数 値 解 析 17－6
（1986. 7. 4）

実係数多項式用のDurand－Kerner法

安藤 茂
津田塾大学

Durand-Kerner 法の拡張として、多項式を2次と1次の因子に因数分解するアルゴリズムを提案する。1次因子への分解は、実係数の場合でも複素計算が必要であるが、2次因子への分解ならば実数計算のみで実行できるので実係数問題の解法として用いるのが目的である。多項式の因数分解は因子の係数に関する連立方程式をとくことと考えられるので、Durand-Kerner 法ににた、Newton 反復がもちいられる。
出発値は、Aberth のものを若干修正したものを提案する。かんたんな PASCAL プログラムと実行例をも紹介する。

Durand-Kerner Method For Real-Coefficient Polynomials.

Shigeru Ando
(Tsuda college, Kodaira-si, 187, Japan)

A numerical algorithm for factorization of polynomials into linear and quadratic factors is set forth as a simple generalyzation of Durand-Kerner method. Whereas linear polynomial factorization is only possible in the complex field even if the coefficients are real, quadratic factorization of real polynomials can be performed within the real field. Since polynomial factorization can be thought of as solving system of equations of the factor coefficients, Durand-Kerner-like Newtonian iteration is adopted.
A choice of modified-Aberth's starting values is proposed and experimental results with a simple PASCAL coding is reported.

1. はじめに

多項式の1次因子への分解をNewton反復でおこなおうというのがDurand-Kerner法である．これには実係数であっても複素計算が必要であるが，これに対し，2次因子への分解ならば実数計算でできるではないかとゆうのが，われわれの発想である．マイコンで，BASIC や PASCAL でコーディングしようとするとき，複素計算の道具をいちいち用意するのはあんまりうれしくない．とゆうのがそもそもの動機である。

2. アルゴリズム

与えられた多項式 $F(x)$ に対し，

$$F(x) = \prod_{i=1}^{\ell} F_i(x) \qquad \cdots\cdots \ (1)$$

をみたすような，各 $F_i(x)$ の係数をもとめることを考える。（$F(x)$, $F_i(x)$ はすべて monic にとることにする。） これを，各 $F_i(x)$ の係数に関する連立方程式とみて Newton 法をもちいるわけであるが，その定式化は，多項式計算のかたちのままおこなうことにする。すなわち，

いま，$F_1(x), \cdots, F_\ell(x)$ が近似的に (1) をみたしているとして，

$$G(x) = F(x) - \prod_{i=1}^{\ell} F_i(x) \qquad \cdots \ (2)$$ とおく。

ここで $$G(x) + dG(x) = 0 \qquad \cdots \ (3)$$

となるような $dF_i(x)$ $(i=1,\cdots,\ell)$ をもとめ，$F_i^*(x) = F_i(x) + dF_i(x)$ を新しい $F(x)$ とする… とゆうのが，ここでの Newton 法である。

積の微分により (3) は

$$F(x) - \prod_{i=1}^{\ell} F_i(x) - \sum_{i=1}^{\ell} \prod_{j\neq i} F_j(x)\, dF_i(x) \qquad \cdots \ (4)$$ となる。

$F_i(x)$ $i=1,\ldots,\ell$ が互いに共通因子をもたないとき，(4) は，実は，各 $F_i(x)$ を法として成立てばよい。mod $F_i(x)$ で (4) は，

$$F(x) - \prod_{j\neq i} F_j(x)\, dF_i(x) \equiv 0 \qquad (\mathrm{mod}\ F_i(x))$$ となる。

$F_i(x)$ が1次式 $x-t$ のときは，

$$F(x) \equiv F(t), \quad F_j(x) \equiv F_j(t), \quad dF_i(x) = -dt \qquad (\mathrm{mod.}\ F_i(x))$$

より (4) は $$F(t) + \prod_{j\neq i} F_j(t)\, dt_i \equiv 0 \qquad (\mathrm{mod}\ F_i(x))$$ となるが，両辺は0次式であるから，"$\equiv$ mod $F_i(x)$" は "$=$" となり，新しい t は，

$$t + dt = t - F(t) \Big/ \prod_{j\neq i} F_j(t) \qquad \cdots\cdots \ (5)$$

で求められる。もしも他の $F_j(x)$ もすべて1次式であるならこれはDK法にほかならない。

次に，$F_i(x)$ が2次因子 $x^2 + px + q$ の場合をかんがえる。

$F(x) \equiv ax+b$,　$F_j(x) \equiv r_j x + s_j$　$(\mathrm{mod.}\ x^2+px+q)$
とするとき (5) は

$$ax+b \equiv \prod_{j \neq i} (r_j x + s_j)(dp\, x + dq) \quad \cdots (6)$$

となり、これを dp, dq についてとけばよい。容易にたしかめられるように、
方程式、$ax+b \equiv (rx+s)(a^* x + b^*)$　$(\mathrm{mod}\ x^2+px+q)$
は

$$\begin{cases} a^* = (as-br)/D \\ b^* = (arq + (s-rp)b)/D, \end{cases} \qquad D = s^2 - qrs + r^2$$

によってとかれる。
(6) における各 r_j, s_j を r, s として、つぎつぎと、とうきびのかわをめくるようにといてゆけば、最終的に dp, dq がもとめられることがわかる。
この過程で $D=0$ となるのは、$F_i(x)$ と $F_j(x)$ が共通因子をもつときである。

以上のようにして、任意の次数の因子に対する Newton 反復が構成されることがわかるが、われわれはここで、実係数多項式の実2次因子分解にもんだいをしぼる。すなわち、n 次実多項式 $F(x) = x^n + C_{n-1}x^{n-1} + \cdots + C_1 x + C_0$ に対し

$n=2m$ のとき　$F(x) = \prod_{i=1}^{m} (x^2+p_i x+q_i)$ を

$n=2m+1$ のとき　$F(x) = (x-t)\prod_{i=1}^{m}(x^2+p_i x+q_i)$ をとくのである。

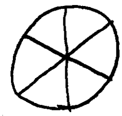
3. 出発値のえらびかた。
出発値はさまざまに考えられるが、われわれはここでは Aberth の出発値をもとに考えることにする。そのもとで以下をそのまま踏襲することにする。

1. 根の和（重心）が 0 となるように変数変換 $x = \frac{1}{n}C_{n-1} + \tilde{x}$ をほどこす。 i.e. $F(x) = F(\frac{1}{n}C_{n-1} + \tilde{x}) = \tilde{F}(\tilde{x}) = \sum_{i=1}^{n} \tilde{C}_i \tilde{x}^i$
（以後 $\tilde{F}, \tilde{C}_i, \tilde{x}$ をあらためて、それぞれ F, C_i, x とかく。）
2. $x^n - \sum_{i=1}^{n} |a_i| x^i$ の唯一の正根 r を Newton 法で求める。
（r は、根の絶対値の bound であることが知られている。）
3. 原点中心、半径 r の円周上に、出発値を等間かくに分布させる。

DKA 法の場合、複素共役ペアをさけるいみで、実軸に関して非対称に分布させる。これは、実根の場合に生ずる特異性（共役ペア同士で話しあいがつかない）をさけるためである。しかしわれわれのばあい、1.）2次因子でつかまえるため2実根をわける必要がない。2.）もともと実2次因子であつかうため非対称にしようと思ってもできない。などの理由により、実軸に関し対称にとる。（共役ペア $r\exp(\pm i\theta)$ に対応する2次因子は $x^2 - 2r\cos\theta \cdot x + r^2$ ）
容易にわかる理由により、われわれは以下の出発値を採用した。（$F(0)$ の符号により、正根、負根の個数の偶奇がきまるというのがポイント。）

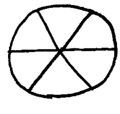
1. $n=2m$, $F(0)>0$ のとき

$$F_i(x)=x^2-2r\cos\left(\frac{2i-1}{n}\pi\right)x+r^2 \quad (i=1,\dots,m)$$



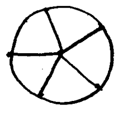
2. $n=2m$ $F(0)<0$ のとき

$$F_i(x)=\begin{cases} x^2-2r\cos\dfrac{2i\pi}{n}x+r^2 & (i=1,\dots,m-1) \\ x^2-r^2 & (i=m) \end{cases}$$



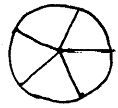
3. $n=2m+1$ $F(0)>0$ のとき

$$F_i(x)=\begin{cases} x^2-2r\cos\left(\dfrac{2i-1}{n}\pi\right)x+r^2 & (i=1,\dots,m) \\ x+r & (i=m+1) \end{cases}$$



4. $n=2m+1$ $F(0)<0$ のとき

$$F_i(x)=\begin{cases} x^2-2r\cos\dfrac{2i\pi}{n}x+r^2 & (i=1,\dots,m) \\ x-r & (i=m+1) \end{cases}$$



4. プログラムと実行例

機械はPC9801F, i8087付, 言語はTurbo PASCAL ver 3.0, 8087用。マイコンではポピュラーな世界である。

各部分の説明

① 配列 C は多項式の係数 $C_i=C[n-i]$, 配列 V は因子 $F_i(x)$ の係数をいれる。 $p_i=V[2*i-1]$, $q_i=V[2*i]$. 奇数次の場合 $t=V[n]$

② 因子係数配列 $V[i]$ を $V[i]+dV[i]$ におきかえる procedure

③ $ax+b\equiv(rx+s)(a^*x+b^*)$ (mod. x^2+px+q) をといて, $a\leftarrow a^*$, $b\leftarrow b^*$

④ $F(x)\equiv ax+b$ (mod. x^2+px+q) なる a, b を求める

⑤ $x^2+p_jx+q_j\equiv(p_j-p)x+(q_j-q)$ (mod. x^2+px+q)

⑥ 奇数次の場合 因子 $x-t$ に対する処理.

⑦ 多項式打ち出し procedure

⑧ 多項式の係数読込み

⑨ 変数変換 $x=\tilde{x}+C_{n-1}/n$ をおこなって根の重心が0にくるように。

⑩ $2\max|C_{n-i}|^{1/i}$ を出発値として, $x^n-|C_{n-1}|x^{n-1}-\dots-|C_1|x-|C_0|$ の正根 r をもとめる。

⑪ 原点中心 半径 r の円周上に根の出発値を分布させ、それに対応する、各 $F_i(x)$ の係数を配列 V にいれる. 詳細は前節.

⑫ realDK をよびながら, iteration をおこなう。収束判定は組みこんでいない。1回ごとにキー入力応答を待ち、ESC キーが入力されると終る. 画面で収束状況をみながら人間が判定する.

⑬ 結果の各因子から, 2次方程式をとくなどして 根を計算, ⑨ での変数変換のもどしをして出力.

その他. プリンタのコントロールコマンド (上下そえ字等) は機種により異るので注意, ここでは NEC NM9300S 用.

```
    type vector=array[1..80]of real;
1   var c,v:vector;i,j,k,n:byte; p,q,d,r,s,t:real; inc:char;
2   procedure realDK(var c,v:vector;n:integer);
      var i,j:integer; dv:vector; a,b,p,q,d,t:real;
3     procedure solve(r,s:real); begin
        d:=s*s-p*r*s+q*r*r; t:=(a*s-b*r)/d; b:=(a*r*q+(s-r*p)*b)/d; a:=t  end;
      begin
      for i:=1 to n div 2 do begin p:=v[2*i-1]; q:=v[2*i]; a:=1; b:=c[1];
4       for j:=2 to n do begin t:=b-a*p;b:=c[j]-a*q; a:=t end;
        for j:=1 to n div 2 do if j<>i then solve(v[2*j-1]-p, v[2*j]-q);
        if odd(n) then solve(1,-v[n]);      5
        dv[2*i-1]:=a; dv[2*i]:=b                                   end;
6     if odd(n) then begin t:=v[n]; a:=1; for j:=1 to n do a:=a*t+c[j];
        for j:=1 to n div 2 do a:=a/(t*t+v[2*j-1]*t+v[2*j]); dv[n]:=-a end;
      for i:=1 to n do v[i]:=v[i]+dv[i]                               end;

7   procedure printpol(var c:vector;n:integer);var i:integer; sgn:char;
      procedure xx(i:integer);begin
        write(lst,' X');if i>1 then write(lst,^Z'U',i,^['Q ') end;
      begin  xx(n);for i:=1 to n do if abs(c[i])>1E-10 then begin
        if c[i]>0 then sgn:='+' else sgn:='-';write(lst,sgn,abs(c[i]):7:3);
        if i<n then xx(n-i)end;writeln(lst) end;
    Begin
    writeln(lst,^J^['Q');write('n = ');readln(n);
8   for i:=1 to n do begin write('c',n-i,' = ');readln(c[i]) end; printpol(c,n);
9   s:=-c[1]/n;if s<>0 then begin for i:=n downto 1 do begin p:=1;
      for j:=1 to i do begin p:=c[j]+p*s; c[j]:=p end end; printpol(c,n) end;
10  r:=0;for i:=1 to n do if c[i]<>0 then
      begin t:=2*exp(ln(abs(c[i]))/i);if t>r then r:=t end;
    repeat write(lst,r:8:3);p:=-abs(c[1])+r;q:=1; i:=1;
      while i<n do begin q:=p+q*r;i:=i+1;p:=-abs(c[i])+p*r end;
      r:=r-p/q until p/q<1e-8;    writeln(lst,^J);
11  t:=3.14159265357979*2/n; k:=0;
    if c[n]>0 then begin if odd(n) then v[n]:=-r;
      for i:=1 to n div 2 do begin v[2*i-1]:=-2*r*cos(t*i-t/2);v[2*i]:=r*r end end
    else begin if odd(n) then v[n]:=r else begin v[n-1]:=0; v[n]:=-r*r end;
      for i:=1 to (n-1)div 2 do begin v[2*i-1]:=-2*r*cos(t*i);v[2*i]:=r*r end end;
12  repeat k:=k+1;write(lst,k:3,' ');
      for i:=1 to n do write(lst,v[i]:8:3);writeln(lst);
      realDK(c,v,n); read(kbd,inc) until inc=^[;
    if s<>0 then writeln(lst,'C'^Z'L',n-1,^['Q/',n,' = ',s:10:6);
13  for i:=1 to n div 2 do begin p:=v[2*i-1]; q:=v[2*i]; d:=p*p-4*q;
      if d<0 then writeln(lst,s-p/2:15:10,' ±'^['Q',sqrt(-d)/2:15:10,' i')
      else begin d:=sqrt(d);if p>0 then d:=-d;t:=(-p+d)/2;
                  writeln(lst,s+t:15:10,s+q/t:19:10) end  end;
    if odd(n) then writeln(lst,s+v[n]:15:10);
    End.
```

例1 $F(x)=(x^{14}+1)(x^2-0.01)$

X^{16} - 0.010 X^{14} + 1.000 X^2 - 0.010

2.000 1.875 1.758 1.648 1.545 1.449 1.359 1.275 1.198 1.128 1.070 1.027 1.006 1.002 1.001 1.001

1	-1.850	1.003	-1.416	1.003	-0.766	1.003	-0.000	1.003	0.766	1.003	1.416	1.003	1.850	1.003	0.000	-1.003
2	-1.690	0.793	-1.416	0.879	-0.833	0.965	-0.000	1.000	0.833	0.965	1.416	0.879	1.690	0.793	0.000	-0.757
3	-1.721	0.673	-1.603	0.972	-0.873	0.999	-0.000	1.000	0.873	0.999	1.603	0.972	1.721	0.673	0.000	-0.038
4	-2.092	1.186	-1.531	0.957	-0.867	0.999	0.000	1.000	0.867	0.999	1.531	0.957	2.092	1.186	-0.000	0.034
5	-1.960	1.014	-1.565	0.995	-0.868	1.000	0.000	1.000	0.868	1.000	1.565	0.995	1.960	1.014	-0.000	-0.001
6	-1.950	1.000	-1.564	1.000	-0.868	1.000	-0.000	1.000	0.868	1.000	1.564	1.000	1.950	1.000	-0.000	-0.010
7	-1.950	1.000	-1.564	1.000	-0.868	1.000	-0.000	1.000	0.868	1.000	1.564	1.000	1.950	1.000	-0.000	-0.010

```
  0.9749279122 ±  0.2225209340 i
  0.7818314825 ±  0.6234898019 i
  0.4338837391 ±  0.9009688679 i
-4.7411597E-031 ± 1.0000000000 i
 -0.4338837391 ±  0.9009688679 i
 -0.7818314825 ±  0.6234898019 i
 -0.9749279122 ±  0.2225209340 i
  0.1000000000    -0.1000000000
```

例2　$F(x)=(x^{14}-1)(x^2-0.01)$

X^{16} - 0.010 X^{14} - 1.000 X^2 + 0.010

2.000 1.875 1.758 1.648 1.545 1.449 1.359 1.275 1.198 1.128 1.070 1.027 1.006 1.002 1.001 1.001

1	-1.964	1.003	-1.665	1.003	-1.113	1.003	-0.391	1.003	0.391	1.003	1.113	1.003	1.665	1.003	1.964	1.003
2	-1.742	0.766	-1.587	0.832	-1.165	0.925	-0.435	0.991	0.435	0.991	1.165	0.925	1.587	0.832	1.742	0.766
3	-1.266	0.286	-1.789	0.881	-1.267	0.993	-0.445	0.999	0.445	0.999	1.267	0.993	1.789	0.881	1.266	0.286
4	-1.147	0.121	-1.796	1.017	-1.247	0.997	-0.445	1.000	0.445	1.000	1.247	0.997	1.796	1.017	1.147	0.121
5	-1.106	0.103	-1.799	0.998	-1.247	1.000	-0.445	1.000	0.445	1.000	1.247	1.000	1.799	0.998	1.106	0.103
6	-1.100	0.100	-1.802	1.000	-1.247	1.000	-0.445	1.000	0.445	1.000	1.247	1.000	1.802	1.000	1.100	0.100
7	-1.100	0.100	-1.802	1.000	-1.247	1.000	-0.445	1.000	0.445	1.000	1.247	1.000	1.802	1.000	1.100	0.100

1.0000000000		0.1000000000
0.9009688679	±	0.4338837391 i
0.6234898019	±	0.7818314825 i
0.2225209340	±	0.9749279122 i
-0.2225209340	±	0.9749279122 i
-0.6234898019	±	0.7818314825 i
-0.9009688679	±	0.4338837391 i
-1.0000000000		-0.1000000000

例3　$F(x)=(x-1)(x-2)(x-3)\cdots\cdots(x-14)(x-15)$

X^{15} -120.000 X^{14} +6580.000 X^{13} -218400.000 X^{12} +4899622.000 X^{11} -78558480.000 X^{10} +928095740.000 X^9 -8207628000.000 X^8 +54631129553.000 X^7 -272803210680.000 X^6 +1009672107080.000 X^5 -2706813345600.000 X^4 +5056995703824.000 X^3 -6165817614720.000 X^2 +4339163001600.000 X-1307674368000.000

X^{15} -140.000 X^{13} +7462.000 X^{11} -191620.000 X^9 +2475473.000 X^7 -15291640.000 X^5 +38402064.000 X^3 -25401600.000 X　← 変数変換後

23.664 22.169 20.786 19.511 18.341 17.274 16.315 15.469 14.753 14.193 13.826 13.668 13.640 13.640 13.640

1	-24.921	186.043	-18.254	186.043	-8.430	186.043	2.851	186.043	13.640	186.043	22.070	186.043	26.683	186.043	13.640
2	-24.365	173.739	-18.239	164.697	-8.665	144.754	2.955	137.460	13.832	155.341	21.774	170.721	25.953	174.944	13.244
3	-23.056	152.649	-16.878	137.921	-7.875	124.915	2.684	120.574	12.645	131.015	20.427	145.810	24.646	156.689	12.591
4	-21.677	133.721	-15.848	119.077	-7.342	106.701	2.495	103.336	11.846	112.190	19.203	126.962	23.179	138.000	11.856
5	-20.400	117.561	-14.927	103.163	-6.894	90.801	2.341	87.750	11.154	96.213	18.093	111.032	21.791	121.613	11.156
6	-19.229	103.661	-14.077	89.373	-6.492	77.028	2.204	74.104	10.519	82.415	17.070	97.222	20.518	107.568	10.513
7	-18.157	91.678	-13.290	77.394	-6.122	65.110	2.080	62.254	9.933	70.461	16.126	85.214	19.360	95.573	9.928
8	-17.178	81.344	-12.564	66.999	-5.782	54.803	1.965	51.996	9.390	60.104	15.254	74.785	18.313	85.334	9.398
9	-16.286	72.436	-11.895	57.989	-5.469	45.900	1.860	43.136	8.889	51.139	14.453	65.743	17.369	76.605	8.921
10	-15.476	64.763	-11.280	50.192	-5.180	38.222	1.763	35.500	8.426	43.390	13.718	57.914	16.524	69.180	8.494
11	-14.745	58.161	-10.716	43.453	-4.914	31.617	1.673	28.939	8.000	36.707	13.046	51.143	15.774	62.893	8.118
12	-14.086	52.485	-10.199	37.640	-4.670	25.953	1.590	23.321	7.607	30.957	12.433	45.293	15.115	57.608	7.791
13	-13.494	47.607	-9.725	32.638	-4.445	21.115	1.513	18.532	7.246	26.027	11.873	40.245	14.546	53.215	7.515
14	-12.961	43.410	-9.292	28.347	-4.238	17.004	1.442	14.473	6.914	21.819	11.363	35.894	14.066	49.628	7.293
15	-12.479	39.785	-8.896	24.684	-4.049	13.536	1.377	11.059	6.609	18.248	10.896	32.151	13.675	46.782	7.132
16	-12.039	36.650	-8.534	21.574	-3.875	10.635	1.316	8.215	6.328	15.241	10.467	28.939	13.374	44.631	7.038
17	-11.645	34.014	-8.202	18.955	-3.715	8.240	1.261	5.879	6.069	12.737	10.072	26.196	13.164	43.151	7.004
18	-11.330	32.017	-7.895	16.768	-3.568	6.295	1.209	3.996	5.831	10.680	9.707	23.881	13.046	42.323	7.000
19	-11.124	30.749	-7.608	14.964	-3.432	4.757	1.161	2.521	5.611	9.026	9.387	22.030	13.005	42.036	7.000
20	-11.025	30.152	-7.342	13.536	-3.306	3.587	1.117	1.415	5.404	7.735	9.152	20.777	13.000	42.000	7.000
21	-11.001	30.009	-7.132	12.558	-3.186	2.757	1.075	0.651	5.212	6.789	9.033	20.166	13.000	42.000	7.000
22	-11.000	30.000	-7.025	12.101	-3.080	2.248	1.036	0.204	5.066	6.221	9.002	20.010	13.000	42.000	7.000
23	-11.000	30.000	-7.001	12.004	-3.014	2.035	1.008	0.028	5.007	6.022	9.000	20.000	13.000	42.000	7.000
24	-11.000	30.000	-7.000	12.000	-3.000	2.001	1.000	0.001	5.000	6.000	9.000	20.000	13.000	42.000	7.000
25	-11.000	30.000	-7.000	12.000	-3.000	2.000	1.000	0.000	5.000	6.000	9.000	20.000	13.000	42.000	7.000
26	-11.000	30.000	-7.000	12.000	-3.000	2.000	1.000	0.000	5.000	6.000	9.000	20.000	13.000	42.000	7.000

$C_{14}/15$ = 8.000000

14.0000000000	13.0000000000
12.0000000000	11.0000000000
10.0000000000	9.0000000000
7.0000000000	8.0000000000
5.0000000000	6.0000000000
3.0000000000	4.0000000000
1.0000000000	2.0000000000
15.0000000000	

参考文献

[1] 伊理正夫：数値計算，朝倉書店 (1981)

[2] 伊理正夫他：大域的収束性をもつ代数方程式の解法，数理解析研究所講究録 339 (1978)

[3] Aberth, O.: Iteration Method for All Zeros of a Polynomial Simultaneously, Math. Comp, vol. 27 pp 339-344 (1973)