

二分法とニュートン法における解の収束の速さについて

25G1065 塩澤匠生

2025 年 7 月 3 日

1 はじめに

自然界の現象をモデル化するのに方程式が用いられる．(1) 式に示されるオームの法則 [1] や (2) 式に示される等速直線運動 [2] については線形の方程式でモデル化できるが (3) 式に示されるダイオードの電流電圧特性 [3] や (4) 式に示される人口増加モデルなどに用いられるロジスティック方程式 [4] については線形であるとモデル化できないため非線形方程式のまま扱う必要がある．

$$V = IR \quad (1)$$

$$x = v_0 t + x_0 \quad (2)$$

$$I(V) = I_0 \left[\exp \left(\frac{eV}{k_B T} \right) - 1 \right] \quad (3)$$

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K} \right) \quad (4)$$

一般の非線形方程式は解析的に解けないため数値計算に頼る必要がある．非線形方程式の解を数値計算によって求める手法として中間値の定理によって解を求める二分法 [5] と接線を利用して解を求めるニュートン法 [5] というものがある．

今回はニュートン法と二分法どちらの方が速く解が収束するかを調べる．調べることでどちらの手法の方が解を速く求められるかが分かる．解が収束する速さを求める手法として c 言語でのプログラムを用いる．

2 手法

2.1 方程式の説明

本レポートでは，(5) 式に示される関数 $f(x)$ の $f(x) = 0$ の解を二分法とニュートン法により求める．この関数を図視すると図 1 のようになる．

$$f(x) = \cos x - x^2 \quad (5)$$

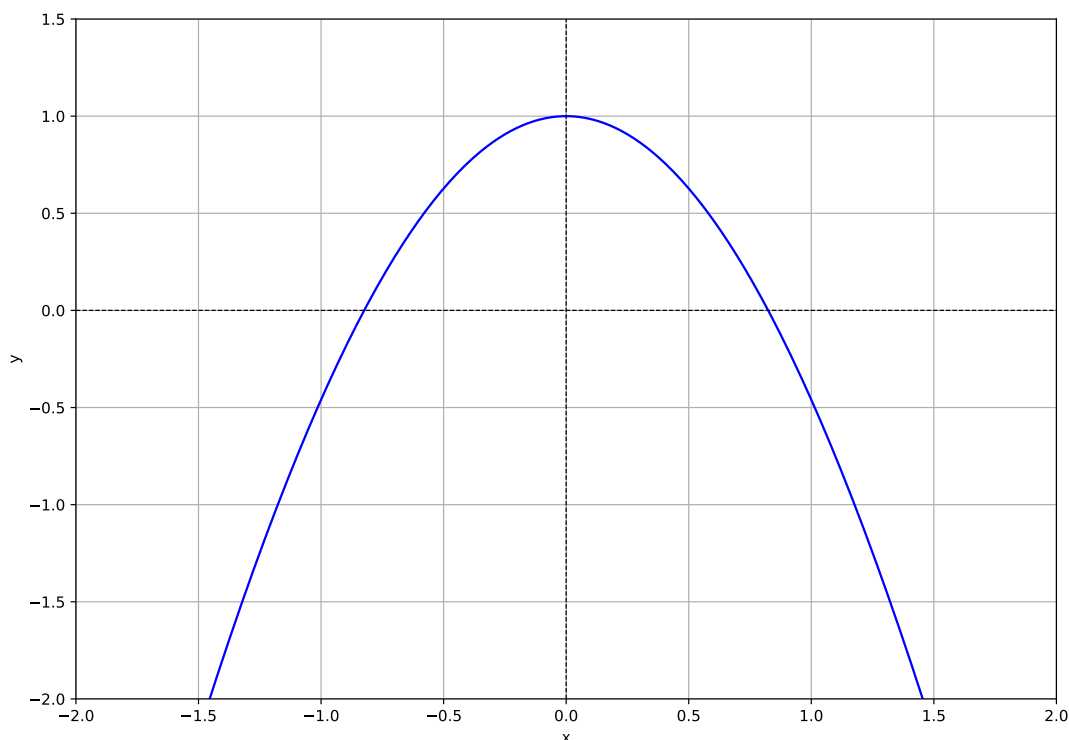


図1 $f(x) = \cos x - x^2$ のグラフ

2.2 2分法・ニュートン法の説明

まず，二分法についての解説を行う．二分法とは中間値の定理を用いて方程式の解を求める方法である．中間値の定理とは，関数 $y = f(x)$ が連続であるとき x 軸上に異なる2点 a, b (ただし, $a < b$ とする) を考えると， $f(x)$ は2点 $(a, f(a)), (b, f(b))$ を通る．この時， d を $f(a)$ と $f(b)$ の間の数とすれば，関数は連続であるから， $f(c) = d$ を満たす点 c が少なくとも1つ a と b の間に存在するというものである．この中間値の定理を応用し， $f(a)$ と $f(b)$ が異符号の場合について考える．まず，区間の midpoint $c = (a + b)/2$ を求める．もし $f(a)$ と $f(c)$ が異符号，すなわち $f(a)f(c) < 0$ であれば解は区間 $[a, c]$ 内に存在するため b を c に更新する．そうでなければ解は区間 $[c, b]$ 内に存在するため a を c に更新する．この操作を反復し，解を含む区間の幅を狭めていくことで解の近似値を得る手法が二分法である．

次にニュートン法についての解説を行う．ニュートン法とは，関数の接線を利用して方程式 $f(x) = 0$ の解を求める数値計算手法である．まず，解の初期近似値 x_0 を定める．次に，点 $(x_0, f(x_0))$ における $f(x)$ の接線と x 軸との交点を求め，これを新たな近似値 x_1 とする． n 番目の近似値 x_n が与えられたとき，次の近似値 x_{n+1} は以下の (6) 式に示される漸化式によって計算される．

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

この計算を $|x_{n+1} - x_n|$ が十分に小さくなるまで反復することで， $f(x) = 0$ の解の近似値を得る．

2.3 プログラムの仕様

まず、二分法により方程式の解を求めるプログラムの解説を行う。二分法により方程式の解を求める関数として `double bisection(double a, double b, double tol)` 関数を定義した。この関数は引数として a の座標、 b の座標、許容誤差をもつ。区分が間違っていたら -1 を返し、区分があっていたら a と b の中点 c を求め、もし c の値が負の数なら b の値を c とし、そうでなければ a の値を c とすることで、 a と c の距離を許容誤差 tol を下回るまで狭めていく。このループ中にコンソールと csv 形式のファイルにループ毎にループ回数とループ毎の a と c の距離を出力し、解が収束したら c の値を返す関数になっている。この関数を使用することで何回のループ処理で解を求めることができたかと、どのような過程で値が収束していったかと解の値を調べる。

次にニュートン法により方程式の解を求めるプログラムの解説を行う。ニュートン法により方程式の解を求める関数として `double newton(double x0, double tol, int max_iter)` 関数を定義した。この関数は引数として初期の近似解 x_0 と許容誤差 tol と最大ループ回数 max_iter をもつ。まず最初に $f(x_n)/df(x_n)$ によって得られる値を求めそのループ時の近似解 x_n と $f(x_n)$ における接線と x 軸との交点の距離 h を求める。そして x_n から h の値を引くことで次の近似解 x_{n+1} を求める。この処理を h の値が許容誤差 tol よりも小さくなり収束したと判断されるかループ回数が最大ループ回数 max_iter を上回り解が収束しないと判断されるまで繰り返す。このループ中にコンソールと csv 形式のファイルにループ回数とループ毎の h の値を出力し、解が収束したらその時の近似解の値を返す関数になっている。この関数を使用することで何回のループ処理で解を求めることができたかと、どのような過程で値が収束していったかと解の値を調べる。

2.4 評価指標

今回は解を求めるまでの処理のループ回数が少ないかと解の誤差をより速く縮められているかを速く解を求められるということの評価指標とする。

3 結果

プログラムを実行した結果得られた二分法とニュートン法の推定解が収束していく様子と繰り返し回数あたりの誤差を図 2 に示す。

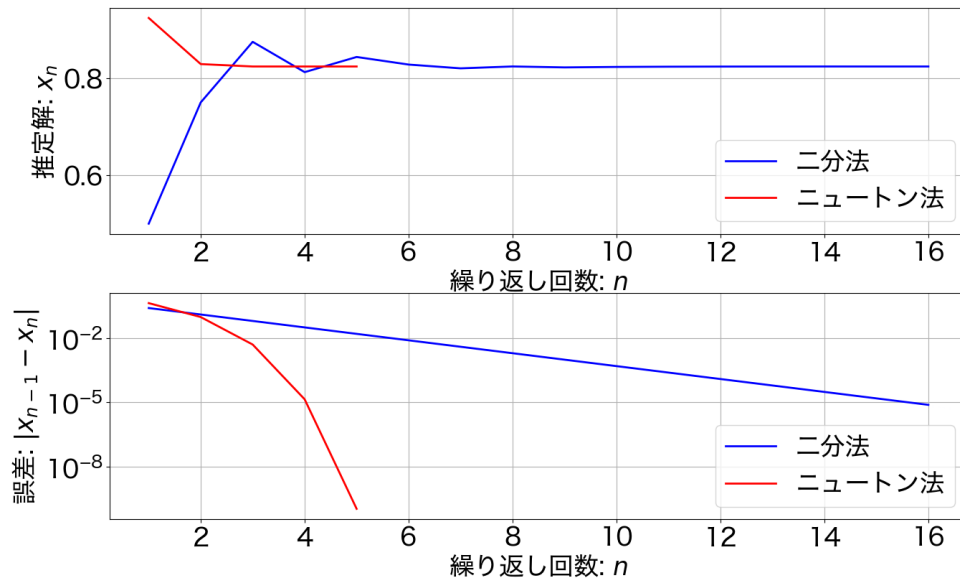


図2 上図：推定解が収束していく様子/下図：繰り返し回数あたりの誤差

結果、解はどちらも約 0.824 に収束し、二分法の解が収束するまでのループ回数は 16 回で、誤差は最終的に 10^{-5} になることと、ニュートン法の解が収束するまでのループ回数は 5 回で、誤差は最終的に 10^{-10} になることが分かった。

4 おわりに

4.1 まとめ

今回は二分法とニュートン法のどちらのほうが方程式の解をより速く求める事ができるか調査した。二分法とニュートン法それぞれの方法を使って解を求めるプログラムを制作し、それぞれ値が収束するまでに何回反復計算を行ったか調べた。結果、二分法は 16 回、ニュートン法は 5 回繰り返し計算を行っていることがわかった。

4.2 考察

まず、二分法とニュートン法のどちらが方程式の解をより速く求められるかの考察を行う。評価指標にある解を求めるのにかったループ回数に関してはニュートン法はループ回数が 5 回、二分法はループ回数が 16 回であることからニュートン法のほうが速く解を導けると考えることができる。また、もう一つの評価指標である解の誤差をより速く縮められているかという点についても誤差が 10^{-5} になるまでニュートン法はループ回数が 4 回、二分法はループ回数が 16 回であることからやはりニュートン法のほうが速く解を求められるということが分かる。この結果は早稲田大学の「非線形方程式の解法 (1) —[5]」という資料にある二分法より高速に解に収束することが多いという記述と一致するため、結果は妥当であると考えられる。

次に、計算回数が指定されていた場合どちらのほうが解に近づくことができるかについての考察を行う。図 2 の下図から誤差が小さくなる様子の比較を行うと二分法はほぼ一定の間隔で緩やかに誤差が小さくなってい

るがニュートン法は二回目以降誤差が急激に小さくなる事がわかる。このことから計算の回数が指定されている場合であっても二回以上の試行回数が与えられるならニュートン法のほうが速く解に近づくことができるという事が考えられる。

今後の調査の課題についての考察を行う。今回は計算の反復回数を評価指標としたが、一回の反復で行われる計算の回数は考慮していない。二分法は一回のループで行う処理が a 点と b 点の中間を求め、その中間の値が正か負かで値を更新するというシンプルなものだが、ニュートン法は $f(x_n)/df(x_n)$ によってそのループ時の近似解 x_n と $f(x_n)$ における接線と x 軸との交点の距離 h を求め、 x_n から h の値を引くことで次の近似解 x_{n+1} を求める処理が必要であり、二分法に比べて一回のループで行う処理の回数が多い。そのためニュートン法のほうが少ないループ回数で解を求められるとしても全体の処理時間に関しては確実にニュートン法のほうが速く処理が終わっているかについての検証を行っていないということが今回の手法の問題点であると考えられる。また、今回は (5) 式を用いての検証しか行っていないため、他の関数の場合でも同様にニュートン法の方が速く解が収束するかの検証も行っていないため、この点についても今後検証を行う必要があると考えられる。

4.3 結論

今回行った調査の結果、ニュートン法の方が少ないループ回数で非線形方程式の解を求められるということが分かった。そのため非線形方程式の解を速く求めたい時はニュートン法を用いた方が良いと言える。しかし、今回の調査ではあくまでも計算のループ回数の検証しか行っていないため、実際に計算にかかる時間については考慮していないため、今後の検証ではループ回数だけでなく 1 回のループにかかる時間や他の関数で検証しても同様の事が言えるかについての検証も行い、確実にニュートン法のほうが速く解を求められるか検証する必要がある。

参考文献

- [1] 近畿大学, オームの法則, 2025 年 7 月 2 日閲覧, <https://www.phys.kindai.ac.jp/laboratory/kondo/lectures/ichil>
- [2] 兵庫県立大学, 速さ (等速直線運動, 変位), 2025 年 7 月 2 日閲覧, <https://www.eng.u-hyogo.ac.jp/msc/msc12/HIT/html/rikigaku/pdf/Chapter-1.pdf>
- [3] 広島大学, 非線形素子 固体の電子を制御せよ, 2025 年 7 月 2 日閲覧, https://home.hiroshima-u.ac.jp/ino/lecture/SSP1slide12_ino2017.pdf
- [4] 九州産業大学, ロジスティック曲線, 2025 年 7 月 2 日閲覧, <https://design.kyusan-u.ac.jp/OpenSquareJP/?LogisticCurve>
- [5] 早稲田大学, ー 非線形方程式の解法 (1) ー, 2025 年 7 月 2 日閲覧, https://yuuka.w.waseda.jp/cpro2/Cpro2_10th.pdf