Bul.St.Univ.Baia Mare
Seria B, Matematică-Informatică, Vol.VII, Nr.1-2, 141-147

ASUPRA DETERMINARII RADACINILOR REALE SI COMPLEXE ALE UNUI POLINOM CU AJUTORUL CALCULATORULUI

Gheorghe ARDELEAN

In cele ce urmeaza, ne propunem studiul unei metode si elaborarea unui program la calculator, pentru calculul aproximativ al radacinilor unui polinom cu coeficienti reali si de variabila complexa, metoda ce deriva din metoda generala Newton-Raphson.

Pentru aceasta, sa consideram polinomul :

$$f(z)= a_1 z^n + a_2 z^{n-1} + \dots + a_n z + a_{n+1} \qquad (1)$$

unde $a_1, a_2, \dots, a_{n+1}$ sint coeficienti reali, iar $z=x+iy$ este o variabila complexa ($x,y \in R$; $i=\sqrt{-1}$).

Metoda generala Newton-Raphson pentru functii reale de variabila reala conduce la sirul aproximantelor solutiei ecuatiei $f(x)=0$, dat de relatia de recurenta :

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \qquad (2)$$

O functie $f:E \longrightarrow C$, cu $E \subset C$ se numeste <u>functie complexa de variabila complexa.</u>

In studiul acestor functii, se considera multimea C organizata ca spatiu metric, cu distanta euclidiana. Cum C si RxR coincid ca multimi, nu se va face nici o deosebire intre spatiile metrice (C,d) si (RxR,d) unde d este distanta euclidiana.

Multimea $E \subset C$ pe care este definita functia f, poate fi asimilata cu multimea punctelor $(x,y) \in (RxR)$, corespunzatoare punctelor $(x+iy) \in E$. Aceasta multime o notam tot cu E. Deci functia f poate fi asimilata cu o functie vectoriala definita pe E, cu valori in RxR.

Fie u si v componentele acestei functii :

$$f(z)=u(x,y)+iv(x,y),\ z=x+iy \in E.$$

In acest mod, functiei complexe f i se asociaza o pereche ordonata (u,v) unica, formata din functiile reale u=Re(f), v=Im(f) definite pe aceeasi multime E.

Reciproc, fiind date functiile $u:E \longrightarrow R$ si $v:E \longrightarrow R$, perechii ordonate (u,v) i se asociaza o singura functie $f:E \longrightarrow C$, asimilabila cu functia vectoriala $(u,v):E \longrightarrow RxR$.

Functiile u si, respectiv v, se numesc <u>partea reala</u> si respectiv, <u>partea imaginara</u> a functiei f.

Conditia de derivabilitate a functiei reale f din relatia (2) revine in cazul functiei complexe f=(u,v) la sistemul de conditii de monogenitate Cauchy-Riemann :

$$\left(\frac{\partial u}{\partial x}\right)_{(x,y)} = \left(\frac{\partial v}{\partial y}\right)_{(x,y)}$$

$$\left(\frac{\partial u}{\partial y}\right)_{(x,y)} = -\left(\frac{\partial v}{\partial x}\right)_{(x,y)} \qquad \forall (x,y)\in E. \tag{3}$$

adica la conditia ca functia f sa fie olomorfa pe multimea E.

Fara a prezenta demonstratia, precizam faptul ca orice functie polinomiala de variabila complexa este o functie olomorfa.

Polinomului f(z) din (1) ii vom asocia functia de variabila complexa, notata cu f :

$$f: C \longrightarrow C$$

$f(z)=u(x,y)+iv(x,y)$ unde $z=x+iy$ si pentru care in plus avem

$$f'(z)=\frac{\partial u}{\partial x}(x,y)+i\frac{\partial v}{\partial x}(x,y) \tag{4}$$

Sirul aproximantelor solutiei z a ecuatiei f(z)=0 va fi dat de relatia de recurenta

$$z_n=z_{n-1}-\frac{f(z_{n-1})}{f'(z_{n-1})} \tag{5}$$

sau

$$x_n+iy_n=x_{n-1}+iy_{n-1}-\frac{u(x_{n-1},y_{n-1})+iv(x_{n-1},y_{n-1})}{\frac{\partial u}{\partial x}(x_{n-1},y_{n-1})+i\frac{\partial v}{\partial x}(x_{n-1},y_{n-1})} \tag{6}$$

si cu precizarea ca in relatiile ce urmeaza functiile $u, v, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}$ sint calculate in (x_{n-1},y_{n-1}), prin inmultirea cu conjugata numitorului in relatia (6) se obtine :

$$x_n+iy_n=x_{n-1}+iy_{n-1}-\frac{(u+iv)\left(\frac{\partial u}{\partial x}-i\frac{\partial v}{\partial x}\right)}{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2} \tag{7}$$

Efectuind inmultirea la numarator, se obtine :

$$x_n+iy_n= x_{n-1}+ iy_{n-1} - \frac{\left(u\frac{\partial u}{\partial x}+v\frac{\partial v}{\partial x}\right)+i\left(v\frac{\partial u}{\partial x}-u\frac{\partial v}{\partial x}\right)}{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2} \tag{8}$$

Identificind acum in cei doi membri ai relatiei de mai sus partile reale si respectiv partile imaginare, se obtin relatiile :

$$x_n = x_{n-1} - \frac{u\frac{\partial u}{\partial x} + v\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2}$$

$$y_n = y_{n-1} - \frac{v\frac{\partial u}{\partial x} - u\frac{\partial v}{\partial x}}{\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2} \qquad (9)$$

Asadar, dispunem acum de relatii de recurenta pentru determinarea termenilor sirului aproximantelor solutiei $\bar{z}=\bar{x}+i\bar{y}$ a ecuatiei $f(z)=0$. Procesul iterativ (9) demareaza cu o valoare initiala $z1=x1+iy1$ data, si continua pina la satisfacerea urmatoarei conditii de oprire pentru o radacina :

$$\frac{|x_{n+1}-x_n|+|y_{n+1}-y_n|}{|x_{n+1}|+|y_{n+1}|} < eps \qquad (10)$$

unde eps este o valoare aleasa suficient de mica, dar pozitiva. In acest caz valoarea radacinii aproximative $\bar{z}$ va fi considerata ca fiind $x_{n+1}+iy_{n+1}$.

Pentru a calcula radacinile urmatoare ale polinomului (1) se pleaca de la valoarea radacinii precedente determinata, rezolvind polinomul de grad (n-1), apoi (n-2), ... ,(n-i), ... ai carui coeficienti se determina prin relatiile :

$$\begin{aligned} & b_1 = a_1 \\ & \vdots \\ & b_j = a_j + x_i\, b_{j-1} \qquad j=2,3,\dots,n \end{aligned} \qquad (11)$$

in cazul unei radacini reale x_i, sau prin relatiile

$$\begin{aligned} & b_1 = a_1 \\ & b_2 = a_2 + 2x_i\, b_1 \\ & \vdots \\ & b_j = a_j + 2x_i\, b_{j-1} - (x_i^2 + y_i^2)\, b_{j-2} \qquad j=3,4,\dots,n-1 \end{aligned} \qquad (12)$$

in cazul unei radacini complexe x_i+iy_i. In acest ultim caz radacina conjugata se obtine direct, fara calcul iterativ.

In continuare, avind in vedere faptul ca metoda prezentata urmeaza a fi programata la calculator, vom deduce formulele de recurenta pentru calculul valorilor $u, v, \frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}$ intr-un punct (x,y). Pentru aceasta, sa consideram din nou functia polinomiala

$$f(z) = a_1 z^n + a_2 z^{n-1} + \dots + a_n z + a_{n+1} \qquad (13)$$

unde $z=x+iy$.

Sa notam $f(z)=u_{n+1}(x,y)+iv_{n+1}(x,y) = s_{n+1}$ (14)

In cele ce urmeaza vom subintelege faptul ca functiile u_i, v_i sint calculate in punctul (x,y).

Din relatiile (14) deducem

$$s_{n+1}= a_{n+1} + z(u_n+iv_n)$$

sau

$$s_{n+1}= a_{n+1} + (x+iy)(u_n+iv_n)$$

sau

$$s_{n+1}= a_n + (xu_n+yv_n)+i(yu_n+xv_n) \qquad (15)$$

si tinind cont de relataia (14), obtinem :

$$u_{n+1}= a_{n+1} + xu_n- yv_n$$

$$v_{n+1}= yu_n+ xv_n \qquad (16)$$

cu $u_1=a_1$, $v_1=0$

relatii ce reprezinta formulele de recurenta pentru calculul lui $f(z)=u_{n+1}(x,y)+iv_{n+1}(x,y)$.

Pentru a deduce acum relatii de recurenta pentru calculul lui $\frac{\partial u}{\partial x}$ si $\frac{\partial v}{\partial x}$ in punctul (x,y), vom deriva in raport cu x in relatiile (16) si vom obtine :

$$\frac{\partial u_{n+1}}{\partial x} = u_n + x\frac{\partial u_n}{\partial x} - y\frac{\partial v_n}{\partial x}$$

$$\frac{\partial v_{n+1}}{\partial x} = v_n + y\frac{\partial u_n}{\partial x} + x\frac{\partial v_n}{\partial x} \qquad (17)$$

Se mai observa ca :

$$s_2= a_1z+a_2= a_1(x+iy) + a_2= a_1x+a_2+ia_1y.$$

Deci

$$u_2(x,y)=a_1x+a_2$$

$$v_2(x,y)=a_1y$$

de unde se obtine

$$\frac{\partial u_2}{\partial x} = a$$

$$\frac{\partial v_2}{\partial x} = 0. \qquad (18)$$

Relatiile (17) si (18) constituie formule de recurenta pentru calculul derivatelor partiale $\frac{\partial u}{\partial x}(x,y)$ si $\frac{\partial v}{\partial x}(x,y)$.

Pe baza celor prezentate s-a elaborat programul REZPOL, in limbajul Turbo Pascal, versiunea 5.5, care accepta ca date :

- n – gradul polinomului (<31)
- a_i, $i=1,n+1$ – coeficientii polinomului;

si ofera ca rezultate :

- x_i, $i=1,n$ – tabloul partilor reale ale radacinilor
- y_i, $i=1,n$ – tabloul partilor imaginare ale radacinilor.

```
program REZPOL;{ determinarea radacinilor reale si complexe }
                        { ale unui polinom }
    uses crt;
    label 1,2,3,4,5,10;
    var a,b:array[1..31] of real;
        x,y:array[1..30] of real;
        x1,y1,x2,y2,u,du,v,dv,u1,du1,v1,dv1,rx,ry,d,s:real;
        m,n,i,j,k,l,it1,it2:byte;
        complex:boolean;

    begin
        clrscr;
1:      write('grad polinom m=');read(m);
        if (m<2) or (m>30) then
            begin
                write('               grad eronat');writeln;
                goto 1;
            end;
        writeln;
        n:=m+1;
        for i:=1 to n do
            begin
                write('a(',i,')=');read(a[i]);writeln;
            end;
        for i:=1 to n do { se citesc coeficientii }
            begin
                x[i]:=0.0;
                y[i]:=0.0;
                b[i]:=a[i];
            end;
        j:=1;{ se va determina prima solutie x(1),y(1) }
        x1:=1.0;{ valoarea initiala de pornire }
        y1:=1.0;{          z0=1+i                }
        while n>1 do
          begin
             complex:=false;{ radacina presupusa reala }
             k:=0;{ se incearca prima valoare de pornire }
             it1:=0;
             it2:=0;
```

```
2:               u:=b[1];
                 v:=0.0;{ relatiile (18) }
                 du:=b[1];{ si conditiile initiale de la (16) }
                 dv:=0.0;
                 for i:=2 to n do
                 begin
                     u1:=b[i]+x1*u-y1*v;{ relatiile (16) }
                     v1:=y1*u+x1*v;

                     if i<n then
                     begin
                         du1:=u1+x1*du-y1*dv;{ relatiile (17) }
                         dv1:=v1+y1*du+x1*dv;
                         du:=du1;
                         dv:=dv1;
                         u:=u1;
                         v:=v1;
                     end;
                 end;
                 rx:=u1*du+v1*dv;{ numaratorii de la (9) }
                 ry:=v1*du-u1*dv;
                 d:=dv*dv+du*du;{ numitorul de la (9) }
                 if abs(d)<=1.0e-8 then goto 3;
                 x2:=x1-rx/d;{ relatiile de recurenta (9) }
                 y2:=y1-ry/d;{ pentru determinarea radacinii }
                 if abs(x2-x1)+abs(y2-y1)<(abs(x2)+abs(y2))*1.0e-3
                    then goto 4;{ conditia (10) }
3:               it1:=it1+1;{ se contorizeaza numarul de iteratii }
                 if it1<100 then
                    begin
                        x1:=x2;
                        y1:=y2;
                        goto 2;
                    end
                            else
                    begin
                        x1:=1.0;{ se incearca a doua valoare }
                        x2:=2.0;{     de pornire  z0=1+2i }
                        k:=k+1;
                        if k<=1 then goto 2;
writeln;write('O solutie nu s-a putut determina dupa 100 de iteratii');
                                exit;
                     end;

4:            it2:=it2+1;
              if it2<=10 then
              begin
                   x1:=x2;{ se imbunatateste solutia }
                   y1:=y2;{ cu inca 10 iteratii       }
              end;
        if abs(y2)<=abs(x2)*1.0e-6 then goto 5;{ radacina reala}
            x[j]:=x2;{ se retine conjugata radacinii }
            y[j]:=-y2;
```

```
        j:=j+1;{ se trece la urmatoarea radacina }
        n:=n-1;{ scade numarul coeficientilor polinomului cit }
        complex:=true;{ radacina complexa }
5:      x[j]:=x2;{ se retine radacina determinata }
        y[j]:=y2;
        j:=j+1;{ se trece la urmatoarea radacina}
        n:=n-1;if n<=1 then goto 10;{ SFIRSIT }
        if not complex then
          begin
               d:=x2;{ se pregateste utilizarea }
               s:=0.0;{     relatiei (11) }
          end
                         else
          begin
               d:=2*x2;{ se pregateste utilizarea }
          end;s:=x2*x2+y2*y2;{ relatiei (12) }
          b[2]:=b[2]+d*b[1];{ se determina coeficientii polinomului }
          if n=2 then         {                                    cit      }
               begin
                    x[j]:=-b[2]/b[1];{ ecuatie de gradul 1 }
                    goto 10;{ SFIRSIT }
               end;
          for l:=2 to n do
               b[l+1]:=b[l+1]+d*b[l]-s*b[l-1];{ relatiile (11)/(12) }
      end;{ while }
10:   clrscr;{ afisare rezultate }
      for i:=1 to m do
           writeln('x(',i,')=',x[i]:8:3,'      y(',i,')=',y[i]:8:3);
   end.{ PROGRAM }
```

BIBLIOGRAFIE

1. B.Demidovici — Elémentes de calcul numérique
   Editions Mir, Moscou, 1973.

2. W.S.Dorn, D.D.Mc.Cracken — Metode numerice cu programe in FORTRAN IV
   Editura Tehnica, Bucuresti, 1976.

3. Ion Gh.Sabac — Matematici speciale
   Editura Didactica si Pedagogica, Bucuresti, 1981.

ABSTRACT

A numerical method (that comes from the general method Newton-Raphson) to estimate the real roots and the complex roots of a polynom is presented. It appears too,a Turbo Pascal v5.5 program, that implements tis method on the computer.

UNIVERSITY OF BAIA MARE

4800 Baia Mare

ROMANIA.