応用プログラミングB

第4回ーB課題 **5SE-26　渡部泰生**

課題Bを以下のプログラムで解決した。

```
// 2020.05.14 課題4-B
// 4B26watanabe.c
// Made by Taiki Watanabe(5SE-26)
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

#define f(x) (atan(x - 1))
#define df(x) (1 / (1 + ((x - 1) * (x - 1))))

int main(void)
{
    double const error_threshold = 0.01;
    int count = 0;
    double x_new, x, error = 1;
    x = -1.5;

    while (error > error_threshold)
    {
        count++;
        x_new = x - f(x) / df(x);
        error = fabs(x_new - x);

        printf("【計算回数】%d回\n", count);
        printf("f(x)=%.15lf\n", f(x));
        printf("df(x)=%.15lf\n", df(x));
        printf("x_new=%.15lf\n", x_new);
        printf("error=%.15lf\n\n", error);

        x = x_new;

        if (count == 5)
        {
            printf("打ち切り\nerror=%.15lf\n", error);
            printf("x=%.15lf\n", x);
            exit(EXIT_SUCCESS);
        }
    }

    printf("終了:%d回\n", count);
    printf("x=%.15lf\n", x);

    return 0;
}
```

実行結果は以下のようになった。

```
~/Documents/Activities/学校/応用プログラミングB/4_20200514/b master*
❯ ./4B26watanabe
【計算回数】1回
f(x)=-1.190289949682532
df(x)=0.137931034482759
x_new=7.129602135198354
error=8.629602135198354

【計算回数】2回
f(x)=1.409078295657527
df(x)=0.025925526831170
x_new=-47.221397357834526
error=54.350999493032880

【計算回数】3回
f(x)=-1.550061616867805
df(x)=0.000429866586541
x_new=3558.691845027006821
error=3605.913242384841396

【計算回数】4回
f(x)=1.570515245684664
df(x)=0.000000079006588
x_new=-19874723.316293656826019
error=19878282.008138682693243

【計算回数】5回
f(x)=-1.570796276479733
df(x)=0.000000000000003
x_new=620471839689134.125000000000000
error=620471859563857.500000000000000

打ち切り
error=620471859563857.500000000000000
x=620471839689134.125000000000000
```

考察

上記の実行結果から、ニュートン法の正しい実行結果が得られたことがわかる。
今回のプログラムでは、計算を5回行って既定の誤差に収まらない場合は計算を打ち切る処理を挿入した。この打ち切りのプログラムをwhile文内に記述し、プログラムを正常終了させることで、計算の打ち切り時に結果が2回出てしまうことを防いだ。
前回の課題も振り返って、実際に二分法とニュートン法の違いについて知ることができた。
二分法に比べて、計算回数が少なく、プログラムもより簡潔に記述できることがわかった。

この2回の課題から、3次の収束のできるベイリー法も時間のある際に実装しようと思いました。