

ルンゲ・クッタ法による連立非線形方程式の数値解法

2008MI287 吉田 雅俊

指導教員：杉浦 洋

1 はじめに

本論文では、常微分方程式の数値解法であるルンゲ・クッタ法を用いた連立非線形方程式の解法であるサンドの方法について研究する。非線形方程式の代表的な解法に反復法とホモトピー法があり、その関連について研究する。その際、サンドの方法について着目する。

サンドの方法はもとの方程式の解の任意の初期値に対して、それが解曲線の始点となるようなホモトピー方程式を設定し、それをルンゲ・クッタ法の1ステップで解くことを発案した。それにより解曲線の終点、すなわちもとの方程式の近似値が得られる。この近似値は初期値の改良と考えられる。この操作を繰り返すことで新しい反復法が得られる。

これをもとに数値実験を行い、その結果について考察する。

2 非線形方程式

変数 x_1, x_2, \dots, x_n に対する非線形方程式を

$$\begin{cases} f_1(x_1, \dots, x_n) = 0, \\ \vdots \\ f_n(x_1, \dots, x_n) = 0 \end{cases}$$

とする。これを簡単に、

$$f(x) = \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = 0, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \quad (1)$$

と書く。

2.1 反復法

$f(x) = 0$ の解を x^* とするとき、解 x^* に収束する点列 $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ を生成し、十分大きな k で $x^{(k)}$ を近似解として、採用する方法を反復法という。代表的な反復法として、ニュートン法がある。方程式 (1) に対して、初期近似 $x^{(0)} \cong x^*$ をとり、

$$x^{(k+1)} = x^{(k)} + d^{(k)}, \quad (2)$$

$$d^{(k)} = - \left(\frac{\partial f}{\partial x}(x^{(k)}) \right)^{-1} f(x^{(k)}) \quad (k \geq 0) \quad (3)$$

で x^* に収束すると期待される列 $x^{(k)}$ ($k \geq 1$) を生成する方法をニュートン法という。ここで $\frac{\partial f}{\partial x}$ は f のヤコビアン行列

$$\frac{\partial f}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix} \quad (4)$$

である。

2.2 ホモトピー法

一般に反復法は、初期近似 $x^{(0)}$ の取り方が難しい。 $x^{(0)}$ が解 x^* に近ければ、反復法は収束するが、遠いときには収束が保証されない。その欠点を克服する方法がホモトピー法である。

ホモトピー法では、簡単に解がわかる方程式 $g(x) = 0$ を補助的に設定する。そしてパラメーター t を含むホモトピー方程式

$$F(t, x) = 0 \quad (0 \leq t \leq 1) \quad (5)$$

を作る。ただし、

$$F(0, x) = g(x), \quad F(1, x) = f(x) \quad (6)$$

を満たすものとする。

具体例として本研究で使用したサンドのホモトピー方程式を紹介する。適当な初期値 $x^{(0)}$ をとりホモトピー方程式を

$$F(t, x) = f(x) - (1-t)f(x^{(0)}) = 0 \quad (7)$$

とする [2]。 $0 \leq t \leq 1$ に対する (5) の解を $x(t)$ と書くと

$$F(0, x^{(0)}) = f(x^{(0)}) - f(x^{(0)}) = 0 \quad (8)$$

となり、 $x(0) = x^{(0)}$ である。また、 $x(1)$ は $f(x) = 0$ の解であり、 $x(t)$ ($0 \leq t \leq 1$) は、 $x(0)$ と $x(1)$ を結ぶ \mathbb{R}^n の曲線となることが期待される。何からの方法で、始点 $x(0)$ から終点 $x(1)$ まで曲線 $H: x(t)$ ($0 \leq t \leq 1$) をたどることができれば $f(x) = 0$ の解 $x(1) = x^*$ が求まる。

1つの方法は、曲線 H を微分方程式の数値解法によりたどることである。(7) に解 $x(t)$ を代入すると

$$f(x(t)) - (1-t)f(x^{(0)}) = 0 \quad (9)$$

が成立する。両辺を t で偏微分して、

$$\left(\frac{\partial f}{\partial x} \right) \frac{dx}{dt} + f(x^{(0)}) = 0 \quad (10)$$

これより、 $x(t)$ に関する微分方程式

$$\frac{dx}{dt} = - \left(\frac{\partial f}{\partial x} \right)^{-1} f(x^{(0)}) \quad (11)$$

を得る。 $x(0) = x^{(0)}$ は既知なので、これは初期値問題である。これをルンゲ・クッタ法などの常微分方程式初期値問題の解法を用いて解き、 $x(1)$ を求める。

3 ルンゲ・クッタ法

ルンゲ・クッタ (R.K.) 法の1ステップは、 $x = x_0$ における初期値 $y_0 = y(x_0)$ から、 $x_1 = x_0 + h$ における解 $y_1 = y(x_1)$ の近似解 Y_1 を計算するアルゴリズムである。以下に代表的な3つの R.K. 法を示す.[1]

3.1 オイラー公式

オイラー公式は1段1次法である.

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(x_0, \mathbf{y}_0), \\ \mathbf{Y}_1 &= \Phi(x_0, \mathbf{y}_0; h) = \mathbf{y}_0 + h\mathbf{k}_1 = \mathbf{y}_1 + O(h^2). \end{aligned} \quad (12)$$

3.2 ホイン公式

ホイン公式は2段2次法である.

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(x_0, \mathbf{y}_0), \quad \mathbf{k}_2 = \mathbf{f}(x_0 + h, \mathbf{y}_0 + h\mathbf{k}_1), \\ \mathbf{Y}_1 &= \Phi(x_0, \mathbf{y}_0; h) = \mathbf{y}_0 + \frac{h}{2}(\mathbf{k}_1 + \mathbf{k}_2) = \mathbf{y}_1 + O(h^3). \end{aligned} \quad (13)$$

3.3 古典的ルンゲ・クッタ公式

古典的 R.K. 公式は4段4次法である.

$$\begin{aligned} \mathbf{k}_1 &= \mathbf{f}(x_0, \mathbf{y}_0), \quad \mathbf{k}_2 = \mathbf{f}(x_0 + \frac{h}{2}, \mathbf{y}_0 + \frac{h}{2}\mathbf{k}_1), \\ \mathbf{k}_3 &= \mathbf{f}(x_0 + \frac{h}{2}, \mathbf{y}_0 + \frac{h}{2}\mathbf{k}_2), \quad \mathbf{k}_4 = \mathbf{f}(x_0 + h, \mathbf{y}_0 + h\mathbf{k}_3), \\ \mathbf{Y}_1 &= \Phi(x_0, \mathbf{y}_0; h) = \mathbf{y}_0 + \frac{h}{6}(\mathbf{k}_1 + 2(\mathbf{k}_2 + \mathbf{k}_3) + \mathbf{k}_4) \\ &= \mathbf{y}_1 + O(h^5). \end{aligned} \quad (14)$$

4 サンドの方法

サンドの方法は(7)をステップ幅1の1ステップで解き,新しい近似解とホモトピー方程式を作る.これを繰り返し,解を求める反復法であることがわかった.具体的にアルゴリズムを書くと

$$\mathbf{x}^{(k+1)} = \Phi(0, \mathbf{x}^{(k)}; 1) \quad (k \geq 0) \quad (15)$$

となる. 解曲線をたどる微分方程式は,

$$\frac{dx}{dt} = - \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^{-1} \mathbf{f}(\mathbf{x}^{(k)}) \quad (16)$$

である.[2] サンド法の収束次数は用いる R.K. 法の次数 + 1 という理論がある. ちなみに, R.K. 法として, オイラー法を用いたものはニュートン法と等価で2次ある.

5 数値実験・結果

2変数 x, y の2次方程式

$$\mathbf{f}(x, y) = (9x^2 + 16y^2 - 25, 16x^2 + 9y^2 - 25) = (0, 0) \quad (17)$$

をホモトピー法とサンドの方法で数値実験した.

5.1 ホモトピー法

真の解は $(x^*, y^*) = (\pm 1, \pm 1)$ の4つである. 20通りの初期値 $(x_0, y_0) = (5 \cos \frac{\pi}{10}(l+1/2), 5 \sin \frac{\pi}{10}(l+1/2))$ ($0 \leq l \leq 19$) を用いた. ステップ数 $n = 20$, ステップサイズ $h = 1/20$ のときに描かれたホモトピー曲線を図1に示す. すべてのホモトピー曲線は初期値から解に到達している. x 軸, y 軸の近くから出発して得られた解は精度が低い. 座標軸上でヤコビアン行列が特異になるからだと思われる.

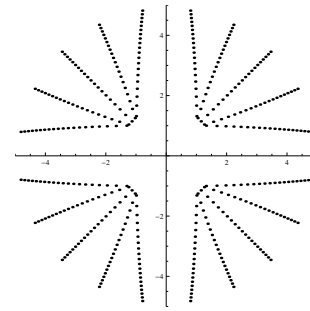


図1 ホモトピー法

k	オイラー法 e_k	ホイン法 e_k	古典的 R.K. 法 e_k
1	1.57	4.80×10^{-1}	3.55×10^{-2}
2	4.80×10^{-1}	2.81×10^{-3}	1.06×10^{-9}
3	7.78×10^{-2}	7.70×10^{-12}	2.78×10^{-47}
4	2.81×10^{-3}	4.40×10^{-46}	3.43×10^{-235}
5	3.93×10^{-6}	4.71×10^{-183}	9.92×10^{-1175}
6	7.70×10^{-12}	6.13×10^{-731}	2.00×10^{-5872}
7	2.97×10^{-23}	1.76×10^{-2922}	0
8	4.40×10^{-46}	0	0
9	9.70×10^{-92}	0	0
10	4.71×10^{-183}	0	0

表1 サンド法の収束状況

5.2 サンドの方法

オイラー法, ホイン法, 古典的 R.K. 法を用いた. まず, 初期値を $x_0 = (5 \cos(\pi/20), 5 \sin(\pi/20))$ とする. 収束の様子を詳しく見るために, 反復回数 k と誤差の2-ノルム e_k を調べたそれを示したのが表1である. 収束が急速であるため, Mathematica の多倍長演算機能を用い, 計算桁数を10000桁に設定した. この実験ではオイラー法は2次収束, ホイン法は4次収束, 古典的 R.K. 法は5次収束と判定された. しかし, この実験でホイン法は理論から予想される3次ではなく4次となった.

6 おわりに

本研究では, 非線形方程式の解法であるホモトピー法とサンドの方法について研究した. サンドの方法は結局, 解曲線を1ステップでたどるホモトピー法である.

ホモトピー法は安定して解に近づく. しかし, 高精度が要求される場合には, ステップ数が多くなり計算時間がかかる.

サンド法の収束は解の近傍ではきわめて急速であるが, 解から遠い場合は個性的で非常にゆっくり解に近づいたり, ジャンプして別の解に収束したりする. しかしほとんどの場合, ホモトピー法より速く, 解に収束する. 概して, 次数の高い R.K. 法を用いた方が収束は安定する.

参考文献

- [1] 杉浦洋:『数値計算の基礎と応用』. サイエンス社, 東京, 1997.
- [2] C. Suzuki: Proceedings of the 2009 Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, IEEE Computer Society, pp.575 - 579(2009).