

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

Autor(en): **Jeger, M.**

Objektyp: **Article**

Zeitschrift: **Elemente der Mathematik**

Band (Jahr): **35 (1980)**

Heft 3

PDF erstellt am: **24.09.2024**

Persistenter Link: <https://doi.org/10.5169/seals-34681>

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.

Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.

Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

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

(Fortsetzung)

6. Anwendungen

Die Bestimmung des grössten gemeinsamen primitiven Teilers ganzzahliger Polynome stellt sich gelegentlich als isoliertes Problem. Daneben ist sie aber auch als Teilproblem in manchen Anwendungen anzutreffen. Dazu gehören z. B. die in der Einleitung erwähnten algebraischen *Verfahren zur Eingrenzung der reellen Nullstellen eines Polynoms* mit rationalen Koeffizienten. Wir wollen anschliessend noch kurz auf diese Anwendung des euklidischen Algorithmus eingehen und insbesondere das vorliegende Programm noch mit einer entsprechenden Ergänzung versehen.

Das anvisierte algebraische Verfahren besteht darin, dass man zunächst ein vorgegebenes ganzzahliges Polynom $a(x)$ von mehrfachen Nullstellen befreit und hernach für das reduzierte Polynom $\hat{a}(x)$ die sogenannte *Sturmsche Kette* bestimmt. Beide Schritte beruhen auf dem euklidischen Algorithmus.

a) Befreiung eines Polynoms $a(x)$ von mehrfachen Nullstellen

Es sei $\omega(x)$ ein irreduzibler Faktor von $a(x)$ und $\omega^r(x)$ die höchste Potenz, die in $a(x)$ enthalten ist. Es ist dann

$$a(x) = \omega^r(x) \cdot \beta(x), \quad \text{wobei } \omega(x) \text{ und } \beta(x) \text{ teilerfremd.} \quad (6.1)$$

Für die Ableitung von $a(x)$ folgt aus (6.1)

$$\begin{aligned} a'(x) &= r \omega^{r-1}(x) \cdot \omega'(x) \beta(x) + \omega^r(x) \cdot \beta'(x) \\ &= \omega^{r-1}(x) (r \omega'(x) \beta(x) + \omega(x) \beta'(x)). \end{aligned}$$

Der Klammerausdruck auf der rechten Seite ist sicher nicht durch $\omega(x)$ teilbar, denn $g(\omega') = g(\omega) - 1$. Daher ist

$$a'(x) = \omega^{r-1}(x) \cdot \gamma(x), \quad \text{wobei } \omega(x) \text{ und } \gamma(x) \text{ teilerfremd.} \quad (6.2)$$

Ist nun

$$a(x) = \omega_1^{f_1}(x) \cdot \omega_2^{f_2}(x) \cdots \omega_s^{f_s}(x)$$

die Zerfällung von $a(x)$ in irreduzible Faktoren, dann ist

$$\hat{a}(x) = \omega_1(x) \cdot \omega_2(x) \cdots \omega_s(x)$$

ein Polynom mit denselben, aber jetzt nur noch einfachen Nullstellen. Der grösste gemeinsame Teiler von $a(x)$ und $a'(x)$ ist wegen (6.1) und (6.2)

$$\mu(x) = \omega_1^{r_1-1}(x) \omega_2^{r_2-1}(x) \cdots \omega_s^{r_s-1}(x),$$

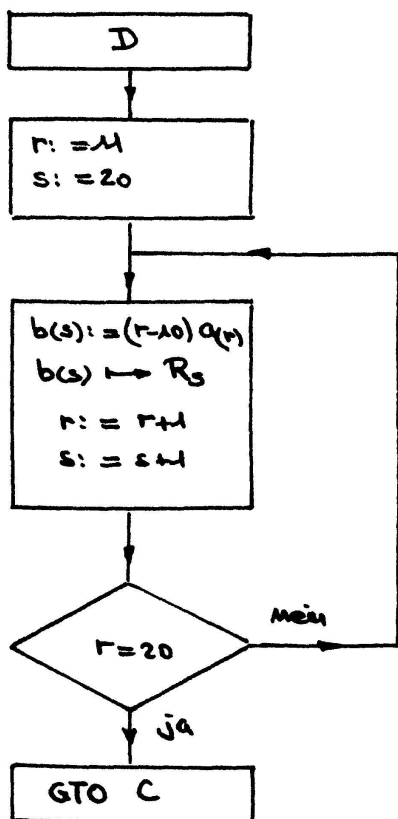
so dass also

$$a(x) = \mu(x) \cdot \hat{a}(x). \tag{6.3}$$

Der Übergang von $a(x)$ zu $\hat{a}(x)$ - d.h. die Befreiung des Polynoms $a(x)$ von mehrfachen Nullstellen - erfordert also nur eine euklidische Ketten-Division und eine einfache Division, nicht aber die Bestimmung der einzelnen irreduziblen Faktoren. Der Prozess verläuft somit ganz im massgebenden Polynomring, im vorliegenden Falle also in $[Q[x]; +, \cdot]^5$.

Der Prozess der Befreiung eines Polynoms von mehrfachen Nullstellen legt nun nahe, das vorliegende Programm noch so zu ergänzen, dass nach Eingabe von $a(x)$ ins Polynom-Register I vorerst $a'(x)$ berechnet und ins Polynom-Register II eingespeichert wird.

Flussdiagramm



Figur 6

Rechnerprogramm

510	76	Lb1	530	95	=
511	14	D	531	72	STO Ind
512	01	}	532	05	05
513	01		533	69	Op
514	42	STO	534	24	24
515	04	04	535	69	Op
516	02	}	536	25	25
517	00		20	537	02
518	42	STO	538	00	20
519	05	05	539	32	$x \geq t$
→ 520	73	RCL Ind	540	43	RCL
521	04	04	541	04	04
522	65	*	542	67	$x = t$
523	53	(543	13	C
524	43	RCL	544	61	GTO
525	04	04	545	05	}
526	75	-	546	20	
527	01	}			
528	00		10		
529	54)			

Der Start der Rechnung erfolgt über die Taste D.

b) Die Sturmsche Kette

Von dem aus Genf stammenden Mathematiker *Charles Sturm* (1803-1855) stammt ein bemerkenswertes Verfahren zur Eingrenzung der reellen Nullstellen des Polynoms

$$a(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0 \quad \text{mit} \quad a_k \in R$$

5) Vgl. etwa [4].

in einem vorgegebenen Intervall $[c_1, c_2]$. Dazu ist vorerst das zugehörige Polynom $\hat{a}(x)$ zu bestimmen und hernach mit

$$\lambda_0(x) = \hat{a}(x) \quad \text{und} \quad \lambda_1(x) = \hat{a}'(x)$$

gemäss

$$\lambda_{k-1}(x) = \lambda_k(x) \sigma_k(x) - \lambda_{k+1}(x); \quad \lambda_{k+1}(x) = 0 \quad \text{oder} \quad g(\lambda_{k+1}) < g(\lambda_k)$$

die Polynom-Kette

$$\lambda_0(x), \quad \lambda_1(x), \quad \lambda_2(x), \quad \dots, \quad \lambda_s(x) \tag{6.4}$$

zu berechnen. Bezeichnet nun $w(c)$ die Anzahl der Vorzeichenwechsel in der Folge der Funktionswerte an der Stelle c

$$\lambda_0(c), \quad \lambda_2(c), \quad \lambda_2(c), \quad \dots, \quad \lambda_s(c),$$

dann ist für $c_1 < c_2$ die Zahl

$$w(c_1) - w(c_2)$$

gleich der Anzahl reeller Nullstellen von $a(x)$ im Intervall $[c_1, c_2]$.

Bei der Berechnung von $w(c)$ sind allenfalls verschwindende Glieder in der Folge der Funktionswerte zu streichen.

Die Polynomfolge (6.4) wird die Sturmsche Kette zum Polynom $\hat{a}(x)$ genannt. Für den Beweis des eben zitierten Satzes von Sturm sei der Leser auf die entsprechende Spezialliteratur verwiesen⁶⁾.

Im Falle eines Polynoms $\hat{a}(x)$ mit ganzzahligen Koeffizienten kann man nun für die Bestimmung der Sturmschen Kette auf unsern Algorithmus zurückgreifen. Da nur die Vorzeichen der Funktionswerte benötigt werden, kann man nämlich die Polynomfolge (6.4) ersetzen durch die Folge der zugehörigen assoziierten primitiven Polynome

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

wobei sich $\varphi_k(x)$ und $\lambda_k(x)$ jeweils durch einen positiven Faktor unterscheiden. Diese modifizierte Sturmsche Kette liefert aber gerade der im Abschnitt 4 beschriebene Algorithmus.

Da $\hat{a}(x)$ und $\hat{a}'(x)$ teilerfremd sind, endet diese Kette stets mit $\varphi_s(x) = \pm 1$.

Beispiel 3

Es sollen die reellen Nullstellen des Polynoms

$$a(x) = 8x^6 + 12x^5 + 22x^4 - 15x^3 - 48x^2 - 28x - 5$$

durch Angabe geeigneter Intervalle eingegrenzt werden.

6) Vgl. etwa [4] und [5].

Zunächst erhält man zu $\varphi_0(x) = a(x)$ eine Sturmsche Kette, die mit dem Polynom

$$\varphi_4(x) = -4x^2 - 4x - 1$$

schliesst. $a(x)$ besitzt demnach mehrfache Nullstellen. Die Division von $a(x)$ durch das Polynom

$$\mu(x) = 4x^2 + 4x + 1$$

ergibt den Quotienten

$$\hat{a}(x) = 2x^4 + x^3 + 4x^2 - 8x - 5.$$

Die Koeffizienten von $\hat{a}(x)$ leuchten bei der Durchführung des euklidischen Algorithmus mit $a(x)$ und $\mu(x)$ nacheinander kurz auf (Bemerkung 4 auf Seite 40, 1. Teil). Für $\hat{a}(x)$ erhält man schliesslich die Sturmsche Kette

$$\begin{aligned} \hat{a}(x) = \varphi_0(x) &= 2x^4 + x^3 + 4x^2 - 8x - 5 \\ \varphi_1(x) &= 8x^3 + 3x^2 + 8x - 8 \\ \varphi_2(x) &= -61x^2 + 200x + 152 \\ \varphi_3(x) &= -2056x - 1077 \\ \varphi_4(x) &= -1. \end{aligned}$$

Daraus liest man die folgende Vorzeichenverteilung ab:

c	$\varphi_0(c)$	$\varphi_1(c)$	$\varphi_2(c)$	$\varphi_3(c)$	$\varphi_4(c)$	$w(c)$
$-\infty$	+	-	-	+	-	3
-1	+	-	-	+	-	3
0	-	-	+	-	-	2
+1	-	+	+	-	-	2
+2	+	+	+	-	-	1
$+\infty$	+	+	-	-	-	1

Sie lässt darauf schliessen, dass $a(x)$ total 2 reelle Nullstellen hat; eine davon liegt im Intervall $[-1, 0]$, die andere im Intervall $[1, 2]$.

Die Figur 7 zeigt den Ausdruck des Rechners zum Beispiel 3. Die Rechenzeiten betragen für die drei Bestandteile der Rechnung der Reihe nach 290, 110 und 230 Sekunden.

7. Polynome mit reellen Koeffizienten

Wie das Beispiel 2 deutlich belegt, ist der obere Plafond der Ganzzahligkeit bei einem Taschen-Rechner sehr rasch erreicht. Es liegt daher nahe, den euklidischen Algorithmus auch noch auf Polynome mit reellen Koeffizienten auszudehnen. Dies ist ohne weiteres durch eine geeignete Modifikation des vorliegenden Programmes

```

POLYNOM-KETTE:
  0.
    8.
   12.
   22.
  -15.
  -48.
  -28.
   -5.

  1.
   48.
   60.
   88.
  -45.
  -96.
  -28.

  2.
 -116.
  268.
  723.
  464.
   92.

  3.
-11744.
-18204.
 -9396.
-1615.

  4.
   -4.
   -4.
   -1.

      ENDE
    
```

```

POLYNOM-KETTE:
  0.
    8.
   12.
   22.
  -15.
  -48.
  -28.
   -5.

  1.
   4.
   4.
   1.

      ENDE
    
```

Bei diesem Prozess leuchten nacheinander die Zahlen

2048
1024
4096
-8192
-5120

auf. Das zugehörige primitive Quotientenpolynom hat die Koeffizienten

2
1
4
-8
-5

```

POLYNOM-KETTE:
  0.
    2.
    1.
    4.
   -8.
   -5.

    1.
    8.
    3.
    8.
   -8.

    2.
   -61.
   200.
   152.

    3.
  -2056.
 -1077.

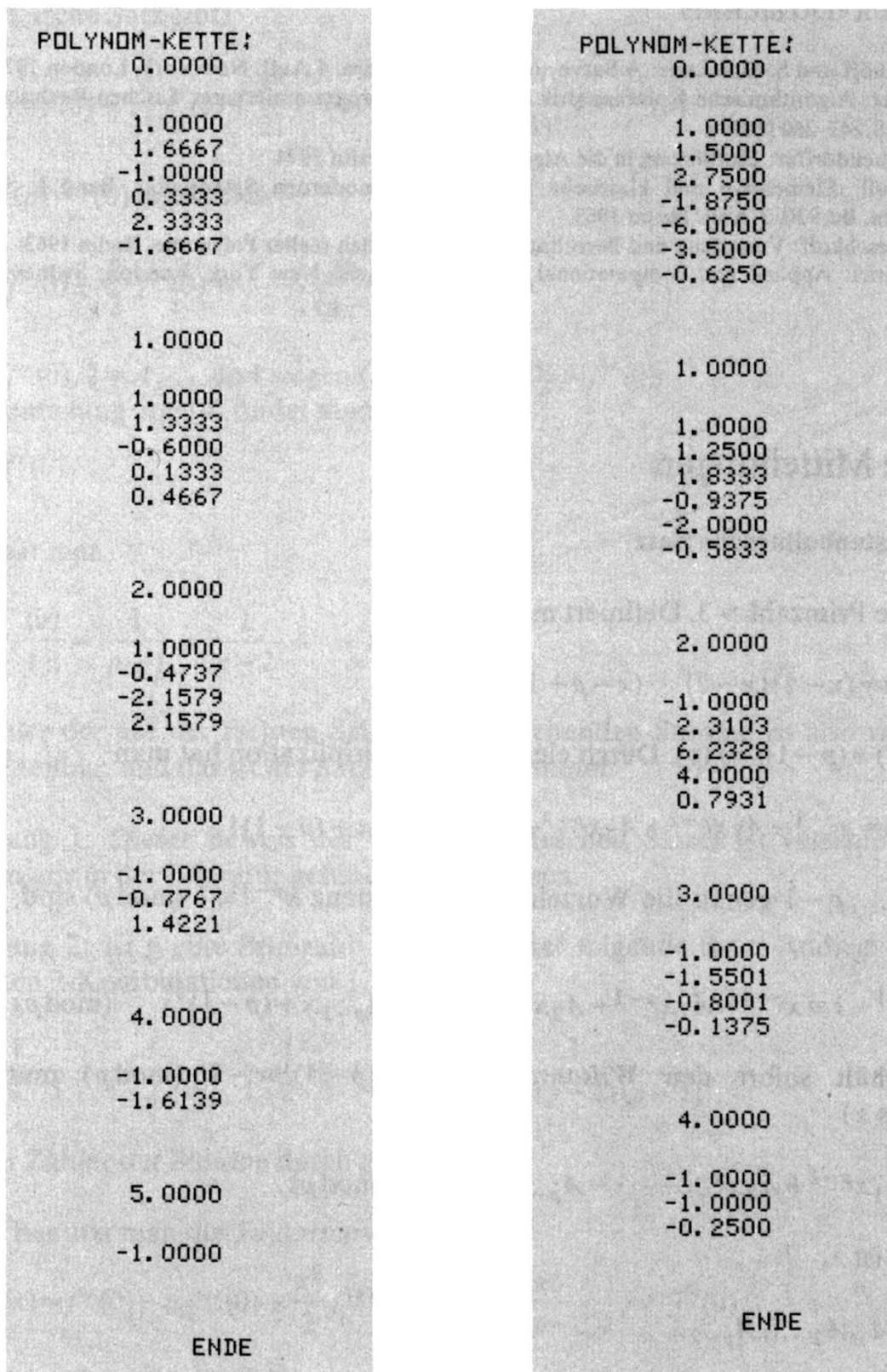
    4.
   -1.

      ENDE
    
```

Figur 7

möglich. Man kann etwa an die Stelle der bis jetzt verwendeten primitiven Polynome sogenannte *normierte Polynome* treten lassen, deren Leitgliedkoeffizient +1 oder -1 ist. Dazu ist im wesentlichen nur das Unterprogramm B' entsprechend abzuändern.

Die letzten Bemerkungen sind primär als Anregung für den interessierten Leser gedacht. Wir verzichten auf eine vollständige Beschreibung des neuen Programmes und führen nur zwei Rechenbeispiele an. Die Figur 8 zeigt in der linken Spalte



Figur 8

die Sturmsche Kette zum Polynom $a(x) = 3x^5 + 5x^4 - 3x^3 + x^2 + 7x - 5$ aus dem Beispiel 2. Sie lässt sich jetzt zu Ende führen. In der rechten Spalte ist nochmals der erste Teil der Rechnung zum Beispiel 3 durchgeführt. Um ein möglichst übersichtliches Protokoll zu erhalten, wurde der Rechner im Zustand Fix 4 betrieben.

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 Götschen, Bd.930, 3. Aufl. Berlin 1963.
- 5 N. Obreschkoff: Verteilung und Berechnung der Nullstellen reeller Polynome. Berlin 1963.
P. Henrici: Applied and computational complex Analysis. New York, London, Sydney, Toronto 1974.

Kleine Mitteilungen

Der Wolstenholmesche Satz

Sei p eine Primzahl > 3 . Definiert man

$$f(x) = (x-1)(x-2)\cdots(x-p+1),$$

so ist $f(0) = (p-1)! = f(p)$. Durch elementare Multiplikation hat man

$$f(x) = x^{p-1} - A_1 x^{p-2} + A_2 x^{p-3} - \cdots - A_{p-2} x + (p-1)! \quad (1)$$

Da $1, 2, \dots, p-1$ genau die Wurzeln der Kongruenz $x^{p-1} \equiv 1 \pmod{p}$ sind, so folgt aus (1)

$$x^{p-1} - 1 \equiv x^{p-1} - A_1 x^{p-2} + A_2 x^{p-3} - \cdots - A_{p-2} x + (p-1)! \pmod{p}. \quad (2)$$

Man erhält sofort den *Wilsonschen Satz* $(p-1)! \equiv -1 \pmod{p}$ und ferner (für jedes x)

$$-A_1 x^{p-2} + A_2 x^{p-3} - \cdots - A_{p-2} x \equiv 0 \pmod{p}.$$

Deshalb ist

$$p \mid A_1, A_2, \dots, A_{p-2}. \quad (3)$$

Weiterhin hat man durch Differentiation

$$f'(x) = (x-2)\cdots(x-p+1) + \cdots + (x-1)\cdots(x-p+2)$$

und daraus

$$f'(0) = -f'(p) = -\{1 \cdot 2 \cdots (p-2) + \cdots + 2 \cdot 3 \cdots (p-1)\}.$$