2変数連立方程式の全解列挙アルゴリズム

日大生産工(学部)　○茂木　渉
日大生産工　　篠原　正明

１．　はじめに

前論文「2変数再帰形はさみうち法の表計算」では、はさみうち法の2変数連立方程式への拡張と表計算インプレメントについて説明し、初期区間（前論文5節中STEP1の条件を満たすかどうか）により解に収束しない場合がある。また、区間内に複数解が存在するときには、どの解が求められるか予測ができず､全ての解を同時に求めることはできない。

本論では、複数解を持つ2変数非線形連立方程式において、一般的に困難とされる全解列挙を行うアドインソフトをMicrosoft ExcelのVBA(Visual Basic for Applications)により開発したので、そのアルゴリズムと実装例を示す。

２．　基本アルゴリズム

前論文と同様に、解法アルゴリズムには、微分不可・不連続を含む場合においても適当な区間を与えることにより1つの解に確実に収束する点から、はさみうち法（二分法）をベースとし、これを全ての解を求められるように拡張する。

３．　全解列挙法

全解列挙をするためには初期区間（単純なる広域区間で良い）を細かい区間に分割し、その区間全てに対して再帰形はさみうち法を適用する方法で行う。以下に1変数1方程式及び2変数2方程式の場合のアルゴリズムを示す。ただし、分割された区間(x_l, x_u) (y_l, y_u)に解があれば、$f(x_{new}, y_{new}) \cdot f(x_u, y_u) \le 0$または$f(x_l, y_l) \cdot f(x_{new}, y_{new}) \le 0$を満たすほど細かい区間であると仮定する。

また、全区間に対して再帰形はさみうち法を適用するため、計算時間がかかることが予想されるので、2変数2方程式では、再帰形はさみうち法の余分な計算を修正する。

３.１　1変数1方程式

変数xについての方程式(１)について考える。

$$f(x) = 0 \qquad (1)$$

STEP1.　xの初期区間を設定する。
$(x_{\min}, x_{\max})$

STEP2.　xの初期区間を細かい範囲に分割する。
$(x_{\min}, \cdots, x_l, x_u, \cdots, x_{\max})$

STEP3.　区間(x_l, x_u)において、
$x_{new} = (x_l + x_u)/2$とする。

STEP4.　$f(x_{new}) \cdot f(x_u) \le 0$ならば、$x$の区間を$(x_{new}, x_u)$として$x_{new}$を$x_l$に、そうでなければ$x$の区間を$(x_l, x_{new})$として$x_{new}$を$x_u$に更新する。

STEP5.　区間(x_l, x_u)が十分小さいならば更新を終了し、そうでなければSTEP3へ。
また､このときの$f(x)$が十分小さいならば解として出力する。

STEP6.　x_l, x_uを次の区間に移し、STEP3へ。
全ての区間について調べ終わったら終了する。

３.２　2変数2方程式

変数x, yについての連立方程式(２), (３)について考える。

$$f(x, y) = 0 \qquad (2)$$
$$g(x, y) = 0 \qquad (3)$$

STEP1.　x, yの初期区間を設定する。
$(x_{\min}, x_{\max})$、$(y_{\min}, y_{\max})$

All-solution Enumeration Algorithm for Two-variable Simultaneous Equation

Wataru MOGI and Masaaki SHINOHARA

STEP2. x, yの初期区間を細かい範囲に分割する。
$(x_{\min}, \cdots, x_l, x_u, \cdots, x_{\max})$,
$(y_{\min}, \cdots, y_\alpha, y_\beta, \cdots, y_{\max})$

STEP3. STEP12～14のx_0をx_uと置いて計算したy_0をy_uとする。

STEP4. $f(x_u, y_u)$を計算し、これをf_1とする。

STEP5. $x_{new} = (x_l + x_u)/2$とおく。

STEP6. STEP12～14のx_0をx_{new}と置いて計算したy_0をy_{new}とする。

STEP7. $f(x_{new}, y_{new})$を計算し、これをf_2とする。

STEP8. $f_1 \cdot f_2 \le 0$ならば、xの区間を(x_{new}, x_u)としてx_{new}をx_lに更新する。
そうでなければxの区間を(x_l, x_{new})としてx_{new}をx_uに更新し、f_1をf_2とする。
区間(x_l, x_u)が十分小さいならば更新を終了し、そうでなければSTEP3へ。

STEP9. このときの$f(x, y)$及び$g(x, y)$が十分小さいならばx, yを解とする。

STEP10. y_α, y_βを次の区間に移し、STEP3へ。区間$(y_{\max-1}, y_{\max})$まで調べ終われば STEP11へ。

STEP11. x_l, x_uを次の区間に移し、STEP3へ。区間$(x_{\max-1}, x_{\max})$まで調べ終わったら終了する。

STEP12. $x = x_0$と置く。$g(x_0, y)$はyについての1変数方程式と考えられる。

STEP13. 区間(y_α, y_β)において、
$y_\gamma = (y_\alpha + y_\beta)/2$とする。

STEP14. $g(x_0, y_\gamma) \cdot g(x_0, y_\beta) \le 0$ならば、$y$の区間を$(y_\gamma, y_\beta)$として$y_\gamma$を$y_\alpha$に、そうでなければ$y$の区間を$(y_\alpha, y_\gamma)$として$y_\gamma$を$y_\alpha$に更新する。
区間(y_α, y_β)が十分小さいならば、このときのyをy_0とする。そうでなければSTEP13へ。

４． 全解列挙プログラムの仕様設定

全解列挙プログラムを作成するにあたっては、以下のような仕様を設定した。

・Microsoft ExcelのVBAを用いて作成する。これは、Sheet上プログラミングではn回のFor文やDo-Loop文などが使用できず、全てをセルの代入文及び関数で作成しようとするとセルの数も膨大となるためである。また、ExcelのVBAならば、セルに数値を書き込むことで、関数評価も容易に行える。

・3.2節の、はさみうち法をベースとしたアルゴリズムに沿ってプログラムを組む。

・区間(x_l, x_u)及び区間(y_α, y_β)は、差が10^{-8}以下で更新を終了する。

・$f(x, y)$、$g(x, y)$は10^{-4}で十分収束したと考える。

・ユーザインタフェースにはGoal Seek機能、及び参考文献[２]の多変数Goal Seek機能のものを踏襲する。

５． 実装例

以下の2変数連立方程式で実行を確認する。

例1.

$$f(x, y) = 2x - y + 3 \qquad (4)$$
$$g(x, y) = x + 3y - 5 \qquad (5)$$

収束値　$x = -0.5714$、$y = 1.8571$

例2.

$$f(x, y) = |x| - y \qquad (6)$$
$$g(x, y) = \max\{0.5x + 2, -x + 3\} - y \qquad (7)$$

収束値１　$x = 4$、$y = 4$
収束値２　$x = 4$、$y = 4$
（同じ収束値が2パターン求められる）

例3.

$$f(x, y) = x^2 + y^2 - 25 \qquad (8)$$
$$g(x, y) = y - \max\{0.5x + 2, -x + 3 - y\} \qquad (9)$$

収束値１　$x = -3.7081$、$y = 3.3541$
収束値２　$x = 3.3761$、$y = 3.6881$

例4.

$$f(x, y) = y^3 - 4(x^3 + 1) \qquad (11)$$
$$g(x, y) = e^y - e^x - e^{-x} - e^{1/y} - 3 \qquad (12)$$

収束値　$x = 1.0596$、$y = 2.0613$

例5.

$$f(x, y) = x^2 + (y-1)^2 - 4 \qquad (13)$$
$$g(x, y) = (x-3)^2 + (y-1)^2 - 4 \qquad (14)$$

収束値１　$x = 1.5$、$y = 2.3229$
収束値２　$x = 1.5$、$y = -0.3229$
収束値３　$x = 1.5$、$y = 2.3229$
（同じ収束値が2パターン求められる）

また、3.2節STEP9を条件として収束値を出力させているので当然であるが、それぞれの

$f(x,y)$、$g(x,y)$は0に近づいている

６．終わりに

6節の実装例からも分かるように、例1～5の2変数連立方程式における全ての解を求めることができた。

しかし、分割した全ての区間において再帰形はさみうち法を行っているため、計算時間が非常にかかるのが問題点である。

計算時間の具体例を挙げると、x及びyの初期区間は共に$(-5,5)$、分割範囲も共に0.5とした場合、計算に要した時間は約12分であり、分割範囲を共に0.25とした場合、計算に要した時間は約50分である。このことから推測すると、仮に初期区間の範囲をそれぞれn倍に広げると計算時間はn^2倍に、分割範囲もそれぞれ$1/n$倍すると計算時間は同じくn^2倍となってしまう（したがってこの場合であれば、計算時間はn^4倍）。よって、基本アルゴリズムは完成したが、何らかの措置を施さない限り実用はできない。

この点の改良案を次の「全解列挙アルゴリズムの高速化」で提案する。

７．　参考文献

[１] 川口博子、篠原正明：「非線形連立方程式の再帰形解法」『第35回 日本大学生産工学部学術講演会数理情報部会講演概要』(2002.12)、pp67-70

[２] 川口博子、篠原正明：「不連続・微分不可関数を含む非線形連立方程式の多変数Goal Seek機能」『第38回 日本大学生産工学部学術講演会数理情報部会講演概要』(2005.12)、pp75-78

[３] 茂木渉、篠原正明：「2変数再帰形はさみうち法の表計算」『第40回 日本大学生産工学部学術講演会数理情報部会講演概要』(2007.12)

付録

[全解列挙メインプログラム]

```
Dim xAnswer(9) As Double
Dim yAnswer(9) As Double
Dim n As Integer

Function Main()

  '変数の宣言
  Dim xUp As Double
  Dim xLow As Double
  Dim xNew As Double
  Dim xMax As Double
  Dim xMin As Double
  Dim xKukan As Double
  Dim yUp As Double
  Dim yLow As Double
  Dim yNew As Double
  Dim yMax As Double
  Dim yMin As Double
  Dim yKukan As Double

  Dim f1 As Double
  Dim f2 As Double
  Dim g1 As Double
  Dim g2 As Double

  '初期区間と区間分割範囲の設定
  xMax = 5
  xMin = -5
  yMax = 5
  yMin = -5
  xKukan = 0.5
  yKukan = 0.5

  '発見した解の個数の初期設定
  n = 0

  'はさみうち法実行プログラム
  For i = xMin To (xMax - xKukan) Step
      xKukan

    For j = yMin To (yMax - yKukan) Step
        yKukan

      xLow = i
      xUp = xLow + xKukan
      Range(UserForm1.Variable1.Value)
          .Value = xUp

      yLow = j
      yUp = yLow + yKukan

      Do
        yNew = (yLow + yUp) / 2
        Range(UserForm1.Variable2.Value)
            .Value = yNew
        g1 = Range(UserForm1
            .FormulaCell2.Value).Value
        Range(UserForm1.Variable2.Value)
            .Value = yUp
        g2 = Range(UserForm1
            .FormulaCell2.Value).Value

        If (g1 * g2) <= 0 Then
          yLow = yNew
        Else
          yUp = yNew
        End If

      Loop While ((yUp - yLow) >
          0.00000001)
      Range(UserForm1.Variable2
          .Value).Value = yUp
```

```
f1 = Range(UserForm1
     .FormulaCell1.Value).Value

 Do
    xNew = (xLow + xUp) / 2
    Range(UserForm1.Variable1
         .Value).Value = xNew

    yLow = j
    yUp = yLow + yKukan

    Do
       yNew = (yLow + yUp) / 2
       Range(UserForm1.Variable2
            .Value).Value = yNew
       g1 = Range(UserForm1
            .FormulaCell2.Value).Value
       Range(UserForm1.Variable2
             .Value).Value = yUp
       g2 = Range(UserForm1
            .FormulaCell2.Value).Value

       If (g1 * g2) <= 0 Then
          yLow = yNew
       Else
          yUp = yNew
       End If

    Loop While ((yUp - yLow) >
          0.00000001)

    Range(UserForm1.Variable2
         .Value).Value = yNew
    f2 = Range(UserForm1
         .FormulaCell1.Value).Value

    If (f1 * f2) <= 0 Then
       xLow = xNew
    Else
       xUp = xNew
       f1 = f2
    End If

Loop While ((xUp - xLow) >
     0.00000001)

 Range(UserForm1.Variable1
      .Value).Value = xNew
Range(UserForm1.Variable2.Value)
      .Value = yNew
f1 = Range(UserForm1
      .FormulaCell1.Value).Value
g1 = Range(UserForm1
      .FormulaCell2.Value).Value

If (Abs(f1) < 0.001) And
      (Abs(g1) < 0.001) Then
    xAnswer(n) = xNew
          yAnswer(n) = yNew
          n = n + 1
       End If

    Next j

 Next i

 Unload UserForm1

 If (n > 0) Then
    MsgBox n & "個の解が見つかりました。"
    UserForm2.Show
 End If

End Function
```

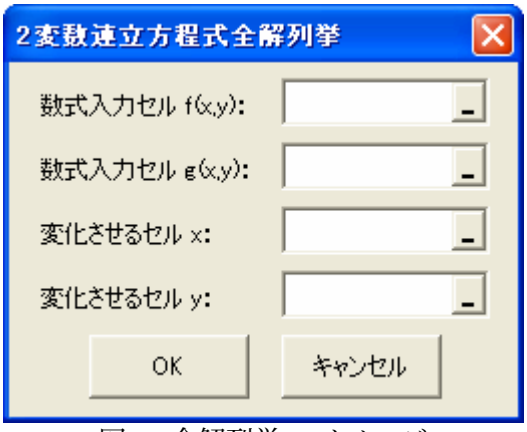

図1. 全解列挙UIイメージ

[解の出力プログラム]

```
Function Answer(AnswerCell1
    , AnswerCell2)

 For k = 0 To n - 1 Step 1
    Range(AnswerCell1).Activate
    ActiveCell.Offset(k, 0)
         .Value = xAnswer(k)
    Range(AnswerCell2).Activate
    ActiveCell.Offset(k, 0)
         .Value = yAnswer(k)
 Next k

 End

End Function
```

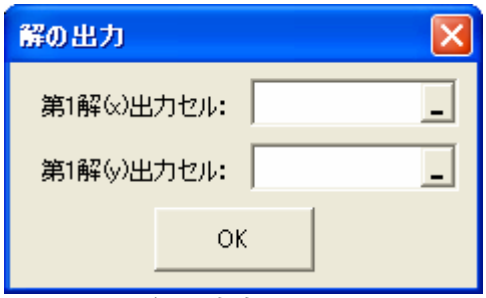

図2. 解の出力UIイメージ