

Jürgen Schwarz

# Behandlung von Polynomen – Teil 2\*

## Stichwörter:

Polynom, HP-BASIC, Laplace-Transformation, Approximation, Ausgleichsrechnung

Im ersten Teil wurde die Darstellung von Polynomen mit reellen und komplexen Koeffizienten im Computer behandelt und es wurden SUB-Programme zur Addition, Subtraktion, Multiplikation und Division von Polynomen abgeleitet.

## Hardware:

HP 9845 B mit SP ROM

## 5 Nullstellenbestimmung von Polynomen

Ist man vor die Aufgabe gestellt, die Nullstellen (Wurzeln) eines Polynoms zu bestimmen, d.h. die Aufgabe  $f(x) = 0$  zu lösen, so liefert der Fundamentalsatz der Algebra die Aussage, daß dabei  $n$  Lösungen zu suchen sind: Jede Gleichung  $n$ -ten Grades besitzt  $n$  reelle oder komplexe Wurzeln, wobei die  $k$ -fachen Wurzeln  $k$ -mal gezählt werden.

Sei  $x_1$  eine Wurzel von  $f(x) = 0$ , so ist  $f(x)$  durch  $(x - x_1)$  teilbar. Das aus dieser Division folgende Polynom  $f^* = f(x) : (x - x_1)$  nennt man Deflationspolynom. Es hat einen um 1 niedrigeren Grad als das Ausgangspolynom. Dieses Verfahren kann man fortführen bis zur Lösung  $x_n$ . Entsprechend kann jedes Polynom als Produkt

$$f(x) = a_n x^n + \dots + a_0 = a_n (x - x_1) (x - x_2) \dots (x - x_n); \quad x_i, a_i, \in \mathbb{C} \quad (10)$$

dargestellt werden.

Sind alle Koeffizienten  $a_0, a_1, \dots, a_n$  reell, dann treten nur reelle und konjugiert komplexe Wurzeln auf. Das heißt, ist  $x_1 = a + ib$  eine Nullstelle des Polynoms, dann ist der konjugiert komplexe Wert  $x_2 = a - ib$  ebenfalls eine Nullstelle und zwar mit derselben Vielfachheit. Daraus folgt, daß Polynome mit nur reellen Koeffizienten und ungeradem Grad  $n$  wenigstens eine reelle Nullstelle aufzuweisen haben und entsprechende Polynome mit nur konjugiert komplexen Nullstellen nur geraden Grades sein können.

Zur Bestimmung der Wurzeln eines Polynoms ist man im allgemeinen auf Iterationsverfahren angewiesen. Bis zu einem Grad des Polynoms  $n = 4$  existieren Lösungsformeln, für  $n > 4$  sind solche aber nicht aufstellbar.

Die beiden hier beschriebenen Iterationsverfahren bilden nach der Berechnung einer Nullstelle das Deflationspolynom. Von diesem wird dann die nächste Nullstelle berechnet, bis die Ordnung des letzten Polynomes  $n = 1$  ist. Diese Vorgehensweise garantiert aber keine Lösungen mit optimaler Genauigkeit. Besser ist es, die so gewonnenen Lösungen als Startwerte für eine iterative Lösung mit dem Ausgangspolynom  $f(x)$  zu verwenden, besonders in den Fällen, in denen das Abdividieren die Kondition verschlechtert hat. Dieser [3] entnommene Gedanke sei hier aber nicht weiter ausgeführt.

### 5.1 Nullstellenbestimmung bei Polynomen mit reellen Koeffizienten und nur reellen Nullstellen

Oft treten Polynome auf, von denen man aus der Vorgeschichte weiß, daß sämtliche Nullstellen einfach und reell sind. Ein Beispiel dafür sind aus elektrischen Schaltungen mit nur einer Art von Energiespeichern (RC- bzw. RL-Schaltungen) abgeleitete Polynome, bei denen nur einfache Nullstellen auf der negativen reellen Achse auftreten [4].

Zur schnellen Nullstellenbestimmung ist hier das Newtonverfahren besonders geeignet. Dieses allgemein bekannte Verfahren berechnet die Nullstellen einer Funktion  $f(x)$  nach der Iterationsvorschrift

$$x^{(v+1)} = x^{(v)} - \frac{f(x^{(v)})}{f'(x^{(v)})} \quad (11)$$

Die Iteration wird beendet, wenn die Änderung der  $x$ -Werte ein bestimmtes Maß unterschreitet.

Listing 5 zeigt die Realisierung des Newtonverfahrens in einem SUB-Programm. Der Algorithmus wurde in Anlehnung an [5] gestaltet.

Die Nullstellen seien in absteigender Größe geordnet

$$x_1 > x_2 > x_3 > \dots > x_n \quad (12)$$

Jürgen Schwarz, Wiesbadener Str. 59 c, 1000 Berlin 33  
\* Teil 1: s. CAL – Comp.-Anw. Lab. 3 (1985) 45.

```

10  SUB Newton(INTEGER N_polynom,REAL R_koeff(*),R_wurzel(*),Epsilon)
20  !
30  ! SUB-Programm zur Berechnung der Nullstellen (Wurzeln) eines Polynoms
40  ! f(x) vom Grad n mit reellen Koeffizienten und nur reellen Nullstellen
50  !
60  ! Eingabedaten:
70  !   n_polynom           ... Grad des Polynoms
80  !   r_koeff(0:n_polynom) ... Koeffizienten des Polynoms
90  !   epsilon             ... Genauigkeitsschranke für die Nullstellen
100 ! Ergebnis:
110 !   r_wurzel(1:n_polynom) ... Nullstellen (Wurzeln) des Polynoms
120 !
130 !   Programmierer:   Jürgen Schwarz
140 !   Programm-Name:   Newton
150 !   Datum/Variante:  05.11.84 / 02
160 !   Speichermedium:  Kassetten 57/58
170 !
180  INTEGER J,K,N
190  REAL A_a,Delta_x,Eps,X
200  REAL A(0:N_polynom),B(0:N_polynom),C(1:N_polynom)
210  !
220  ! Datensicherung
230  !
240  N=N_polynom
250  REDIM R_koeff(0:N_polynom)
260  MAT A=R_koeff
270  MAT R_wurzel=ZER
280  REDIM R_wurzel(1:N_polynom)
290  !
300  ! Berechnung des Wertebereiches und der erforderlichen Genauigkeit
310  !
320  A_a=0
330  FOR J=0 TO N-1
340   A_a=MAX(A_a,ABS(A(J)))
350  NEXT J
360  A_a=A_a/ABS(A(N))
370  X=A_a+1
380  Eps=Epsilon*X
390  !
400  ! Beginn der Lösung
410  !
420  WHILE N>1
430   C(N)=B(N)=A(N)
440   FOR K=0 TO 2           ! zweimalige Nachiteration
450    REPEAT
460     FOR J=N-1 TO 1 STEP -1
470      B(J)=A(J)+X*B(J+1)
480      C(J)=B(J)+X*C(J+1)
490    NEXT J
500    B(0)=A(0)+X*B(1)
510    IF C(1)/A(N)<=0 THEN
520     PRINT "Fehler im SUB-Programm Newton!"
530     PRINT "Ergebnis konvergiert nicht!"
540     PAUSE
550    END IF
560    Delta_x=-B(0)/C(1)
570    X=X+Delta_x
580    DISP "N =";N;"   Delta_x =";Delta_x;"   Eps =";Eps
590    UNTIL ABS(Delta_x)<Eps
600  NEXT K
610  R_wurzel(N)=X           ! Lösung gefunden
620  FOR J=0 TO N-1
630   A(J)=B(J+1)           ! Reduktion der Ausgangsgleichung
640  NEXT J
650  N=N-1
660  END WHILE
670  X=-A(0)/A(1)           ! Lösung für den letzten Schritt bzw. n = 1
680  R_wurzel(1)=X
690  !
700  SUBEND

```

**Listing 5: Nullstellenberechnung von Polynomen mit reellen Koeffizienten und nur einfachen reellen Nullstellen.**

Als Anfangsnäherung wird ein Wert  $x^{(0)} > x_1$  gewählt, mit dem die Iteration begonnen wird. Zur Berechnung eines geeigneten Startwertes  $x^{(0)}$  ist die Ungleichung

$$|z_j| < 1 + \max_i \left| \frac{a_i}{a_n} \right|, i = 0(1)n-1 \quad (13)$$

geeignet, die aussagt, daß alle – reellen und komplexen – Wurzeln  $z_j$  eines Polynoms in der komplexen Zahlenebene innerhalb eines Kreises mit dem Radius  $|z_j|$  liegen. Weiß man vor Rechenbeginn, daß sämtliche Wurzeln des Polynoms negativ sind, dann wird die Iteration zweckmäßigerweise mit  $x^{(0)} = 0$  begonnen. Die Zeilen 320 – 380 werden dann in der Programmmodifikation Newton\_\_mod durch die Anweisungen

```
320 X = 0
330 Eps = Epsilon
```

ersetzt.

Die Berechnung der Funktionswerte  $f(x)$  und  $f'(x)$  erfolgt mit dem vollständigen Horner-Schema. Nach einem Durchlauf durch die Zeilen 460 – 500 enthält B(0) den Wert von  $f(x)$  und C(1) den Wert von  $f'(x)$ . Ist  $x$  zugleich eine Nullstelle von  $f(x)$ , dann enthält das Feld B(\*) die Elemente des Deflationspolynoms [die Koeffizienten des durch die Nullstelle dividierten Polynoms:  $f^*(x) = f(x) / (x - x_1)$ ], welches zur Berechnung der nächsten Nullstellen verwendet wird.

Aus der Abarbeitungsbedingung (12) folgt, daß  $a_n$  und  $f'(x)$  für  $x \geq x_1$  gleiches Vorzeichen haben müssen. Für beliebig hohes  $x$  wird das Vorzeichen von  $f(x)$  nur vom Vorzeichen von  $a_n$  bestimmt. Da aber rechts von  $x_1$  bedingungsgemäß keine Nullstelle sein darf, gilt

$$\left. \frac{f'(x)}{a_n} \right|_{x \geq x_1} > 0, \quad (14)$$

wenn das Polynom nur einfache reelle Wurzeln hat. (14) kann darum als Konvergenzkriterium verwendet werden (Zeile 510).

Beim Vorhandensein von konjugiert komplexen Wurzeln des Ausgangspolynoms  $f(x)$  erfolgt ein Abbruch der Iteration. Dasselbe gilt i.a. auch bei mehrfachen Nullstellen, da dann  $f'(x = x_1 = x_2) = 0$  ist und Rundungsfehler bzw. das Versagen der Iterationsvorschrift (11) einen Abbruch bewirken.

## 5.2 Nullstellenbestimmung bei beliebigen Polynomen

Im folgenden soll ein Nullstellenberechnungsverfahren beschrieben werden, welches in der Literatur nur selten erwähnt wird, welches sich aber in der rechen-technischen Praxis als sehr stabil und gut konvergierend herausgestellt hat [6]. Zudem ist es universell einsetzbar. Es gestattet auch die Berücksichtigung von Polynomen mit komplexen Koeffizienten.

Den Käufern des Computers HP 9845 B (Hersteller: Hewlett Packard) wird ein Programm, welches nach diesem ŠILJAK'schen Verfahren arbeitet, mitgeliefert. Dieses Programm wurde, um es besser verständlich zu machen, aus seiner unstrukturierten Form in eine strukturierte Form gebracht und in deutscher Sprache kommentiert. Listing 6 zeigt das Programm, dessen Funktion nun erläutert werden soll.

Der Wert des Polynoms

$$f(z) = u + iv = \sum_{j=0}^n (a_j + ib_j) z^j \quad (15)$$

an einer beliebigen Stelle  $z = x + iy$  hat einen Realteil  $u$  und einen Imaginärteil  $v$ . Eine Nullstelle des Polynoms liegt dann vor, wenn sowohl  $u$  als auch  $v$  null sind. Mit

$$\begin{aligned} u &= u(x, y) = 0 \\ v &= v(x, y) = 0 \end{aligned} \quad (16)$$

ist damit die Aufgabe  $f(z) = 0$  in ein zweidimensionales nichtlineares Gleichungssystem umgewandelt worden. Zur Lösung dieses Gleichungssystems lassen sich viele Methoden verwenden; hier wird das Verfahren des steilsten Abstiegs (Gradientenverfahren) angewandt. Dazu wird die Funktion

$$F = F(x, y) = u^2(x, y) + v^2(x, y) \quad (17)$$

gebildet. Die Aufgabe, die Nullstellen von  $F$  zu finden, ist äquivalent zu der Aufgabe des Gleichungssystems (16) zu lösen und auch äquivalent zu  $f(z) = 0$ .

Um die Eigenschaften von  $F(x, y)$  zu ermitteln, muß man eine Kurvendiskussion durchführen. Da die Funktion einen polynomialen Zusammenhang mit  $x$  und  $y$  hat, d.h. daß sich  $F$  aus einer Summe von verschiedenen bewerteten Potenzen und Potenzprodukten von  $x$  und  $y$  zusammensetzt [vgl. Gl. (33) u. (34)], ist der Definitionsbereich in der komplexen Ebene nicht beschränkt. Der Wertevorrat liegt zwischen null und plus unendlich. Nullstel-

len liegen, wie unmittelbar aus Gl. (17) ersichtlich, bei  $u = v = 0$ . Notwendige Bedingung für das Auftreten von Extremwerten ist, daß die ersten Ableitungen null werden. Mit

$$\begin{aligned} F_x &= \frac{\partial F}{\partial x} = 2(u u_x + v \cdot v_x) \\ F_y &= \frac{\partial F}{\partial y} = 2(u u_y + v \cdot v_y) \end{aligned} \quad (18)$$

kann man die Ableitungen allgemein angeben. Nun ist aber  $f(z)$  eine reguläre analytische Funktion in der gesamten komplexen Ebene und damit gelten die CAUCHY-RIEMANSCHEN Differentialgleichungen für  $u$  und  $v$ :

$$\begin{aligned} u_x(x, y) &= v_y(x, y) \\ u_y(x, y) &= -v_x(x, y) \end{aligned} \quad (19)$$

Diese beiden Differentialgleichungen in Gl. (18) eingesetzt und nulliert, ergeben die Gleichungen

$$\begin{aligned} F_x &= 2(u u_x + v v_x) = 0 \\ F_y &= 2(v \cdot u_x - u \cdot v_x) = 0 \end{aligned} \quad (20)$$

Gl. (20) hat zwei prinzipielle Lösungen, nämlich

$$1. \text{ Lösung: } u = v = 0 \quad (21)$$

$$2. \text{ Lösung: } u_x = v_x = 0 \quad (22)$$

Nun muß noch geprüft werden, ob beide Lösungen Extremwerte liefern. Hinreichende Bedingung dafür ist, daß die JACOBI-Matrix in dem entsprechenden Punkt definit ist.

Das bedeutet für eine zweidimensionale Funktion die Erfüllung der Ungleichung

$$F_{xx} \cdot F_{yy} - F_{xy}^2 > 0 \quad (23)$$

Zur Prüfung berechnen wir die zweiten Ableitungen

$$\begin{aligned} F_{xx} &= 2(u_x^2 + u \cdot u_{xx} + v \cdot v_{xx} + v_x^2) \\ F_{xy} &= 2(v \cdot u_{xx} - u \cdot v_{xx}) \\ F_{yy} &= 2(u_x^2 - u \cdot u_{xx} - v \cdot v_{xx} + v_x^2) \end{aligned} \quad (24)$$

unter Verwendung der Beziehungen

$$\begin{aligned} u_{xx} &= v_{yy} = -u_{yy} \\ v_{yy} &= u_{xx} = -v_{xx} \end{aligned} \quad (25)$$

die durch weitere partielle Differentiation von Gl. (19) gewonnen werden und die die

```

10  SUB Siljak<INTEGER N,I_max,REAL R_k(*),I_k(*),Eps_w,Eps_f,R_w(*),I_w(*)>
20  !
30  ! *****
40  ! *** Nullstellen-Berechnungsprogramm für Polynome n-ten Grades ***
50  ! *** mit komplexen Koeffizienten r_k(*) und i_k(*) ***
60  ! ***
70  ! *** Rechenverfahren mit siljak'schen Koeffizienten ***
80  ! *** Ausführung in strukturierter Programmierung ***
90  ! ***
100 ! *** Eingangsgr.: n - Grad des Polynoms ***
110 ! *** I_max - Maximalzahl der Iterationen ***
120 ! *** zum Finden einer Nullstelle ***
130 ! *** r_k(0:n) - Realteil der Polynomkoeffizienten ***
140 ! *** i_k(0:n) - Imaginärteil der Polynomkoeffizienten ***
150 ! *** eps_w - zulässige Abweichung für die Wurzeln ***
160 ! *** eps_f - zulässige Abweichung für Real- und ***
170 ! *** Imaginärteile des Polynoms f = f(z) ***
180 ! ***
190 ! *** Ausgangsgr.: r_w(1:n) - Realteil der Nullstellen (Wurzeln) ***
200 ! *** i_w(1:n) - Imaginärteil der Nullstellen ***
210 ! ***
220 ! *** Programmierer: Jürgen Schwarz ***
230 ! *** Programm-Name: Siljak ***
240 ! *** Datum/Variante: 24.10.84 / 03 ***
250 ! *** Speichermedium: Kassetten 57/58 ***
260 ! ***
270 ! *****
280 !
290 INTEGER Boo,I,K,L,M
300 REAL A,B,C,D,Delta_x,Delta_y,F,F_alt,P,Q,U,V,X,Y,Z
310 REAL R_koeff(0:N),I_koeff(0:N),X_siljak(0:N),Y_siljak(0:N)
320 !
330 PRINTER IS 16
340 STANDARD
350 !
360 LOOP
370 Boo=(N<=0) OR (Eps_w<=0) OR (Eps_f<=0) OR (I_max<=0)
380 EXIT IF NOT Boo
390 ! Test auf falsche Eingangsdaten.
400 ! Druck einer Fehlermeldung und Unterbrechung des Programms.
410 ! Korrigiere die Daten entsprechend und führe die Rechnung fort.
420 PRINT LIN(2),"Fehler im Subprogramm Siljak. Unzulässige Daten."
430 PRINT LIN(1),"n =";N,"eps_w =";Eps_w
440 PRINT "eps_f =";Eps_f,"I_max =";I_max,LIN(2)
450 PAUSE
460 END LOOP
470 REDIM R_k(0:N),I_k(0:N),R_w(1:N),I_w(1:N)
480 !
490 L=N ! Schutz der Ausgangsdaten
500 MAT R_koeff=R_k ! Schutz der Ausgangsdaten
510 MAT I_koeff=I_k ! Schutz der Ausgangsdaten
520 MAT R_w=(9.999999E99) ! Vorinitialisierung
530 MAT I_w=(9.999999E99) ! Vorinitialisierung
540 !
550 WHILE L>1
560 Y_siljak(0)=I=0
570 Y=Y_siljak(1)=X_siljak(0)=1 ! Startwert der Berechnung
580 X=X_siljak(1)=.1 ! Startwert der Berechnung
590 GOSUB Siljak ! Berechnung der Siljak-Koeffizienten
600 REPEAT
610 F_alt=F
620 M=Q=P=0
630 I=I+1
640 FOR K=1 TO L
650 P=P+K*(R_koeff(K)*X_siljak(K-1)-I_koeff(K)*Y_siljak(K-1)) ! u_x
660 Q=Q+K*(R_koeff(K)*Y_siljak(K-1)+I_koeff(K)*X_siljak(K-1)) ! v_x
670 NEXT K
680 Z=P*P+Q*Q ! Z=(du/dx)^2+(dv/dx)^2
690 Delta_x=- (U*P+V*Q)/Z ! delta_x: Änderung des Realteils
700 Delta_y=(U*Q-V*P)/Z ! delta_y: Änderung des Imaginärteils
710 LOOP
720 M=M+1 ! Zähler für die Anzahl der Viertelungen
730 DISP "Siljak: L =";L,"I =";I,"M =";M
740 !
750 X_siljak(1)=X+Delta_x ! neue Wurzel-Approximation
760 Y_siljak(1)=Y+Delta_y ! neue Wurzel-Approximation
770 GOSUB Siljak ! Neuberechnung der Siljak-Koeffizienten
780 ! Test, ob die Werte von u und v kleiner als zulässigen sind,
790 ! wenn m > 20 und der neue Wert von F größer als der alte ist.
800 Boo=(F)=F_alt) AND (M>20) AND (ABS(U)<=Eps_f) AND (ABS(V)<=Eps_f)
810 EXIT IF Boo ! Wurzel gefunden

```

Listing 6: Nullstellenberechnungsverfahren nach ŠILJAK.

# SERIEN

```

820      ! Test, ob die Werte für die Schrittweiten delta_x und delta_y
830      ! kleiner als die zulässige Toleranz der Wurzeln sind.
840      Boo=(F<F_alt) AND (ABS(Delta_x)<Eps_w) AND (ABS(Delta_y)<Eps_w)
850      EXIT IF Boo      ! Wurzel gefunden
860      EXIT IF F<F_alt      ! Berechnung eines neuen Gradienten
870      ! Ist der neue Wert von F größer als der alte Wert, dann wird die
880      ! Schrittweite geviertelt. Die maximale Anzahl der Viertelungen
890      ! ist gleich 20. Wird dieser Wert überschritten, dann findet ein
900      ! Test auf Einhaltung der Funktionstoleranzen von u und v statt.
910      ! Werden diese nicht eingehalten, dann erfolgt eine Fehlermeldung.
920      Boo=(M>20) AND (ABS(U)<=Eps_f) AND (ABS(V)<=Eps_f)
930      EXIT IF Boo      ! Wurzel gefunden
940      EXIT IF M>20      ! Abbruch der Rechnung und Fehlermeldung
950      Delta_x=.25*Delta_x      ! Viertelung der Schrittweite
960      Delta_y=.25*Delta_y      ! Viertelung der Schrittweite
970      END LOOP
980      IF (M>20) AND NOT Boo THEN
990      ! Druck einer Fehlermeldung und Unterbrechung des Programms.
1000     PRINT LIN(2),"Fehler im Subprogramm Siljak. Die Schrittweite"
1010     PRINT "wurde 20 mal geviertelt und die zulässige Toleranz für"
1020     PRINT "die Funktionswerte u und v ist noch überschritten."
1030     PRINT LIN(1),"eps_f=";Eps_f,"u=";U,"v=";V,LIN(2)
1040     PAUSE
1050     END IF
1060     WHILE (I>I_max) AND NOT Boo
1070     ! Druck einer Fehlermeldung und Unterbrechung des Programms
1080     PRINT LIN(2),"Fehler im Subprogramm Siljak. Die maximal"
1090     PRINT "zulässige Zahl von Iterationen ist überschritten."
1100     PRINT LIN(1),"I=";I,"I_max=";I_max,LIN(2)
1110     PAUSE
1120     END WHILE
1130     X=X_siljak(1)
1140     Y=Y_siljak(1)
1150     UNTIL Boo
1160     R_w(L)=X      ! Realteil der berechneten Wurzel
1170     I_w(L)=Y      ! Imaginärteil der berechneten Wurzel
1180     !
1190     ! Division durch die gefundene Nullstelle
1200     ! Berechnung neuer Koeffizienten r_koeff(*) und i_koeff(*)
1210     A=R_koeff(L)
1220     B=I_koeff(L)
1230     R_koeff(L)=I_koeff(L)=0
1240     FOR K=L-1 TO 0 STEP -1
1250         C=R_koeff(K)
1260         D=I_koeff(K)
1270         R_koeff(K)=A+X*R_koeff(K+1)-Y*I_koeff(K+1)
1280         I_koeff(K)=B+X*I_koeff(K+1)+Y*R_koeff(K+1)
1290         A=C
1300         B=D
1310     NEXT K
1320     L=L-1      ! Neuer Grad des Polynoms
1330     END WHILE
1340     DISP
1350     !
1360     ! Algebraische Berechnung der (letzten) Nullstelle (für L=1)
1370     Z=R_koeff(1)*R_koeff(1)+I_koeff(1)*I_koeff(1)
1380     R_w(1)=-((R_koeff(0)*R_koeff(1)+I_koeff(0)*I_koeff(1))/Z)
1390     I_w(1)=(R_koeff(0)*I_koeff(1)-R_koeff(1)*I_koeff(0))/Z
1400     DISP
1410     SUBEXIT
1420     !
1430     Siljak: !
1440     ! Subroutine zur Berechnung der Siljak-Koeffizienten:
1450     ! (X_siljak(1) + i * Y_siljak(1))^n = X_siljak(n) + i * Y_siljak(n)
1460     Z=X_siljak(1)*X_siljak(1)+Y_siljak(1)*Y_siljak(1)
1470     FOR K=0 TO L-2
1480         X_siljak(K+2)=2*X_siljak(1)*X_siljak(K+1)-Z*X_siljak(K)
1490         Y_siljak(K+2)=2*X_siljak(1)*Y_siljak(K+1)-Z*Y_siljak(K)
1500     NEXT K
1510     ! Subroutine zur Berechnung des Polynoms:
1520     ! f(z) = f ( X_siljak(1) + i * Y_siljak(1) ) = u + i * v
1530     U=V=0
1540     FOR K=0 TO L
1550         U=U+R_koeff(K)*X_siljak(K)-I_koeff(K)*Y_siljak(K)      ! Realteil
1560         V=V+R_koeff(K)*Y_siljak(K)+I_koeff(K)*X_siljak(K)      ! Imaginärteil
1570     NEXT K
1580     ! F(z) = F ( X_siljak(1) + i * Y_siljak(1) ) = u^2 + v^2
1590     F=U+U+V+V
1600     RETURN
1610     !
1620     SUBEND

```

# SERIEN

LAPLACE'sche Differentialgleichung beinhalten. Gl. (24) in die Ungleichung (23) eingesetzt ergibt

$$(u_x^2 + v_x^2)^2 - (u^2 + v^2) \cdot (v_{xx}^2 + u_{xx}^2) > 0 \quad (25)$$

als hinreichende Bedingung für Extremwerte. Daraus folgt wiederum, daß nur die 1. Lösung nach Gl. (21) Extremwerte liefert (und zwar Minima, da  $F_{xx} = F_{yy} = 2(u_x^2 + v_x^2) > 0$ ) und die 2. Lösung nach Gl. (22) sicher weder ein Minimum noch ein Maximum hat, da die Ungleichung (25) nicht erfüllt wird. An den Lösungen von Gl. (22) hat die Fläche  $F = F(x, y)$  die Form eines Sattels.

Die hier abgebildeten Eigenschaften von  $F = F(x, y)$  zeigen, daß die Nullstellen von  $F$  zugleich die einzigen Minima der Funktion sind. Um diese Nullstellen laufen die Höhenlinien  $F = F(x, y) = \text{const.}$  in konzentrischen Kreisen. Durch

$$\vec{n} = \text{grad } F = \begin{pmatrix} F_x \\ F_y \end{pmatrix} \quad (26)$$

ist ein Vektor in Richtung der Normalen auf diesen Höhenlinien gegeben, der nach außen zeigt. Wandert man von einem beliebigen Punkt  $(x, y)$  in negativer Richtung zu diesen Gradienten, d.h. auf den Projektionen des stärksten Abstiegs der Funktion  $F$ , dann erreicht man bei einer geeignet gewählten Schrittweite eine neue, bessere Näherung für die gesuchte Nullstelle [3]. Die Iterationsvorschrift für das allgemeine Gradientenverfahren lautet daher

$$\vec{x}^{(v+1)} = \vec{x}^{(v)} - \mu \cdot \text{grad } F(\vec{x}^{(v)}); \quad \mu > 0, \text{ reell.} \quad (27)$$

In [3] wird für den Faktor  $\mu$  der Vorschlag

$$\mu = \frac{F(\varphi^{(v)})}{[\text{grad } F(\varphi^{(v)})]^2} \quad (28)$$

gemacht, der bei linearem  $F(x, y)$  mit einem Schritt zur Nullstelle führen würde. Zur Lösung des hier vorliegenden Problems ist ein doppelt so hoher Wert für  $\mu$  rechenzeit-sparend (etwa um den Faktor 3!). Die Iterationsvorschrift im zweidimensionalen Fall lautet allgemein

$$\begin{aligned} x^{(v+1)} &= x^{(v)} - \frac{2F \cdot F_x}{F_x^2 + F_y^2} \\ y^{(v+1)} &= y^{(v)} - \frac{2F \cdot F_y}{F_x^2 + F_y^2} \end{aligned} \quad \begin{matrix} x = x^{(v)} \\ y = y^{(v)} \end{matrix} \quad (29)$$

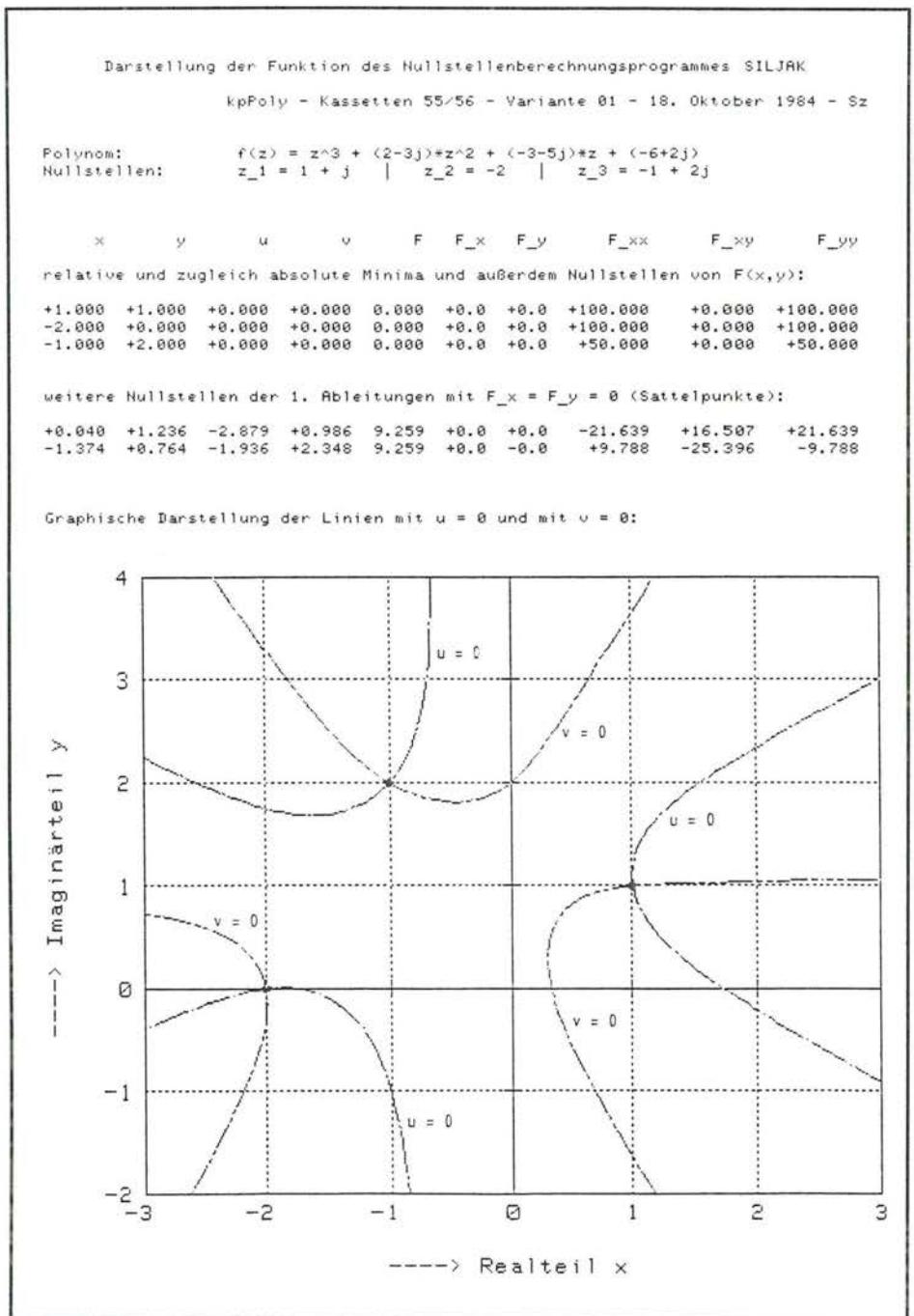


Abb. 1: Linien, bei denen der Realteil  $u$  bzw. der Imaginärteil  $v$  des Polynoms  $f(z)$  null ist.

und speziell mit den Gleichungen (18) und (19)

$$\begin{aligned} x^{(v+1)} &= x^{(v)} - \frac{u \cdot u_x + v \cdot v_x}{(u_x^2 + v_x^2)} \\ y^{(v+1)} &= y^{(v)} + \frac{u \cdot v_x - v \cdot u_x}{(u_x^2 + v_x^2)} \end{aligned} \quad \begin{matrix} x = x^{(v)} \\ y = y^{(v)} \end{matrix} \quad (30)$$

Ist der Wert von  $F(x^{(v+1)}, y^{(v+1)})$ , größer als der zuvor ermittelte Wert, wird die Schritt-

weite von Gl. (30) so lange geviertelt, bis ein kleinerer Wert gefunden ist.

Bevor auf die Berechnung der einzelnen Werte von  $F$ ,  $u_x$  und  $v_x$  eingegangen wird, sei das bisher abgeleitete an einem Beispiel verdeutlicht. Als willkürliches Beispiel dient das Polynom

$$f(z) = z^3 + (2 - 3i)z^2 + (-3 - 5i)z - 6 + 2i \quad (31)$$

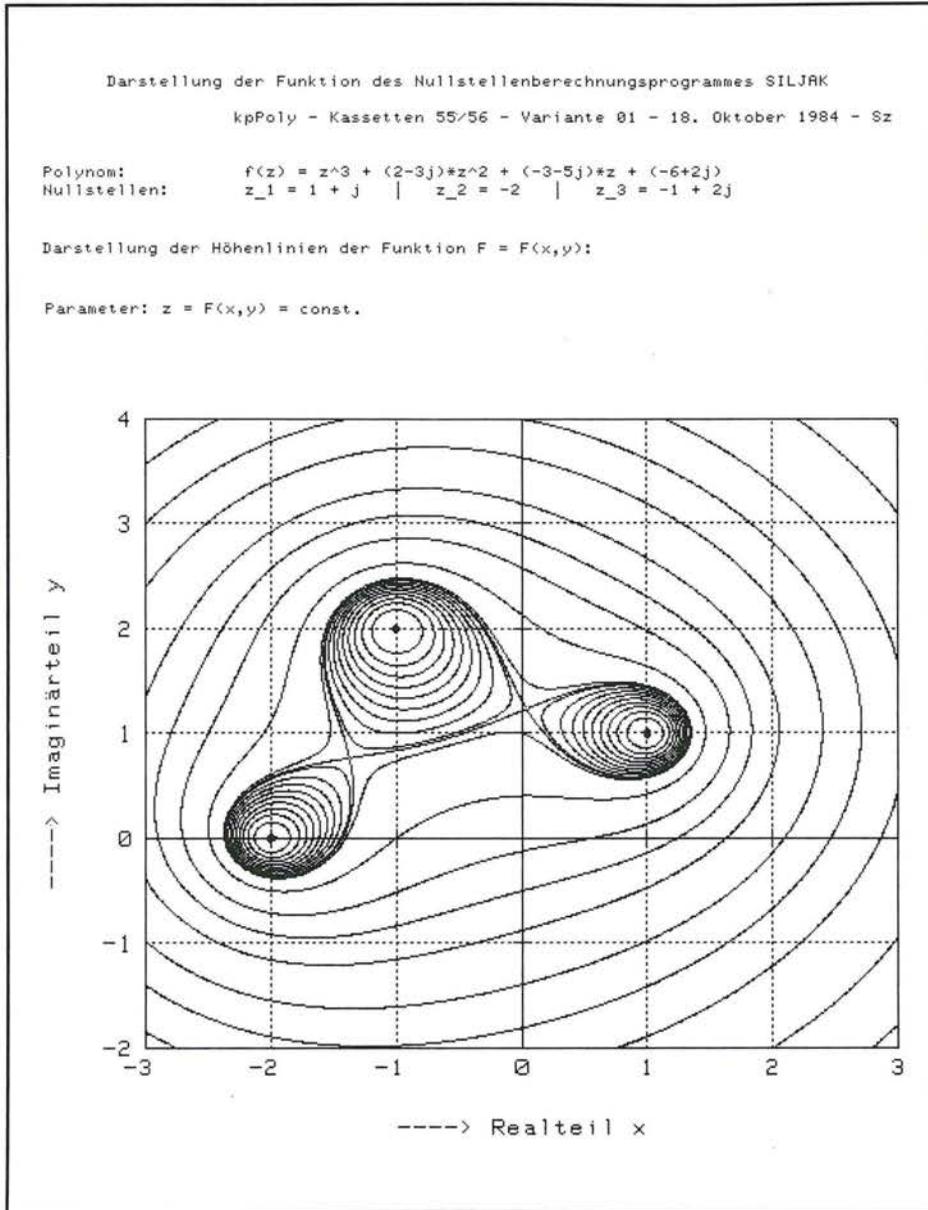


Abb. 2: Höhenlinien der Funktion  $F = F(x, y)$

mit den drei Nullstellen

$$z_1 = 1 + i \quad z_2 = -2 \quad z_3 = -1 + 2i. \quad (32)$$

Für den Realteil von  $f(z)$  gilt

$$u = x^3 + 2x^2 - 3x + 6xy - 3xy^2 + 5y - 2y^2 - 6 \quad (33)$$

und für den Imaginärteil erhält man

$$v = -y^3 + 3y^2 - 3y + 4xy + 3x^2y - 5x - 3x^2 + 2. \quad (34)$$

Abb. 1 zeigt den Verlauf der Linien mit  $u = 0$  und mit  $v = 0$  in der komplexen Ebene. In den Schnittpunkten der Linien liegen die Nullstellen von  $f(z)$ .

Die Ableitungen von  $u$  und  $v$  ergeben sich zu

$$u_x = v_y = 3x^2 + 4x - 3 + 6y - 3y^2 \quad (35)$$

$$u_y = -v_x = 6x - 6xy + 5 - 4y. \quad (36)$$

Während die Lösungen von Gl. (21) bereits bekannt sind, hat die Gl. (22) in dem Beispiel zwei reelle Lösungen für  $x$  und  $y$

$$\begin{aligned} x_1 &= \frac{1}{2}\sqrt{2} - \frac{2}{3} & y_1 &= 1 + \frac{1}{6}\sqrt{2} \\ x_2 &= -\frac{1}{2}\sqrt{2} - \frac{2}{3} & y_2 &= 1 - \frac{1}{6}\sqrt{2}, \end{aligned} \quad (37)$$

d.h. die Sattelpunkte. In beiden Punkten ergibt sich ein Wert von  $F = \frac{250}{27} = 9,259$ . Höhenlinien mit einem kleineren Wert umlaufen die Nullstellen einzeln, solche mit einem größeren Wert umlaufen die Nullstellen gemeinsam (Abb. 2). Schließlich zeigt Abb. 3 den Verlauf der Iteration bei verschiedenen Startwerten  $x^{(0)}, y^{(0)}$ . Man sieht deutlich die Wahl der Schrittichtung, die senkrecht auf den Höhenlinien steht. Liegt der Startwert zufällig auf einem Sattelpunkt, dann versagt das Verfahren.

Zur Berechnung der Werte der ŠILJAK-Funktion

$$z^j = (x + iy)^j = X_j + iY_j \quad (38)$$

werden die Rekursionsformeln (aus [6])

$$\begin{aligned} X_{j+2} &= 2x X_{j+1} - (x^2 + y^2) X_j \\ Y_{j+2} &= 2x Y_{j+1} - (x^2 + y^2) Y_j \end{aligned} \quad (39)$$

verwendet mit den Startwerten

$$X_0 = 1 \quad Y_0 = 0 \quad X_1 = x \quad Y_1 = y. \quad (40)$$

Damit erhält man

$$\begin{aligned} u &= \sum_{j=0}^n (a_j X_j - b_j Y_j) \\ v &= \sum_{j=0}^n (a_j Y_j + b_j X_j) \end{aligned} \quad (41)$$

$$u_x = \frac{\partial u}{\partial x} = \sum_{j=1}^n j(a_j X_{j-1} - b_j Y_{j-1})$$

$$v_x = \frac{\partial v}{\partial x} = \sum_{j=1}^n j(a_j Y_{j-1} + b_j X_{j-1}).$$

Das in Listing 6 wiedergegebene Programm ist so weit kommentiert, daß hier keine weiteren Erläuterungen erfolgen müssen.

## Literatur

- [3] G. JORDAN-ENGELN und F. REUTTER, „Numerische Mathematik für Ingenieure“. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich (1982).
- [4] W. CAUER, „Die Verwirklichung von Wechselstromwiderständen vorgeschriebener Frequenzabhängigkeit“. Archiv Elektrotechnik 17 (1926) 355–388.
- [5] R. ZURMÜHL, „Praktische Mathematik für Ingenieure und Physiker“. Springer-Verlag, Berlin, Heidelberg, New York (1965).
- [6] J. B. MOORE, „A Convergent Algorithm for Solving Polynomial Equations“. J. Assoc. Comp. Machinery 14 (1967) 311–315.

Darstellung der Funktion des Nullstellenberechnungsprogrammes SILJAK

kpPoly - Kassetten 55/56 - Variante 01 - 18. Oktober 1984 - Sz

Polynom:  $f(z) = z^3 + (2-3j)z^2 + (-3-5j)z + (-6+2j)$   
 Nullstellen:  $z_1 = 1 + j$  |  $z_2 = -2$  |  $z_3 = -1 + 2j$

Darstellung der Einzelschritte bei der Nullstellensuche:

Parameter: Startwert der Iteration

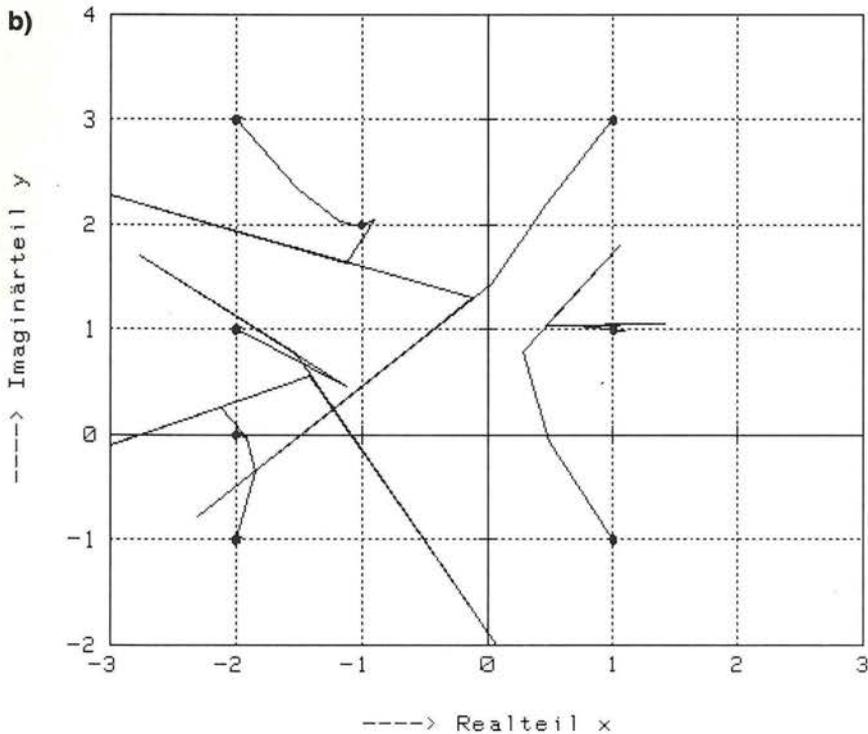
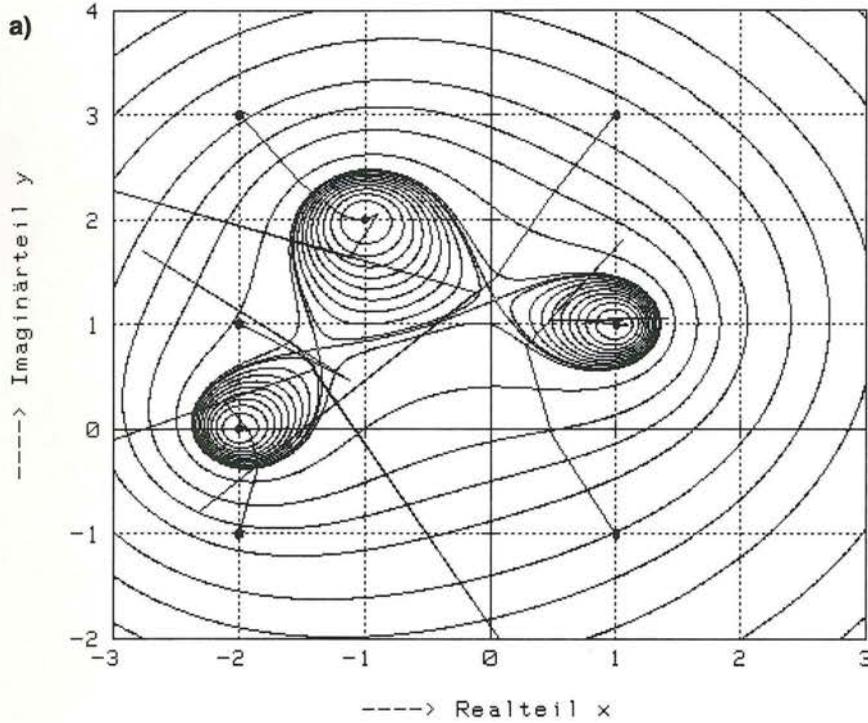


Abb. 3: Verlauf der Iteration bei der Nullstellensuche.

a) Darstellung mit Höhenlinien.

b) Darstellung der einzelnen Schritte.

## Abstract

Jürgen Schwarz

### **45** **Behandlung von Polynomen** **Teil 1**

Gemessene Vorgänge werden vor der Weiterverarbeitung im Computer oft durch geeignete Approximationen ersetzt. Als Ansatz für die Approximationsfunktion dienen oft Polynome oder Exponentialsummen, wobei letztere zweckmäßigerweise in den Bildbereich der Laplace-Transformation transformiert werden. Dabei entstehen ebenfalls Polynome. Dieser Artikel enthält einige Subprogramme zur Behandlung von Polynomen.