工学基礎実験実習 やさしいC言語
最終レポート

氏名：重近大智
学生番号：09501527

出題日: 2019年7月9日
提出日: 2019 年7 月16 日
締切日: 2019年7月23日

1　はじめに

本レポートでは，ニュートン法を用いて，与えられた4つの課題を解く．まず，ニュートン法の式を反復するC言語プログラムの作成方針を示す．次に，プログラムについて説明し，実験結果を示す．最後に考察とまとめを行う．一連のレポートはLaTeXを用いて作成した．

2　プログラム作成方針

3　プログラムの説明

4　実験

課題で与えられた関数に関する図，また用いたC言語プログラムのソースはレポートの最後にまとめて表示する．

表 1: 1 つ目の方程式に関する反復回数と求められた解の近似値

反復回数	求められた解の近似値
1	1.1657479108
2	1.1449182997
3	1.1447299036
4	1.1447298858
5	1.1447298858

4.1 課題 1

課題 1 で与えられた方程式は次の 3 つである．

$$\sin e^x = 0 \tag{1}$$

$$x^3 - 3x - 2 = 0 \tag{2}$$

$$x^3 - x^2 - x + 1 = 0 \tag{3}$$

まず，1 つ目の方程式 (式 1) について考える．方程式の解は，$f(x) = \sin \pi = 0$ と考えると，$e^x = \pi$ となるから，$x = \log \pi$ が解である．導関数は $f\prime(x) = e^x \cos e^x$ であるから，ニュートン法の反復式は次のようになる．

$$x_{k+1} = x_k - \frac{\sin e^{x_k}}{e^{x_k} \cos e^{x_k}} \tag{4}$$

初期値 x_0 を 1 とし，C 言語プログラムで式 4 の反復計算を行った．結果を表 1 に示す．

次に，2 つ目の方程式 (式 2) について考える．方程式の解は，$f(x) = x^3 - 3x - 2 = 0$ と考えると，$x = -1$ が解の 1 つとなる．

導関数は $f\prime(x) = 3x^2 - 3$ であるから，ニュートン法の反復式は次のようになる．

$$x_{k+1} = x_k - \frac{x_k^3 - 3x_k - 2}{3x_k^2 - 3} \tag{5}$$

初期値 x_0 を -8 とし，C 言語プログラムで式 5 の反復計算を行った．結果を表 2 に示す．ただし，収束するまで回数を要したため，最後の 5 回のみを表示する．

次に，3 つ目の方程式 (式 3) について考える．方程式の解は，$f(x) = x^3 - x^2 - x + 1 = 0$ と考えると，$x = 1$ が解の 1 つとなる．

導関数は $f\prime(x) = 3x^2 - 2x - 1$ であるから，ニュートン法の反復式は次のようになる．

$$x_{k+1} = x_k - \frac{x_k^3 - x_k^2 - x_k + 1}{3x_k^2 - 2x_k - 1} \tag{6}$$

表 2: 2 つ目の方程式に関する反復回数と求められた解の近似値

反復回数	求められた解の近似値
28	-1.0000000712
29	-1.0000000349
30	-1.0000000189
31	-1.0000000111
32	-1.0000000045

表 3: 3 つ目の方程式に関する反復回数と求められた解の近似値

反復回数	求められた解の近似値
27	1.0000001053
28	1.0000000527
29	1.0000000263
30	1.0000000132
31	1.0000000066

初期値 x_0 を 6 とし，C 言語プログラムで式 6 の反復計算を行った．結果を表 3 に示す．ただし，収束するまで回数を要したため，最後の 5 回のみを表示する．

4.2 課題 2

課題 2 で与えられた方程式は次のとおりである．

$$x^3 - 2x - 5 = 0 \tag{7}$$

この方程式 (式 7) について考える．導関数は，$f\prime(x) = 3x^2 - 2$ であるから，ニュートン法の反復式は次のようになる．

$$x_{k+1} = x_k - \frac{x_k^3 - 2x_k - 5}{3x_k^2 - 2} \tag{8}$$

課題 1 とは異なり，初期値 x_0 は 0 と決められている．また，収束の様子を詳しく調べるため，k，x_k，$f(x_k)$，および $f\prime(x_k)$ の値を表示する．このために，課題 1 とは異なる C 言語プログラムを使用した．

5　プログラムの作成に関する考察

6　まとめ

参考文献

[1] Cox D.A., Little J. and O'Shea D., *Using Algebraic Geometry*, Springer, 2005.

[2] http://mathworld.wolfram.com/NewtonsMethod.html

7　課題で出題された関数の図

次のページにまとめている.

8　C言語プログラムのソース

課題1で用いたプログラムのソース (50行) これは，式4の反復計算に用いたプログラムだが，式5，式6の計算では，元の関数，導関数，初期値を変更して使用した.

```
 1 /*
 2   ニュートン法による方程式の解法
 3
 4   コンパイル方法：gcc -lm newton-method-1.c
 5 */
 6
 7 #include <stdio.h>  /*printf を利用するのに必要*/
 8 #include <math.h>   /*各種算術関数のために必要*/
 9
10 /* f(x) = 0 の x を求める問題となる関数 */
11 double f(double x)
12 {
13     return sin(exp(x));
14 }
15
```

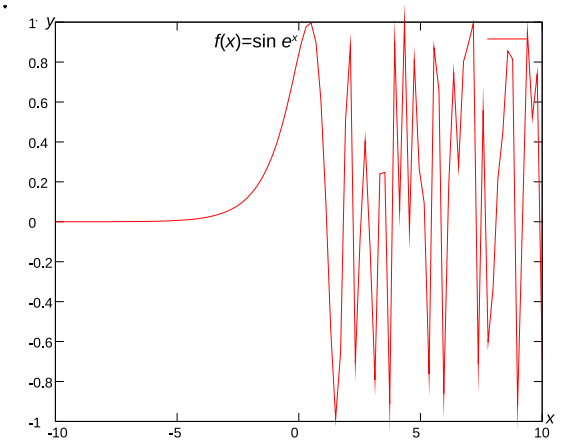

図 1: 式 1 の図

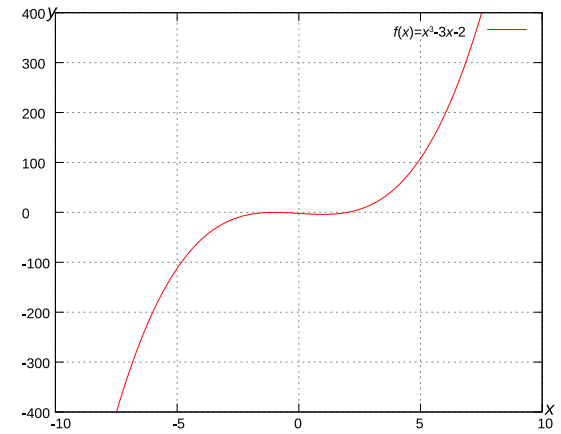

図 2: 式 2 の図

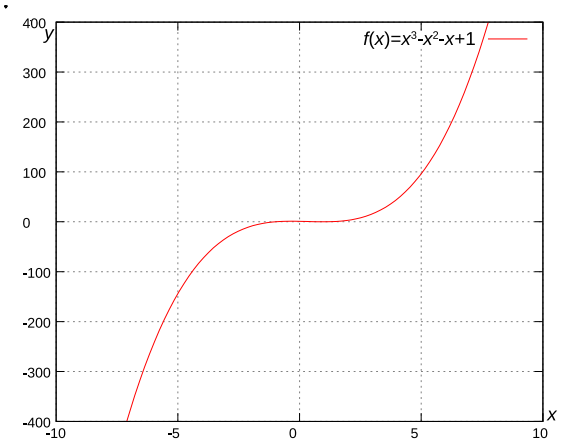

図 3: 式 3 の図

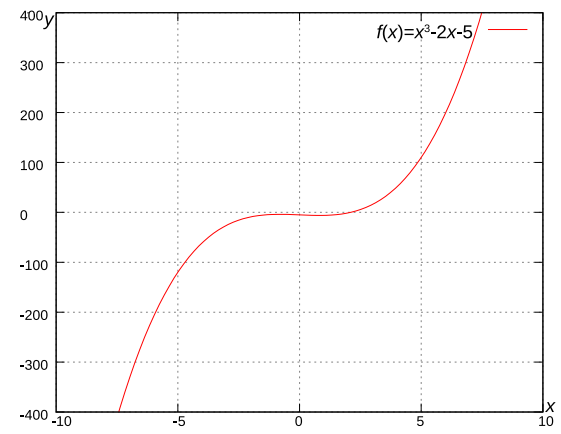

図 4: 式 7 の図

```
16 /* f の一次導関数*/
17 double f1(double x)
18 {
19     return exp(x) * cos(exp(x));
20 }
21
22 /* ニュートン法の繰り返し関数 */
23 double newton(double xk)
24 {
25     return xk - (f(xk) / f1(xk));
26 }
27
28 main()
29 {
30
31     /* 各種初期値設定 */
32     int n = 0;
 /* 繰り返し回数の初期値 */
33     double ans = log(3.14159265358979323846264338327950288419716939937510582097494459); /* 真の x */
34     double delta = 3E-16;
 /* 許容誤差 3E-16 */
35
36     double xk = 1;
 /* 初期値 */
37
38     /* 開始メッセージを表示 */
39     printf("Newton method program start.\n");
40
41     /* fabs(x) … x の絶対値を求める関数*/
42     while(fabs(f(xk)) > delta){ /* 収束条件：f(x) が delta 以下になる */
43         n = n + 1;
44         xk = newton(xk);
45         printf("n:%d xk:%.10f f(xk):%.10f ans-xk:%.10f\n"
```

```
, n, xk, f(xk), ans - xk);
    46      }
    47
    48      /* 終了メッセージを表示 */
    49      printf("done.\n");
    50  }
```

課題 2 で用いたプログラムのソース (49 行)

```
     1  /*
     2    ニュートン法による方程式の解法
     3
     4    コンパイル方法：gcc -lm newton-method-1.c
     5  */
     6
     7  #include <stdio.h>   /*printf を利用するのに必要*/
     8  #include <math.h>    /*各種算術関数のために必要*/
     9
    10  /* f(x) = 0 の x を求める問題となる関数 */
    11  double f(double x)
    12  {
    13      return x * x * x - 2 * x - 5;
    14  }
    15
    16  /* f の一次導関数*/
    17  double f1(double x)
    18  {
    19      return 3 * x * x - 2;
    20  }
    21
    22  /* ニュートン法の繰り返し関数 */
    23  double newton(double xk)
    24  {
    25      return xk - (f(xk) / f1(xk));
    26  }
    27
```

```
28 main()
29 {
30
31     /* 各種初期値設定 */
32     int n = 0;             /* 繰り返し回数の初期値 */
33     double delta = 3E-15; /* 許容誤差 3E-15 */
34
35     double xk = 0;         /* 初期値 */
36
37     /* 開始メッセージを表示 */
38     printf("Newton method program start.\n");
39
40     /* fabs(x) … x の絶対値を求める関数*/
41     while(fabs(f(xk)) > delta){ /* 収束条件：f(x) が delta 以下に
なる */
42         n = n + 1;
43         xk = newton(xk);
44         printf("k:%d xk:%f f(xk):%f f'(xk):%f\n", n, xk, f(xk),
 f1(xk));
45     }
46
47     /* 終了メッセージを表示 */
48     printf("done.\n");
49 }
```