

Zeitschrift:	Elemente der Mathematik
Herausgeber:	Schweizerische Mathematische Gesellschaft
Band:	35 (1980)
Heft:	2
Artikel:	Zur Behandlung des euklidischen Algorithmus bei Polynomen mit einem programmierbaren Taschen-Rechner
Autor:	Jeger, M.
DOI:	https://doi.org/10.5169/seals-34679

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 13.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

ELEMENTE DER MATHEMATIK

Revue de mathématiques élémentaires – Rivista di matematica elementare

*Zeitschrift zur Pflege der Mathematik
und zur Förderung des mathematisch-physikalischen Unterrichts*

El. Math.

Band 35

Heft 2

Seiten 25–48

Basel, 10. März 1980

Zur Behandlung des euklidischen Algorithmus bei Polynomen mit einem programmierbaren Taschen-Rechner

Aus der Algebra sind verschiedene Verfahren bekannt, die das Auffinden von reellen Nullstellen bei Polynomen stark vereinfachen und die allesamt auf dem *euklidischen Algorithmus im Polynomring über dem Körper der reellen Zahlen* beruhen. Sie haben aber für den Praktiker den Nachteil, dass sie ohne adäquate Rechen-Prothese nur schwerfällig zu handhaben sind. Im folgenden wird gezeigt, dass schon ein programmierbarer Taschen-Rechner mit der heute üblichen Kapazität bei der Bewältigung solcher Probleme sehr nützlich sein kann.

Der Beitrag verfolgt aber noch ein zweites Ziel. Er möchte gleichzeitig darauf hinweisen, dass mit einem programmierbaren Taschen-Rechner im Mathematikunterricht neue Akzente gesetzt werden können. Durch den vermehrten Einbau von Beispielen und Anwendungen wird eine gezielte *Pflege des algorithmischen Denkens* möglich. Im Zeitalter der Computer-Mathematik ist der Umgang mit Algorithmen eine nicht mehr vernachlässigbare Komponente der mathematischen Ausbildung, und zwar sowohl auf der Gymnasialstufe wie auch auf der universitären Ebene.

Diese Note befasst sich insbesondere mit dem euklidischen Algorithmus im Polynomring über dem Körper der rationalen und über dem Körper der reellen Zahlen. Bei den Polynomen mit rationalen Koeffizienten besteht bekanntlich die Möglichkeit, den euklidischen Algorithmus mit ganzen Zahlen zu beschreiben. In didaktischer Hinsicht weisen Algorithmen im Bereich der Ganzahligkeit mancherlei Vorteile auf. So treten dort überhaupt keine Konvergenzfragen auf, und Genauigkeitsüberlegungen drängen sich höchstens von der Kapazität des Rechners her auf. Algorithmen, die sich vollumfänglich im Ring der ganzen Zahlen bewegen, begegnet man vor allem in der Kombinatorik¹⁾ und in der elementaren Zahlentheorie. Die euklidische Ketten-Division im Polynomring über dem Körper der rationalen Zahlen ist ein weiteres interessantes Beispiel, das seine Wurzel in der elementaren Algebra hat. Der Taschen-Rechner ist aber auch in der Lage, den euklidischen Algorithmus im Polynomring über dem Körper der reellen Zahlen zu verkraften. Es wird sich herausstellen, dass man sehr oft auch bei Polynomen mit rationalen Koeffizienten auf diesen allgemeineren Prozess angewiesen ist.

1) Vgl. [2].

Der euklidische Algorithmus bei Polynomen führt auf ein Programm, das die Kapazität heute verfügbarer, grösserer programmierbarer Taschen-Rechner weitgehend ausschöpft. Man kann daran sehr hübsch die Leistungsfähigkeit solcher Geräte demonstrieren. Dies ist ein weiterer Aspekt der vorliegenden Note.

1. Der euklidische Algorithmus im Ring der ganzen Zahlen

Der euklidische Algorithmus stützt sich auf die Tatsache, dass zu zwei Zahlen $a, b \in \mathbb{Z}, b \neq 0$ stets eine eindeutig bestimmte, nichtnegative ganze Zahl q' existiert, so dass

$$|b|q' \leq |a| < |b|(q' + 1). \quad (1.1)$$

Aus

$$q' \leq \left\lceil \frac{a}{b} \right\rceil < q' + 1$$

entnimmt man, dass

$$q' = \left[\left\lceil \frac{a}{b} \right\rceil \right]^2$$

ist. Mit (1.1) gilt zugleich

$$0 \leq |a| - |b|q' = r' < |b|. \quad (1.2)$$

Aus der Beziehung (1.2) schliesst man nun auf die folgende endliche *Kette von Divisionen mit Rest*

$$\left. \begin{array}{lll} |a| = |b|q' & + r'_1 & \text{mit } 0 < r'_1 < b \\ |b| = r'_1 q'_1 & + r'_2 & \text{mit } 0 < r'_2 < r'_1 \\ r_1 = r'_2 q'_2 & + r'_3 & \text{mit } 0 < r'_3 < r'_2 \\ \vdots & & \\ \vdots & & \\ r'_{s-2} = r'_{s-1} q'_{s-1} + r'_s & \text{mit } 0 < r'_s < r'_{s-1} \\ r'_{s-1} = r'_s q'_s & & \end{array} \right\} \quad (1.3)$$

Die Folge der Reste ist nämlich monoton abnehmend

$$r'_1 > r'_2 > r'_3 > \cdots > r'_s > 0.$$

2) $[x]$ bezeichnet die grösste ganze Zahl kleiner oder gleich x .

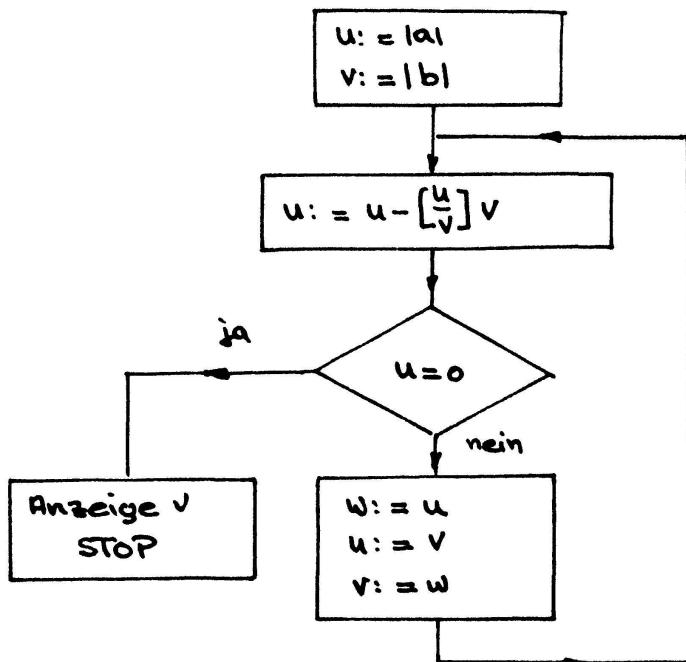
Daher muss nach endlich vielen Schritten der Rest 0 erscheinen; die Kette bricht also sicher ab. Es ist nun offenbar

$$d/a \wedge d/b \Leftrightarrow d/b \wedge d/r'_1 \Leftrightarrow d/r'_1 \wedge d/r'_2 \Leftrightarrow \dots \Leftrightarrow d/r'_{s-1} \wedge d/r'_s \Leftrightarrow d/r'_s.$$

Für den *grössten gemeinsamen positiven Teiler* von a und b entnimmt man daraus

$$t = (a, b) = (b, r'_1) = (r'_1, r'_2) = \dots = (r'_{s-1}, r'_s) = r'_s. \quad (1.4)$$

Die Berechnung von t mit einem Taschen-Rechner kann etwa anhand des Flussdiagrammes in der Figur 1 erfolgen.



Figur 1

2. Euklidische Ringe

Mit dem euklidischen Algorithmus bewegt man sich generell in einem sogenannten *euklidischen Ring*. Im Hinblick auf das anvisierte Ziel seien in diesem Abschnitt die wichtigsten Fakten über euklidische Ringe kurz zusammengestellt.

Ein kommutativer Ring $[F; +, \cdot]$ mit Einselement, der frei von Nullteilern ist, heisst bekanntlich ein *Integritätsbereich*. Die Teiler des Einselementes nennt man *Einheiten*. Zwei Elemente $a, b \in F$, die sich nur um eine Einheit e als Faktor unterscheiden, werden *assoziiert* genannt. Wegen

$$a = eb \Leftrightarrow b = e^{-1}a$$

teilen sich assoziierte Elemente gegenseitig, d. h. es gilt zugleich $a|b$ und $b|a$.

Teilbarkeitsaussagen in einem Integritätsbereich sind daher ihrem Wesen nach stets Ein Integritätsbereich $[F; +, \cdot]$ wird als *euklidischer Ring* bezeichnet, wenn sich

jedem Element $a \in F, a \neq 0$ eine nichtnegative ganze Zahl $g(a)$ zuordnen lässt, die folgende Eigenschaften aufweist:

(E₁) $g(ab) \geq g(a)$ für alle $a, b \neq 0$.

(E₂) Zu $a, b \in F, b \neq 0$ gibt es immer zwei Elemente $q, r \in F$, so dass $a = bq + r$ mit $r = 0$ oder $r \neq 0$ und $g(r) < g(b)$.

Das einfachste Beispiel ist der *Ring der ganzen Zahlen* $[Z; +, \cdot]$. Die Einheiten in $[Z; +, \cdot]$ sind die Zahlen $+1$ und -1 ; eine Klasse von assoziierten Elementen besteht daher – falls $a \neq 0$ ist – aus den beiden Zahlen a und $-a$. Ferner ist

$$a \mapsto g(a) = |a|$$

eine Abbildung mit den Eigenschaften (E₁) und (E₂). Für ganze Zahlen $a, b \neq 0$ gilt nämlich

$$|ab| = |a||b| \geq |a|,$$

und aus der Beziehung (1.2)

$$|a| = |b|q' + r' \quad \text{mit} \quad 0 \leq r' < b$$

kann stets auf das Vorhandensein einer Zerfällung

$$a = bq + r \quad \text{mit} \quad q = \pm q' \quad \text{und} \quad r = \pm r' \tag{2.1}$$

geschlossen werden, die (E₂) erfüllt.

Die Eigenschaften (E₁) und (E₂) garantieren in einem euklidischen Ring $[F; +, \cdot]$ den Prozess der Ketten-Division; zu $a, b \in F, b \neq 0$ gibt es eine abbrechende Kette von Divisionen mit Rest:

$$\left. \begin{array}{l} a = bq_0 + r_1 \quad \text{mit} \quad g(r_1) < g(b) \\ b = r_1q_1 + r_2 \quad \text{mit} \quad g(r_2) < g(r_1) \\ r_1 = r_2q_2 + r_3 \quad \text{mit} \quad g(r_3) < g(r_2) \\ \vdots \\ r_{s-2} = r_{s-1}q_{s-1} + r_s \quad \text{mit} \quad g(r_s) < g(r_{s-1}) \\ r_{s-1} = r_sq_s \end{array} \right\} \tag{2.2}$$

Dass die Konstruktion nach endlich vielen Schritten abbrechen muss, geht aus den Ungleichungen

$$g(b) > g(r_1) > g(r_2) > \cdots > g(r_{s-1}) > g(r_s) > 0$$

hervor. r_s bezeichnet den letzten von Null verschiedenen Rest.

Wie im Abschnitt 1 schliesst man auch hier, dass

$$d/a \wedge d/b \Leftrightarrow d/b \wedge d/r_1 \Leftrightarrow \dots \Leftrightarrow d/r_{s-1} \wedge d/r_s \Leftrightarrow d/r_s.$$

Zu jedem Teiler d von r_s gehört ein Element d' , so dass $r_s = dd'$ ist. Aufgrund von (E₁) ist dann

$$g(r_s) = g(dd') \geq g(d).$$

Das Element r_s repräsentiert also die *Klasse assoziierter gemeinsamer Teiler von a und b mit dem grösstmöglichen Wert der Funktion g*. Diese Klasse wird der *grösste gemeinsame Teiler von a und b* genannt.

Ein Element $a \neq 0$ aus dem euklidischen Ring $[F; +, \cdot]$ hat stets die vorhandenen Einheiten und die zu a assoziierten Elemente zu Teilern. Man nennt sie die *trivialen Teiler von a*.

Ein Element p , das nicht Einheit ist und nur triviale Teiler aufweist, heisst ein *Primelement* des betreffenden euklidischen Ringes.

In einem euklidischen Ring gilt der Satz von der eindeutigen Zerfällung in Primfaktoren:

Jedes von 0 verschiedene Element a, das nicht Einheit ist, besitzt eine Darstellung als Produkt von endlich vielen Primelementen. Diese Zerfällung ist eindeutig bis auf die Reihenfolge der Faktoren und ihre Ersetzung durch assoziierte Elemente.

Da dieser Satz für die folgenden Überlegungen nur von untergeordneter Bedeutung ist, verzichten wir auf eine Wiedergabe des Beweises. Der Leser sei auf die einschlägige Algebraliteratur verwiesen³⁾.

Es sei noch erwähnt, dass die in einem euklidischen Ring geforderte Zerfällung

$$a = bq + r \quad \text{mit} \quad r = 0 \quad \text{oder} \quad g(r) < g(b)$$

im allgemeinen nicht eindeutig ist. So bestehen etwa im Ring $[Z; +, \cdot]$ zu $a = 17$ und $b = 5$ die Zerlegungen

$$17 = 5 \cdot 3 + 2 \quad \text{und} \quad 17 = 5 \cdot 4 - 3,$$

die beide (E₂) genügen.

3. Der Polynomring über einem Körper

Wir betrachten jetzt Polynome

$$a(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

mit Koeffizienten aus einem vorgegebenen Körper $[K; +, \cdot]$. Der höchst vorkommende Exponent n heisst der *Grad*, und $a_n x^n$ ist das sogenannte *Leitglied des*

3) Vgl. etwa [3], S. 120.

Polynoms. Bezeichnet $K[x]$ die Menge aller Polynome mit Koeffizienten aus K , dann ist das Verknüpfungsgebiilde $[K[x]; +, \cdot]$ stets ein euklidischer Ring. Er wird der *Polynomring über dem Körper K* genannt.

Nullelement dieses Ringes ist das Polynom $v(x) = 0$, Einselement das Polynom $\varepsilon(x) = 1$. Mit

$$g(a) = \text{grad}(a)$$

lässt sich jedem Polynom $a(x) \neq v(x)$ eine nichtnegative ganze Zahl zuordnen, welche die Eigenschaften (E₁) und (E₂) hat. Es gilt nämlich die Gradregel

$$g(a \cdot \beta) = g(a) + g(\beta) \geq g(a). \quad (3.1)$$

Ferner gibt es zu $a(x), \beta(x) \in K[x]$ und $\beta(x) \neq v(x)$ stets zwei Polynome $\kappa(x)$ und $\rho(x)$, so dass

$$a(x) = \beta(x) \cdot \kappa(x) + \rho(x) \quad \text{mit} \quad \rho(x) = v(x) \quad \text{oder} \quad g(\rho) < g(\beta). \quad (3.2)$$

Man kann zwei solche Polynome mit dem üblichen *Divisionsalgorithmus* erhalten, der sich bekanntlich in $K[x]$ immer bis zu einem Restpolynom fortsetzen lässt, dessen Grad kleiner als $g(\beta)$ ist. So bekommt man etwa in $[Q[x]; +, \cdot]$ für

$$a(x) = 5x^3 + 2x^2 - 3x + 4 \quad \text{und} \quad \beta(x) = 3x^2 + 2x - 1$$

$(5x^3 + 2x^2 - 3x + 2) : (3x^2 - 2x + 1)$ $\begin{array}{r} 5x^3 - \frac{10}{3}x^2 + \frac{5}{3}x \\ \hline \frac{16}{3}x^2 - \frac{14}{3}x + 2 \\ \hline \frac{16}{3}x^2 - \frac{22}{9}x + \frac{16}{9} \\ \hline -\frac{10}{9}x + \frac{2}{9} \end{array}$	$\begin{array}{c cc} \kappa(x) & \frac{5}{3}x + \frac{16}{9} \\ \hline \rho(x) & -\frac{10}{9}x + \frac{2}{9} \end{array}$
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------

Es ist somit

$$\underbrace{5x^3 + 2x^2 - 3x + 2}_{a(x)} = \underbrace{(3x^2 - 2x + 1)}_{\beta(x)} \underbrace{\left(\frac{5}{3}x + \frac{16}{9} \right)}_{\kappa(x)} + \underbrace{\left(-\frac{10}{9}x + \frac{2}{9} \right)}_{\rho(x)}.$$

Die Einheiten im euklidischen Ring $[K[x]; +, \cdot]$ sind die Teiler des Einselementes. Ist das Polynom $a(x)$ eine Einheit, dann gibt es dazu ein Polynom $\delta(x)$, so dass $a(x) \cdot \delta(x) = 1$ ist. Daraus folgt

$$g(a \cdot \delta) = g(a) + g(\delta) = 0$$

und dies impliziert

$$g(a) = g(\delta) = 0.$$

Dadurch sind die von Null verschiedenen Elemente des Grundkörpers K gekennzeichnet. Eine Klasse von assoziierten Elementen in $K[x]$ besteht daher aus einem Polynom $a(x)$ und allen seinen Vielfachen $f \cdot a(x)$ mit $f \in K \setminus \{0\}$.

Die Primelemente von $K[x]$ werden in der Algebra auch *irreduzible Polynome* genannt.

4. Der euklidische Algorithmus im Polynomring über dem Körper der rationalen Zahlen

Im Polynomring über dem Körper $[Q; +, \cdot]$ gibt es in jeder Klasse von assoziierten Elementen auch solche mit ganzzahligen Koeffizienten. Darunter befinden sich zwei besonders ausgezeichnete, nämlich diejenigen mit teilerfremden ganzzahligen Koeffizienten. Man nennt sie *primitive Polynome*.

So heissen etwa die zu

$$a(x) = \frac{3}{4}x^3 + \frac{1}{7}x^2 - \frac{5}{2}x + \frac{2}{5}$$

assoziierten primitiven Polynome

$$a_1(x) = 105x^3 + 20x^2 - 350x + 56 \quad \text{und} \quad a_2(x) = -105x^3 - 20x^2 + 350x - 56.$$

In der Ketten-Division kann man sich nun bei allen vorkommenden Polynomen auf einen der beiden ausgezeichneten Repräsentanten festlegen. Der Algorithmus läuft dann, was die Koeffizienten anbetrifft, vollständig im Ring $[Z; +, \cdot]$ ab. Zugleich erreicht man mit primitiven Polynomen gerade auch noch eine optimale Ausnutzung des Rechners, weil dann der Prozess gewissermassen auf der untersten Stufe der Ganzahligkeit abläuft.

Anschliessend soll ein Programm beschrieben werden, das mit einem Taschenrechner realisiert werden kann.

Als Kernproblem bei der Aufstellung eines solchen Programmes stellt sich zunächst die folgende Aufgabe. Der Divisionsalgorithmus liefert für zwei primitive Polynome $\alpha(x)$ und $\beta(x)$ eine Zerfällung

$$\alpha(x) = \beta(x) \kappa(x) + \rho(x) \quad \text{mit} \quad \rho(x) = 0 \quad \text{oder} \quad g(\rho) < g(\beta). \quad (4.1)$$

$\kappa(x)$ und $\rho(x)$ sind im allgemeinen nicht ganzzahlig. Es ist nun $\alpha(x)$ durch ein geeignetes assoziiertes ganzzahliges Polynom $\tilde{\alpha}(x)$ zu ersetzen, so dass aus dem Divisionsalgorithmus eine ganzzahlige Zerfällung

$$\bar{a}(x) = \beta(x) \alpha^*(x) + \rho^*(x) \quad \text{mit} \quad \rho^*(x) = 0 \quad \text{oder} \quad g(\rho^*) < g(\beta) \quad (4.2)$$

hervorgeht.

Ist nun

$$\begin{aligned} a(x) &= a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0, \\ \beta(x) &= b_n x^n + \cdots + b_1 x + b_0, \end{aligned}$$

dann ist

$$\bar{a}(x) = h \cdot a(x) \quad \text{mit} \quad h = |b_n|^{m-n+1} \quad (4.3)$$

immer ein Polynom mit der angestrebten Eigenschaft. Man kann dies mit der folgenden Überlegung verifizieren.

$\alpha(x)$ und das assoziierte Polynom $\alpha^*(x)$ haben den Grad $m-n$ und weisen daher $m-n+1$ Glieder auf. Der Divisionsalgorithmus besteht somit aus $m-n+1$ Schritten. Der erste Schritt ist nun offensichtlich ganzzahlig, denn die Leitglieder der Polynome $\bar{a}(x)$ und $\beta(x)$ lauten

$$|b_n|^{m-n+1} a_m x^m \quad \text{und} \quad b_n x^n.$$

Aber auch der zweite Schritt verläuft ganzzahlig; die Leitglieder der massgebenden Polynome sind nämlich

$$|b_n|^{m-n} (|b_n| a_{m-1} - a_m \operatorname{sign}(b_n)) x^{m-1} \quad \text{und} \quad b_n x^n.$$

Mit jedem weiteren Schritt nimmt nun der Exponent von $|b_n|$ im Leitglied des jeweiligen Restpolynoms um 1 ab. Man schliesst daraus, dass der Divisionsalgorithmus tatsächlich bis zum letzten Schritt ganzzahlig bleibt.

Wir wollen dies noch an unserem früheren Beispiel aufzeigen. Für

$$a(x) = 5x^3 + 2x^2 - 3x + 2 \quad \text{und} \quad \beta(x) = 3x^2 - 2x + 1$$

ist $h = 3^2 = 9$ und somit

$$\bar{a}(x) = 45x^3 + 18x^2 - 27x + 18.$$

Der Divisionsalgorithmus führt dann auf

$$\begin{array}{r} (45x^3 + 18x^2 - 27x + 18):(3x^2 - 2x + 1) \\ \underline{45x^3 - 30x^2 + 15x} \\ 48x^2 - 42x + 18 \\ \underline{48x^2 - 32x + 16} \\ -10x + 2 \end{array} \quad \begin{array}{c|c} \alpha^*(x) & 15x + 16 \\ \hline \rho^*(x) & -10x + 2 \end{array}$$

Wie das vorliegende Beispiel zeigt, ist $\rho^*(x)$ nicht notwendigerweise primitiv. Man erhält ein assoziiertes primitives Polynom zu $\rho^*(x)$, indem man die Koeffizienten durch ihren grössten gemeinsamen positiven oder negativen Teiler dividiert.

Ist $\varphi(x)$ ein ganzzahliges Polynom, dann bezeichnen wir fortan mit $\underline{\varphi}(x)$ das zugehörige primitive Polynom, das sich von $\varphi(x)$ um einen positiven Faktor unterscheidet.

Im Hinblick auf eine möglichst vielseitige Verwendbarkeit legen wir nun den euklidischen Algorithmus zu zwei ganzzahligen Polynomen $a(x)$ und $\beta(x)$ wie folgt fest:

$$\begin{aligned}\varphi_0(x) &:= a(x) \\ \varphi_1(x) &:= \beta(x) \\ \bar{\varphi}_{k-1}(x) = \varphi_k(x) \sigma_k(x) - \varphi_{k+1}^*(x) &\quad \text{mit } \varphi_{k+1}^*(x) = 0 \quad \text{oder} \quad g(\varphi_{k+1}^*) < g(\varphi_k) \\ \varphi_{k+1}(x) &:= \underline{\varphi}_{k+1}^*(x)\end{aligned}\tag{4.4}$$

$\bar{\varphi}_{k-1}(x)$ ist das Polynom, das vermöge der zuvor beschriebenen Konstruktion aus $\varphi_{k-1}(x)$ und $\varphi_k(x)$ hervorgeht und die ganzzahlige Division mit Rest garantiert.

Der Algorithmus liefert eine endliche Kette von Polynomen

$$\varphi_0(x), \quad \varphi_1(x), \quad \varphi_2(x), \quad \dots, \quad \varphi_s(x).$$

Das letzte Glied ist ein grösster gemeinsamer primitiver Teiler von $a(x)$ und $\beta(x)$. Sind diese beiden Polynome teilerfremd, dann endet die Kette mit $\varphi_s(x) = \pm 1$.

Das negative Vorzeichen von $\varphi_{k+1}^*(x)$ in der Rekursionsformel von (4.4) dient dazu, dass der Algorithmus auch für die Sturmsche Nullstellenzählung bei einem Polynom herangezogen werden kann (siehe Abschnitt 6).

5. Ein Programm für den Rechner Texas TI 59

Im Hinblick auf die beschränkte Kapazität eines Taschen-Rechners (Stellenzahl, Anzahl der Speicher) ist das folgende Programm für ganzzahlige Polynome $a(x)$ und $\beta(x)$ mit Graden $m, n \leq 9$ ausgelegt. Das Programm ist speziell auf den Rechner Texas TI 59 zugeschnitten, der über die notwendige Anzahl von Programmschritten verfügt und zudem eine Speicherung des relativ langen Programmes auf Magnetkarten gestattet. Zum TI 59 ist ausserdem ein Drucker entwickelt worden; es lag daher nahe, den sukzessiven Ausdruck der Ergebnisse ins Programm einzubeziehen. Wer nur über den Rechner verfügt, kann das Programm leicht so modifizieren, dass der Prozess nach jedem Glied in der durch $a(x)$ und $\beta(x)$ definierten Polynom-Kette anhält.

In den Polynomen $a(x)$ und $\beta(x)$ stecken aufgrund der verabredeten Beschränkung höchstens je 10 Koeffizienten, nämlich a_0, a_1, \dots, a_9 und b_0, b_1, \dots, b_9 . Diese werden in den Speichern R_{10} bis R_{19} und R_{20} bis R_{29} untergebracht. Die Speicher R_{00} bis R_{09} sind beim Rechner TI 59 mit einer programmverkürzenden Speicherarithmetik ausgerüstet; sie sind aus diesem Grunde für die benötigten Rechen-Parameter reserviert.

Zur Vereinfachung der Programmbeschreibung werden die Speicher R_{10} bis R_{19} fortan als Polynom-Register I, die Speicher R_{20} bis R_{29} als Polynom-Register II bezeichnet.

Die folgende Zusammenstellung zeigt die verwendete Speicherzuweisung.

R_{00}	R_{01}	R_{02}	R_{03}	R_{04}	R_{05}	R_{06}	R_{07}	R_{08}	R_{09}
m	n	p	$d = m - n$	r	s	k	h, q	u	v
(t)	(t)	(j)	(a, b)						

R_{10}	R_{11}	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}	R_{17}	R_{18}	R_{19}
a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9

R_{20}	R_{21}	R_{22}	R_{23}	R_{24}	R_{25}	R_{26}	R_{27}	R_{28}	R_{29}
b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9

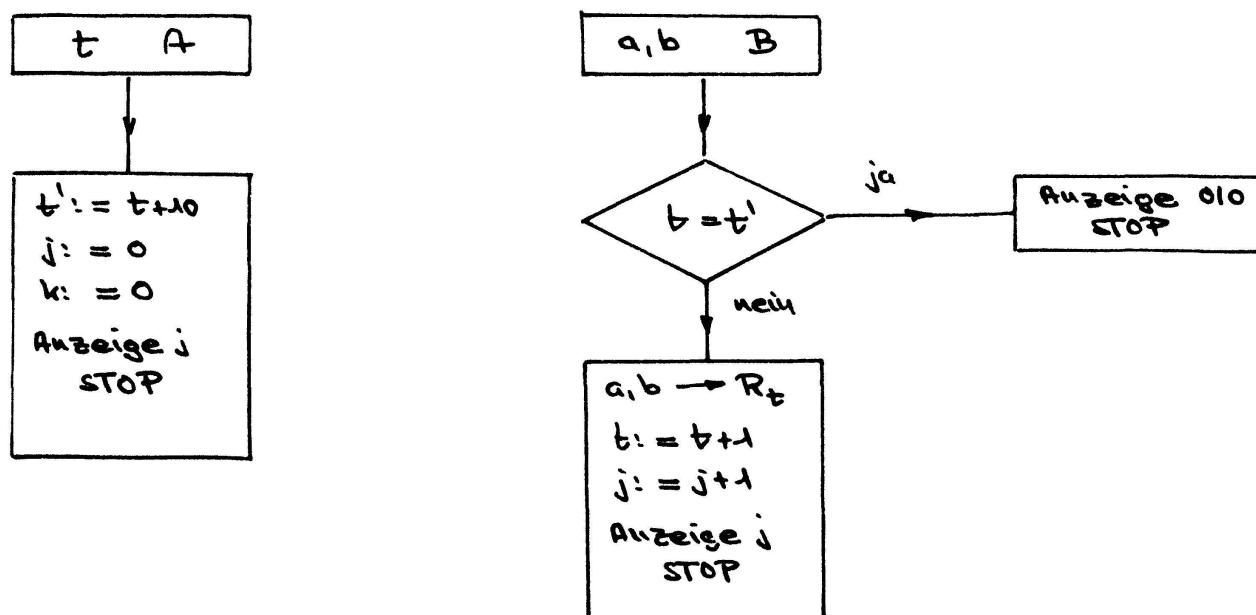
Die eingeführten Parameter haben folgende Bedeutung:

r ist der Index im Polynom-Register I: $a(r) = a_{r-10}$; gespeichert in R_r ,
 s ist der Index im Polynom-Register II: $b(s) = b_{s-20}$; gespeichert in R_s ,
 k ist der Index in der Kette der φ -Polynome gemäss (4.4).

u und v werden bei der Berechnung des grössten gemeinsamen Teilers für die jeweiligen Koeffizienten im Polynom-Register I benötigt.

Die beiden ersten Programmabschnitte, beginnend mit Lbl A und Lbl B dienen zur Eingabe der Polynome $a(x)$ und $\beta(x)$ in die Polynom-Register I und II. Dazu sind die eingeklammerten Parameter erforderlich.

Das Eingabekonzept geht aus dem folgenden Flussdiagramm hervor.



Figur 2

Das zugehörige Rechnerprogramm lautet:

000	76	Lbl	016	76	Lbl	032	69	Op
001	11	A	017	12	B	033	20	20
002	42	STO	018	42	STO	034	69	Op
003	00	00	019	03	03	035	22	22
004	85	+	020	43	RCL	036	43	RCL
005	01	10	021	01	01	037	02	02
006	11		022	32	$x \geq t$	038	91	R/S
007	95	=	023	43	RCL	039	00	0
008	42	STO	024	00	00	040	85	+
009	01	01	025	67	$x = t$	041	65	*
010	00	0	026	00	039	042	91	R/S
011	42	STO	027	39				
012	02	02	028	43	RCL			
013	42	STO	029	03	03			
014	06	06	030	72	STO Ind			
015	91	R/S	031	00	00			

Dementsprechend läuft die Eingabe von $\alpha(x)$ und $\beta(x)$ wie folgt ab:

Eingabe	Taste	Anzeige	Eingabe	Taste	Anzeige
10	A	0	20	A	0
a_0	B	1	b_0	B	1
a_1	B	2	b_1	B	2
.		.	.		.
.		.	.		.
a_9	B	10	b_9	B	10

Falls der Grad 9 erreicht ist, erscheint bei nochmaligem Drücken der Taste B in der Anzeige eine blinkende 0.

Das Hauptprogramm liefert zu $\alpha(x)$ und $\beta(x)$ die Kette der primitiven Restpolynome

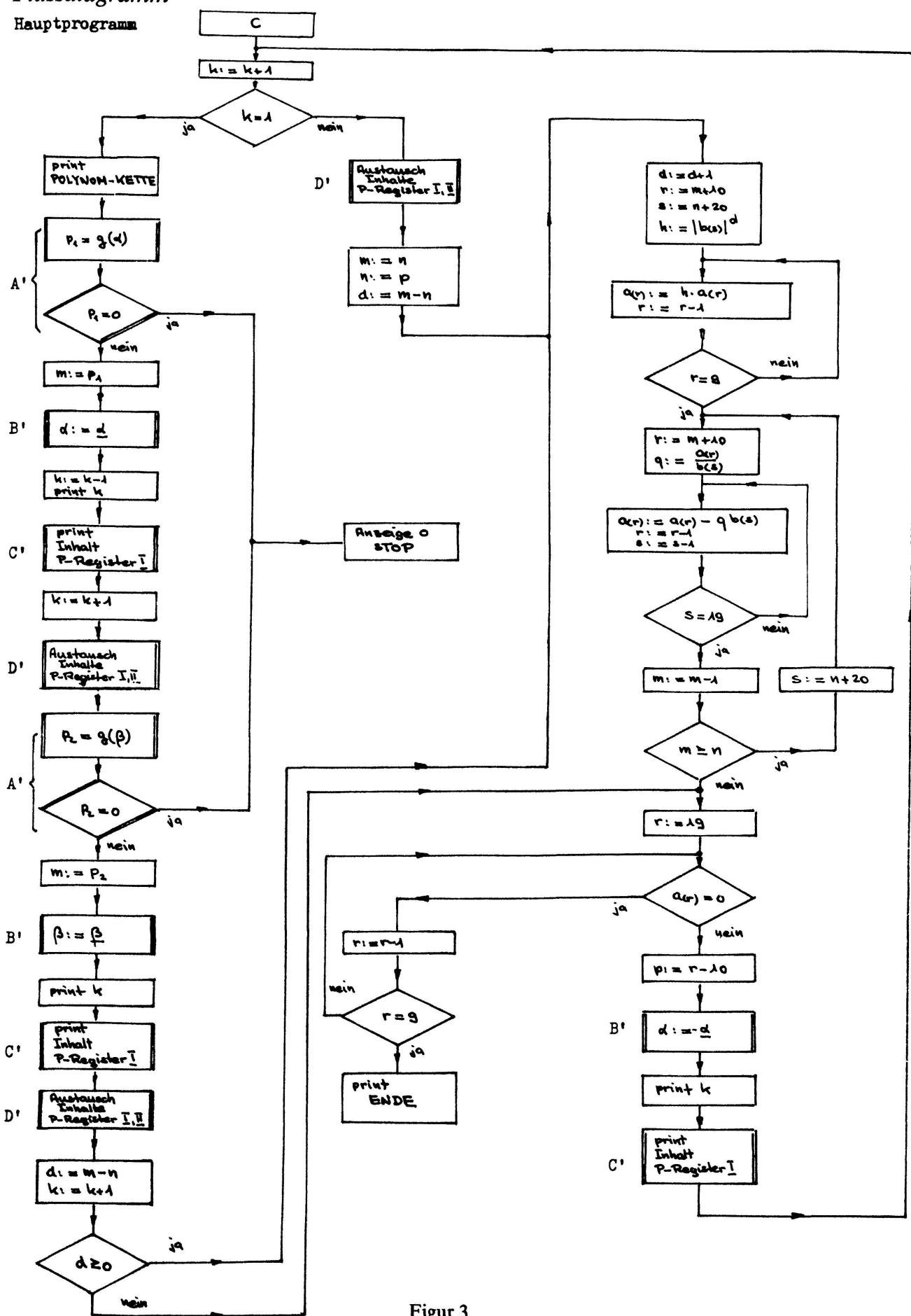
$$\varphi_0(x), \quad \varphi_1(x), \quad \dots, \quad \varphi_s(x)$$

mit $\varphi_0(x) = \alpha(x)$ und $\varphi_1(x) = \beta(x)$. Es wird abgerufen über die Taste C.

Für ein Polynom $\varphi_k(x)$ der Kette wird vorerst die Nummer k ausgedruckt; anschliessend werden dann die Koeffizienten aufgelistet, und zwar nach fallendem Index geordnet. Das Prozessende wird mit dem Wort ENDE angezeigt.

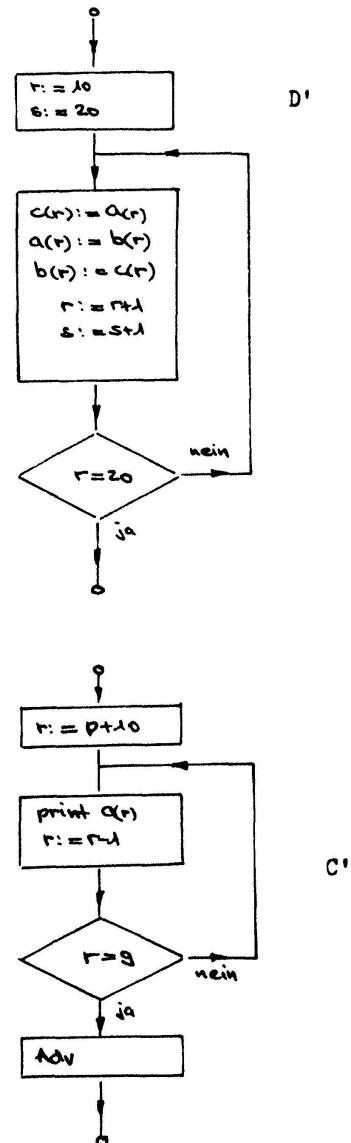
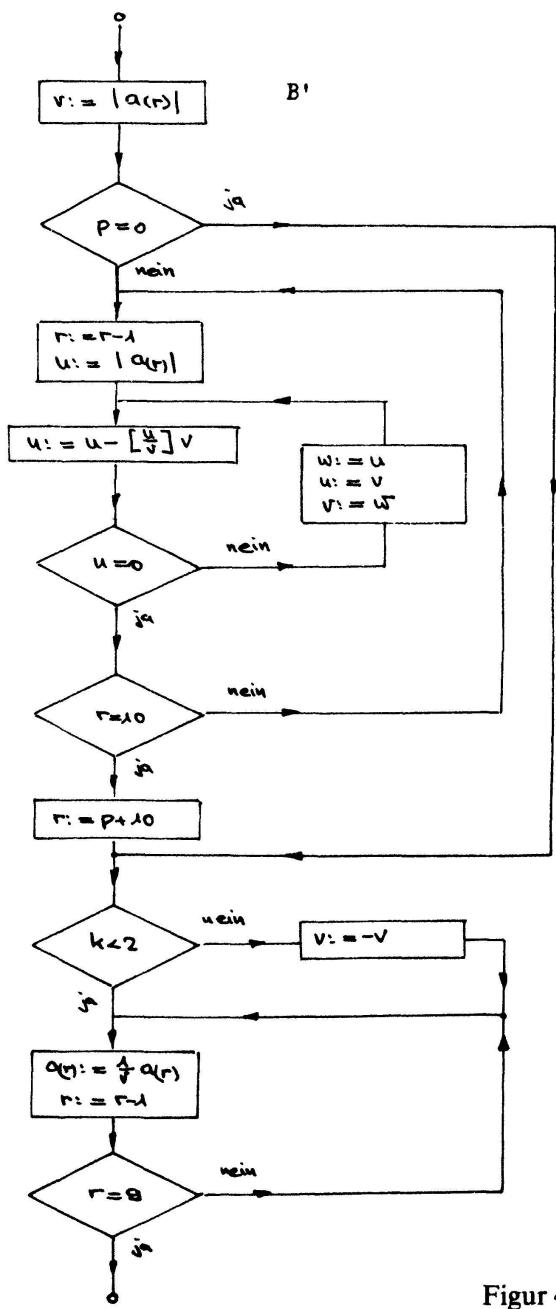
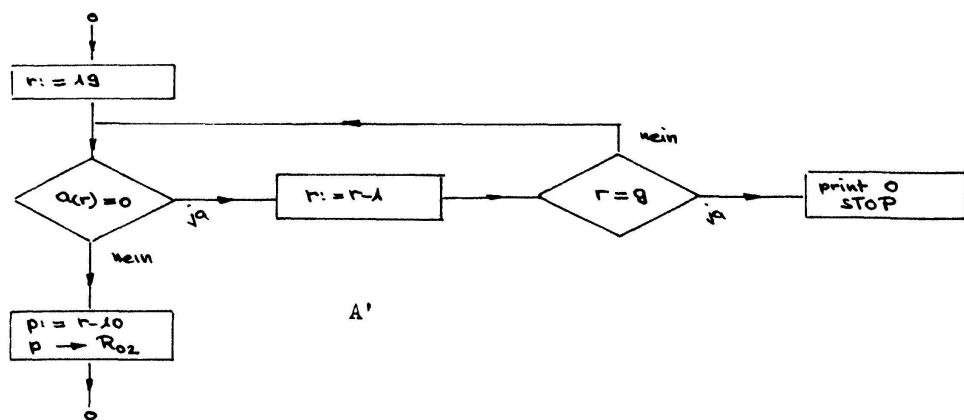
Flussdiagramm

Hauptprogramm



Figur 3

Unterprogramme



Figur 4

Rechnerprogramm

Speicherverteilung 559/49 (5 Op 17)

Hauptprogramm

043	76	Lb1	095	71	SBR	140	02	248	192	95	=
044	13	C	096	16	A'	141	48		193	42	STO
045	69	Op	097	42	STO	→ 142	69	Op	194	04	04
046	26	26	098	00	00	143	23	23	195	73	RCL Ind
047	01	1				144	43	RCL	196	04	04
048	32	$x \geq t$	099	71	SBR	145	00	00	197	55	:
049	43	RCL	100	17	B'	146	85	+	198	73	RCL Ind
050	06	06	101	69	Op	147	01	10	199	05	05
051	22	INV	102	36	36	148	00		200	95	=
052	67	$x = t$	103	43	RCL	149	95	=	201	42	STO
053	03	302	104	06	06	150	42	STO	202	07	07
054	02		105	99	Prt	151	04	04	203	66	PAUSE
055	69	Op	106	98	Adv	152	43	RCL	→ 204	73	RCL Ind
056	00	00				153	01	01	205	05	05
057	03	3	107	71	SBR	154	85	+	206	65	*
058	03	3	108	18	C'	155	02	20	207	43	RCL
059	03	3	109	69	Op	156	00		208	07	07
060	02	2	110	26	26	157	95	=	209	95	=
061	02	2				158	42	STO	210	22	INV
062	07	7	111	71	SBR	159	05	05	211	74	SUM Ind
063	04	4	112	19	D'	160	73	RCL Ind	212	04	04
064	05	5				161	05	05	213	69	Op
065	03	3	113	71	SBR	162	50	x	214	34	34
066	01	1	114	16	A'	163	45	y^x	215	69	Op
067	69	Op	115	42	STO	164	43	RCL	216	35	35
068	01	01	116	01	01	165	03	03	217	01	19
069	03	3				166	85	+	218	09	
070	02	2	117	71	SBR	167	93	0.1	219	32	$x \geq t$
071	03	3	118	17	B'	168	01		220	43	RCL
072	00	0	119	43	RCL	169	95	=	221	05	05
073	02	2	120	06	06	170	59	Int	222	22	INV
074	00	0	121	99	Prt	171	42	STO	223	67	$x = t$
075	02	2	122	98	Adv	172	07	07	224	02	204
076	06	6				→ 173	43	RCL	225	04	
077	01	1	123	71	SBR	174	07	07	226	69	Op
078	07	7	124	18	C'	175	64	Prd Ind	227	30	30
079	69	Op				176	04	04	228	43	RCL
080	02	02	125	71	SBR	177	69	Op	229	01	01
081	03	3	126	19	D'	178	34	34	230	32	$x \geq t$
082	07	7	127	29	CP	179	09	9	231	43	RCL
083	03	3	128	69	Op	180	32	$x \geq t$	232	00	00
084	07	7	129	26	26	181	43	RCL	233	22	INV
085	01	1	130	43	RCL	182	04	04	234	77	$x \geq t$
086	07	7	131	00	00	183	22	INV	235	02	248
087	06	6	132	75	-	184	67	$x = t$	236	48	
088	02	2	133	43	RCL	185	01	173	237	43	RCL
089	00	0	134	01	01	186	73		238	01	01
090	00	0	135	95	=	→ 187	43	RCL	239	85	+
091	69	Op	136	42	STO	188	00	00	240	02	20
092	03	03	137	03	03	189	85	+	241	00	
093	69	Op	138	22	INV	190	01	10	242	95	=
094	05	05	139	77	$x \geq t$	191	00		243	42	STO

244	05	05	264	04	04	→ 284	43	RCL	302	71	SBR
245	61	GTO	265	22	INV	285	04	04	303	19	D'
246	01		266	67	x=t	286	75	-	304	43	RCL
247	87	{ 187	267	02		287	01	{ 10	305	01	01
→ 248	01		268	52	{ 252	288	00	{ 10	306	42	STO
249	09	{ 19	269	69	Op	289	95	=	307	00	00
250	42	STO	270	00	00	290	42	STO	308	43	RCL
251	04	04	271	01	1	291	02	02	309	02	02
→ 252	29	CP	272	07	7				310	42	STO
253	73	RCL Ind	273	03	3	292	71	SBR	311	01	01
254	04	04	274	01	1	293	17	B'	312	43	RCL
255	22	INV	275	01	1	294	43	RCL	313	00	00
256	67	x=t	276	06	6	295	06	06	314	75	-
257	02	{ 284	277	01	1	296	99	Prt	315	43	RCL
258	84	{ 284	278	07	7	297	98	Adv	316	01	01
259	69	Op	279	69	Op				317	95	=
260	34	34	280	03	03	298	71	SBR	318	42	STO
261	09	9	281	69	Op	299	18	C'	319	03	03
262	32	x ≥ t	282	05	05	300	61	GTO	320	61	GTO
263	43	RCL	283	91	R/S	301	13	C	321	01	
									322	42	{ 142

Unterprogramme

323	76	Lb1	356	42	STO	389	44	SUM	422	32	x ≥ t
324	16	A'	357	02	02	390	08	08	423	43	RCL
325	01	{ 19	358	92	INV SBR	391	29	CP	424	06	06
326	09	{ 19	359	76	Lb1	392	43	RCL	425	22	INV
327	42	STO	360	17	B'	393	08	08	426	77	x ≥ t
328	04	04	361	73	RCL Ind	394	67	x=t	427	04	{ 434
→ 329	29	CP	362	04	04	395	04	{ 404	428	34	{ 434
330	73	RCL Ind	363	50	x	396	04	{ 404	429	43	RCL
331	04	04	364	42	STO	397	48	Exc	430	09	09
332	22	INV	365	09	09	398	09	09	431	94	+/-
333	67	x=t	366	29	CP	399	42	STO	432	42	STO
334	03	{ 350	367	43	RCL	400	08	08	433	09	09
335	50	{ 350	368	02	02	401	61	GTO	→ 434	43	RCL
336	69	Op	369	67	x=t	402	03	{ 379	435	09	09
337	34	34	370	04		403	79		436	22	INV
338	09	9	371	21	{ 421	→ 404	01	{ 10	437	64	Prd Ind
339	35	x ≥ t	→ 372	69	Op	405	00	{ 10	438	04	04
340	43	RCL	373	34	34	406	32	x ≥ t	439	69	Op
341	04	04	374	73	RCl Ind	407	43	RCL	440	34	34
342	22	INV	375	04	04	408	04	04	441	09	9
343	67	x=t	376	50	x	409	22	INV	442	32	x ≥ t
344	03	{ 329	377	42	STO	410	67	x=t	443	43	RCL
345	29	{ 329	378	08	08	411	03	{ 372	444	04	04
346	00	0	→ 379	55	:	412	72	{ 372	445	22	INV
347	99	Prt	380	43	RCL	413	43	RCL	446	67	x=t
348	91	R/S	381	09	09	414	02	02	447	04	{ 434
349	68	Nop	382	95	=	415	85	+	448	34	{ 434
→ 350	43	RCL	383	59	Int	416	01	{ 10	449	92	INV SBR
351	04	04	384	65	*	417	00	{ 10	450	68	Nop
352	75	-	385	43	RCL	418	95	=			
353	01	{ 10	386	09	09	419	42	STO			
354	00	{ 10	387	95	=	420	04	04			
355	95	=	388	22	INV	→ 421	02	2			

451	76	Lb1	466	09	9	478	76	Lb1	494	69	Op
452	18	C'	467	32	$x \geq t$	479	19	D'	495	24	24
453	43	RCL	468	43	RCL	480	01	{ 10	496	69	Op
454	02	02	469	04	04	481	00	} 10	497	25	25
455	85	+	470	22	INV	482	42	STO	498	02	{ 20
456	01	{ 10	471	67	$x = t$	483	04	04	499	00	} 20
457	00	} 10	472	04	{ 461	484	02	{ 20	500	32	$x \geq t$
458	95	=	473	61	} 461	485	00	} 20	501	43	RCL
459	42	STO	474	98	Adv	486	42	STO	502	04	04
460	04	04	475	98	Adv	487	05	05	503	22	INV
→461	73	RCL Ind	476	92	INV SBR	→488	73	RCL Ind	504	67	$x = t$
462	04	04	477	68	Nop	489	04	04	505	04	{ 488
463	99	Prt				490	63	Exc Ind	506	88	} 488
464	69	Op				491	05	05	507	92	INV SBR
465	34	34				492	72	STO Ind	508	68	Nop
						493	04	04	509	68	Nop

Bemerkungen:

1. Das Programm schliesst auch den Fall $g(\alpha) < g(\beta)$ ein. Die Polynom-Kette beginnt dann mit

$$\varphi_0(x) = \beta(x); \quad \varphi_1(x) = \alpha(x); \quad \varphi_2(x) = -\beta(x).$$

2. Im Unterprogramm B' wird der positive ggT der Koeffizienten $a_p, a_{p-1}, \dots, a_1, a_0$ rekursiv bestimmt gemäss

$$(a_p, a_{p-1}, \dots, a_1, a_0) = ((\dots((a_p, a_{p-1}), a_{p-2}), \dots, a_1), a_0).$$

Sobald ein Zwischenergebnis $v=1$ vorliegt, steht $(a_p, a_{p-1}, \dots, a_1, a_0) = 1$ fest, und man könnte die Berechnung des ggT an dieser Stelle abbrechen. Mit einem entsprechenden bedingten Sprung liesse sich also hier die Rechenzeit noch etwas verkürzen. Um den Aufbau des Programmes möglichst einsichtig zu belassen, wurde von einer solchen Ergänzung bewusst Abstand genommen.

3. Der vorliegende Algorithmus setzt voraus, dass die auftretenden Zwischenwerte exakt ganzzahlig sind. Die Schritte 166 bis 170 garantieren die Ganzzahligkeit nach Anwendung der Operation y^x , die im allgemeinen nur gerundete Werte liefert⁴⁾.
4. Der Programmschritt 203 bewirkt, dass bei der Division mit Rest jeweils die Koeffizienten von $\sigma_k(x)$ kurz aufleuchten, und zwar nach abnehmendem Index geordnet. Er wurde im Hinblick auf die anschliessend diskutierten Anwendungen aufgenommen.

Beispiel 1

$$\begin{aligned} a(x) &= \varphi_0(x) = 3x^7 + x^6 - 8x^5 - 8x^4 + 20x^3 + 13x^2 - 35x + 14, \\ \beta(x) &= \varphi_1(x) = 6x^5 + 2x^4 - 19x^3 + 40x^2 - 47x + 18. \end{aligned}$$

Der Ausdruck des Rechners ist in der linken Spalte der Figur 5 wiedergegeben. Man

4) So erhält man mit dem Rechner TI 59 etwa bei der Berechnung von $[2^4]$ unter Verwendung der Operation y^x den falschen Wert 15.

entnimmt daraus als grössten gemeinsamen Teiler das primitive Polynom $-3x^2 + 5x - 2$. Die Rechenzeit beträgt rund 300 Sekunden.

POLYNOM-KETTE:

0.	
3.	
1.	
-8.	
-8.	
20.	
13.	
-35.	
14.	
	1.
	15.
	20.
	-9.
	2.
	7.
	-19.
	40.
	-47.
	18.
	2.
	19.
	-9.
	-41.
	41.
	3.
	-4357.
	-3384.
	6196.
	4.
	-47023.
	-75889.
	4.
	-3.
	5.
	-2.

ENDE

Beispiel 1

POLYNOM-KETTE:

0.	
3.	
5.	
-3.	
1.	
7.	
-5.	
	1.
	15.
	20.
	-9.
	2.
	7.
	2.
	19.
	-9.
	-41.
	41.
	3.
	-4357.
	-3384.
	6196.
	4.
	-47023.
	-75889.

Beispiel 2

Figur 5

Der Algorithmus läuft nicht in jedem Falle völlig problemlos ab. Es ist durchaus möglich, dass der obere *Plafond der Ganzzahligkeit* erreicht wird. Man erkennt dies

daran, dass der Rechner auf Exponentialanzeige umstellt. Dies trifft beim folgenden Beispiel 2 zu.

Beispiel 2

$$\begin{aligned} a(x) &= \varphi_0(x) = 3x^5 + 5x^4 - 3x^3 + x^2 + 7x - 5, \\ \beta(x) &= \varphi_1(x) = 15x^4 + 20x^3 - 9x^2 + 2x + 7. \end{aligned}$$

Der Ausdruck des Rechners ist – soweit er ganzzahlig ist – in der rechten Spalte von Figur 5 festgehalten. Bei $\varphi_4(x)$ angelangt, hat der Rechner als nächstes mit dem Faktor $h = 47023^2 = 2213043849$ das zu $\varphi_3(x)$ assoziierte Polynom zu bestimmen. Da schon h eine 10stellige Zahl ist, gerät man mit diesem Schritt über den Bereich der Ganzzahligkeit hinaus. (Fortsetzung im nächsten Heft.)

M. Jeger, Mathematisches Seminar, ETH Zürich

LITERATURVERZEICHNIS

- 1 G. Birkhoff und S. Mac Lane: A Survey of modern Algebra, 4. Aufl. New York, London 1977.
- 2 M. Jeger: Algorithmische Kombinatorik auf der Stufe programmierbarer Taschen-Rechner. ZAMP, Heft 2, S. 243–260 (1979).
- 3 R. Kochendörffer: Einführung in die Algebra, 4. Aufl. Berlin 1974.
- 4 W. Krull: Elementare und klassische Algebra vom modernen Standpunkt, Band I. Sammlung Göschen, Bd. 930, 3. Aufl. Berlin 1963.
- 5 N. Obreschkoff: Verteilung und Berechnung der Nullstellen reeller Polynome. Berlin 1963.

Kleine Mitteilungen

Die Cantorsche Abbildung ist ein Borel-Isomorphismus

Die Cantorsche Abbildung benutzt man im allgemeinen, um zu zeigen, dass $J = [0, 1]$ und $J \times J$ dieselbe Mächtigkeit haben. Sie ist folgendermassen definiert (vgl. [2], S. 66): Jedes $x \in J$ hat eine eindeutige Darstellung $x = \sum_{n=1}^{\infty} 2^{-a_1 - \dots - a_n}$, wobei $\{a_n\}$ eine Folge von natürlichen Zahlen ist. Dies gilt, da sich x in eindeutiger Weise in einen dyadischen Bruch mit unendlich vielen Einsen entwickeln lässt. Die Darstellung liefert für jedes $n \in \mathbb{N}$ [\mathbb{N} bezeichnet die natürlichen Zahlen (ohne Null)] eine Abbildung X_n von J nach \mathbb{N} , die durch $X_n(x) = a_n$ gegeben ist. Für alle $x, y \in J$ sei $f(x, y) \in J$ definiert vermöge $X_{2n-1}(f(x, y)) = X_n(x)$ und $X_{2n}(f(x, y)) = X_n(y)$, $n \in \mathbb{N}$. f ist eine bijektive Abbildung von $J \times J$ auf J . Hausdorff ([2], S. 377) nennt sie Cantorsche Abbildung. Er sagt, sie röhre im Prinzip von Cantor her (vgl. [1]).

Es ist wohlbekannt, dass es keine stetige bijektive Abbildung von $J \times J$ auf J gibt ([2], S. 377). Also ist f nicht stetig. Es ist nicht schwierig zu zeigen, dass die Unstetigkeitsstellen von f auf abzählbar vielen Hyperebenen $\mathbf{R} \times \{y\}$ bzw. $\{x\} \times \mathbf{R}$ liegen, wobei $x, y \in \{z \in J : z = k \cdot 2^{-l}, k, l \in \mathbb{N}\}$ gilt und \mathbf{R} die reellen Zahlen bezeichne. Das Ziel dieser Note ist es, einen elementaren Beweis dafür zu geben, dass f ein Borel-Isomorphismus ist.