

# 計算物理学 II 第 9 回：非線形方程式の解法

ver. 2020/7/26

五次以上の高次の多項式や、複雑な関数では代数解が存在しない場合が多くあります。今回は、一般的な非線形方程式  $f(x) = 0$  の解  $x$  を数値的に求める方法について説明します。

## 1 二分法

まずは  $x$  が一次元の場合、 $f(x) = 0$  を考えます。一次元の問題の場合はまずは関数のグラフを書いてみて解のおおよその位置を検討することが大事です。 $f(x)$  は図 1 のような関数であったとします。求めたいのは  $x$  軸との交点での  $x$  の値です。二分法 (bisection method) ではまず解の存在する区間を見つけ、その区間のサイズを狭めて近似解に収束させます。関数  $f(x)$  の連続な領域で、 $x = x_1, x_2$  の二点で  $f(x_1)$  と  $f(x_2)$  の符号が異符号であるものが見つかったとします。図 1 の例では

- $x_1, x_2$  で  $f(x_1) < 0, f(x_2) > 0$

であり、 $f(x)$  が連続であることから、解は  $x_1 < x < x_2$  にあります。続いてこの中点  $x_3 = (x_1 + x_2)/2$  を考えます。 $f(x_3)$  の符号は正負のどちらかですが、

- $f(x_1)$  と  $f(x_3)$  の符号が同じであれば、 $x_3$  のほうが  $x_1$  よりも解に近いので解は  $x_3 < x < x_2$  の範囲に存在
- $f(x_2)$  と  $f(x_3)$  の符号が同じであれば、 $x_3$  のほうが  $x_2$  よりも解に近いので解は  $x_1 < x < x_3$  の範囲に存在

どちらの場合でももともとの解の存在する区間  $x_1 < x < x_2$  を半分に制限することができました。プログラムの中で  $x_1$  または  $x_2$  を  $x_3$  で置き換え、再び中点を考えてその符号がどちらと同じかを調べます。図 1 では  $f(x_3) > 0$  であるため、プログラムの中で  $x_2$  を  $x_3$  で置き換えます。続いて  $f(x_4) < 0$  であるため、 $x_1$  を  $x_4$  で置き換え。 $f(x_5) > 0$  より  $x_3$  を  $x_5$  で置き換えます。このように中点を新しい境界の値とすることで解の存在する領域を半分ずつ狭めてゆき、反復法によって解を求めます。二分法は他の方法に比べて収束に時間がかかりますが (収束の精度は  $k$  回行って  $(x_2 - x_1)/2^k$ )、確実に収束するため信頼性が高いです。

## 2 はさみうち法・割線法 (セカント法)

二分法では  $x_1, x_2$  の中点を用いて解の存在する領域を狭めていきましたが、中点ではなく、 $[x_1, x_2]$  内の任意の点でも構いません。はさみうち法 (false position method/regular falsi method) では  $(x_1, f(x_1)), (x_2, f(x_2))$  を通る直線が  $x$  軸と交わる点を中点の代わりに用います。このような直線は

$$y = \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) + f(x_1) \quad (1)$$

と書けますので、 $x$  軸との交点  $x_3$  は

$$x_3 = x_1 - \frac{x_2 - x_1}{f(x_2) - f(x_1)} f(x_1) \quad (2)$$



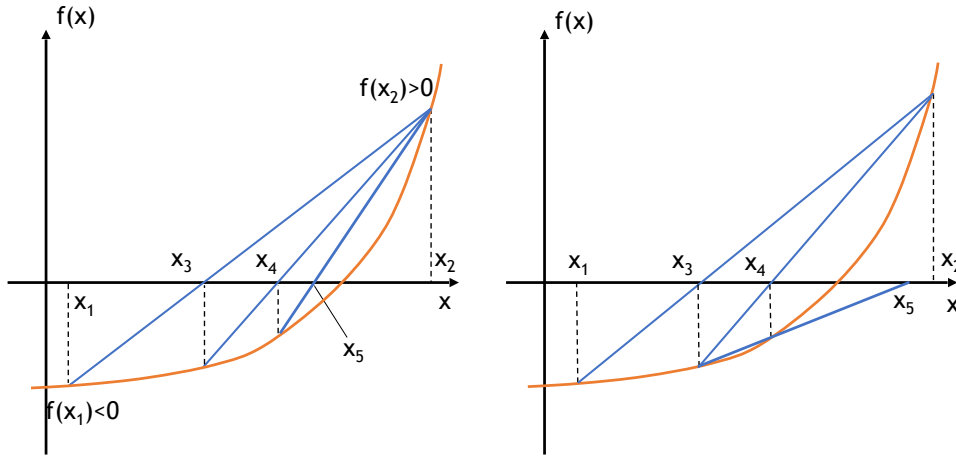


図2 はさみうち法(左図)と割線法(右図)

となります。 $\Delta x$ の一次までで近似し、 $f(x) = 0$ とすると

$$\Delta x = -\frac{f(x_n)}{f'(x_n)} \quad (5)$$

となります。つまり、 $x_{n+1}$ として

$$x_{n+1} = x_n + \Delta x = x_n - \frac{f(x_n)}{f'(x_n)} \quad (6)$$

として  $x$  の値を更新します。これは図3のように  $x = x_n$  での  $f(x)$  の接線が  $x$  軸と交わる点を  $x_{n+1}$  とすることに对应します。Newton法では  $x_n$  での微係数の値を計算する必要があります。 $f(x)$  が多項式などの場合で解析的に  $f'(x)$  が計算できる場合はそれを用いればよいですが、 $f(x)$  が複雑で直接の微分が不可能である場合(例えば行列要素が  $x$  の関数になっている行列の固有値や固有ベクトルを用いて  $f(x)$  が求められる場合、など)は数値微分を用いて  $f'(x_n)$  を評価します。このとき数値微分でやったような複雑な公式を用いる必要はなく、 $[f(x_n + \delta) - f(x_n)]/\delta$ 、あるいは  $[f(x_n + \delta) - f(x_n - \delta)]/(2\delta)$  など近似するので十分です。後者のほうがよい近似ですが、一点あたり  $f(x)$  を3回計算する必要があります。Newton法は割線法よりも早く収束するため非常によく使われますが、初期値の選び方を間違えると失敗します。特に微係数が0に近くなる場合( $f(x)$  が  $x$  軸に対して平行に近い場合)は(5)式から  $\Delta x$  が非常に大きくなるため、 $x$  軸との交点が離れた場所に飛んでしまいます。階段関数に近いような振る舞いを示す関数の場合はNewton法では収束させることができず、二分法を用いる必要があります。

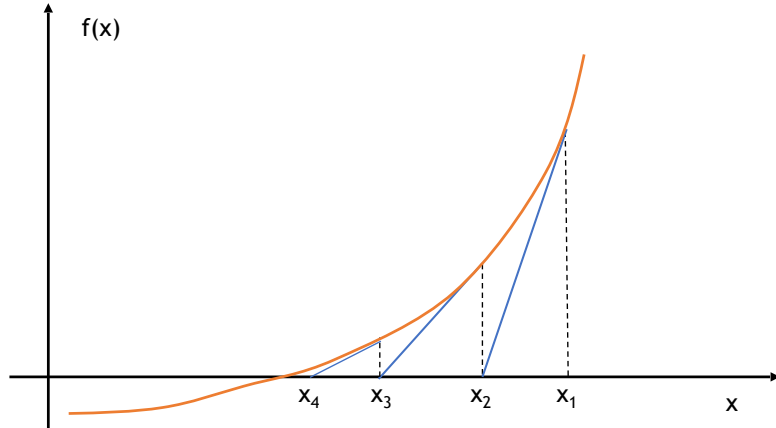


図3 Newton法

## 4 多次元の Newton 法

多変数の場合 ( $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ ) を考えます。 $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})\}$  を  $n$  個の関数として

$$f_1(x_1, x_2, \dots, x_n) = 0, \quad (7)$$

$$f_2(x_1, x_2, \dots, x_n) = 0, \quad (8)$$

⋮

$$f_n(x_1, x_2, \dots, x_n) = 0, \quad (9)$$

まとめてベクトルの形で書くと

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \quad (10)$$

を満たす方程式を解く場合を考えます。二分法のような考え方は次元の場合は解を含む領域を設定できてよいのですが多次元の場合は応用が難しくなります。一方で Newton 法であれば、多変数関数となっても同じアルゴリズムを拡張して用いることが可能です。

まず  $k$  回目の近似解  $\mathbf{x}^{(k)} = \{x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}\}$  が求まったとします。これを用いて  $k+1$  回目の近似解  $\mathbf{x}^{(k+1)}$  を Newton 法によって求めます。 $\mathbf{x}^{(k)}$  周りで  $\mathbf{f}(\mathbf{x})$  を Taylor 展開すると

$$f_i(\mathbf{x}^{(k)} + \Delta \mathbf{x}^{(k)}) = f_i(\mathbf{x}^{(k)}) + \sum_{j=1}^n \frac{\partial f_i(\mathbf{x}^{(k)})}{\partial x_j} \Delta x_j^{(k)} + \mathcal{O}((\Delta \mathbf{x}^{(k)})^2) \quad (11)$$

となります。ここで Jacobian を

$$J_{ij}(\mathbf{x}^{(k)}) \equiv \frac{\partial f_i(\mathbf{x}^{(k)})}{\partial x_j} \quad (12)$$

と書くとベクトルで

$$\mathbf{f}(\mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)}) = \mathbf{f}(\mathbf{x}^{(k)}) + J(\mathbf{x}^{(k)}) \cdot \Delta\mathbf{x}^{(k)} + \mathcal{O}((\Delta\mathbf{x}^{(k)})^2) \quad (13)$$

と書けます。この Taylor 展開を 1 次まで行った  $\mathbf{f}(\mathbf{x})$  がゼロとなる点として  $\mathbf{x}^{(k+1)}$  を選ぶと

$$\Delta\mathbf{x}^{(k)} = -J^{-1}(\mathbf{x}^{(k)}) \cdot \mathbf{f}(\mathbf{x}^{(k)}), \quad (14)$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)} = \mathbf{x}^{(k)} - J^{-1}(\mathbf{x}^{(k)}) \cdot \mathbf{f}(\mathbf{x}^{(k)}) \quad (15)$$

となります。Jacobian の計算は式 (12) が代数的に計算できない場合は数値微分によって計算し、その逆行列を求めて式 (14) に用います。逆行列の数値計算法についてはこの授業では扱いませんが、興味のある方は前年度の資料を参照してください。

## 5 Broyden 法

Newton 法は多次元の非線形方程式でも使えますが、Jacobian とその逆行列の計算の必要があり、代数的に偏微分が行えない複雑な方程式の場合や、数値微分を行わないといけない場合はこれらの計算コストが高くなってしまふことがあります。Broyden 法はセカント法を多次元に拡張したもので、Jacobian を計算することなく反復を行うことができます。

Newton 法での Jacobian  $J_{ij}(\mathbf{x}^{(k)})$  を行列  $B^{(k)}$  で近似するとします。そうすると、式 (14) は

$$B^{(k)} \cdot \Delta\mathbf{x}^{(k)} = -\mathbf{f}(\mathbf{x}^{(k)}) \quad (16)$$

となり、与えられた  $B^{(k)}$  を用いて  $\Delta\mathbf{x}^{(k)} \equiv \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$  が決まります。(初期値  $B^{(1)}$  は Jacobian を使うか、単位行列を使う場合もあります) 続いて  $B^{(k+1)}$  を多次元セカント法で求めるとすると、これは

$$B^{(k+1)} \cdot \Delta\mathbf{x}^{(k)} = \Delta\mathbf{f}^{(k)} \quad (17)$$

となります。ここで  $\Delta\mathbf{f}^{(k)} = \mathbf{f}(\mathbf{x}^{(k+1)}) - \mathbf{f}(\mathbf{x}^{(k)})$  です。この式は一次元の場合はセカント法に対応し、 $B^{(k+1)}$  は  $(x^{(k)}, f(x^{(k)}))$  と  $(x^{(k+1)}, f(x^{(k+1)}))$  を通る直線の傾きを表します。多次元の場合も同様の意味を持ちますが、一次元の場合と違い、これを解いて一意に  $B^{(k+1)}$  を求めることはできません。Broyden 法では

$$B^{(k+1)} = B^{(k)} + \frac{(\Delta\mathbf{f}^{(k)} - B^{(k)} \cdot \Delta\mathbf{x}^{(k)}) \otimes \Delta\mathbf{x}^{(k)}}{\Delta\mathbf{x}^{(k)} \cdot \Delta\mathbf{x}^{(k)}} \quad (18)$$

(ただし  $(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j$ ) とすることで、式 (17) を満たす行列  $B^{(k+1)}$  が前の反復での値  $B^{(k)}$  および  $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}, \mathbf{f}(\mathbf{x}^{(k)}), \mathbf{f}(\mathbf{x}^{(k+1)})$  の値から計算することができます。Broyden 法の計算に必要なのはこの逆行列で

$$(B^{(k+1)})^{-1} = (B^{(k)})^{-1} + \frac{[\Delta\mathbf{x}^{(k)} - (B^{(k)})^{-1} \cdot \Delta\mathbf{f}^{(k)}] \otimes \Delta\mathbf{x}^{(k)} \cdot (B^{(k)})^{-1}}{\Delta\mathbf{x}^{(k)} \cdot (B^{(k)})^{-1} \cdot \Delta\mathbf{f}^{(k)}} \quad (19)$$

で与えられます。

## 6 演習問題

第9回分の演習問題のサンプルプログラムは Fortran, C 言語それぞれ

```
$ wget https://wwwnucl.ph.tsukuba.ac.jp/~hinohara/compphys2-20/src/fortran9.tgz
```

```
$ wget https://wwwnucl.ph.tsukuba.ac.jp/~hinohara/compphys2-20/src/c9.tgz
```

とすることでダウンロードできます。解凍は

```
$ tar zxvf fortran9.tgz
```

```
$ tar zxvf c9.tgz
```

関数  $f(x) = \sin\left(\frac{x}{4}\right) - \cos\left(\frac{x}{4}\right)$  について

(22)  $f(x) = 0$  の  $0 < x < 10$  の範囲内にある解を二分法で求めよ。

(23)  $f(x) = 0$  の解を割線法で求めよ。

(24)  $f(x) = 0$  の解を Newton 法で求めよ。

関数  $f(x, y) = x^2 + y^2 - 10$ ,  $g(x, y) = x - y^3$  について

(25)  $f(x, y) = 0$ ,  $g(x, y) = 0$  の解を多次元 Newton 法を用いて求めよ。

(26)  $f(x, y) = 0$ ,  $g(x, y) = 0$  の解を Broyden 法を用いて求めよ。