九州大学学術情報リポジトリ
Kyushu University Institutional Repository

2元連立方程式の電子計算機による1解法

竹松，正樹

木村，剛三
有明工業高等専門学校教授

山下，巖

https://doi.org/10.15017/4743452

出版情報：應用力學研究所所報. 29, pp.61-66, 1968. 九州大学応用力学研究所
バージョン：
権利関係：

KYUSHU UNIVERSITY

寄　　書

2元連立方程式の電子計算機による1解法

竹　松　正　樹
木　村　剛　三*
山　下　　巌

1. 層流安定の問題に関連して，（複素）固有値方程式を解く必要に迫られた．この方程式を分離すると，実変数の2元連立方程式に帰着できるけれども，関数自身が複雑で，いくつかのパラメーターを含み，加えて超越方程式であることから，代数方程式のこれまでに見られるような解法では，到底手に負えなくなり，このプログラムを考案した．

従来の解法では，根の第1近似をなんらかの方法で予測して，それの精度を高めていくというやり方であった．たとえば，1つの超越方程式の根を求める場合にも，なんらかの方法で根の第1近似を求め，それから出発して Newton-Raphson 法あるいは regula falsi（左右挟撃）法により近似の精度を高めるという方法で，いずれにしても，根の第1近似を予測し，関数の微係数を必要とする煩わしさを免れえない．まして連立方程式ともなれば，根の第1近似の予測など困難である．

そこで，我々は regula falsi 法を用いて，自動的に根の第1近似をピックアップして，その精度を高めていくという方法で，根の第1近似も関数の微係数も必要とすることなしに，ただ単に機械的に関数の符号のみを調べていく方法を考えた．従って，そういう点では数学的デリカシィーに欠けるが，関数の性質には，比較的無頓着に使用することができるので，非常に応用範囲の広い実用的なプログラムである．その上，物理・工学の分野では，このような種類の問題には数多く遭遇するであろうと思うので，このプログラムをここに公表する．

2. 我々が実際扱った問題の原形ともいうべきものを定式化すると，

問題:

$$\begin{cases} f_1(x,y)=0 \\ f_2(x,y)=0 \end{cases}$$
$$0 \leqq x, y \leqq 1$$

の根を，1×10^{-4} の精度で求める．

ただし，f_1, f_2（1価連続）は初等関数に限る必要はなく，コンピューターで扱いうる関数であれば，この方法で解くことができる．

3. プログラム

*有明工業高等専門学校教授，機械工学教室

```
begin real  H , P , XZ , YZ , XV , YV , HABA ;
 integer    I , IK , JC , JN , JV , KETA , SGR , SGI ;
 array      X , Y [0 : 4 , 1 : 20] , FB , FA [0 : 10 , 1 : 2] ;
 procedure   KEISAN (X , Y , F) ;
    real  X , Y ;     array  F ;
 begin      F [I , 1] :=f1(x , y) ;  F [I , 2] :=f2 (x , y) ;  end ;
 procedure   SAGASÕ ;
 begin      XZ : =X [IK-1 , JC] ;  YZ : =Y [IK-1 , JC] ;
  for       I: =0    step 1    until 10    do
  begin     XV : =XZ+H * FLÕAT (I) ;
            KEISAN (XV , YZ , FA) ;
  end ;
  for       YV: =YZ+H  step H  until YZ+HABA  do
  begin     FB [I , 1] : =FA [I , 1] ;  FB [I , 2] : =FA [I , 2] ;
            XV: =XZ+H * FLÕAT (I) ;
            KEISAN (XV , YV , FA) ;
  end ;
  for       I : =0    step 1    until 9    do
  begin     SGR : =SGI : =0 ;
    if      FB [I , 1] * FA [I , 1]≦0   ∨  FA [I , 1] * FA [I+1 , 1]≦0
    ∨       FA [I+1 , 1] * FB [I+1 , 1]≦0    then  SGR : =−1 ;
    if      FB [I , 2] * FA [I , 2]≦0   ∨  FA [I , 2] * FA [I+1 , 2]≦0
    ∨       FA [I+1 , 2] * FB [I+1 , 2]≦0    then  SGI : =−1 ;
    if      SGR * SGI=1    then
   begin    JV : =JV+1 ;
            X [IK , JV] : =X Z+H * FLÕAT (I) ;
            Y [IK , JV] : =YV−H ;
     if     IK=KETA    then
    begin   CRLF (5) ;  SPACE (10) ;  PRINT ('CÕEF . =') ;
            PRINT (1.6 , XZ+H * FLÕAT (I));  CRLF (2);
                        PRINT ('LAMDA=');
            PRINT (1.6 , YV-H) ;  CRLF (2) ;
            SPACE (20) ;  PRINT ('(REALbbPART)') ;
            SPACE (40) ;  PRINT ('(IMAG . b PART)') ;
            CRLF (2) ;  SPACE (10) ;  PRINT (6 , FA [I , 1]) ;
            SPACE (4) ;  PRINT (6 , FA [I+1 , 1]) ;
            SPACE (20) ;  PRINT (6 , FA [I , 2]) ;
            SPACE (4) ;  PRINT (6 , FA [I+1 , 2]) ;
            CRLF (2) ;  SPACE (10) ;  PRINT (6 , FB [I , 1]) ;
            SPACE (4) ;  PRINT (6 , FB [I+1 , 1]) ;
            SPACE (20) ;  PRINT (6 , FB [I , 2]) ;
            SPACE (4) ;  PRINT (6 , FB [I+1 , 2]) ;
    end ;
   end ;
  end ;
end  SAGASÕ ;
```

```
HAJIME :
  READ (P , KETA) ;
  H : =P ;  X [0 , 1] : =0.0 ;  Y [0 , 1] : =0.0 ;  JV : =1 ;
  for  IK : =1  step 1  until  KETA  do
  begin  JN : =JV ;  H : =0.1 * H ;
         HABA : =10.1 * H ;  JV : =0 ;
    if   IK=KETA    then
    begin  PAGESKIP (1) ;  CRLF (3) ;
           SPACE (5) ;  PRINT ('MESHbWIDTHbH=') ;
           PRINT (1.6 , H) ;
    end ;
    for  JC : =1  step  1  until  JN  do
         SAGASŌ ;
  end ;
end ;
(データー)
  1.0,  4 ,
```

・パラメーターの説明.

H;　mesh の幅.

P;　x, y の変域の上限.

X, XZ, XV;　変数 x.

Y, YZ, YV;　変数 y.

HABA;　与えられた変域の y 方向の等分数.

I;　与えられた変域の x 方向の等分数.

IK, KETA;　求める根の桁数.

JC, JN;　procedure SAGASŌ の実行回数.

SGR, SGI;　判断文に入る前では, SGR : =SGI : =0 で判断をして yes であれば, SGR : =−1, SGI : =−1 とおきかえられる判定用のパラメーター.

FB;　各格子点の下側での $f_1(x,y)$, $f_2(x,y)$ の関数値.

FA;　各格子点の上側での $f_1(x,y)$, $f_2(x,y)$ の関数値.

array の変域で

0 : 4　は桁数より決定されるもの.

1 : 20　は関数の性質より決定されるもの.

0 : 10　は x 方向の等分数より決定されるもの.

1 : 2　は procedure KEISAN の本体の関数の個数より決定されるもの.

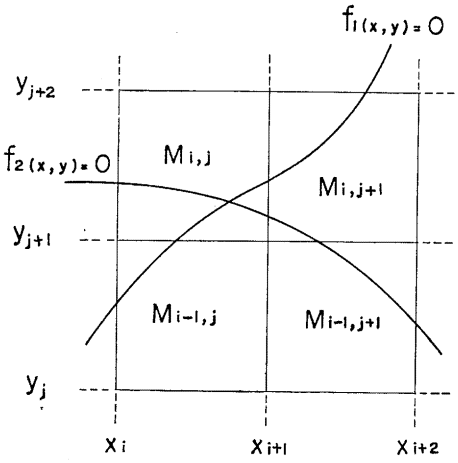
・プログラムの説明.

この解法は, 本質的には regula falsi 法の応用に他ならず, 先づ指定された (x-y) 面の領域を適当な大きさの mesh に分ける. それぞれの mesh $M_{i,j}$, $M_{i,j+1}$, ……に属する各格子点における $f_1(x,y)$, $f_2(x,y)$ の値を計算し, 相隣る格子点の符号を比較して $f_1(x,y)=0$, $f_2(x,y)=0$ の軌跡が通るかどう

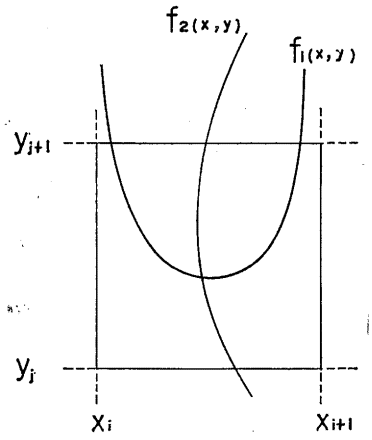
かを判定し，それぞれの軌跡が同時に通る mesh（第1図でいえば，$M_{i,j}$，$M_{i,j+1}$）を拾いあげ，mesh の左下すみの格子点 (x_i, y_{j+1})，(x_{i+1}, y_{j+1}) で代表させ，メモリーに記憶させる．こうして $(x\text{-}y)$ 面の領域のすべての mesh について，同じことを調べていく．次に，その拾いあげた mesh（1つとは限らない）について，同様の操作をくり返し，根の存在する領域に収束させていく．

なお，SAGASŌ の最終段階（IK が指定した桁数に達した時）に達すれば，格子点の値とその点における関数の値をプリントさせるようにプログラムしてるので，必要であれば，この関数値をもとに，graphical にその精度を一段高めることもできるし，また，重根であるか否かも，この方法で判定することができる．

以上説明したことからも明らかなように，この方法は単に関数の符号のみを判定して，根を求めていくのであるから，mesh の幅が適当でなかったり，急激に変化する関数においては，第2図の $f_1(x,y)$



第　1　図



第　2　図

のようなことがおこり，その mesh を拾い出さない。従って，この方法の適用は，いわゆる“おとなしい”関数に限られる．また，

$$\left.\frac{\partial f_i}{\partial x}\right|_{\substack{x=x_0\\y=y_0}} = \left.\frac{\partial f_i}{\partial y}\right|_{\substack{x=x_0\\y=y_0}} = 0 \quad (i=1 \text{ or } 2)$$

となるような根 (x_0, y_0) は求めることができないので注意する必要がある．

・プログラムのテスト．

(問題1)

$$\begin{cases} xy - 0.1 = 0 \\ y - kx = 0 \end{cases}$$

の $k = 0.25,\ 0.50,\ 0.75,\ 1.00$ に対する根 $(x, y \geqq 0)$ を求める．

(結　果)

最終 mesh の大きさを 0.001 で打切ったので，1つの単根に対し $f_1(x,y) = 0$，$f_2(x,y) = 0$ が

同時に通る mesh を2～3ケ拾い出すけれども，各格子点における関数値が計算してあるから，どの mesh に根が属するかは容易に判定できる．また，このプログラムではプログラムの便宜上，各格子点の左下すみの座標をもってその mesh を代表させたが，根の属する mesh の中心座標をもってその解（xおよびy）とすれば，真の解との誤差は± 0.0005の範囲におさえられるので，そのxおよびyを第1表に記す．

第　1　表

k	x	y
0.25	0.6325	0.1585
0.50	0.4475	0.2235
0.75	0.3655	0.2735
1.00	0.3165	0.3165

（問題2）

$$z^6-2z^5+6z^4-58z^3+633z^2-200z+2500=0$$

の複素根を求める．先づ，これを $z=x+iy=re^{i\theta}$ とおき，実，虚部に分離して，

$$\begin{cases} r^6\cos 6\theta-2r^5\cos 5\theta+6r^4\cos 4\theta-58r^3\cos 3\theta+633r^2\cos 2\theta-200r\cos\theta+2500=0 \\ r^5\sin 6\theta-2r^4\sin 5\theta+6r^3\sin 4\theta-58r^2\sin 3\theta+633r\sin 2\theta-200\sin\theta=0 \end{cases}$$

のr,θ面で，$0\leqq\theta\leqq\pi$なる根を求める．

最後にもとの x, y にかえす．

（結　果）

この場合のxおよびyは最終 mesh の大きさが0.0000009なので，真の解との誤差は± 0.00000045の範囲である．

第　2　表

x	y
0.00000005	± 2.0000035
4.0000035	± 3.0000035
-2.9999955	± 4.0000035

4.　このプログラムは関数の性質に左右されることなく，機械的に根を求めることができるというところに最大の特徴を有し，応用範囲が広いことから，比較的多くの人に対して有効な手段となると思い，これを我々のみのプログラムとせず，広く計算機利用者に公表し，計算機利用者の便宜をはかり，実用に供されれば，我々としても，この上ない光栄である．また，このプログラムが，これで完璧でないのは言うまでもないが，今後の改良上の指針として，ユーザーのアイデアおよび批判などうかがえれば望外の幸いである．

なお，このプログラムの発表を示唆され，色々御教示をいただいた九州大学応用力学研究所の岡部淳一教授に深甚の謝意を表する．

最後に，このノートの問題1，問題2の計算は九州大学中央計数施設の OKITAC 5090H で行なわれたことを付記しておく．

（昭和43年6月29日 受理）