + 1}, \; k \neq -1$ .

Wegen $\int \frac{a_{-1}}{z} dz = a_{-1}\ln z$ fehlt $a^*_0$ in:

$$g(z) = \int\left(\sum_{k = k_0}^{\infty} a_k z^k\right)dz = \sum_{\substack{k = k_0 \\ k \neq 1}}^{\infty} a^*_{k + 1} z^{k + 1} + a_{-1}\ln z = \sum_{\substack{k = k_0 + 1 \\ k \neq 0}}^{\infty} a^*_k z^k + a_{-1}\ln z .$$

Von dieser Ausnahme abgesehen gilt analog zu Lemma 2.4-1 (Teil h) $a^*_{k+m} = R^*(k)a^*_k$ mit $R^*(k) = \frac{k}{k+m}R(k-1)$ rational, d. h. eine Rekursionsgleichung der Form (2.4-1). ❑

## 4.2 Besondere Ergebnisse

Alle bisher vorgestellten Methoden für die Entwicklung von Funktionen in ihre Reihendarstellung zeichnen sich durch einfache und direkte Lösungswege aus. Bei der Suche nach einer Reihenentwicklung einer Funktion können sie damit in vielen Fällen eine Alternative zu den großen Nachschlagebüchern für Reihen [GR80; Ha75] darstellen. Dies gilt um so mehr, da der Umgang mit dieser Literatur teilweise viel Übung erfordert und oft Neben- und Umrechnungen zu den angegebenen Formeln nötig sind: Allein für $F = \sum_{k=0}^{\infty} \frac{(a)_k(c+a)_k}{k!(c)_k}x^k$ gibt Hansen acht verschiedene Darstellungen zugehöriger Funktionen an [H75, S. 182, Formel 10.9.5], u. a.:

$$\begin{aligned} f(x) &= (1-x)^{-a}\sum_{k=0}^{n}\frac{(a)_k(-n)_k}{k!(c)_k}\left(\frac{x}{x-1}\right)^k \\ &= \frac{(a)_n}{(c)_n}(1-x)^{-a-n}\sum_{k=0}^{n}\frac{(c-a)_k(-n)_k}{k!(1-a-n)_k}(1-x)^k \quad . \end{aligned}$$

Darüberhinaus werden bei der methodischen Entwicklung von Funktionen automatisch u. U. interessante Nebenerkenntnisse gewonnen, z. B. mit der Klassifizierung der Rekursionsgleichung (hypergeometrischer / rekurrenter Ansatz). Für das Beispiel $f(z) = \left(\frac{\arcsin z}{z}\right)^2$ ergibt sich:

$$f(z) = \sum_{k=0}^{\infty}\frac{4^k(k!)^2}{(1+k)(1+2k)}z^{2k} \quad .$$

Gradshteyn-Ryzhik gibt dieses Ergebnis ziemlich genauso an [GR80, S. 52, F. 1.645.2]:

$$(\arcsin x)^2 = \sum_{k=0}^{\infty}\frac{2^{2k}(k!)^2x^{2k+2}}{(2k+1)(k+1)} \quad ,$$

demgegenüber notiert Hansen diesen Sachverhalt folgendermaßen [Ha75, S. 77, F. 5.18.4]:

$$\frac{2}{x}\left(\sin^{-1}\left(\frac{1}{2}x^{1/2}\right)\right)^2 = \frac{1}{2}\sum_{k=0}^{\infty}\frac{k!}{(3/2)_k(k+1)}\left(\frac{x}{4}\right)^k = \sum_{k=0}^{\infty}\frac{(k!)^2}{(2k+2)!}x^k \quad .$$

Der hypergeometrische Ansatz führt zum Ausdruck

$$f(z) = {}_3F_2\left(\left.\begin{matrix}1 & 1 & 1 \\ & 3/2 & 2\end{matrix}\right| z^2\right).$$

Wird jedoch $g(z) = \frac{\arcsin z}{z}$ mit dem hypergeometrischen Ansatz entwickelt, folgt

$$g(z) = {}_2F_1\left(\begin{matrix}1/2 & & 1/2\\ & 3/2 & \end{matrix}\middle| z^2\right).$$

Offensichtlich gilt:

$$ {}_3F_2\left(\begin{matrix}1 & & 1 & & 1\\ & 3/2 & & 2 & \end{matrix}\middle| z^2\right) = \left({}_2F_1\left(\begin{matrix}1/2 & & 1/2\\ & 3/2 & \end{matrix}\middle| z^2\right)\right)^2,$$

und somit liegt hier ein Spezialfall der sog. Clausen Formel [GKP88, S. 241] vor:

$$ {}_3F_2\left(\begin{matrix}2a & 2b & a+b\\ a+b+1/2 & & 2a+2b\end{matrix}\middle| x\right) = \left({}_2F_1\left(\begin{matrix}a & b\\ a+b+1/2 & \end{matrix}\middle| x\right)\right)^2 \text{, setze } a = b = \frac{1}{2}.$$

In einigen Fällen kommt es durchaus vor, daß durch die methodische Entwicklung der Funktion Ergebnisse gefunden werden, die in den Tabellenwerken weniger bekannt sind. Der hypergeometrische Ansatz entwickelt $f(z) = e^{\operatorname{arcsin} z}$ in:

$$f(z) = \sum_{k=0}^{\infty} \frac{4^k \prod_{j=1}^{k}\left(\frac{1}{4} + (j-1)^2\right)}{(2k)!} z^{2k} + \sum_{k=0}^{\infty} \frac{4^k \prod_{j=1}^{k}\left(\frac{1}{2} - j + j^2\right)}{(2k+1)!} z^{2k+1} .$$

Eine allgemeine Formel für die Reihenkoeffizienten fehlt bei Gradshteyn-Ryzhik völlig, dort findet sich lediglich [GR80, S. 22, F. 1.216]:

$$e^{\operatorname{arcsin} x} = 1 + x + \frac{x^2}{2!} + \frac{2x^3}{3!} + \frac{5x^4}{4!} + \dots .$$

Hansen schreibt allerdings ohne eine Aufteilung in geraden und ungeraden Anteil [Ha75, S. 210, F. 10.49.33]:

$$\sum_{k=0}^{\infty} \frac{(i/2)_{k/2}}{k!(i/2+1)_{-k/2}} (-ix)^k = \exp\left(\sin^{-1}\left(\frac{x}{2}\right)\right) .$$

Eine andere wesentliche Bedeutung gewinnen die Entwicklungsmethoden durch die Entwicklung zusammengesetzter Funktionen. Für $f(z) = \operatorname{arcsin}(z) + \operatorname{arcsin}(-z)$ ergibt sich $f(z) = 0$. Die Entwicklungsmethode findet dieses Ergebnis dabei ohne Wissen der Symmetriebeziehung $\operatorname{arcsin}(-z) = -\operatorname{arcsin}(z)$ . Mit

$$\operatorname{arcsin} z = \sum_{k=0}^{\infty} \frac{(1/4)^k z^{1+2k} (2k)!^2}{k!^2 (1+2k)!} \quad \text{und} \quad \operatorname{arcsin}(-z) = \sum_{k=0}^{\infty} -\frac{(1/4)^k z^{1+2k} (2k)!^2}{k!^2 (1+2k)!}$$

wird das leicht einsehbar.

In weiteren Fällen kommt es zu bemerkenswerten Nachweisen, die nicht mit einer Symmetriebeziehung zusammenhängen. Für

$$f(z) = e^{\operatorname{arcsinh} z} - \sqrt{1+z^2}$$

ergibt sich die Reihendarstellung $f(z) = z$, womit $e^{\operatorname{arcsinh} z} = z + \sqrt{1+z^2}$ bzw. $\operatorname{arcsinh} z = \ln(z + \sqrt{1+z^2})$ gezeigt ist.

Die methodische Entwicklung eignet sich ebenfalls zur Gewinnung von nützlichen Umformungen. Für

$$f(x) = \sin(x+y-z) + \sin(x-y+z) + \sin(-x+y+z)$$

ergibt sich:

$$\sum_{k=0}^{\infty} \frac{(-1)^k (2\cos(y-z) - \cos(y+z))}{(1+2k)!} x^{1+2k} + \sum_{k=0}^{\infty} \frac{(-1)^k \sin(y+z)}{(2k)!} x^{2k} \quad ,$$

und damit die trigonometrische Beziehung:

$$\begin{aligned} &\sin(x+y-z) + \sin(x-y+z) + \sin(-x+y+z) \\ &= \sin(x)(2\cos(y-z) - \cos(y+z)) + \cos(x)\sin(y+z) \quad . \end{aligned}$$

Selbst wenn eine Reihendarstellung durch geeignete Umformungen schnell möglich ist, kann die methodische Entwicklung dagegen zu sehr einfachen Reihendarstellungen führen. Dies zeigt sich z. B. bei mehrfach symmetrischen Funktionen:

$$e^z - 2e^{-z/2}\cos\left(\frac{\sqrt{3}z}{2} + \frac{\pi}{3}\right) = \sum_{k=0}^{\infty} \frac{3}{(1+3k)!} z^{1+3k} \quad ,$$

[W90, S. 47-48; Ha75, S. 50, F. 5.10.28],

und bei

$$\frac{1}{2}\ln\left(\frac{1+z}{1-z}\right) - \arctan(z) = \sum_{k=0}^{\infty} \frac{2}{3+4k} z^{3+4k} \quad ,$$ [Ha75, S. 23, F. 5.4.22].

Ihren größten Vorteil entfalten diese Entwicklungsmethoden durch ihre vollständige Algorithmisierung und Zusammenfassung für eine durch Computer unterstützte Nutzung.

# 5 Algorithmisierung der Methoden

## 5.1 Struktur des Algorithmus

Für eine automatisierte Entwicklung von Funktionen in eine formale Laurentreihe wird es wichtig, Zeitpunkt und Voraussetzungen für die Anwendungsmöglichkeiten verschiedener Entwicklungsmethoden festzulegen. Die Ansätze aus Kapitel 3 und 4.1 können als programmierbare Einzelbestandteile eines umfassenden Algorithmus verstanden werden. Da die optimale Anordnung für den Computer von Hardware- und Softwarefaktoren beeinflußt ist, wird hier lediglich ein Strukturierungsentwurf vorgestellt. Aus mathematischer Sicht geht es dabei vielmehr um die Einsatzmöglichkeiten einer Methode bei unterschiedlichen Objekten (z. B. $f(z)$ oder $f'(z)$) und eine mathematisch korrekte und sinnvolle Kriterienwahl für Algorithmusverzweigungen.

Zur einfacheren Darstellung und Diskussion sei der Algorithmus in drei Teile aufgegliedert: Der erste Teil gibt Einsatzmöglichkeiten für den rationalen Ansatz an (s. Abb. 5.1-1). Im zweiten Teil folgt die Behandlung der Funktionen, für die keine rekursive Koeffizientengleichung vom hypergeometrischen Typ gefunden wurde, also insbesondere die Stellung des rekurrenten Ansatzes (s. Abb. 5.1-2). Spezielle Ansätze zur erweiterten Nutzung des hypergeometrischen Ansatzes werden im dritten Teil angeordnet (s. Abb. 5.1-3). Es gilt dabei folgende Symbolik: Ein Rhombus stellt

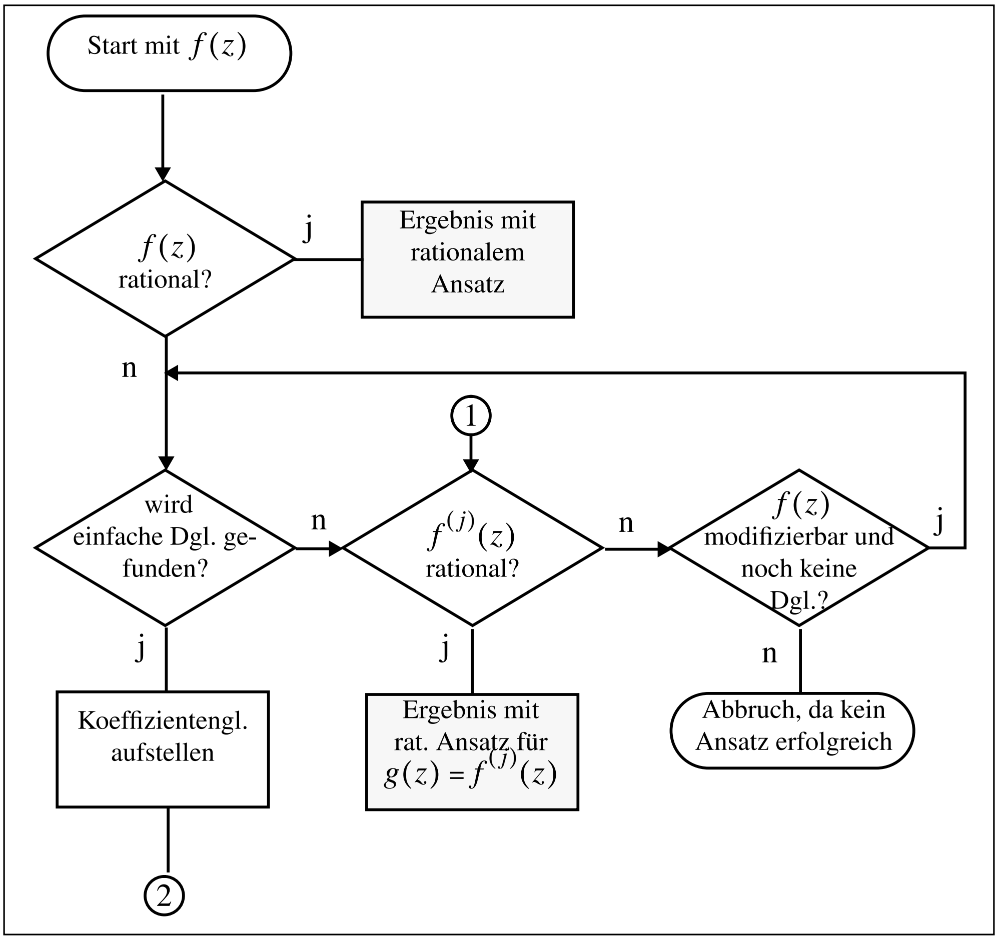


Abb. 5.1-1: Strukturierung der Methoden (Teil 1)

eine bedingte Verzweigung dar. Die beiden zugehörigen Ausgänge sind mit “j” (ja) bzw. “n” (nein) markiert und als Antwortalternativen auf die im Rhombus enthaltene Frage zu verstehen. Mit Nummern markierte Kreise symbolisieren Ein- und Aussprungstellen zwischen den drei Algorithmusteilen. Rechtecke sind unverzweigte Algorithmusteile und nur dann grau ausgefüllt, falls sie bereits zu einer gesuchten Lösung führen.

Zu Teil 1:

Der rationale Ansatz ist an den Anfang des Algorithmus gestellt, da er die Konstruktion einer einfachen Differentialgleichung nicht benötigt. Die Anwendung des rationalen Ansatzes bei Ableitungen von $f(z)$ profitiert dagegen bereits von den Zwischenrechnungen ($f(z)$, $f'(z)$, …) der Differentialgleichungssuche und erfolgt deshalb ggf. erst im Anschluß daran. Es gibt aber auch Beispiele wie $f(z) = \frac{z}{1-z^5}$, bei denen der rationale Ansatz aufgrund seiner komplexen Partialbruchzerlegung aufwendiger werden kann als der hypergeometrische Ansatz.

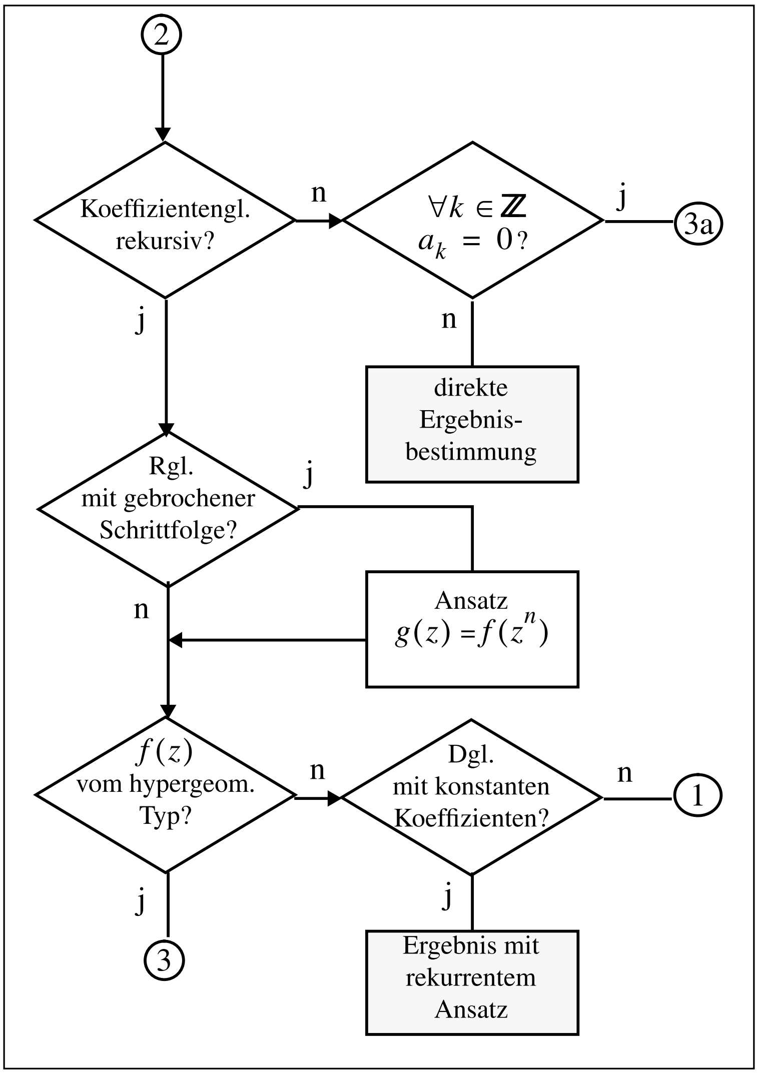


Abb. 5.1-2: Strukturierung der Methoden (Teil 2)

Zu Teil 2:

In bestimmten Fällen führt die konstruierte Differentialgleichung auf eine nicht-rekursive Koeffizientengleichung ($a_k = c, c \in \mathbb{C}$). Wird dann $\forall k \in \mathbb{Z}$: $a_k = 0$ festgestellt, kann $f(z)$ in keine formale Laurentreihe entwickelt werden, und es wird wie bei einer logarithmischen Singularität mit dem Ansatz $g(z) = f'(z)$ fortgefahren (z. B. $f(z) = \ln(z)$ ). Ansonsten ist unter Berücksichtigung der gefundenen Bedingungsgleichung für die Koeffizienten eine direkte Lösung durch Angabe einer ggf. aus nur endlich vielen Gliedern bestehenden “formalen Laurentreihe” möglich. Sofern der rationale Ansatz nicht an den Algorithmusanfang gestellt wurde, finden sich hier insbesondere Polynome und Monome (z. B. $1/z^2$ ) wieder.

In Kapitel 4.1 ist am Beispiel $f(z) = e^{\sqrt{z}}$ deutlich geworden, daß Rekursionsgleichungen modifiziert werden müssen, um auch den Anteil von $f(z)$ zu erfassen, der nicht in einer formalen Laurentreihe mit nur ganzzahligen Exponenten von Potenzen von $z$ enthalten ist, also z. B.

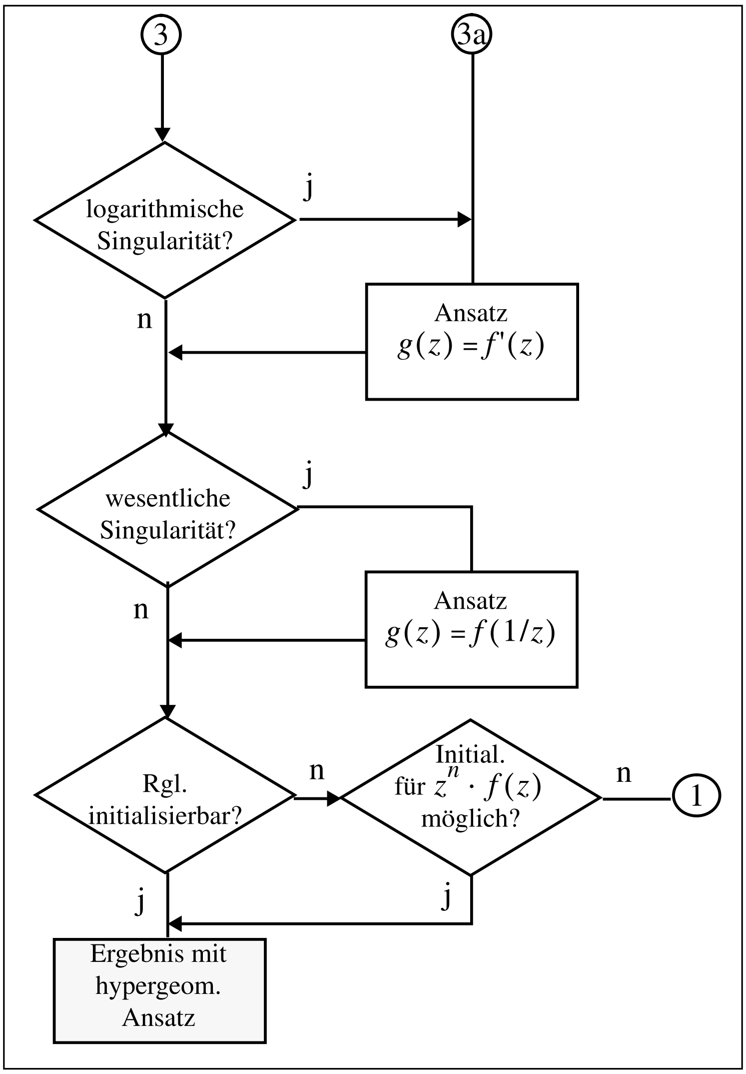


Abb. 5.1-3: Strukturierung der Methoden (Teil 3)

$$\sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{k+1/2} \ .$$

Die Entscheidung über den Ansatz $g(z) = f(z^n)$ kann dabei anhand der vorliegenden Rekursionsgleichung von $f(z)$ getroffen werden. Aus dem Beweis von Lemma 4.1-1 (Teil b) geht hervor, daß beim Übergang $f(z) \to f(z^{1/n})$ für die rationalen Funktionen $R^*(k) = \frac{P^*(k)}{Q^*(k)}$ und $R(k)$ aus den Rekursionsgleichungen von $f(z^{1/n})$ und $f(z)$, d. h. $a^*_{k+m/n} = R^*(k)a^*_k$ und $a_{k+m} = R(k)a_k$, die Beziehung gilt:

$$R^*(k) = R(nk) \qquad \text{bzw.} \qquad R^*\left(\frac{k}{n}\right) = R(k) \ .$$

Sofern $Q^*$ nicht-ganzzahlige rationale Nullstellen besitzt, liegt eine Rekursionsgleichung mit gebrochener Schrittfolge vor (Puiseux-Reihe), und dies führt zum Ansatz $g(z) = f(z^n)$.

Nachdem sichergestellt ist, daß die Rekursionsgleichung alle Anteile der zu erstellenden formalen Laurentreihe umfaßt, wird das weitere Verfahren vom Typ der Rekursionsgleichung abhängig. Hat die Rekursionsgleichung die Form $a_{k+m} = \sum_{j=0}^{n} h_j(k) a_{k+j}$ mit $1 \le n < m$, wobei für mindestens zwei der dabei auftretenden rationalen Funktionen $h_j(k) \ne 0$ gilt, kann der hypergeometrische Ansatz nicht genutzt werden. Gilt für die darin auftretenden $h_j(k) \equiv c_j \in \mathbb{C}$, besteht ein Lösungsweg durch den rekurrenten Ansatz.

Der rekurrente Ansatz wird insbesondere auch bei Funktionen benutzt, deren Differentialgleichung ausschließlich konstante Koeffizienten aufweisen (vgl. Lemma 3.5-1). Dieser Fall ist natürlich bereits mit dem Vorliegen der Differentialgleichung identifizierbar. Die spätere Einordnung dieser Abzweigung soll aber eine vergleichende Betrachtung der Rekursionsgleichungen mit bzw. ohne Hilfe der Koeffizientensubstitution $b_k = k!a_k$ ermöglichen.

Kann weder der hypergeometrische noch der rekurrente Ansatz benutzt werden, bleibt im Rahmen der hier untersuchten Methoden die Suche nach einer rationalen Ableitung für den rationalen Ansatz (Teil 1 des Gesamtalgorithmus).

<u>Zu Teil 3:</u>

Bevor mit dem hypergeometrischen Ansatz der Übergang von einer Rekursionsgleichung zur Reihendarstellung von $f(z)$ Sinn macht, ist es nötig, eine Initialisierung für die aufgestellte Rekursionsgleichung zu finden. Bei der Entwicklung von $f(z)$ in eine formale Laurentreihe um eine logarithmische oder wesentliche Singularität gelingt dies nicht, ohne mit einer Transformation von $f(z)$ zu arbeiten. Der entsprechende Hilfsansatz ist abhängig vom Typ der Singularität, die dabei benutzten Transformationen müssen in umgedrehter Reihenfolge wieder aufgehoben werden. Ein notwendiges Kriterium für das Vorliegen einer logarithmischen Singularität, also z. B. für den Fall

$$f(z) = \ln(z) + \sum_{k=k_0}^{\infty} a_k z^k ,$$

bietet sich bei Betrachtung einer einfachen Differentialgleichung von $f(z)$. Wegen

$$f^{(n)}(z) = (-1)^{n+1}(n-1)!z^{-n} + \sum_{k=k_0}^{\infty} k(k-1)\dots(k-n+1)a_k z^{k-n} \quad ,$$

muß für jede einfache Differentialgleichung (2.3-2) von $f(z)$ vom Grad $Q \in \mathbb{N}$ der polynomiale Koeffizient von $f^{(Q)}(z)$ die Nullstelle $z = 0$ haben, d. h. die Differentialgleichung die Form

$$z \cdot p_Q(z) \cdot f^{(Q)}(z) + \sum_{j=0}^{Q-1} p_j(z) \cdot f^{(j)}(z) = 0$$

besitzen, wobei $p_0(z), \dots, p_Q(z)$ Polynome sind. Sofern keine logarithmische Singulatität vorliegt, gibt es für die formale Laurentreihe mit endlichem Nebenteil noch den Lösungsansatz $g(z) = f(1/z)$.

Neben den in Kapitel 4.1 vorgestellten Ansätzen $g(z) = f'(z)$ und $g(z) = f(1/z)$ ist es wegen einer Polstelle der Ordnung $n \in \mathbb{N}$ oft erst mit dem Ansatz

$$g(z) = z^n \cdot f(z) = \sum_{k=0}^{\infty} a_k z^k$$

möglich, eine Initialisierung der Koeffizientengleichung herzustellen. Sofern die gefundene Rekursionsgleichung der Koeffizienten von $f(z)$ jedoch nur für $k \ge 0$ gilt, muß dies aufgrund der vorgenommenen Verschiebung der Koeffizienten später durch Addition eines separat zu berechnenden aus endlich vielen Summanden bestehenden Anteils ausgeglichen werden:

$$\sum_{k=0}^{n-1} a_k \cdot z^{k-n} .$$

Verhilft keiner der genannten Transformationen zur Anwendung des hypergeometrischen Ansatzes, so folgt eine Untersuchung der Ableitungen in Hinblick auf den rationalen Ansatz. Der rekurrente Ansatz benötigt wie der hypergeometrische Ansatz die Initialisierungskoeffizienten $b_0, \dots, b_{N-1}$ (vgl. Satz 3.5-3) und kann daher nicht herangezogen werden.

## 5.2 Funktionalität der Computer Algebra Systeme

Nachdem eine Algorithmusstruktur für eine automatisierte Nutzung der untersuchten Methoden festgelegt ist, stellt sich die Frage nach der Programmierbarkeit dieses Algorithmus und insbesondere nach der Leistungsfähigkeit bereits realisierter Entwicklungsmethoden in den verfügbaren Computer Algebra Systemen. Während die über 20jährige Geschichte, das Angebot und die breite Anwendung heutiger Computer Algebra Systeme kaum Zweifel an der Eignung für eine symbolischen Programmierung entstehen lassen, überraschen ihre äußerst geringen Fähigkeiten bei der Entwicklung von Funktionen in eine formale Laurentreihe.

Für eine Beurteilung der Funktionalität der Computer Algebra Systeme soll hier nur ein kurzer Überblick über ihre verbreitetsten Vertreter gegeben werden. Zu den frühen aus den siebziger Jahren stammenden Computer Algebra Systemen gehören REDUCE und MACSYMA. Beide bieten einem Nutzer bereits mit ihren Sprachelementen eine große Flexibilität und weite Einsatzmöglichkeit. Mit ihnen lassen sich wahrscheinlich zwar fast alle Probleme angehen, die Nutzung dieser Vielfalt bedeutet aber gleichzeitig einen größeren Aufwand bei ihrer Erlernung und Bedienung [Hä89]. Seit Anfang der achtziger Jahre gibt es MAPLE und MUMATH (seit 1988 in DERIVE übergegangen). Beide Systeme hatten das Ziel, möglichst wenig Speicherkapazität zu beanspruchen, ohne dabei die grundlegende Funktionalität und Effizienz zu verlieren. DERIVE macht dabei gegenüber MAPLE mehr Kompromisse zugunsten geringerer Speicheranforderungen und auf Kosten anspruchsvollerer Funktionen. Hinzu kam als Randbedingung, dem Benutzer ohne langes Literaturstudium eine einfache und schnell erlernbare Bedienungsoberfläche bereitzustellen [Hä89; M91]. Eine Neuentwicklung, die die Mächtigkeit von REDUCE und MACSYMA sowie die Einsatzmöglichkeiten und den Bedienungskomfort von MAPLE und DERIVE in sich zu vereinigen versucht, ist MATHEMATICA. Ende der achtziger Jahre auf den Markt gekommen, ist es durch die an ein mathematisches Axiomensystem erinnernde regelbasierte Programmiersprache (pattern matching / Mustererkennung), seine graphischen Fähigkeiten und hohe Kompatibilität mit derzeitigen Hardwareanforderungen und Softwareprodukten gekennzeichnet [AS89].

Neben diesen fünf genannten Computer Algebra Systemen existiert eine unüberschaubare Vielfalt weiterer Softwareentwicklungen zur symbolischen Programmierung. Allein in [FWH91] werden weitere 16 symbolische Programmsysteme vorgestellt. Die meisten unter ihnen sind dabei als Werkzeug für spezielle Anwendungen konzipiert, z. B. für kommutative Algebra oder Zahlentheorie. Da sich keines insbesondere bei Anwendungen in der reellen und komplexen Analysis auf dem heutigen Markt durchgesetzt hat, werden in dieser Arbeit nur die bekannten kommerziell vertriebenen Computer Algebra Systeme näher betrachtet.

Auf den ersten Blick finden sich bei allen der fünf bekanntesten Computer Algebra Systeme Befehle, die Funktionen in eine Reihe entwickeln können. In der Regel handelt es sich dabei jedoch nur um die Bestimmung einer aus endlich vielen Gliedern bestehenden Partialsumme

| | Berechnung von Partialsummen | Rechnen mit Reihenobjekten | Reihenobjekte aus Dgl./Rgl. | geschlossene Reihendarstellung |
|---|---|---|---|---|
| DERIVE | x | - | - | - |
| REDUCE | x | x | - | - |
| MATHEMATICA | x | x | - | - |
| MAPLE | x | x | x | - |
| MACSYMA | x | x | - | x |

Tabelle 5.2-1: Vergleich kommerzieller Computer Algebra Systeme

$\sum_{k=k_0}^{k_1} a_k z^k$ dieser Reihe mit fest anzugebenden Grad $k_1 \in \mathbb{N}_0$ ("Truncated Series") und nicht um eine geschlossene Darstellung der Form (2.1-2). Für die Berechnung solcher "abbrechender Reihen" werden dabei unterschiedliche Begriffe verwendet, z. B. "Taylor" (Derive, Maple, Macsyma) oder "Series" (Maple, Mathematica), bei Reduce gibt es dafür keine Standardfunktionen, sondern die beiden vergleichbaren Anwenderprogramme "ps" (truncated power series) und "Taylor".

Ein zweites Merkmal der Computer Algebra Systeme ist, ihre Fähigkeit, Funktionen in interne Reihenobjekte ("Series Objects") zu überführen, die zwar nicht dargestellt aber miteinander verknüpft werden können. Nach einer Rechnung mit solch einem internen Reihenobjekt ist es erneut möglich, zu beliebig aber festem Grad eine Partialsumme des resultierenden internen Reihenobjekts auszugeben, ohne die Ungenauigkeit des Rechnens mit Partialsummen zu erhalten. Dabei erfolgt keine Verknüpfung formaler Laurentreihen, sondern der ursprünglichen Funktionen, die zu den internen Reihenobjekten geführt haben. Für Reduce und Mathematica ist damit die Unterstützung von formalen Laurentreihen ausgeschöpft, während sich Derive bereits allein mit der Berechnung von Partialsummen begnügt (vgl. Tabelle 5.2-1).

Ein Bezug zum Algorithmus aus Kapitel 5.1 kann nur bei Maple und Macsyma gesehen werden. In Maple ermöglicht das "powseries" Programmpaket [Ch88, S. 317 - 326] die Festlegung und Verknüpfung formaler Potenzreihen, die durch Angabe einer Koeffizientenformel mit initialen Koeffizienten (z. B.: powcreate(t(n)=1/n!, t(0)=1) ), oder einer linearen Differentialgleichung mit Nebenbedingungen (z. B.: powsolve({diff(y(x),x)=y(x), y(0)=1}) ) definiert sind. Eine geschlossene Darstellung der Form (2.1-1) für diese formalen Potenzreihen wird auch hier nicht angeboten. Dies deutet darauf hin, daß es sich um Implementierungen im Sinne der Norman-Methode (s. Kapitel 1) handeln kann.

Das einzige Computer Algebra System, das angibt, eine Funktion in die geschlossene Form einer Potenzreihe zu überführen ("Series Generation"), ist Macsyma. Das Kommando "powerseries" liefert für Beispiele wie $\exp(x)$, $\sin(x)$ oder $\frac{1}{(1-x)^\alpha}$ tatsächlich die geschlossene Form der entsprechenden Potenzreihe, aber in anderen Fällen wird schnell klar, auf welcher Grundlage dies passiert, so ergibt sich z. B.

$$\exp(x) \cdot \sin(x) = \left( \sum_{k=0}^{\infty} \frac{x^k}{k!} \right) \cdot \left( \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!} \right) \quad ,$$

oder Macsyma scheitert völlig bei der Potenzreihenangabe (z. B. $\frac{1}{x^2+2x+2}$, $\arctan(x)$ ) [C91, S. 199 - 209]. Es kann deshalb davon ausgegangen werden, daß mit "powerseries" keine algorithmisierte Entwicklung der Funktion in eine formale Reihe erfolgt, sondern eher eine Art elektronisches Nachschlagewerk bekannter Reihenentwicklungen vorliegt, auf die die zu entwickelnden Terme zurückgeführt werden.

Ein anderer nur bei Macsyma vorgefundener Ansatz für eine Potenzreihenentwicklung wird mit dem "series" Programmpaket angeboten. Bei Eingabe einer linearen gewöhnlichen Differentialgleichung wird eine Rekursionsgleichung für die Koeffizienten bestimmt und ggf.

eine geschlossene Reihendarstellung. Das klappt zwar bei $f''(x) + f(x) = 0$ für $f(x) = \sin(x)$, scheitert jedoch bereits an $f'(x) - f(x) = 0$ für $f(x) = e^x$.

Dieser Exkurs in die Funktionalität heute verfügbarer Computer Algebra Systeme zeigt, wie wenig an eine algorithmisierte Entwicklung von Funktionen in eine formale Laurentreihe gedacht wurde. Bei den ansonsten gebotenen großartigen Möglichkeiten von Computer Algebra Systemen und dem enormen Konkurrenzdruck ist das ziemlich unverständlich. Daß die prinzipielle und problemlose Möglichkeit für die symbolische Programmierung der in dieser Arbeit vorgestellten Methoden zur Entwicklung von Funktionen in formale Laurentreihen besteht, soll anhand einer experimentellen Implementierung mit dem Computer Algebra System MATHEMATICA im nächsten Kapitel deutlich gemacht werden. Die Entscheidung für MATHEMATICA ist hier von geringer Bedeutung und nur durch seine derzeit stark gewachsene Verbreitung begründet.

## 5.3 Experimentelle Implementierung mit MATHEMATICA

Bei einer Implementierung mathematischer Methoden werden neben der in Kapitel 5.1 gestellten Frage zur Festlegung von Algorithmusstruktur und -verzweigungskriterien noch weitere mathematisch zu behandelnde Antworten erforderlich. Dies liegt einerseits an beschränkten Fähigkeiten der Programmiersprache, wodurch bisher nicht vermutete Lücken im Algorithmus erkannt werden, als auch der notwendigen Begrenzung von iterativen Prozeduren durch geeignete Parameterwahl (z. B. Suchtiefe bei der Konstruktion einer Differentialgleichung). Da einige dieser Fragestellungen z. B. erst durch die Zwänge der Programmierung oder einer intensiven automatisierten Anwendung des Algorithmus bei unterschiedlichsten Funktionen entdeckt werden, ist es sinnvoll, die untersuchten Methoden in einer experimentellen Implementierung nutzbar zu machen. Die dabei gefundenen Beobachtungen vervollständigen die Methodenanalyse und sind darüberhinaus eine zusätzliche Verifikationsmöglichkeit für den Algorithmus.

Das Computer Algebra System MATHEMATICA [Wo91] stellt aufgrund der im vorherigen Kapitel geführten Diskussion ein geeignetes Arbeitswerkzeug für solch eine experimentelle Untersuchung dar. Eine Implementierung des in Kapitel 5.1 strukturierten Algorithmus wird durch MATHEMATICA-Funktionen unterstützt, die algebraische und trigonometrische Umformungen und Verknüpfungen von Termen vornehmen (z. B. Simplify, Factor, Expand, GCD, TrigReduce, TrigExpand, PolynomialQuotient, PolynomialLCM), Funktionswerte berechnen, analytische Operationen durchführen (D[erivative], Integrate, Limit) und lineare Gleichungssysteme (Solve) lösen. Dabei kann in der Regel insbesondere in bezug auf die Lösung linearer Gleichungssysteme von mathematisch anerkannten und erfolgreichen Verfahren ausgegangen werden.

Keine oder nur unzureichende Unterstützung durch MATHEMATICA findet der Algorithmus bei spezifischen aber grundsätzlichen Problemen wie:

- Berechnung einer komplexen Partialbruchzerlegung analog zum Algorithmus 3.2-2 fehlt bei MATHEMATICA.
- Berechnung von Pochhammersymbolen: Obwohl sich MATHEMATICA mit seiner Funktion “Pochhammer” vielversprechend gibt, Pochhammersymbole zu berechnen, ist deren bereitgestellte Fähigkeit keine ausreichende Antwort auf die Vielfalt, der mit dem hypergeometrischen Ansatz erzeugten Pochhammersymbole. Hier müssen zahlreiche Vereinfachungsregeln aufgestellt werden, z. B. mit $n$, $k$, $a \in \mathbb{N}$:

$(1/2)_k = \dfrac{2k!}{4^k \cdot k!}$ , aus Beispiel 3.4-2,

$(n/2)_k = \dfrac{(n-1)_{2k}}{4^k \cdot ((n-1)/2)_k}$ , z. B. für $(3/2)_k$ aus Beispiel 2.4-4 und 3.4-2,

$(1/3)_k \cdot (2/3)_k = \dfrac{(3k)!}{k! \cdot 27^k}$ , z. B. bei $f(z) = e^{\sqrt[3]{z}}$,

$(a)_k = \dfrac{(a+k-1)!}{(a-1)!}$ , wegen $(y)_k$ bei $f(z) = (1+z)^{-y}$ ,

$$(-a)_k = (-a)(-a+1)\ldots(-a+(k-1)) = (-1)^k a(a-1)\ldots(a-(k-1)) = \frac{(-1)^k a!}{(a-k)!},$$

wegen $(-y)_k$ bei $f(z) = (1+z)^y$ .

Mit $(a)_k = k!\binom{a+k-1}{k} = (-1)^k k!\binom{-a}{k}$ können die Pochhammersymbole auch durch die vertrauteren Binominalkoeffizienten ersetzt werden.

- Zusätzliche Vereinfachungen: Nach Berechnung der allgemeinen Formel für die Koeffizienten der formalen Laurentreihe $\sum_{k=k_0}^{\infty} a_k z^k$ fehlen MATHEMATICA viele Kenntnisse zu deren Vereinfachung. Z.B. entstehen sehr häufig Terme der Form

  $z^n - \bar{z}^n$ mit $z \in \mathbb{C}$

  (vgl. Beispiele 3.2-3 und 3.5-4), für die eine zusätzliche Regel bekanntgegeben werden muß:

  $z^n - \bar{z}^n = 2i \cdot r^k \cdot \sin(k \cdot \phi)$ , mit $z = r(\cos\phi + i \cdot \sin\phi)$ .

Für die spätere Verwendung der formalen Laurentreihe ist es sinnvoll, eine einheitliche Ergebnisform festzulegen. Hierunter fällt das Bestreben, alle Summanden in die geschlossene Form der formalen Laurentreihe einzubeziehen oder - falls das nicht geht - den Anfangswert der Indexvariable zu normieren, z. B.:

$$f(z) = z + \arctan(z) = 2z + \sum_{k=1}^{\infty} \frac{(-1)^k}{2k+1} z^{2k+1} = z + \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} z^{2k+1} .$$

Es gilt auch, ein nach Lemma 2.4-3 aus $m$ ($m$-fach symmetrischen) Summen bestehendes Ergebnis wieder zusammenzufassen, insbesondere wegen des in Kapitel 4.1 eingeführten Ansatzes $g(z) = f(z^n)$ für eine “formale Laurentreihe” mit gebrochener Schrittfolge, z. B.:

$$f(z) = e^{\sqrt{z}} = \sum_{k=0}^{\infty} \frac{1}{(2k)!} z^k + \sum_{k=0}^{\infty} \frac{1}{(2k+1)!} z^{k+1/2} = \sum_{k=0}^{\infty} \frac{1}{k!} z^{k/2} .$$

Dabei handelt es sich vor allem um algebraische Umformungen, die mit Hilfe der Mustererkennung (pattern matching) gelöst werden und nicht das Anliegen der hier untersuchten

Methoden sind.

Aus den iterativen Algorithmusteilen ergibt sich spätestens mit einer experimentellen Implementierung die Frage nach Anfang und insbesondere Länge der Iterationsintervalle. Im vorliegendem Algorithmus ist die Angabe von $N_{max} \in \mathbb{N}$ zur Begrenzung der Suche einer einfachen Differentialgleichung mit dem Algorithmus 3.3-1 erforderlich, die Festlegung von Intervallanfang und -länge, $k_{init}$ und $n_{len}$, bei der Suche nach einer Initialisierung der Rekursionsgleichung im hypergeometrischen Ansatz sowie $n_{max} \in \mathbb{N}$ für ein maximales $n$ im speziellen Ansatz $g(z) = z^n \cdot f(z)$ (s. Abb. 5.1-3).

Ein großer Wert von $N_{max}$ bzw. $n_{max}$ als auch ein langes Intervall $[k_{init}, \ldots, k_{init} + n_{len}]$ können eine erhebliche Verlängerung der Bearbeitungszeit zur Folge haben und müssen deshalb sorgfältig gewählt sein. Für die Konstruktion einer einfachen Differentialgleichung ist

$$N_{max} := 4$$

geeignet, da mit zunehmender Ordnung der Differentialgleichung die Schwierigkeit zunimmt, daß die entsprechende Rekursionsgleichung mit dem hypergeometrischen oder rekurrenten Ansatz gelöst werden kann.

Im hypergeometrischen Ansatz wird von einer Rekursionsgleichung der Form $a_{k+m} = R(k)a_k$ mit $R = \frac{P}{Q}$ ausgegangen. Damit ist die Initialisierung dieser Rekursion bei $k_{init} \in \mathbb{Z}$ erst dann sinnvoll, falls es kein $k^* \in \mathbb{Z}$ gibt mit $k^* \geq k_{init}$ und $R(k^*) = 0$, d. h. es wird gefordert:

$$k_{init} > \max\{k \in \mathbb{Z} \quad | P(k) = 0\} .$$

Mit dem Beweis von Lemma 2.3-1 entsteht die Forderung $a_{k_{init}} \neq 0$. Die Situation

$$\forall k \in \{k_{init}, \ldots, k_{init} + n_{len}\} : a_k = 0$$

wird mit zunehmenden $n_{len}$ seltener, so daß ein kleiner Wert für $n_{len}$, z. B.

$$n_{len} := 3 ,$$

als angebracht erscheint. Eine mathematisch sinnvolle Abschätzung für $n_{max} \in \mathbb{N}$ ist nur bedingt möglich. Schließlich versucht der Ansatz $g(z) = z^n \cdot f(z)$, Funktionen wie z. B. $f(z) = \frac{\sin(z)}{z^{n_0}}$, $n_0 \in \mathbb{N}$, trotz einer Singularität bei $z = 0$ in eine formale Laurentreihe zu entwickeln. Die Festlegung

$$n_{max} := 10$$

ist dabei beliebig, da ein großes $n_0$ keinen Einfluß hat auf die Schwierigkeit, eine einfache Differentialgleichung zu konstruieren, die dann auf eine mit dem rekurrenten oder hypergeometrischen Ansatz lösbare Rekursionsgleichung führt.

In Fällen wie $f(z) = \frac{\sin(z)}{z^{100}}$, $f(z) = z^{100} \cdot \sin(z)$ oder auch $f(z) = z^N \cdot \sin(z)$ mit unbestimmtem $N \in \mathbb{N}$ wird es sinnvoller, den Algorithmus so zu erweitern, daß ausgehend von der mit dem Algorithmus aus Kapitel 5.1 gefundenen formalen Laurentreihe für $\sin(z)$ auch die formale Laurentreihe für $f(z)$ aufgestellt wird. Dies geschieht prinzipiell eher mit Hilfe von

Mustererkennung und ist nicht Ziel der experimentellen Implementierung der hier untersuchten Methoden.

Eine Liste von Funktionen, die unter Anwendung einer mit Mathematica erstellten experimentellen Implementierung der untersuchten Methoden, auf die Möglichkeit hin geprüft wurden, in eine formale Laurentreihe entwickelt zu werden, befindet sich in Anhang A. Die dabei genutzten Mathematica Programme sind im Rechenzentrum des Fachbereichs Mathematik der FU Berlin verfügbar [K92a] und gehören zu den Hilfsmitteln der in dieser Arbeit vorgenommenen Methodenanalyse. Die vom mathematischen Standpunkt aus geführte Diskussion und Benutzung dieses Werkzeugs soll dabei den um die Computer Algebra Systeme gewachsenen Arbeitsbereich des Mathematikers berücksichtigen.

# 6 Weitere Fragestellungen

## 6.1 Umkehrung der Methoden

Ausgehend von den Methoden aus Kapitel 3 ist es naheliegend, einen inversen Algorithmus zur Berechnung der erzeugenden Funktion einer formalen Laurentreihe zu betrachten:

$$\left( F = \sum_{k=k_0}^{\infty} a_k z^k \right) \rightarrow f(z) \,.$$

Dabei soll es nun weniger um eine ebenso ausführliche Erarbeitung wie bei den bereits diskutierten Entwicklungsmethoden gehen, sondern eher um eine aus didaktischen Gründen sinnvolle Intensivierung dieses Methodenverständnisses durch deren Betrachtung in entgegengesetzter Richtung.

Bei dem inversen Entwicklungsalgorithmus geht es prinzipiell um folgende Arbeitsschritte [K92, S. 23]:

<u>ALGORITHMUS 6.1-1:</u>

(a) Bestimmung einer Rekursionsgleichung für die Reihenkoeffizienten mit dem Ansatz:

$$\text{(6.1-1)} \qquad a_k + \sum_{j=1}^{m} p_j(k) a_{k+j} = 0, \qquad \text{mit } p_j(k) \text{ rational.}$$

(b) Aufstellung einer Differentialgleichung durch die Transformation:

$$k^n \cdot a_{k+j} \rightarrow \theta^n \frac{f(z)}{z^j}, \qquad \text{mit } \theta := z\frac{d}{dz}.$$

Aus der Rekursionsgleichung (6.1-1) entsteht damit die Differentialgleichung:

$$\text{(6.1-2)} \qquad f(z) + \sum_{j=1}^{m} p_j(\theta) \frac{f(z)}{z^j} = 0 \,.$$

(c) Bestimmung einer Lösung für die Differentialgleichung (6.1-2) und damit einer Funktion $f(z)$ mit $f(z) = F$. In Abhängigkeit von der vorliegenden Differentialgleichung bleiben für die gesuchte Funktion Parameter unbestimmt (z. B. $f(z) = c \cdot e^z$, $c \in \mathbb{C}$), die unter Beachtung entsprechender Anfangsbedingungen ($f(0) = a_0$, $f'(0) = a_1$, …, $f^{(k)}(0) = k! \cdot a_k$) berechnet werden können.

In Teil (a) von Algorithmus 6.1-1 ist zur gegebenen Koeffizientenfunktion $A$ mit $A(k) = a_k$ eine Gleichung der Form (6.1-1) zu konstruieren, d. h. einer Gleichung mit polynomialen Koeffizienten der Funktionen $A(k), A(k+1), \ldots, A(k+m)$. Diese Aufgabe kann mit dem Algorithmus 3.3-1 behandelt werden.

In Umkehrung zu Lemma 3.4-1 entsteht durch Teil (b) des Algorithmus 6.1-1 nun eine Differentialgleichung für $f(z) = \sum_{k=k_0}^{\infty} a_k z^k$, die der vorgegebenen Rekursionsgleichung (6.1-1) genügen soll. Die angegebene Transformation ist begründet durch:

$$\begin{aligned}\theta^n \frac{f(z)}{z^j} &= \theta^n\left(\sum_{k=k_0}^{\infty} a_k z^{k-j}\right)\\ &= \theta^n\left(\sum_{k=k_0-j}^{\infty} a_{k+j} z^k\right)\\ &= \sum_{k=k_0-j}^{\infty} (k^n \cdot a_{k+j}) z^k, \qquad \text{wegen (2.4-2).}\end{aligned}$$

Ein Beweis für die Gültigkeit der gefundenen Differentialgleichung im Fall einer Funktion vom hypergeometrischen Typ ist mit Satz 2.4-2 bereits gegeben. Es ist aber darauf zu achten, daß bevor die Transformation durch Teil (b) des Algorithmus 6.1-1 durchgeführt wird, die Voraussetzungen im Beweis von Satz 2.4-2 erfüllt sind, d. h. in $Q(k)a_{k+m} = P(k)a_k$ (Gleichung (2.4-3)) müssen die Polynome $P$ und $Q$ so gewählt sein, daß $Q(k_0-1) = \ldots = Q(k_0-m) = 0$ . Ansonsten multipliziere $\forall j \in \{1, \ldots, m\}$ mit $Q(k_0-j) \neq 0$ die Rekursionsgleichung mit $(k-k_0+j)$ .

Der Erfolg von Algorithmus 6.1-1 bleibt aber insbesondere von der Möglichkeit abhängig, die Differentialgleichung (6.1-2) zu lösen. Die Algorithmisierung von Teilen der Theorie gewöhnlicher Differentialgleichungen liegt jedoch außerhalb der Methodenanalyse dieser Arbeit, so daß an dieser Stelle nur die Benutzung von Spezialliteratur [z. B. Ka77] oder in dieser Frage fortgeschrittener Computer Algebra Systeme (z. B. “ode”-Funktion, Ordinary Differential Equation, von MACSYMA) empfohlen wird.

Der Algorithmus 6.1-1 kann als Umkehrung des hypergeometrischen bzw. rekurrenten Lösungsansatzes (Kapitel 3.3 - 3.5) verstanden werden. Eine Methode zur direkten Umkehrung des rationalen Ansatzes aus Kapitel 3.2 müßte sich - wegen der dem rationalen Ansatz zugrundeliegenden Äquivalenzschritte - mit dem Muster der Koeffizientenfunktion der gegebenen formalen Laurentreihe beschäftigen. Im Gegensatz zum Kriterium für die Anwendung des rationalen Ansatzes, nämlich der rationalen Form der zu entwickelnden Funktion $R(z) = \frac{P(z)}{Q(z)}$ mit $P(z)$ und $Q(z)$ Polynome, wäre es für eine direkte Umkehrung wegen (3.2-1) und (3.2-2) notwendig, für die Koeffizienten in $\sum_{k=k_0}^{\infty} a_k z^k$ mit algebraischen Umformungen die folgende Form zu gewinnen:

$$a_k = \sum_{t=1}^{m} \sum_{j=1}^{q_t} \frac{(-1)^j \cdot c_{tj}}{\zeta_t^{j+k}} \binom{j+k-1}{k} \text{ mit } \zeta_t, c_{tj} \in \mathbb{C} \text{ und } m, q_t \in \mathbb{N}.$$

Offensichtlich ist dieses Kriterium im allgemeinen nicht trivial und ohne besondere zu entwickelnde Strategien wenig handhabbar. Da der Algorithmus 6.1-1 auch auf rationale Funktionen $f(z)$ anwendbar ist (vgl. Beispiel 6.1-3), sollte die Umkehrung der Methoden nur von diesem universellen Algorithmus ausgehen.

BEISPIEL 6.1-2:

Der Algorithmus 6.1-1 wird angewandt auf $F = \sum_{k=0}^{\infty} \frac{z^k}{k!}$.

Teil (a): $a_k = \frac{1}{k!}$ und $a_{k+1} = \frac{1}{(k+1)!}$,

Ansatz: $a_k + p_1(k)a_{k+1} = 0$, d. h. $\frac{1}{k!} + p_1(k) \cdot \frac{1}{(k+1)!} = 0$,
und damit folgt bereits: $p_1(k) = -(k+1)$
sowie die Rekursionsgleichung: $a_k - (k+1)a_{k+1} = 0$.

Teil (b): Aus $a_k - ka_{k+1} - a_{k+1} = 0$ folgt:

$$
\begin{aligned}
0 &= f - z\frac{d}{dz}\left(\frac{f}{z}\right) - \frac{f}{z}\\
&= f - z\left(\frac{f'z - f}{z^2}\right) - \frac{f}{z}\\
&= f - f' + \frac{f}{z} - \frac{f}{z}\\
&= f - f'.
\end{aligned}
$$

Teil (c): Lösung der Differentialgleichung unter Berücksichtigung der Bedingung $f(0) = a_0 = 1$ ist offensichtlich $f(z) = e^z$.

BEISPIEL 6.1-3:

Sei $F = \sum_{k=0}^{\infty} (k+1)z^k$ gegeben, also $a_k = k+1$ und $a_{k+1} = k+2$.

Der Ansatz $a_k + p_1(k)a_{k+1} = 0$ bzw. $p_1(k) = -\frac{k+1}{k+2}$ führt auf die Rekursionsgleichung:

$$(k+2)a_k - (k+1)a_{k+1} = 0 .$$

Mit der Transformation aus Teil (b) von Algorithmus 6.1-1 folgt die Differentialgleichung:

$$2 \cdot f + (z-1)f' = 0.$$

Eine Lösung dieser Differentialgleichung mit $f(0) = 1$ ist die rationale Funktion $f(z) = \frac{1}{(1-z)^2}$.

BEISPIEL 6.1-4:

Gegeben sei $F = \sum_{k=0}^{\infty} (1 + (-1)^k)z^k$, d. h. $a_k = 1 + (-1)^k$.

Aus dem Ansatz $a_k + p_1(k)a_{k+1} = 0$ ergibt sich $p_1(k) = -\frac{1+(-1)^k}{1+(-1)^{k+1}} = \frac{1+(-1)^k}{-1+(-1)^k}$ .

Dabei ist $p_1(k)$ nicht-rational (ansonsten wäre $h(k):=(-1)^k$ rational, wegen:

$$p_1(k)=\frac{1+h(k)}{-1+h(k)} \quad \Leftrightarrow \quad h(k) = \frac{1+p_1(k)}{-1+p_1(k)} \quad )$$

und führt damit auf den Ansatz $a_k + p_1(k)a_{k+1} + p_2(k)a_{k+2} = 0$ .

Mit der Lösung $p_1(k) \equiv 0$ und $p_2(k) \equiv -1$ gilt die rekursive Beziehung $a_k = a_{k+2}$.

Auf direktem Weg ergibt sich sofort:

$$\begin{aligned}
0 &= \sum_{k=0}^{\infty} (a_k - a_{k+2})z^k \\
&= \sum_{k=0}^{\infty} a_k z^k - \sum_{k=2}^{\infty} a_k z^{k-2} \\
&= f(z) - \frac{1}{z^2}(f(z) - a_0 - a_1 z)
\end{aligned}$$

$$\Leftrightarrow \quad 0 = z^2 f(z) - f(z) + a_0 + a_1 z$$

$$\Leftrightarrow \quad (1-z^2)f(z) = a_0 + a_1 z = 2$$

$$\Leftrightarrow \quad f(z) = \frac{2}{1-z^2} .$$

Soll der Algorithmus 6.1-1 verfolgt werden, ist zunächst die Rekursionsgleichung mit $(k-k_0+1)(k-k_0+2)$ zu multiplizieren, denn für den polynomialen Koeffizient $Q(k)$ von $a_{k+2}$ gilt $Q(k_0-1) = Q(k_0-2) = 1 \neq 0$ . Damit ergibt sich die Rekursionsgleichung

$$(k+2)(k+1)a_{k+2} = (k+2)(k+1)a_k$$

und nach Transformation die Differentialgleichung

$$(1-z^2)f''(z) - 4zf'(z) - 2f(z) = 0 .$$

Mit den Nebenbedingungen $f'(0) = 0$ und $f''(0) = 0$ folgt die Lösung $f(z) = \frac{2}{1-z^2}$.

## 6.2 Bestimmung des Konvergenzradius

Der Algorithmus 6.1-1 erinnert nicht nur an den Algorithmus aus Kapitel 5.1, sondern beinhaltet auch gleiche Problemstellungen (z. B. Algorithmus 3.3-1 und Teil (a) aus Algorithmus 6.1-1). Solche Überschneidungen zeigen, daß verwandte Fragestellungen mit Bausteinen eines gemeinsamen erweiterten Algorithmus behandelt werden können. Eine vergleichbare Situation besteht bei der Betrachtung des Konvergenzradius der formalen Laurentreihen.

Die Berechnung des Konvergenzradius ist bereits mit dem Vorliegen einer Rekursionsgleichung für die Reihenkoeffizienten möglich und häufig sogar einfacher als anhand einer geschlossenen Formel für $a_k$. Beim rekurrenten Ansatz hat die erzeugende Funktion wegen (3.5-5), (3.5-6) und

(3.5-7) die Form:

$$(6.2\text{-}1) \qquad f(z) = \sum_{j=1}^{N} c_j z^{p_j} e^{\lambda_j z} \text{ mit } c_j, \lambda_j \in \mathbb{C} \text{ und } p_j, N \in \mathbb{N}_0.$$

Die Exponentialfunktionen $e^{\lambda_j z}$ haben den Konvergenzradius $r = \infty$ und damit auch $f(z)$.

Im Fall einer Funktion vom hypergeometrischen Typ besitzt die Rekursionsgleichung die Form (2.4-1). Nach Lemma 2.4-3 kann diese Funktion in $m$ hypergeometrische Reihen aufgeteilt werden:

$$\sum_{k=k_0}^{\infty} a_{mk+s} z^{mk+s} \text{ mit } s = 0, 1, \ldots, (m-1),$$

wobei die Koeffizienten $a_n$ und $a_{n+m}$ mit $n = mk+s$ jeweils aufeinander folgen. Analog zum Quotientenkriterium [W90, S. 96] konvergiert $\sum_{k=k_0}^{\infty} a_{mk} z^{mk}$, falls der Grenzwert existiert von:

$$\lim_{k\to\infty} \left| \frac{a_{mk+m} z^{mk+m}}{a_{mk} z^{mk}} \right| = \left|z^m\right| \cdot \lim_{k\to\infty} |R(mk)|, \text{ bzw. } |z| < \left( \lim_{k\to\infty} |R(mk)| \right)^{-1/m}.$$

Für die anderen Reihen mit $s = 1, \ldots, (m-1)$ gilt dies analog wegen:

$$\lim_{k\to\infty} |R(mk+s)| = \lim_{k\to\infty} |R(mk)|,$$

so daß insgesamt für den Konvergenzradius folgt [He89, S. 825]:

$$(6.2\text{-}2) \qquad r = \frac{1}{\lim\limits_{k\to\infty} \sqrt[m]{|R(mk)|}}.$$

Da $R(mk) = \left( \sum_{j=0}^{a} p_j (mk)^j \right) \Big/ \left( \sum_{j=0}^{b} q_j (mk)^j \right)$ mit $p_j, q_j \in \mathbb{C}$ rational ist, gilt sogar:

$$(6.2\text{-}3) \qquad r = \begin{cases} 0 & \text{, falls } a < b \\ \infty & \text{, falls } a > b \\ \sqrt[m]{\left|\dfrac{q_a}{p_a}\right|} & \text{, falls } a = b \end{cases}.$$

Die Bestimmung des Konvergenzradius mit Hilfe der Rekursionsformel ist sogar dann noch möglich, falls kein Lösungsansatz zu einer geschlossenen Formel für $a_k$ führt. Dies soll an einem Beispiel verdeutlicht werden:

Gegeben sei die Rekursionsgleichung $a_n = P(n)a_{n-1} + Q(n)a_{n-2}$, $P(n)$ und $Q(n)$ rational. Es folgt $\frac{a_n}{a_{n-1}} = P(n) + Q(n)\frac{a_{n-2}}{a_{n-1}}$ und mit $b_n := \frac{a_n}{a_{n-1}}$:

$$b_n = P(n) + Q(n) \cdot \frac{1}{b_{n-1}} \qquad \text{bzw.} \qquad \frac{1}{b_n \cdot b_{n-1}} + \frac{P(n)}{Q(n)} \cdot \frac{1}{b_n} - \frac{1}{Q(n)} = 0 \quad .$$

Nach dem Quotientenkriterium gilt für den Konvergenzradius $r = \lim\limits_{n \to \infty} \frac{1}{b_n}$, so daß sich dieser als Lösung einer quadratischen Gleichung bestimmen läßt:

$$\text{(6.2-4)} \qquad r = \lim_{n \to \infty} \left( -\frac{P(n)}{2 \cdot Q(n)} + \sqrt{\frac{P^2(n)}{4 \cdot Q^2(n)} + \frac{1}{Q(n)}} \right)$$

$$= \lim_{n \to \infty} \left( -\frac{P(n) + \sqrt{P^2(n) + 4 \cdot Q(n)}}{2 \cdot Q(n)} \right) \quad .$$

Bei Vorliegen einer rationalen Funktion $f(z)$ folgt der Konvergenzradius direkt aus $f(z) = \frac{P_0(z)}{Q_0(z)}$ oder - sofern nur eine Ableitung von $f(z)$ rational ist und da der Konvergenzradius von $f(z)$ mit denen seiner Ableitungen übereinstimmt - aus $f^{(k)}(z) = \frac{P_k(z)}{Q_k(z)}$ . In beiden Fällen wird der Konvergenzradius durch die Nullstellen des Nennerpolynoms eingeschränkt, so daß gilt:

$$\text{(6.2-5)} \qquad r := \min\{z_N \mid Q_k(z_N) = 0\} \quad .$$

<u>BEISPIEL 6.2-1:</u>

Sei $f(z) = \sqrt{2 - z^3}$.

Der hypergeometrische Ansatz liefert

$$a_{k+3} = \frac{2k-3}{4k+12} \cdot a_k \text{ und } f(z) = \sum_{k=0}^{\infty} -\frac{(1/8)^k \sqrt{2}(2k)!}{(2k-1)k!^2} z^{3k} \quad .$$

Mit (6.2-3) folgt als Konvergenzradius $r = \sqrt[3]{\frac{4}{2}} = \sqrt[3]{2}$.

# 7 Zusammenfassung und Ausblick

Die in dieser Arbeit vorgestellten Methoden lassen sich zu einem konstruktiven Algorithmus zusammenfassen, der für eine Vielzahl komplexwertiger Funktionen $f(z)$ jeweils eine geschlossene Darstellung durch eine formale Reihe $F$ bestimmt. Die Berücksichtigung von Singularitäten und Puiseux-Reihen vergrößert den Anwendungsbereich der mit dem Algorithmus verbundenen Transformation $f(z) \to F$. Es werden nicht nur formale Laurentreihen im eigentlichen Sinne konstruiert, sondern darüberhinaus formale Reihen der Form:

$$F = P(z) \cdot \ln(z) + \sum_{k=k_0}^{\infty} a_k z^{v \cdot k/n} \text{, mit } n \in \mathbb{N},\ v \in \{1, -1\} \text{, } P \text{ Polynom und } a_k \in \mathbb{C}.$$

Im wesentlichen basiert der Algorithmus auf zwei Grundideen:

(1) Eine komplexwertige rationale Funktion oder deren rationale Ableitung kann durch Partialbruchzerlegung auf Terme zurückgeführt werden, die einer Binominalreihe entsprechen.

(2) Mit einer linearen Differentialgleichung für $f(z)$, die nur polynomiale Koeffizienten besitzt, liegt für die Koeffizienten der formalen Reihe eine Rekursionsformel vor, für die eine nichtrekursive Formel aufgestellt werden muß. Die Existenz und die Form einer solchen Lösungsformel wurde bewiesen für die Rekursionsgleichungstypen:

(a) $a_{k+m} = R(k) \cdot a_k$, mit $R(k)$ rational, und

(b) $a_{k+m} = \sum_{j=0}^{m-1} c_j \cdot a_{k+j}$, mit $c_j \in \mathbb{C}$.

Beide Grundideen können aufgrund ihres konstruktiven Charakters mit einer symbolischen Programmiersprache automatisiert werden. Die Benutzung einer solchen Implementierung zeigt z. B., daß sich insbesondere der Lösungsweg (2) zum Beweis elementarer Identitäten (z. B. $\operatorname{arcsinh}(z) = \ln(z + \sqrt{1+z^2})$ ) oder einfacher trigonometrischer Umformungen und Symmetriebeziehungen eignet. Die umgekehrte Transformation $F \to f(z)$ als auch die Bestimmung des Konvergenzradius steht ebenfalls im engen Zusammenhang mit dem zweiten Lösungsweg, da die Algorithmen zur Konstruktion einer Differentialgleichung bzw. zur Aufstellung einer rekursiven Koeffizientenformel erneut zur Anwendung gelangen.

Mit der Beschränkung der Methoden auf die beiden o. g. Rekursionsgleichungstypen werden der Bedeutung des Algorithmus deutliche Grenzen gesetzt. Eine naheliegende Vergrößerung der Methodenvielfalt, insbesondere um auch Koeffizienten mit Bernoullizahlen zu berücksichtigen, ist sicher sehr lohnend. Anregungen könnten dabei vielleicht weitere Regeln aus der Theorie der erzeugenden Funktionen geben, z. B. [Wi90, S. 39]:

$$f \overset{\text{egf}}{\leftrightarrow} \{a_k\}_0^\infty, g \overset{\text{egf}}{\leftrightarrow} \{b_k\}_0^\infty \Rightarrow \qquad f \cdot g = \sum_{k=0}^{\infty} c_k z^k, \quad \text{mit } c_k = \sum_{n=0}^{k} \binom{k}{n} a_n b_{k-n}.$$

Ausgehend von Algorithmen zur Entwicklung komplexwertiger Funktionen, deren Umkehrung und der Berechnung zugehöriger Konvergenzradien auf der Basis von Differentialgleichungen und rekursiver Koeffizientengleichungen, bestehen zugleich die Grundelemente für einen Algorithmus zur analytischen Fortsetzung komplex differenzierbarer Funktionen. Bei Benutzung eines Computer Algebra Systems läßt sich das Kreiskettenverfahren [BS76, S. 179] automatisieren. Mit der Festlegung einiger Strategien für eine Vergrößerung des Konvergenzbereichs in der komplexen Ebene durch wiederholte analytische Fortsetzung einer Funktion, entsteht somit eine andere interessante Erweiterungsmöglichkeit der Methoden und Algorithmen, die in dieser Arbeit dargestellt wurden.

Mit der Verbreitung der Computer Algebra Systeme gewinnen algorithmisierte Methoden in der Mathematik mehr an Bedeutung. Die zugrundeliegenden Ideen sollten fundiert sein und können deshalb keinesfalls ohne den Beweis ihrer mathematischen Aussage programmiert werden. Für den Mathematiker entsteht hier ein wichtiges Arbeitsfeld, von dessen Ergebnissen neue Impulse für die Mathematik ausgehen können. Der Lohn für diese “neuen” Aufgaben steckt dabei in der Zuverlässigkeit, Schnelligkeit und den Anregungen durch die Arbeit mit Computer Algebra Systemen.

Allmählich werden Computer Algebra Systeme, die bereits heute schon auf nur 300 Gramm schweren sog. Palmtop-PC’s (Maße $16\ \mathrm{cm} \times 8{,}6\ \mathrm{cm} \times 2{,}5\ \mathrm{cm}$ ) arbeiten können [Ku91, S. 23], den herkömmlichen numerischen oder programmierbaren Taschenrechnern eine starke Konkurrenz sein. Während bisherige Konzepte für den Computereinsatz im Mathematikunterricht für die unterschiedlichen Anwendungen in Arithmetik, Geometrie und Sachrechnen verschiedene Software-Werkzeuge benötigen [MPW86, S. 8], stehen nun einheitliche Gesamtwerkzeuge zur Verfügung. In der Bundesrepublik Deutschland ist die Diskussion um deren Bildungswert im Schulfach Mathematik unter Lehrern und Didaktikern sehr intensiv. Diese didaktische Frage kann im Rahmen dieser Arbeit natürlich nicht behandelt werden, es bleibt aber festzuhalten, daß auch kritische Betrachtungen die Benutzung eines Computer Algebra Systems im Mathematikunterricht zumindest im Schulalltag der Sekundarstufe II befürworten und die Aufstellung eines entsprechenden didaktischen Forschungsprogramms anregen [Sch91].

# Anhang A Anwendungsbeispiele

Zur Veranschaulichung der Breite der Anwendbarkeit des in Kapitel 5 diskutierten Algorithmus wird hier eine Auswahl komplexwertiger Funktionen gegeben, zu denen der Algorithmus eine auf eine formale Laurentreihe zurückzuführende Darstellung findet. In der 5. Gruppe sind Funktionen genannt, für die der Algorithmus nicht auf eine formale Laurentreihe führt. Die Beispiele wurden unter Benutzung einer MATHEMATICA Implementierung des Algorithmus [K92a] ausgewählt.

1. Gruppe (Lösung mit dem rationalen Ansatz):

(A.1-1) $$\frac{z}{1-az} = -\frac{1}{a} + \sum_{k=0}^{\infty} a^{k-1} z^k ,$$

(A.1-2) $$\frac{a}{(b-z)^3} = \sum_{k=0}^{\infty} \frac{a(1+k)(2+k)}{2b^{k+3}} z^k ,$$

(A.1-3) Fibonacci-Zahlen:

$$\frac{1}{1-z-z^2} = \sum_{k=0}^{\infty} \frac{(1+\sqrt{5})^{1+k} - (1-\sqrt{5})^{1+k}}{2^{k+1}\sqrt{5}} z^k ,$$

(A.1-4) Beispiel 3.2-3:

$$\frac{1}{z^2+2z+2} = \sum_{k=0}^{\infty} \frac{(-1)^k \sin\left(\frac{\pi(k+1)}{4}\right)}{2^{(k+1)/2}} z^k .$$

Rationaler Ansatz angewandt auf die 1. Ableitung:

(A.1-5) $$\arctan(z+y) = \arctan(y) + \sum_{k=0}^{\infty} \frac{\frac{i}{2}(-1)^k((y-i)^{k+1} - (y+i)^{k+1})}{(k+1)(y^2+1)^{k+1}} z^{k+1} .$$

2. Gruppe (Lösung mit dem hypergeometrischen Ansatz):

Alle Funktionen (2.2-1) - (2.2-11),

(A.2-1) $$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt = \sum_{k=0}^{\infty} \frac{2 \cdot (-1)^k}{\sqrt{\pi}(1+2k)!k!} z^{1+2k} ,$$

(A.2-2) $$e^{z^2} \operatorname{erf}(z) = \sum_{k=0}^{\infty} \frac{2 \cdot 4^k \cdot k!}{\sqrt{\pi}(1+2k)!} z^{1+2k} ,$$

(A.2-3) $$\frac{e^z \sinh(z)}{z^3} = \sum_{k=0}^{\infty} \frac{2^k}{(1+k)!} z^{k-2} ,$$

(A.2-4) $$e^{z+y} = \sum_{k=0}^{\infty} \frac{e^y}{k!} z^k ,$$

(A.2-5) $$z \cdot \operatorname{arccot}(z) = \frac{\pi z}{2} + \sum_{k=0}^{\infty} -\frac{(-1)^k}{2k+1} z^{2k+2} \quad ,$$

(A.2-6) $$(1 + \sqrt{1-z})^y = \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}\right)^k 2^y (-y)_{2k}}{k!(1-y)_k} z^k \quad ,$$

(A.2-7) $$\sin(z)\cosh(z) = \sum_{k=0}^{\infty} \frac{(-4)^k}{(1+4k)} z^{4k+1} + \sum_{k=0}^{\infty} \frac{2(-4)^k}{(3+4k)} z^{4k+3} \quad ,$$

(A.2-8) $$\frac{\sin(3 \cdot \arccos(z))}{\sqrt{1-z^2}} = \sum_{k=0}^{\infty} -\frac{(-4)^k (1+k)!}{(1-k)!(2k)!} z^{2k} = -1 + 4z^2 \quad ,$$

(A.2-9) $$\cos(4 \cdot \arccos(z)) = \sum_{k=0}^{\infty} \frac{2(-4)^k (1+k)!}{(2-k)!(2k)!} z^{2k} = 1 - 8z^2 + 8z^4 \quad ,$$

(A.2-10) $$\ln\left(\frac{1+z}{1-z}\right) = \sum_{k=0}^{\infty} \frac{2}{2k+1} z^{2k+1} \quad .$$

Mit dem Ansatz $g(z) = f(1/z)$ :

(A.2-11) $$\sin\left(\frac{1}{z}\right) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} z^{-(2k+1)} \quad ,$$

(A.2-12) $$e^{-\frac{1}{z^2}} = \sum_{k=0}^{\infty} \frac{4^k k!}{(2k)!} z^{-2k} \quad .$$

Mit dem Ansatz $g(z) = f'(z)$ :

(A.2-13) $$\operatorname{Ci}(z) = -\int_z^{\infty} \frac{\cos(t)}{t} dt = C + \ln(z) + \sum_{k=1}^{\infty} \frac{(-1)^k}{2k(2k)!} z^{2k}$$ , ($C$ = Eulersche Konstante),

(A.2-14) $$\operatorname{arcsech}(z) = \ln(2) - \ln(z) + \sum_{k=1}^{\infty} \frac{\left(\frac{1}{4}\right)^k (2k)!}{2k \cdot k!^2} z^{2k} \quad .$$

Mit dem Ansatz $g(z) = f(z^n)$ :

(A.2-15) $$\operatorname{erf}(\sqrt{z}) = \sum_{k=0}^{\infty} \frac{2 \cdot (-1)^k}{\sqrt{\pi}(k! + 2 \cdot k \cdot k!)!} z^{k+1/2} \quad ,$$

(A.2-16) $$B_z(1/2, 3) = \int_0^z t^{-1/2}(1-t)^2 dt = \sum_{k=0}^{\infty} \frac{4(-1)^k}{(2k+1)(2-k)!k!} z^{k+1/2} \quad ,$$

(A.2-17) $$B_z(1/2, 2)\sqrt{z} = \int_0^z t^{-1/2}(1-t) dt = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1/2)(1-k)!k!} z^{k+1} = 2z - \frac{2z^2}{3} \quad .$$

3. Gruppe (Lösung mit dem rekurrenten Ansatz):

(A.3-1) Beispiel 3.3-3 / 3.5-4:

$$e^z \sin(z) = \sum_{k=1}^{\infty} -\frac{2^{k/2} \sin\left(-\frac{\pi \cdot k}{4}\right)}{k!} z^k \quad ,$$

(A.3-2) $$e^z + \sin(z) = \sum_{k=0}^{\infty} \frac{2 - i^{k+1} + (-1)^k i^{k+1}}{2k!} z^k \quad ,$$

(A.3-3) $$(e^z + \sin(z))^2 = \frac{1}{2} + \sum_{k=0}^{\infty} -\frac{(-2)^k i^k - 4i(1-i)^k + 4i(1+i)^k - (4 \cdot 2^k) + 2^k i^k}{4k!} z^k \quad .$$

4. Gruppe (Direkte Lösung, da höchstens endlich viele Koeffizienten ungleich 0):

(A.4-1) $\arcsin(\sin(z)) = z$ ,

(A.4-2) $\arctan(\cot(z)) = \frac{\pi}{2} - z$ ,

(A.4-3) $(\cos(z)^2 + \sin(z)^2)^2 = 1$ ,

(A.4-4) $\cos(\arcsin(z)) - \sqrt{1-z^2} = 0$ ,

(A.4-5) $\arcsin(z) + i \cdot \operatorname{arcsinh}(i \cdot z) = 0$ .

5. Gruppe (Ohne Lösungsansatz):

(A.5-1) $\sqrt{\frac{1+z}{1-z}}$, wegen der Rekursionsgleichung $a_{k+2} = \frac{ka_k + a_{k+1}}{k+2}$,

(A.5-2) $\arctan(z)^2$ , wegen $a_{k+4} = -\frac{(k^2+k)a_k + (2k^2+8k+8)a_{k+2}}{(k+3)(k+4)}$ ,

(A.5-3) $\frac{e^z \sin(z)}{z}$ , wegen $a_{k+2} = -2\frac{a_k - (2+k)a_{k+1}}{(2+k)(3+k)}$ .

## Anhang B Literaturliste

[AS89] ALMSICK, Maarkus van, SCHULTEN, Klaus: “MATHEMATICA - Mathematik auf Mikrocomputern”, in: *mc*, Heft 11/89, Franzis-Verlag, München 1989, S. 42 - 59.

[BGK92] BEN-ISRAEL, A., GILBERT, R. P., KOEPF, W.: *Analysis mit DERIVE*, Buchmanuskript und Begleitmaterial zur Analysis-Vorlesung an der FU, Berlin 1992.

[BS76] BEHNKE, Heinrich, SOMMER, Friedrich: *Theorie der analytischen Funktionen einer komplexen Veränderlichen*, 3. Auflage, Springer-Verlag, Berlin 1976.

[C91] CAVINESS, B. F., et al.: *An Introduction to Applied Symbolic Computation using MACSYMA*, Technical Report, University of Delaware 1991.

[Ch88] CHAR, Bruce W., et al.: *MAPLE Reference Manual*, Fifth Edition, Watcom Publications, Waterloo (Ontario) 1988.

[DR72] DIEDERICH, Klas, REMMERT, Reinhold: *Funktionentheorie I*, Springer-Verlag, Berlin 1972.

[DST88] DAVENPORT, J. H., SIRET, Y., TOURNIER, E.: *Computer Algebra systems and algorithms for algebraic computation*, Academic Press, London 1988.

[FWH91] FUCHSSTEINER, B., WIWIANKA, W., HERING, K. (Hrsg.): *mathPAD*, Vol. 1, Heft 3, Sonderausgabe zur Jahrestagung der Deutschen Mathematiker Vereinigung (DMV) vom 16. - 20.9.91 in Bielefeld, GH Paderborn 1991.

[G78] GOSPER, R. William: “Decision procedure for indefinite hypergeometric summation”, in: *Proceedings of the National Academy of Sciences of the USA*, Vol. 75, No. 1, National Academy of Sciences, Washington (D. C.) 1978, p. 40 - 42.

[GKP88] GRAHAM, R. L., KNUTH, D. E., PATASHNIK, O.: *Concrete mathematics: a foundation for Computer science*, Addison-Wesley Publishing Co., Reading (Massachusetts) 1988.

[GM90] GILLIGAN, Lawrence G., MARQUARDT, James F.: *Calculus and the DERIVE Program: Experiments with the Computer*, Second Edition, Gilmar Publishing Co., Cincinnati (Ohio) 1990.

[GR80] GRADSHTEYN, Izrail S., RYZHIK, Iosif M.: *Table of Integrals, Series, and Products*, Corrected and Enlarged Edition, Academic Press, New York 1980.

[H74] HENRICI, Peter: *Applied and computational complex analysis*, Volume 1, Wiley-Interscience publication, New York 1974.

[Ha75] HANSEN, Eldon R.: *A table of series and products*, Prentice-Hall, Englewood Cliffs (New York) 1975.

[Hä89] HÄRING, Reinhard: “Computer-Algebra Systeme”, in: *mc*, Heft 11/89, Franzis-Verlag, München 1989, S. 60 - 64.

[He89] HERRON, Isom H.: “The Radius of Convergence of Power Series Solutions to Linear Differential Equations”, in: *The American Mathematical Monthly*, Vol. 96, No. 11, Mathematical Association of America, Washington (D. C.) 1989, p. 824 - 827.

[K92] KOEPF, Wolfram: "Power series in Computer Algebra", erscheint in: *Journal of Symbolic Computation*, Academic Press, London 1992; Preprint: Fachbereich Mathematik Nr. B-92-3, FU Berlin 1992.

[K92a] KOEPF, Wolfram: *MATHEMATICA package 'PowerSeries'*, Fachbereich Mathematik der FU, Berlin 1992.

[Ka77] KAMKE, Erich: *Differentialgleichungen: Lösungsmethoden und Lösungen*, 9. Auf-lage, Teubner-Verlag, Stuttgart 1977.

[Ko83] KOECHER, Max: *Lineare Algebra und analytische Geometrie*, Springer-Verlag, Berlin 1983.

[Ku91] KUTZLER, Bernhard: "Der Mathematik-Assistent - DERIVE Version 2", in: [FWH91], S. 21 - 24.

[Li68] LIU, C. L.: *Introduction to Combinatorial Mathematics*, McGraw-Hill Book Company, New York 1968.

[Lu80] LUEKER, George S.: "Some Techniques for Solving Recurrences", in: *Computing Surveys*, Vol. 12, No. 4, ACM Press, New York 1980, p. 419 - 436.

[M91] MONAGAN, Michael: "Research and Development in MAPLE", in: [FWH91], S. 6 - 9.

[MPW86] MENZEL, K., PROBST, G., WERNER, W.: *Computereinsatz im Mathematik-unterricht, Band 2, Materialien für die Klassenstufen 9 und 10*, Verlage Metzler und Teubner, Stuttgart 1986.

[N75] NORMAN, A. C.: "Computing with Formal Power Series", in: *Transactions on Mathematical Software*, Vol. 1, No. 4, ACM Press, New York 1975, p. 346 - 356.

[S92] SCHEU, Günter: "Entdeckungen in der Menge der Primzahlen mit DERIVE", in: *Praxis der Mathematik, Sekundarstufen 1 und 2*, 34. Jahrgang, Heft 3, Aulis Verlag Deubner & Co., Köln 1992, S. 119 - 122.

[Sch91] SCHÖNWALD, Hans G.: "Zur Evaluation von DERIVE", in: *Didaktik der Mathematik*, 19. Jahrgang, Heft 4, Bayerischer Schulbuch-Verlag, München 1991, S. 252 - 265.

[Si90] SINGER, Michael F.: "Formal Solutions of Differential Equations", in: *Journal of Symbolic Computation*, Vol. 10, No. 1, Academic Press, London 1990, p. 59 - 94.

[T89] TACK, Lutz H.: *Die Methode von Berman zur konformen Abbildung einfach zusammenhängender Gebiete und eine Implementierung des Verfahrens in einem algebraischen Programmsystem*, Diplomarbeit in Mathematik an der Mathematisch-Naturwissenschaftlichen Fakultät der RWTH, Aachen 1989.

[W90] WALTER, Wolfgang: *Analysis I*, 2. Auflage, Springer-Verlag, Berlin 1990.

[W90a] WALTER, Wolfgang: *Gewöhnliche Differentialgleichungen*, 4. Auflage, Springer-Verlag, Berlin 1990.

[Wi90] WILF, Herbert S.: *generatingfunctionology*, Academic Press, Boston (Massachusetts) 1990.

[Wo91] WOLFRAM, Stephen: *MATHEMATICA - A System for Doing Mathematics by Computer*, Second Edition, Addison-Wesley Publishing Co., Redwood City (California) 1991.