任意区間における代数方程式の実根分離について

岡　田　敏　彦*

On the Separation of All Real Roots of a Polynomial Equation on the Arbitrary Interval

Toshihiko OKADA

Abstract

This method finds all real roots of a polynomial equation on the arbitrary interval $[a, b]$ such that $a<b$. Let $f(x)=a_0x^n+a_1x^{n-1}+\cdots+a_n=0$ be a given polynomial whose coefficients are real but $a_0\neq 0$. From this, we construct a sequence of polynomials as follows. That is, $f(x)$, $f'(x)$, $\cdots$, $f^{(n-1)}(x)$. Now we can obtain the root x_{11} of $f^{(n-1)}(x)=0$ immediately because of $f^{(n-1)}(x)$ being linear. If there is x_{11} on the interval $[a, b]$, we divide the interval $[a, b]$ into two parts $[a, x_{11}]$ and $[x_{11}, b]$ by using it. Next we consider whether there is or not a root of $f^{(n-2)}(x)=0$ on each part. If $f^{(n-2)}(a)\cdot f^{(n-2)}(x_{11})$ is negative, then $f^{(n-2)}(x)=0$ has one root on the interval $[a, x_{11}]$. If positive, then one has no root. Now suppose $f^{(n-2)}(x)=0$ has one root on the interval $[a, x_{11}]$. Then by Newton-Raphson method starting with the first approximation $x_1=(a+x_{11})/2$, we can obtain rapidly the root x_{21} of $f^{(n-2)}(x)=0$ on the interval $[a, x_{11}]$. Similarly, on the other interval $[x_{11}, b]$ we can obtain the root x_{22} of $f^{(n-2)}(x)=0$ if there is. By using x_{21} and x_{22}, we divide afresh the interval $[a, b]$ into three parts $[a, x_{21}]$, $[x_{21}, x_{22}]$ and $[x_{22}, b]$. Repeating the similar procedure for these parts, we can obtain the roots of $f^{(n-3)}(x)=0$ of degree greater than $f^{(n-2)}(x)$. In general, on the interval $[a, b]$ we can obtain the real roots of $f^{(i-1)}(x)=0$ by finding those of $f^{(i)}(x)=0$ of degree lower than $f^{(i-1)}(x)$ where $1\leq i\leq n-1$. Finally, continuing the method mentioned above, we can find the real roots of $f(x)=0$ on the interval $[a, b]$, which is divided into some parts by those of $f'(x)=0$.

1. 緒　　言

本方法は任意区間に存在する代数方程式のすべての実根を方程式の次数を減じることなく求めるものである．現在計算機による代数方程式の解法によく用いられている Newton-Raphson 法[1]，Bairstow 法[1]は適当に根の第一近似を設定し，反復修正をくり返して，一つの根あるいは二つの根を見い出していく．根を一つあるいは二つ見い出すたびに与えられた代数方程式の次数を1次あるいは2次づつ減じて，新しい方程式について同様に根を見つけ出す方法をとっている．これらの方法においては根の第一近似の設定が悪いと収束しなかったり，そのため反復プログラムをいつまでも脱出できなくて，第一近似の変更をよぎなくされることがある．また，根を見い出すたびごとに方程式の次数を減じるため，減じた新しい方程式の係数に誤差が累積伝播する．このことは方程式が高次になるほどいちぢるしい．

本方法では代数方程式 $f(x)=0$（次数 n）の実根を，これより1次だけ低次の $f'(x)=0$ を満足する実根 $x_{ij}(i=n-1,\ j=1,2,\cdots,\ k\leq n-1)$ を分割点とし，区間 $[a, b]$ をいくつかの小区間 $[a, x_{i1}]$，$[x_{i1}, x_{i2}]$，$\cdots$，$[x_{ik}, b]$ に分割し，各区間に高々1個存在するようにする．各区間の根への接近は，Newton-Raphson法，二分割挟撃法（Binary search）等によって求める．このような方法によって与えられた代数方程式の次数を減じることなく，また，多重根を持つような場合でも容易に求めることができる．

2. 計 算 法

任意の区間 $[a, b]$ に存在するすべての実根を求めるには，次のような計算過程をとる．以下根といえば

*共通科

特にことわらないかぎり実根とする．

与えられた代数方程式，

$f(x)=a_0x^n+a_1x^{n-1}+\cdots+a_n=0$

より，

$f'(x)=0$，$f''(x)=0$，…，$f^{(i)}(x)=0$，…，

$f^{(n-1)}(x)=0$

をつくり出す．$f^{(n-1)}(x)=0$ は1次だからただちに根を求められる．求めた根 x_{11} によって区間〔a, b〕を二分割し，それを〔a, x_{11}〕，〔x_{11}, b〕とする．各区間には $f^{(n-2)}(x)=0$（次数2）の根が高々1個存在する．すなわち，1次方程式の根 x_{11}によって，2次方程式の根が二つの区間にそれぞれ分離されることになる．

このようにして順次高次の方程式に移っていき，$f^{(i)}(x)=0\ (i=1, 2\cdots, n-1)$ のm個（$m\leq n-i$）の根によって，区間〔a, b〕が $m+1$個に分割される．したがって，区間〔a, b〕において $f'(x)=0$ の根が l 個求まれば，区間〔a, b〕が$l+1$個に分割され，$f(x)=0$ の根が各区間に分離されることになる．次に各区間の根への接近方法について述べる．

$f^{(n-1)}(x)=0$ の根 x_{11} によって区間〔a, x_{11}〕，〔x_{11}, b〕に分割される．小区間〔a, x_{11}〕に $f^{(n-2)}(x)=0$ の根が存在するかどうかは，$f^{(n-2)}(a)$，$f^{(n-2)}(x_{11})$の符号によって決まる．同符号ならば存在しない．異符号ならば1個存在する．存在しないときは次の小区間に移る．

いま，Fig. 1に示すように区間〔a, x_{11}〕に根が存在するとき，2次方程式の一つの真根 x_{21} への接近は Newton-Raphson 法あるいは二分割挟撃法によって行なう．

Newton-Raphson 法：

第1近似として，$x_1=(a+x_{11})/2$ をとり，

第2近似，

$x_2=x_1-f^{(n-2)}(x_1)/f^{(n-1)}(x_1)$

を求める．一般に第 k 近似は

$x_k=x_{k-1}-f^{(n-2)}(x_{k-1})/f^{(n-1)}(x_{k-1})\ (k=2, 3, \cdots)$

となり，

$|(x_k-x_{k-1})/x_k|<\varepsilon$ あるいは，

$|f^{(n-2)}(x_k)|<\varepsilon$　　　　ε：収束判定定数

を満足するならば x_k を根とする．

二分割挟撃法：

まず，$x_1=(a+x_{11})/2$ をとり，$f^{(n-2)}(a)$，$f^{(n-2)}(x_1)$ が同符号のとき，$a=x_1$，異符号のとき，$x_{11}=x_1$ と置換し，根の存在する区間をせばめる．同様にして新しい区間の二分割点 x_2 をとり $f^{(n-2)}(x_2)$の符号をしらべ，さらに区間をせばめていく．すなわち，Fig. 2 に示すように，〔a, x_{11}〕→〔x_1, x_{11}〕→〔x_1, x_2〕→…となっていく．区間〔x_i, x_j〕(i, $j=1, 2, \cdots$, $|i-j|=1$）が十分せばまり，$|f(x_i)|<\varepsilon$ あるいは $|f(x_j)|<\varepsilon$ を満たすとき，x_i あるいは x_j を根とする．または，$x_k=(x_i+x_j)/2$ をとり，$i<j<k$ とすれば，$|(x_k-x_j)/x_k|<\varepsilon$ を満たすならば x_k を根とする．

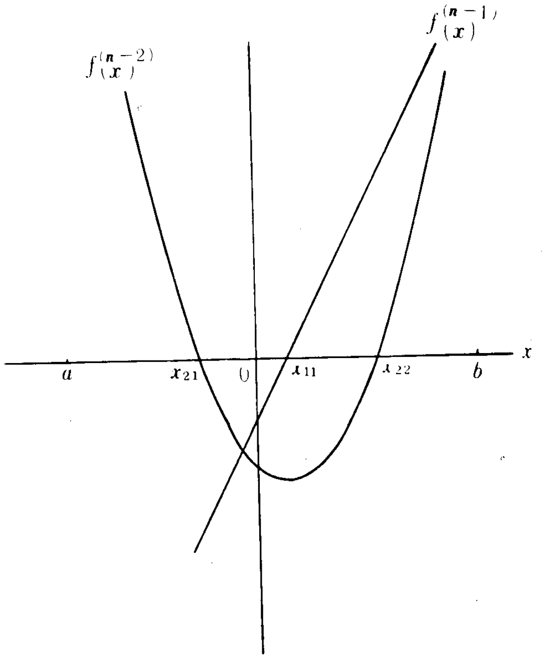

Fig. 1 The root x_{11} of $f^{(n-1)}(x)=0$ (degree 1) divides the interval 〔a, b〕 into two parts 〔a, x_{11}〕 and 〔x_{11}, b〕. A root x_{21} of $f^{(n-2)}(x)=0$ (degree 2) lies between a and x_{11}, the other x_{22} between x_{11} and b

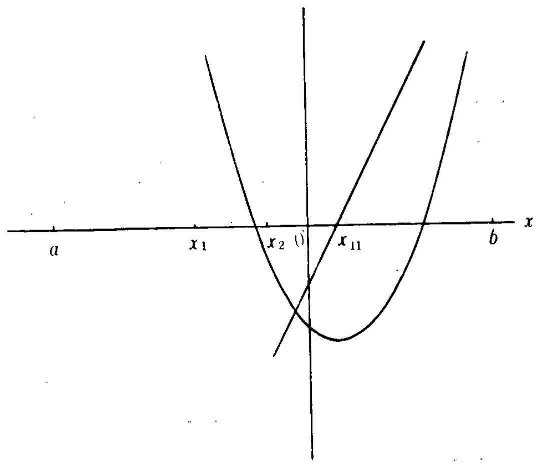

Fig. 2 By the binary search method the intevtal 〔a, x_{11}〕 where the root exists narrows gradually as follows. That is, 〔a, x_{11}〕→〔x_1, x_{11}〕→〔x_1, x_2〕→…

かくして，各区間で求めた根を保存し，一つ高次の方程式の区間〔a, b〕での根を小区間に分離する分割点とする．

以上のような方法により，1次方程式から順次せり上って根を求めていき，最終的に $f(x)=0$ の根を求めることができる．

例　$f(x)=(x+1)(x-1)(x-2)$

$$=x^3-2x^2-x+2=0$$

$$f'(x)=3x^2-4x-1=0$$

$$f''(x)=6x-4=0$$

Fig. 3 に示すように区間〔-2, 3〕において，1次方程式の根 $x_{11}=2/3$ は2次方程式の根 $x_{21}=(2-\sqrt{7})/3$, $x_{22}=(2+\sqrt{7})/3$ を小区間〔-2, $2/3$〕と〔$2/3$, 3〕に分離している．さらに，x_{21}, x_{22} は3次方程式の根 $x_{31}=-1$, $x_{32}=1$, $x_{33}=2$ を小区間〔-2, x_{21}〕，〔x_{21}, x_{22}〕，〔x_{22}, 3〕に分離している

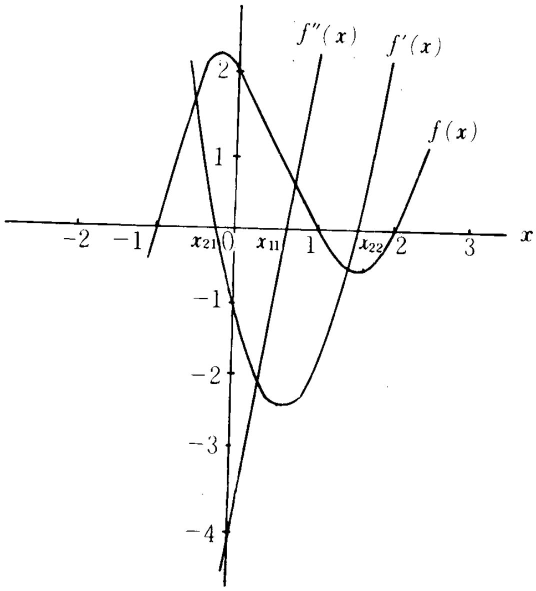

Fig. 3 The separation of the real roots of $f(x)=x^3-2x^2-x+2=0$ *on the interval* 〔-2, 3〕

3. 実 験 結 果

付録に示しているプログラムで FACOM-231 により実験を行なった．Table 1 に示すように近接根，多重根の場合も非常に良好に求めることができている．

Table 1 Experimental results ($\varepsilon=10^{-8}$)

Degree	Coefficients	Results	True roots
4	1, −6, 9, 4, −12[3]	−1.00000000000001E+00 +2.00000000000001E+00 +2.00000000000001E+00 +3.00000000000001E+00	−1.0 2.0 2.0 3.0
4	1, −4.86, 8.8571, −7.173846, 2.1788712[2]	+1.20000000001148E+00 +1.20999999656328E+00 +1.22000000802954E+00 +1.22999999940513E+00	1.20 1.21 1.22 1.23
6	1, −9, 45, −85, 34 74, −100[3]	−1.00000000000001E+00 +1.99999999999999E+00	−1.0 2.0 $3\pm4i$, $1\pm i$
10	1, −55, 1320, −18150 15773, −902055, 3416930, −8409500 12753576, −10628640 3628800	+1.00000000000022E+00 +1.99999999999760E+00 +3.00000000012695E+00 +3.99999999948219E+00 +5.00000000264719E+00 +5.99999998851310E+00 +7.00000000480307E+00 +8.00000000964149E+00 +9.00000000706719E+00 +9.99999999763440E+00	1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0

11	1, −11, 55, −165, 330, −462, 462, −330, 165, −55, 11, −1	+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
		+1.00000000000000E+00	1.0
12	1, −78, 1001, −5005 12870, −19448, 18564, −11628, 4845, −1330, 231, −23, 1 1)	+2.53989777970641E−01	$x_k=\frac{1}{2}\left(1-\cos\frac{2k-1}{25}\pi\right)^{-1}$ ($k=1,2,\cdots\cdots,12$)
		+2.66480958169940E−01	
		+2.89189745666758E−01	
		+3.25557543841587E−01	
		+3.81966011172561E−01	
		+4.70459597509400E−01	
		+6.15294736492605E−01	
		+8.70745329542214E−01	
		+1.37902118690857E+00	
		+2.61803398874930E+00	
		+7.12012217452357E+00	
		+6.34091389484112E+01	
12	1, −10, 44, −110, 165, −132, 0, 132, −165, 110, −44, 10, −1	−1.00000000000001E+00	−1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
		+9.99999999999970E−01	1.0
20	1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, −1	−1.00000000000001E+00	−1.0
		+1.00000000000001E+00	1.0
			$x^{18}+x^{16}+\cdots\cdots+x^2+1$

4. 結　　言

本論文は任意区間の代数方程式の実根を求める一方法について述べたものであるが，区間を大きくとれば当然のことながら与えられた代数方程式のすべての実根を求めることができる．

分割された小区間に高々1個存在する根への接近は Newton-Raphson 法あるいは二分割挾撃法によって求められるが，収束は前者の方がはるかに速い．また，$f(x)=0$ の根を求めるのに $f^{(n-1)}(x)=0$（次数1）の根から順次せり上っていくが，$f'(x)=0$ までの根は単に一つ高次の方程式の根が存在する区間を分割する分割点としての役目を果すのみであるので，収束判定定数 ε は $f'(x)=0$ の根を求めるまではゆるやかでよい．これによっても演算時間を短縮することができる．なお，本方法を使用して九州大学大型計算機 FACOM 230−60 でも行なったところ，Table 1

で示されている例については，すべて1秒以内の時間で得られた．複素根については現在検討中であるが，現段階では，本方法によりすべての実根を求め，与えられた代数方程式を複素根だけを持つ方程式に次数を下げ，それを Bairstow 法などで解いていく方法をとればすべての根が求められる．

参考文献

1) 宇野利雄：計算機のための数値計算，朝倉書店，(1963) P. 94—111

2) 山内，森口，一松：電子計算機のための数値計算法 I，培風館 (1965) P. 61

3) P. Henrici and B. O. Watkins: Communication of the ACM, 8, 572 (1965)

(昭和46年12月15日受理)

付　　録

実験に使用したプログラムを以下に示す．FACOM-231 ALGOL により組まれている．プログラム中の配列宣言は20次まで解けるようにとってあり，収束判定定数 $\varepsilon=10^{-8}$ としている．

入力データの順

$a, b, n, a_0, a_1, \cdots, a_n$

ここに，a, b：区間設定値（実数 $a<b$）

n：次数（整数）

$a_0, a_1, \cdots, a_n$：係数（実数）

```
begin comment reallength:=15;
    integer i, n, I; real a, b; array A, ROOT [1:21];
    procedure POLYN(a, b, n, A, ROOT, I);
    value a, b, n; integer n, I; real a, b; array A, ROOT;
    begin integer i, j, k; real c, d, x;
        array AA [1:21, 1:21], B1, B2[1:21];
        real procedure F(n, i, x, A);
        value n, i, x; integer n, i; real x; array A;
        begin real w; integer j, m;
            m:=n-i+1; w:=A [m, 1];
            for j:=2 step 1 until i+1 do
            w:=w*x+A [m, j];
            F:=w
        end;
        real procedure NEWTON(n, i, x, A);
        value n, i, x; integer n, i; real x; array A;
        begin real x0, x1;
                x0:=x;
        LOOP: x1:=x0-F(n, i, x0, A)/F(n, i-1, x0, A);
                if x1=0 then NEWTON:=0 else
                if abs ((x1-x0)/x1)<10-8 then NEWTON:=x1
                else begin x0:=x1; go to LOOP end
        end;
        for i:=1 step 1 until n+1 do
        AA [1, i]:=A [i] /A [1];
        for i:=1 step 1 until n do
        begin k:=n-i+1; c:=k;
                for j:=1 step 1 until k do
                begin AA [i+1, j]:=c*AA [i, j] /k;
                        c:=c-1.0
                end
```

```
        end ;
        I:=0 ;
        for i:=1 step 1 until n do
        begin B1〔1〕:=a ; B2〔I+1〕:=b ;
            for j:=1 step 1 until I do
            B1〔j+1〕:=B2〔j〕:=ROOT〔j〕;
            k:=n-i+1 ; c:=i ;
            for j:=1 step 1 until i do
            begin AA 〔k+1, j〕:=c*AA 〔k, j〕;
                    c:=c-1.0
            end ;
            k:=I+1 ; I:=0 ;
            for j:=1 step 1 until k do
            begin c:=F(n, i, B1〔j〕, AA) ; d:=F(n, i, B2〔j〕, AA) ;
                if abs (c) <10-11 then begin I:=I+1 ;
                ROOT〔I〕:=B1〔j〕 end else
                if abs(d) <10-11 then begin I:=I+1 ;
                ROOT〔I〕:=B2〔j〕 end else
                if (c>0 and d <0) or (c <0 and d >0) then
                begin I:=I+1 ; x:=0.5* (B1〔j〕 +B2〔j〕) ;
                        ROOT〔I〕:=NEWTON(n, i, x, AA)
                end
            end
        end
    end of POLYN ;
START : Readreal(a) ; Readreal(b) ; Readinteger(n) ;
        for i:=1 step 1 until n+1 do Readreal(A〔i〕) ;
        CRLF(2) ; Prinstring('BETWEEN' ) ; Print(a) ;
        Printstring('AND ' ) ; Print(b) ; CRLF(2) ;
        Printstring('DEGREE=') ; Printinteger(n) ;
        CRLF(2) ; Printstring('COEFFICIENTS' ) ; CRLF(2) ;
        for i:=1 step 1 until n+1 do
        begin Print(A 〔i〕) ;
            if (i div 5) *5 =i then CRLF(2)
        end ;
        CRLF(2) ;
CALL :  POLYN(a, b, n, A, ROOT, I) ;
        if I=0 then begin Printstring('THERE IS NO ROOT.') ;
                          go to END end ;
        Printstring('REAL ROOTS') ; CRLF(2) ;
        for i:=1 step 1 until I do
        begin Print (ROOT 〔i〕) ;
            if (i div 5) *5 =i then CRLF(2)
        end ;
 END :  LFEED
end
```