

全解列挙アルゴリズムの高速化

日大生産工(学部) 〇茂木 渉
日大生産工 篠原 正明

1. はじめに

前論文「2変数連立方程式の全解列挙アルゴリズム」において、2変数連立方程式の全ての解を求める基本的なアルゴリズムとその実装例について説明をした。

ここで、分割された区間の全てに対して、2変数再帰形はさみうち法を実行すれば全ての解を求めることが可能ではあるが、計算時間が膨大にかかるという問題点が挙げられた。

そこで、本論ではアルゴリズムの改良により計算時間を短縮し、実用的なプログラムとすることを目指す。

2. アルゴリズムの改良点

計算時間のかかる1番の原因は、1節のとおり、全ての区間に対してはさみうち法を実行することである。

分割された区間に解が存在しないことがわかれば、無駄なはさみうち法を実行する必要がなくなるため、計算時間を減らせると考えられる。

3. 区分線形近似

広域的に見ると非線形関数でも、限りなく細かく分割された区間ならば線形関数と考えることができる。本アルゴリズムも限りなく細かく区間分割をしているので、区切られた範囲は線形関数と仮定できる。

線形ならば、以下の定理1が成り立つ。

定理1 2変数関数 $f(x, y)$ が分割された長方形の区間の4点 $((x_l, y_l), (x_u, y_l), (x_u, y_u), (x_l, y_u))$ の内部を通るならば、 $f(x_l, y_l) \cdot f(x_u, y_u) \leq 0$ または $f(x_l, y_u) \cdot f(x_u, y_l) \leq 0$ である。

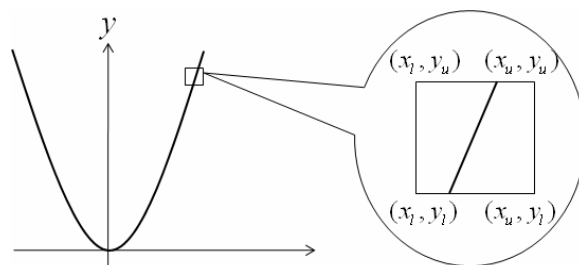


図1. 定理1の区分線形近似

また、 $g(x, y)$ についても同様に言えるので、2方程式では定理2が成り立つ。

定理2 2変数関数 $f(x, y)$ 、 $g(x, y)$ が共に分割された長方形の区間の4点 $((x_l, y_l), (x_u, y_l), (x_u, y_u), (x_l, y_u))$ の内部を通るならば、 $\{f(x_l, y_l) \cdot f(x_u, y_u) \leq 0$ または $f(x_l, y_u) \cdot f(x_u, y_l) \leq 0\}$ かつ $\{g(x_l, y_l) \cdot g(x_u, y_u) \leq 0$ または $g(x_l, y_u) \cdot g(x_u, y_l) \leq 0\}$ である。

計算を行う前に、分割区間が定理2を満たしているかを調べ、満たしていれば少なくとも、2つの関数が区間内を通るので、再帰形はさみうち法を実行する。逆にこの定理が成り立たないならば、区間内に関数が通過せず、再帰形はさみうち法を行う必要がないと言える。

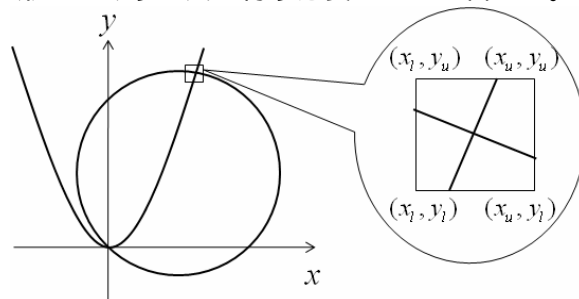


図2. 定理2の区分線形近似

4 . アルゴリズムの追加と修正

3節の定理2を2変数連立方程式の全解列挙アルゴリズムに使用する場合には、前論3.2節のSTEP2とSTEP3の間に以下のSTEP2-2を追加する。また、STEP10、STEP11を以下のように修正する。

STEP2-2. 分割された区間 $(x_l, x_u), (y_\alpha, y_\beta)$ において $\{f(x_l, y_\alpha) \cdot f(x_u, y_\beta) \leq 0$ または $f(x_l, y_\beta) \cdot f(x_u, y_\alpha) \leq 0\}$ かつ $\{g(x_l, y_\alpha) \cdot g(x_u, y_\beta) \leq 0$ または $g(x_l, y_\beta) \cdot g(x_u, y_\alpha) \leq 0\}$ を満たすならば、STEP3へ。そうでなければSTEP10へ。

STEP10. y_α, y_β を次の区間に移しSTEP2-2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、STEP2-2へ。区間 $(x_{\max-1}, x_{\max})$ まで調べ終わったら終了する。

5 . 計算時間の比較

x 及び y の初期区間を共に(-5,5)、分割範囲を共に0.25とした場合、通常的全解列挙アルゴリズムにおける計算時間は、約50分である。同じ条件で、区分線形近似を適用した例1～例5の計算時間を表1に示す。

表1 . 高速全解列挙の計算時間

例1	約15秒
例2	約30秒
例3	約20秒
例4	約25秒
例5	約18秒

通常的全解列挙アルゴリズムに比べ、各段に計算時間を減らせたことがわかる。

6 . 終わりに

5節のとおり、区分線形近似を用いたことにより、計算時間を大幅に減らすことができた。非線形関数を局所的に線形関数と見るという単純なものであるが、この理論の応用面と3変数以上の多変数関数への定理の拡張、及び全解列挙法については今後の課題である。

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)

[4]茂木渉、篠原正明：「2変数連立方程式の全解列挙アルゴリズム」『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)

補足

前論付録の全解列挙プログラムを高速化するには、次の1～3を行う。

1 . 変数f3、f4、g3、g4をdouble型で宣言する。

2 . 1つ目のDo-Loop文の前にある、

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

の間に以下のプログラムを挿入する。

```
yLow = j  
yUp = yLow + yKukan  
Range(UserForm1.Variable1  
.Value).Value = xLow  
Range(UserForm1.Variable2  
.Value).Value = yLow  
f1 = Range(UserForm1  
.FormulaCell1.Value).Value  
g1 = Range(UserForm1  
.FormulaCell2.Value).Value  
Range(UserForm1.Variable2  
.Value).Value = yUp  
f3 = Range(UserForm1  
.FormulaCell1.Value).Value  
g3 = Range(UserForm1  
.FormulaCell2.Value).Value  
Range(UserForm1.Variable1  
.Value).Value = xUp  
f2 = Range(UserForm1  
.FormulaCell1.Value).Value  
g2 = Range(UserForm1  
.FormulaCell2.Value).Value  
Range(UserForm1.Variable2  
.Value).Value = yLow  
f4 = Range(UserForm1  
.FormulaCell1.Value).Value  
g4 = Range(UserForm1  
.FormulaCell2.Value).Value  
If ((f1 * f2 <= 0) Or (f3 * f4 <= 0))  
And ((g1 * g2 <= 0) Or  
(g3 * g4 <= 0)) Then
```

3 . Next j の前にEnd If を挿入する。