KFKI-71-6

Gellai Borbála

POLINOMOK GYÖKEINEK MEGHATÁROZÁSA

Hungarian Academy of Sciences

CENTRAL RESEARCH INSTITUTE FOR PHYSICS

BUDAPEST

KFKI-71-6

POLINOMOK GYÖKEINEK MEGHATÁROZÁSA

Irta
Gellai Borbála
Központi Fizikai Kutató Intézet
Számitástechnikai Osztály

Bevezetés

Az algebrai egyenletek megoldása a numerikus matematika fontos, de sok problémát magába foglaló területe. Fontos, mert a numerikus matematika sok feladata algebrai egyenletek megoldására vezethető vissza, ugyanakkor problematikus is, mert az egyes módszerek kiválasztásánál a legkülönbözőbb szempontok jöhetnek számitásba. Ilyenek például

1. az egyenlet valós vagy komplex együtthatós
2. van-e információ a gyökök elhelyezkedéséről
3. a gyökök kivánt pontossága
4. a módszer konvergenciájának gyorsasága.

Az alábbiakban két módszert ismertetünk, amelyek közül az első valós együtthatós polinomok valós gyökeinek meghatározására szolgál, a második valós együtthatós polinomok valós és komplex gyökeit határozza meg a Bairstow-módszerrel.

I. Valós együtthatós polinomok valós gyökeinek meghatározása a Quotient-Difference módszerrel

Polinomok gyökeinek meghatározására szolgáló ismert módszerek, pl. a Müller-módszer, a Laguerre-módszer, továbbá a Newton-módszer, gyorsan konvergálnak, ha a gyökökre vonatkozólag elég jó kezdeti becslés áll rendelkezésre.

Az itt ismertetésre kerülő Quotient-Difference (Q-D) módszert H. Rutishauser vezette be 1954-ben [1]. A későbbiekben több szerző, közöttük különösen P. Henrici foglalkozott a módszer elméleti [2, 3] és gyakorlati vonatkozásaival egyaránt [4].

E módszer előnye a fent emlitett gyökmeghatározó módszerekkel szemben, hogy a polinom együtthatóiból egyidejüleg számitja ki a polinom valamennyi gyökének kezdőértékeit. Elvileg a módszer alkalmas a gyökök tetszőleges pontossággal való meghatározására, a lassu konvergencia miatt azonban bizonyos lépésszám után a módszer által kapott értékeket kezdőértékekként használhatjuk a gyökök finomitására szolgáló gyorsabb módszerek, pl. a Newton- vagy a Bairstow-módszer alkalmazásánál.

1. A módszer ismertetése

Az algoritmus

Adott az alábbi polinom:

$$p(x) = a_o x^N + a_1 x^{N-1} + \ldots + a_N, \qquad /1/$$

amelynek $a_o, \ldots, a_N$ együtthatói valósak és zérustól különbözőek.

Tekintsük az alábbi táblázatot /N=4 esetre/:

$$\begin{array}{ccccccccc}
 & q_o^{(1)} & & q_o^{(2)} & & q_o^{(3)} & & q_o^{(4)} & \\
e_o^{(o)} & & e_o^{(1)} & & e_o^{(2)} & & e_o^{(3)} & & e_o^{(4)} \\
 & q_1^{(1)} & & \boxed{q_1^{(2)}} & & q_1^{(3)} & & q_1^{(4)} & \\
e_1^{(o)} & & \boxed{e_1^{(1)}} & & \boxed{e_1^{(2)}} & & e_1^{(3)} & & e_1^{(4)} \\
\cdot & q_2^{(1)} & \cdot & q_2^{(2)} & \cdot & q_2^{(3)} & \cdot & q_2^{(4)} & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & \cdot & & \cdot & & \cdot & & \cdot &
\end{array} \qquad /2/$$

A Q-D módszer lényegében a fenti táblázat elemeinek rekurziv módon történő generálása. Például a táblázatban a "bekarikázott" elemet az előző két sor □-el jelölt elemeiből az alábbi rekurziv összefüggéssel számitjuk:

$$q_{n+1}^{(k)} = q_n^{(k)} + \left(e_n^{(k)} - e_n^{(k-1)}\right) \qquad /3/a/$$

$(k = 1,2,\ldots,N;\ n = 0,1,2,\ldots)$.

Az e_n^k elemek generálására szolgáló rekurziv összefüggés az alábbi:

$$e_{n+1}^{(k)} = e_n^{(k)} \frac{q_{n+1}^{(k+1)}}{q_{n+1}^{k}} \qquad /3/b/$$

$(k = 1,2,\ldots, N-1;\ n = 0,1,2,\ldots)$.

A számitáshoz kezdetben ismerni kell az első két sort, valamint az első és utolsó oszlopot. Az első két sor elemeit a polinom együtthatóiból az alábbi hányadosok segitségével kapjuk:

$$q_o^{(1)} = -\frac{a_1}{a_o}, \quad q_o^{(k)} \equiv 0 \quad (k = 2,3,\ldots, N) \qquad /4/$$

$$e_o^{(k)} = \frac{a_{k+1}}{a_k} \quad (k = 1,2,\ldots, N-1). \qquad /5/$$

Az első és utolsó oszlop elemeit zérussal tesszük egyenlővé:

$$e_n^{(o)} \equiv e_n^{(N)} \equiv 0 \quad (n = 0,1,2,\ldots). \qquad /6/$$

Konvergencia-tételek

Jelöljük a polinom gyökeit nagyságszerint csökkenő sorrendben az alábbi módon:

$$|z_1| \geq |z_2| \geq \ldots \geq |z_N|.$$

Fennáll az alábbi két konvergencia-tétel:

a/ Minden olyan "k" indexre, amelyre

$$|z_{k-1}| > |z_k| > |z_{k+1}| \qquad /7/$$

$$\lim_{n\to\infty} q_n^{(k)} = z_k, \quad (k = 1,2, \ldots, N),$$

azaz a /2/ táblázat $q_n^{(k)}$ $(n = 0,1,2,\ldots)$ oszlopa a polinom k-adik gyökéhez konvergál.

b/ Minden olyan "k"-ra, amelyre

$$|z_k| > |z_{k+1}|$$

$$\lim_{n\to\infty} e_n^{(k)} = 0 \quad (k = 1,2,\ldots, N-1), \qquad /8/$$

azaz a táblázat "e oszlopai" zérushoz konvergálnak. Amint a /7/ és /8/ formulákból látható, a módszer csak abszolut értékben különböző gyökök esetében konvergál egyértelmüen a gyökökhöz.

Numerikus stabilitás

A /2/ táblázat jelen módon, tehát sorok szerint történő generálása - az oszlopok szerint történő generálással szemben /lásd [2], 163. old./ -

a kerekitési hibák befolyását a minimálisra csökkenti. Tegyük fel ugyanis, hogy a "q elemek" egy adott sora bizonyos kerekitési hibával terhelt, ekkor a /3/a/ formula alapján számitott "uj q sor" elemeinek abszolut hibái ugyanolyan nagyságrendüek, figyelembevéve a két kerekitett szám összege, ill. különbsége abszolut kerekitési hibájának nagyságára vonatkozó szabályt [2].

Kerekitett számok szorzata, ill. hányadosa relativ hibájára vonatkozó hasonló szabály alapján [2], a /3/b/ formulával generált "uj e sor" relativ hibája az előző "e sor"-éval azonos nagyságrendü.

Itt emlitjük meg, hogy a Q-D algoritmus instabilitásának problémájával kapcsolatban P.Wynn [5] elegendő feltételt állapitott meg.

2. Az ICT számológépre készült eljárás ismertetése és numerikus eredmények

A gyökök meghatározása két fázisban történik:

a/ A Q-D algoritmus segitségével megállapitjuk a gyökök közelitő értékét.

b/ Ezen értékeket kezdőértékekként használva a Newton-módszerrel finomitjuk a gyököket, mindig a polinom eredeti koefficienseit használva.

Az eljárás formális paraméterlistája

Bemenő paraméterek:

degree	a polinom fokszáma
a[0:degree]	tömb, amely a polinom együtthatóit tartalmazza
epszilon	a gyökök kivánt pontosságához szükséges relativ hibakorlát
MAX	a gyökök kezdőértékeinek meghatározásához szükséges "Q-D lépésszám"
maxo	a gyökök finomitásához megengedett maximális "Newton lépésszám"
ALARM	cimke; ha a maxo paraméter értékével meghatározott lépésszám nem elegendő a kivánt pontosság eléréséhez, a vezérlés ezen cimkével jelölt utasitásra történik.

Kimenő paraméterek:

X [1:degree] . . . . tömb, amely a polinom gyökeit tartalmazza

numb [1:degree] . . . . egész tipusu tömb, amelynek rekeszei a megfelelő indexü gyök meghatározásához szükséges tényleges "Newton lépésszámokat" tartalmazzák.

Az 1. táblázatban közöljük néhány polinommal kapcsolatos számolás eredményeit.

Amint az 1. táblázat eredményei mutatják, ha a gyökök jól szeparáltak, a Q-D módszer lépésszámának /MAX formális paraméter/ értékéül elegendő maximálisan 10, s ebben az esetben a Newton-módszer lépésszámát /max paraméter/ is elegendő maximálisan 10-15-nek választani.

Egymáshoz abszolut értékben közel eső vagy egyenlő gyököknél a Q-D módszer nem konvergál kielégitően, az általa szolgáltatott kezdőértékek a Newton-módszer számára nem elég pontosak, ezért az abszolut értékben egyenlő gyököket közel egyenlő abszolut értékü gyökökként szolgáltatja.

Az itt ismertetett eljárással csak valós együtthatós polinomok valós gyökei számithatók, komplex gyökökkel rendelkező polinomok esetén az alábbiakban ismeretett Bairstow-módszert ajánljuk.

II. Valós együtthatós polinomok valós és komplex gyökeinek meghatározása a Bairstow-módszerrel

A módszer egy adott polinom másodfoku tényezőinek meghatározásán alapul.

Adott az alábbi valós együtthatós N-edfoku /$N \geq 2$/ polinom:

$$p(x) = a_o x^N + a_1 x^{N-1} + \ldots + a_N$$

és az $x^2 - ux - v$ másodfoku polinom.

Meghatározandók a $b_o, b_1, \ldots, b_N$ konstansok ugy, hogy az alábbi egyenlőség teljesüljön:

$$p(x) = (x^2 - ux - v)\, q(x) + b_{N-1}(x - u) + b_N \, , \qquad /1/$$

ahol

$$q(x) = b_o x^{N-2} + b_1 x^{N-3} + \ldots + b_{N-2} \; .$$

1. Táblázat.

Probléma	A polinom			Q - D módszer		Newton módszer	
	Fokszáma	Együtthatói	Gyökereinek exakt értékei	-rel kapott közelitő értékek	Lépésszám	-rel kapott finomitott értékek	Lépésszám
1	4	128, -256, 160, -32, 1	0.96194 0.69134 0.30866 0.03806 öt jegyre ismertek	1.017857 0.642337 0.301748 0.038058	5	0.961940 0.691342 0.308658 0.038060	4 3 2 1
2	3	1, 1.0004,-1.0002,-1.0006	-1.00050 1.00010 -1.00000 /öt jegyre ismertek/	-1.000400 0.833417 -0.833416	10	-1.000500 1.000100 -1.000000	3 4 11
3	4	1, - 3, - 12, 52, - 48	-4, 3, 2, 2	-3.678691 2.679985 2.102135 1.896571	20	-4.000000 3.000000 1.999990 1.999990	3 16 25 17
4	3	1, 8, 21, 18	-3, -3, -2	-3.157870 -2.843535 -1.998594	20	-3.000022 -3.000012 -2.000000	13 24 2
5	4	1, -6, 9, 4, -12	3, 2, 2, -1	3.001329 2.102099 1.896409 -0.999837	20	3.000000 1.999992 1.999990 -1.000000	2 18 14 1
6	4	1, 7, 13, -3, -18	-3, -3, -2, 1	-3.155823 -2.845441 -1.998713 0.999977	20	-3.000014 -3.000016 -2.000000 1.000000	19 16 2 1
7	4	1, -5, 6, 4, -8	2, 2, 2, -1	2.187587 1.991433 1.820236 -0.999257	20	1.999729 1.999535 1.999645 -1.000000	31 9 15 1

Az /1/ összefüggés jobboldalán elvégezve a szorzást, a megfelelő x hatványok együtthatóinak összehasonlításával a b_i $(i = 0,1,\ldots, N)$ mennyiségekre az alábbi rekurziv összefüggést kapjuk:

$$b_0 = a_0\,,$$
$$b_1 = a_1 + ub_0$$
$$b_2 = a_2 + ub_1 + vb_0$$
$$\vdots$$
$$b_N = a_N + ub_{N-1} + vb_{N-2} \quad (b_{-1} = b_{-2} = 0). \qquad /2/$$

Az igy kapott b_i $(i = 0,1,\ldots, N)$ koefficiensek természetesen függvényei az u és v változóknak.

Bebizonyitható a következő tétel [2]: Az $x^2 - ux - v$ polinom akkor és csak akkor másodfoku tényezője a $p(x) = a_0x^N + a_1x^{N-1} + \ldots + a_N$ polinomnak, ha $b_{N-1} = b_N = 0$.

A fenti tételből tehát következik, hogy a p(x) polinom másodfoku tényezőinek meghatározása ekvivalens a következő feladattal: határozzuk meg az u és v mennyiségeket ugy, hogy az alábbi két egyenlet egyidejüleg teljesüljön:

$$b_{N-1}(u, v) = 0$$
$$b_N(u, v) = 0 \quad . \qquad /3/$$

Ilyen szimultán egyenletpárok megoldása a Newton-módszerrel Bairstow-módszer néven ismeretes.

A Bairstow-módszer

Alkalmazva a Newton-módszert a /3/ egyenletrendszerre ki kell számitani a

$$\frac{\partial b_{N-1}(u,v)}{\partial u}\,, \quad \frac{\partial b_{N-1}(u,v)}{\partial u}$$
$$\frac{\partial b_N(u,v)}{\partial u}\,, \quad \frac{\partial b_N(u,v)}{\partial v} \qquad /4/$$

parciális deriváltakat. Ezekre a deriváltakra rekurziv összefüggést kapunk a /2/ rekurziv összefüggés u, ill. v szerinti parciális deriválásával.

Alkalmazva a

$$c_n = \frac{\partial b_{n+1}}{\partial u} \qquad /5/$$

jelölést, az u szerinti deriváltakra kapjuk az alábbi összefüggést /figyelembe véve, hogy $\frac{\partial b_o}{\partial u} = 0$/.

$$c_o = b_o ,$$
$$c_1 = b_1 + uc_o ,$$
$$c_2 = b_2 + uc_1 + vc_o ,$$
$$\vdots$$
$$c_n = b_n + uc_{n-1} + vc_{n-2} \qquad \begin{pmatrix} n = 0,1,2,\dots N-1 \\ c_{-2} = c_{-1} = 0 \end{pmatrix}.$$

A v szerinti deriváltra hasonló jelölést bevezetve:

$$d_n = \frac{\partial b_{n+2}}{\partial v} \qquad (\partial b_o/\partial v = \partial b_1/\partial v = 0) \qquad /6/$$

kapjuk:

$$d_o = b_o ,$$
$$d_1 = b_1 + ud_o ,$$
$$d_2 = b_2 + ud_1 + vd_o ,$$
$$\vdots$$
$$d_n = b_n + ud_{n-1} + vd_{n-2} ,$$

$$n = 0,1,2,\dots, N-2 \text{ és } d_{-2} = d_{-1} = 0 .$$

Az /5/ és /6/ alapján a /4/ parciális deriváltakra kapjuk:

$$\frac{\partial b_{N-1}}{\partial u} = c_{N-2} , \quad \frac{\partial b_N}{\partial u} = c_{N-1} ,$$

$$\frac{\partial b_{N-1}}{\partial v} = c_{N-3} , \quad \frac{\partial b_N}{\partial v} = c_{N-2} .$$

Ha az u és v mennyiségek növekményeit δ-val, ill. ε-al jelöljük, ezek a mennyiségek a Newton-módszer alapján kielégitik az alábbi egyenlőségeket:

$$c_{N-2}\delta + c_{N-3}\varepsilon = - b_{N-1}$$

$$c_{N-1}\delta + c_{N-2}\varepsilon = - b_N ,$$

innen átrendezés után ε-ra és δ-ra kapjuk:

$$\delta = \frac{b_N c_{N-3} - b_{N-1} c_{N-2}}{c_{N-2}^2 - c_{N-1} c_{N-3}} , \quad \varepsilon = \frac{b_N c_{N-1} - b_N c_{N-2}}{c_{N-2}^2 - c_{N-1} c_{N-3}} . \qquad /7/$$

Összefoglalva:

Ha adott a

$$p(x) = a_o x^N + a_1 x^{N-1} + \ldots + a_N$$

polinom és az $x^2 - u_o x - v_o$ tetszésszerinti másodrendü faktor, akkor a Bairstow-módszer alkalmazása lényegében az alábbi algoritmussal ekvivalens: meghatározandó az

$$\{x^2 - u_k x - v_k\}$$

másodrendü faktorok sorozata olymódon, hogy minden egyes $k = 0,1,2,\ldots$ értékre meg kell határozni a $\{b_n\} = \{b_n^{(k)}\}$ sorozatot a

$$b_n = a_n + u_k b_{n-1} + v_k b_{n-2}$$

$$n = 0,1,2,\ldots, N \quad (b_{-2} = b_{-1} = 0)$$

rekurziv összefüggés alapján, majd a $\{c_n\} = \{c_n^k\}$ sorozatot a

$$c_n = b_n + u_k c_{n-1} + v_k c_{n-2} , \quad n = 0,1,2,\ldots, N-1$$

$(c_{-2} = c_{-1} = 0)$ összefüggésből.

Ekkor

$$u_{k+1} = u_k + \delta , \quad v_{k+1} = v_k + \varepsilon ,$$

ahol δ és ε a /7/ összefüggés alapján számitott mennyiségek.

Ha $z_1 = z_2$, azaz a másodrendü faktor két gyöke egybeesik /többszörös gyök van/, akkor $p(z_1) = p'(z_1) = 0$, s ekkor /1/ alapján

$$b_{N-1}(z_1 - u) + b_N = 0$$

$$b_{N-1} \qquad\qquad = 0$$

vagyis $b_{N-1} = b_N = 0$. A 6. oldalon emlitett tétel állitása tehát ebben az esetben is érvényes. A Bairstow-módszer ennek alapján többszörös gyökök esetében is jó eredményt ad.

A fent ismertetett módszer alapján készült a következő gépi eljárás, amely valós együtthatós polinomok valós és komplex gyökpárjainak meghatározására szolgál.

Az eljárás formális paraméterlistája

Bemenő paraméterek:

n	a polinom fokszáma
a [0:n]	tömb, amely a polinom együtthatóit tartalmazza
eps	a gyökök pontosságához előirt hibakorlát
K	a gyökök meghatározásához megengedett maximális iterációs lépésszám
ALARM	cimke, ha az iterációs lépésszám a kivánt pontosság elérése előtt eléri K paraméter értékét, a vezérlés ezen cimkével jelölt utasitásra tér át.

Kimenő paraméterek:

x[1:n2], y[1:n2] ahol n2=entier((n+1)/2)	valós tömbök, amelyek komplex gyökpárok esetén a gyökök valós, ill. képzetes részeit tartalmazzák, valós gyökpárok esetén pedig egy adott másodfoku tényező valós gyökeit.
nat[1:n2]	egész tipusu tömb, amelynek rekeszeiben + 1, ill. -1 van, attól függően, hogy a megfelelő indexü gyökpár valós, ill. konjugált komplex gyökpár
m	egész tipusu változó, ha valamelyik gyökpár meghatározásánál a K paraméter a kivánt pontosság előtt elérné maximális értékét, a változó értéke e gyökpár sorszámával egyenlő.

Megjegyzés:

Minthogy a fent ismertetett eljárás gyökpárokat határoz meg, páratlan fokszámu polinom esetén egy zérus gyököt is kapunk.

```
'procedure' realpoly(degree,a,epszilon,MAX,max0,numb,X,ALARM);
'value' degree,epszilon,MAX,max0; 'integer' degree,
MAX,max0; 'label' ALARM;
    'real' epszilon; 'array' a,X;  'integer' 'array' numb;
    'begin' 'integer' n,k,l,rootcount,index,max;
       'real' qmax,x;
       'array' q0,q[1:degree],e0,e,beta[0:degree];
       'procedure' newton(N,r,eps,xo,x,max);
               'value' N,eps;
               'integer' N,max;  'real' eps,xo,x;
               'array' c;
                 'begin'
                  'integer' j,count;
                  'array' alfa,gamma[0:N];
        'comment' eljarastorzs kezdete;
               count:=0;
     A2:alfa[0]:=gamma[0]:=c[0];
          'for' j:=1 'step'    'until' N 'do' 'begin'
         alfa[j]:=c[j]+xo*alfa[j-1];
          gamma[j]:=alfa[j]+xo*gamma[j-1];
        'end';
               x:=xo-alfa[N]/gamma[N-1];
             'if' abs(x-xo)>eps*abs(xo) 'then' 'begin'
               xo:=x;  count:=count+1;
      'if' count=max 'then' 'goto' ALARM; 'goto' A2;
    'end';  max:=count;
   'end' NEWTON;
  'comment' itt kezdodik a realpoly eljarastorzse;
               rootcount:=0;   n:=degree;
         'for' k:=0 'step' 1 'until' n 'do'
  beta[k]:=a[k];
 q0[1]:=-(beta[1]/beta[0]);
         'for' k:=2 'step' 1 'until' n 'do'
q0[k]:=0;
         'for' k:=1 'step' 1 'until' n-1 'do'
e0[k]:=beta[k+1]/beta[k];
e0[0]:=e0[n]:=0;
 'for' l:=   'step' 1 'until' MAX 'do' 'begin'
  e[0]:=e[n]:=0;
           'for' k:=1 'step' 1 'until' n 'do' 'begin'
  q[k]:=q0[k]+(e0[k]-e0[k-1]);
q0[k]:=q[k];
   'end';
'for' k:=1 'step' 1 'until' n-1 'do' 'begin'
e[k]:=e0[k]*q[k+1]/q[k];
e0[k]:=e[k] 'end';
'end' l;
LAB1:    qmax:=0;
         'for' k:=1 'step' 1 'until' n 'do' 'begin'
'if' abs(q[k])>abs(qmax) 'then' 'begin'
  qmax:=q[k];index:=k 'end';
        'end';        max:=max0;

 newton(degree,a,epszilon,qmax,x,max);
               rootcount:=rootcount+1;
                  X[rootcount]:=x;     numb[rootco      :=max;
              'if' rootcount=degree 'then' 'goto'      ;
q[index]:=0; 'goto' LAB1;
LAB2:      'end' REALPOLY;
```

```
'procedure' bairstow(n,a,eps,K,x,y,nat,m,ALARM);
'value' n,eps,K; 'integer' n,m,K;
'label' ALARM; 'integer' 'array' nat;
'real' eps; 'array' a, x,y;
      'begin'
 'integer'i,j,k,n1,n2,m1;
          'real'r0,s0,v0,det0,r1,s1,v1,det1,det2,p,q,
               incrp,incrq,S,T;
        'array'b,c[0:n+1];
          'for'i:=0 'step' 1 'until' n 'do'
               b[i]:=a[i]; b[n+1]:=0;
               n2:=entier((n+1)/2);
               n1:=2*n2;
          'for'm1:=1 'step' 1 'until' n2 'do''begin'
               p:=q:=0;
          'for'k:= 1 'step' 1 'until' K 'do''begin'
         step:'for'i:=0 'step' 1 'until' n1 'do'
               c[i]:=b[i];
          'for'j:=n1-2,n1-4 'do''begin'
          'for'i:=0 'step' 1 'until' j 'do''begin'
               c[i+1]:=c[i+1]-p*c[i];
               c[i+2]:=c[i+2]-q*c[i]'end''end';
         r0:=c[n1]; r1:=c[n1-1];
         s0:=c[n1-2]; s1:=c[n1-3];
         v0:=-q*s1; v1:=s0-s1*p;
         det0:=v1*s0-v0*s ;
'if' abs(det0)<&-10 'then' 'begin'
         p:=p+1; q:=q+1; 'goto' step 'end';
         det1:=s0*r1-s1*r0;
         det2:=r0*v1-v0*r ;
         incrp:=det1/det0; incrq:=det2/det0;
              p:=p+incrp; q:=q+incrq;
'if' abs(r0)<eps'then''begin'
'if'abs(r1)<eps 'then''begin'
'goto'next'end''end';
'if'abs(incrp)<eps 'then' 'begin'
'if' abs(incrq)<eps 'then' 'begin'
'goto' next 'end' 'end';
'if' abs(incrp/p)<eps 'then' 'begin'
'if' abs(incrq/q)<eps 'then' 'begin'
'goto' next 'end' 'end' 'end' ;
m:=m1; 'goto' ALARM;
next:S:=-p/2; T:=S↑2-q;
              'if' T 'ge' 0 'then'
             'begin'
               T:=sqrt(T);
 x[m1]:=S+T;y[m1]:=S-T;
nat[m1]:=1;
     'end';
         'if' T<0 'then'
 'begin' x[m1]:=S; y[m1]:=sqrt(-T); nat[m1]:=-1;
'end';
        'for' j:=0 'step' 1 'until' n1-2 'do''begin'
           b[j+1]:=b[j+1]-p*b[j];
           b[j+2]:=b[j+2]-q*b[j]  'end';
 n1:=n1-2; 'if' n1<  'then' 'begin'
     m:=m1; 'goto' out 'end';
 'if' n1<3 'then' 'begin'
m1:=m1+1;
         p:=b[1]/b[0];  q:=b[2]/b[0];
 'goto' next 'end';
'end' m1;
out:    'end'BAIRSTOW;
```

Numerikus tapasztalat

Az eljárás általában, eltekintve az igen rosszul kondicionált polinomoktól, mindig jó eredményt adott. Rosszul kondicionált polinom volt az eljárás számára egy olyan polinom, amelynél $a_o = 1250162561$ és $a_N = 2$, tehát a polinom koefficiensei között nagyságrendben igen különbözők voltak. Az eljárás itt teljesen hamis eredményt adott.

A megoldott legmagasabb fokszámu polinom 19-edfoku volt, a maximális pontosság eps = 10^{-10}, s ebben az esetben is 50 lépésen belül szolgáltatta a gyököket /K = 50 volt/.

I r o d a l o m

[1] Rutishauser, H.: Der quotient-differenzen algorithmus. Z.f.angew.Math.Phys. 5, 233-251 /1954/

[2] Henrici, P.: Elements of Numerical Analysis. John Wiley and Sons, New York, 1964.

[3] Henrici, P.: The quotient-difference algorithm. Nath.Bur.Standards Applied Math.Ser. 49. US Goverment Printing Off. Washington, D.C. 1958. 23-46.

[4] Henrici, P., Watkins, O.B.: Finding Zeros of Polynomial by the Q-D Algorithm. Com. of the ACM vol. 8. No. 9. 570-574 /1965/

KFKI report-71-6.
Érkezett: 1971. január 26.

Központi Fizikai Kutató Intézet
Könyvtár és Kiadói Osztály

O.v.: Dr. Farkas Istvánné
Szakmai lektor: Zimányi Józsefné
Példányszám: 136 Munkaszám: 5394
Készült a KFKI házi sokszorositójában
F.v.: Gyenes Imre
Budapest, 1971. február hó