

SU-FreeSBIE で実験データを解析する

第2回 C言語で作る超お手軽最小自乗フィットプログラム1：y切片無し

撰南大学 理工学部 電気電子工学科 井上雅彦

参考：http://sprite.eng-scl.setsunan.ac.jp/sst_lab/2009/lms-2.html

まずは前回の復習です。前回は具体的に電流と電圧の測定について勉強しましたが、今回は少し一般化して、設定値 x_i 、測定値 y_i としましょう。表1のようなデータが得られたとします。図1のようにこれら5個のデータポイントをプロットしたとき、これらのデータを原点を通る直線 $y = ax$ (a は比例定数) にフィッティングすることを考えます。

表1：実験結果

(番号)	(設定値)	(測定値)
i	x_i	y_i
1	2.0	9.5
2	4.0	18.0
3	6.0	33.0
4	8.0	45.0
5	10.0	47.0

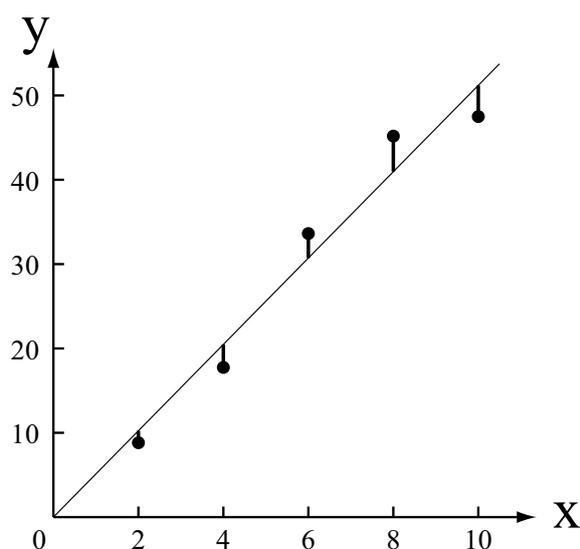


図1：原点を通る直線 $y = ax$ にフィッティングする

各設定値 x_i に対する理論値（直線上の値）は ax_i なので、各データポイントにおける残差（理論値と測定値の差）は $ax_i - y_i$ となります。従って残差自乗和は、

$$\begin{aligned}
 S(a) &= \sum_{i=1}^5 (ax_i - y_i)^2 \\
 &= \sum_{i=1}^5 (a^2 x_i^2 - 2ax_i y_i + y_i^2) \\
 &= a^2 \sum_{i=1}^5 x_i^2 - 2a \sum_{i=1}^5 x_i y_i + \sum_{i=1}^5 y_i^2
 \end{aligned}$$

$S(a)$ は下に凸の二次関数で必ず最小値を持ちます。 $S(a)$ を最小とする a を求めましょう。このとき接線の傾きは0となりますから、

$$\frac{dS(a)}{da} = 0 = 2a \sum_{i=1}^5 x_i^2 - 2 \sum_{i=1}^5 x_i y_i, \quad \therefore a = \frac{\sum_{i=1}^5 x_i y_i}{\sum_{i=1}^5 x_i^2} \quad \dots\dots (1)$$

この(1)式を使って直線の傾き a を求めるプログラムをステップバイステップで作ります。Windowsをお使いの方はVMwarePlayer（フリーウェア）を、またMacOSXをお使いの方はVMwareFusion（有償）を使ってSU-FreeSBIEを起動してください。

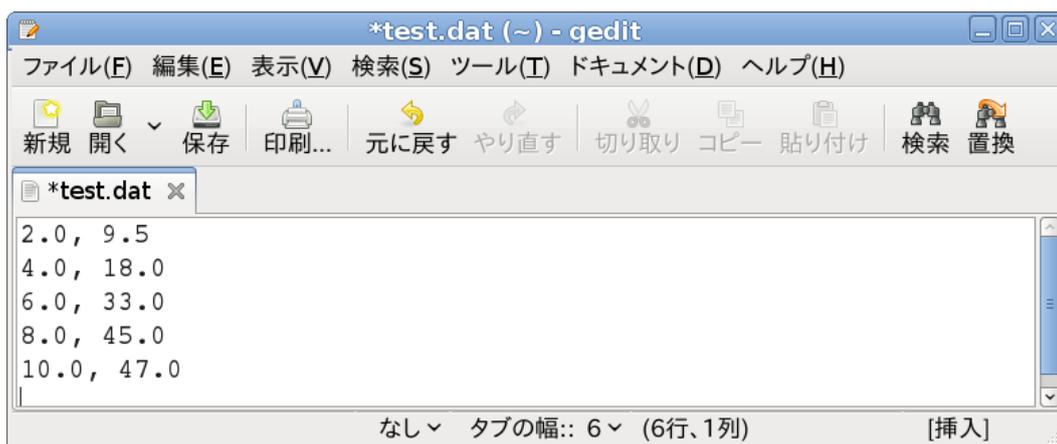


図2：gEdit でデータファイルを作成する

まず、データファイルを作ります。汎用テキストエディタ gEdit のアイコンをクリックして起動します。図2のように、一行に x と y の値を並べて書いてゆきます。 x と y の間はコンマ+スペースで区切ってください。次にデータファイルから数値を読み込み、画面に表示するプログラムを作りましょう。やはり gEdit で「ファイル」→「新規」を選び、図3のように作成し、「ファイル」→「保存」で ファイル名 test.c で保存してください。

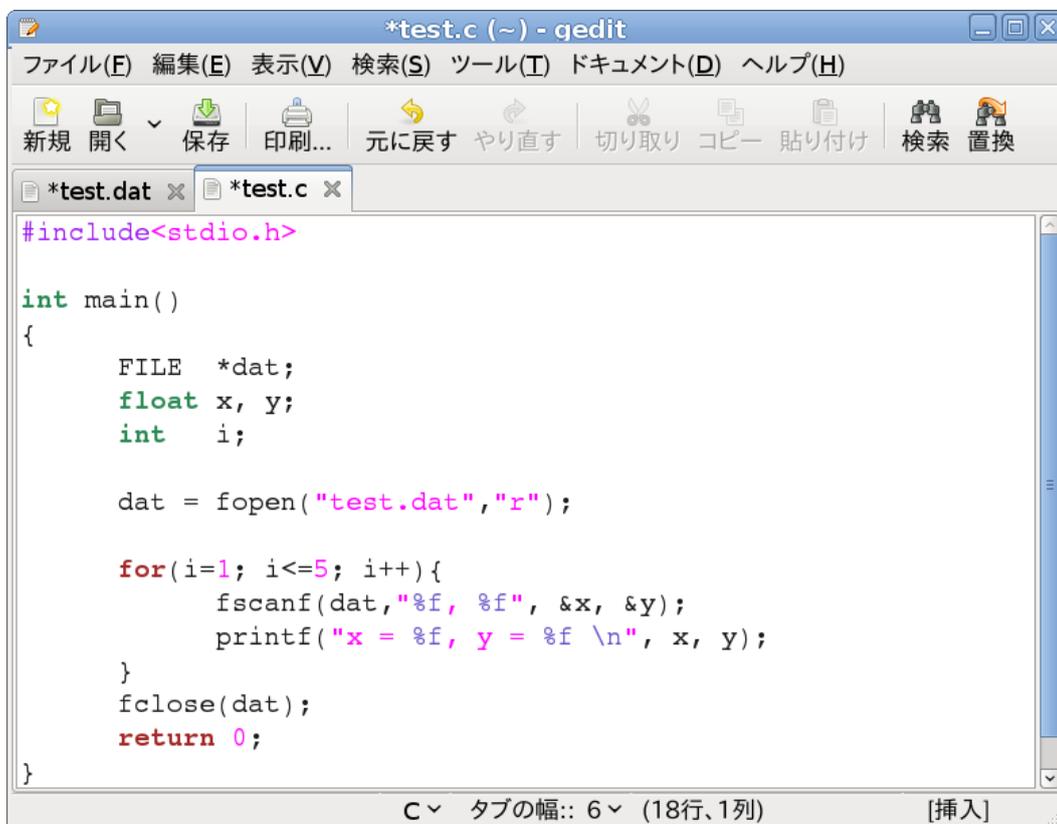


図3：test.c

端末 (terminal) を起動し、コマンドラインから コンパイルを行います。(test.c というソースファイルをコンパイルし、test という名前の実行ファイルを出力する。)

SU-FreeSBIE% cc test.c -o test (ENTER)

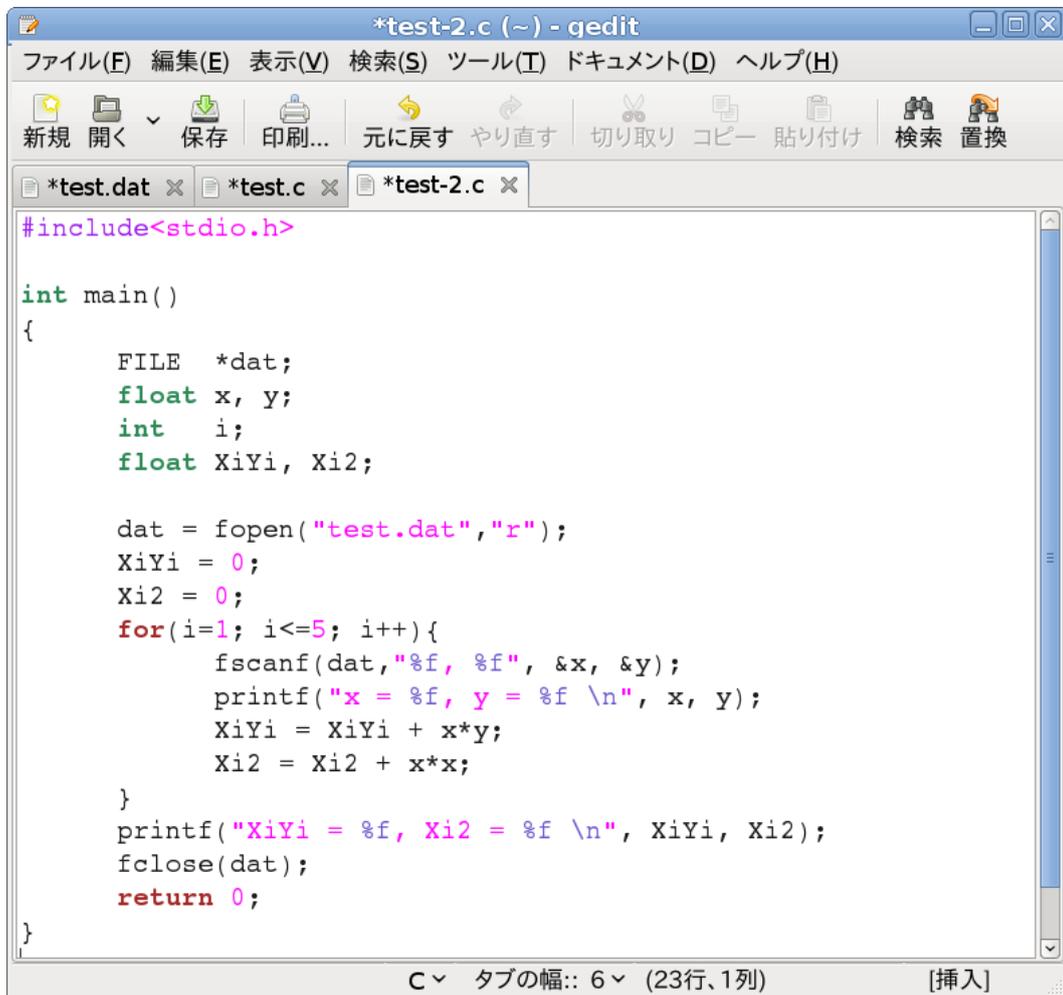
エラーメッセージが出なければ実行ファイル test ができますので、これを実行します。

```
SU-FreeSBIE% ./test (ENTER)
```

次のようにデータが画面表示されれば成功で、データファイルからコンピュータがデータを読み込んでいることが確認できます。

```
x = 2.000000, y = 9.500000
x = 4.000000, y = 18.000000
x = 6.000000, y = 33.000000
x = 8.000000, y = 45.000000
x = 10.000000, y = 47.000000
```

続いてサメンション $\sum_{i=1}^4 x_i y_i$ と $\sum_{i=1}^4 x_i^2 i$ を計算しましょう。これらの数値を表す float 型変数として、XiYi および Xi2 を定義します。これらの数値をまず 0 に初期化しておいて、for 文の中で $x*y$ あるいは $x*x$ を足してゆくことでサメンションを計算します。最後に計算結果を表示しましょう。図4に示すように test.c に6行追加し、「ファイル」→「別名で保存」を選び test-2.c という名前で保存してください。



```
*test-2.c (~) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け 検索 置換
*test.dat x *test.c x *test-2.c x
#include<stdio.h>

int main()
{
    FILE *dat;
    float x, y;
    int i;
    float XiYi, Xi2;

    dat = fopen("test.dat", "r");
    XiYi = 0;
    Xi2 = 0;
    for(i=1; i<=5; i++){
        fscanf(dat, "%f %f", &x, &y);
        printf("x = %f, y = %f \n", x, y);
        XiYi = XiYi + x*y;
        Xi2 = Xi2 + x*x;
    }
    printf("XiYi = %f, Xi2 = %f \n", XiYi, Xi2);
    fclose(dat);
    return 0;
}
```

図4 : test-2.c

コンパイルして test-2 という実行ファイルを作り、実行してください。次のように画面にサメンショ

ンの計算結果が表示されれば成功です。

```
SU-FreeSBIE% cc test-2.c -o test-2 (ENTER)
```

```
SU-FreeSBIE% ./test-2 (ENTER)
```

```
x = 2.000000, y = 9.500000
```

```
x = 4.000000, y = 18.000000
```

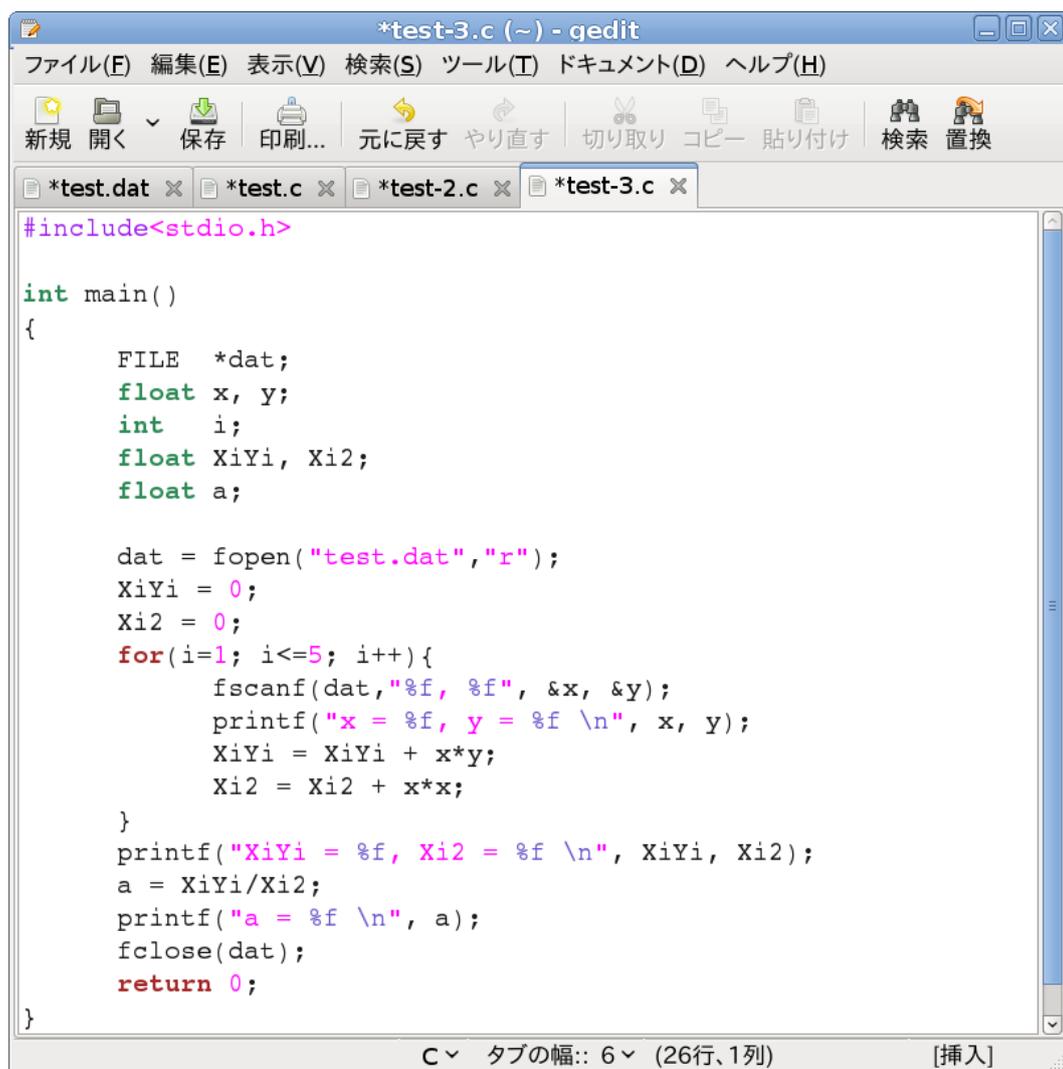
```
x = 6.000000, y = 33.000000
```

```
x = 8.000000, y = 45.000000
```

```
x = 10.000000, y = 47.000000
```

```
XiYi = 1119.000000, Xi2 = 220.000000
```

ではよいよよ仕上げに入ります。直線の傾きを表す float 型変数として a を定義します。(1) 式を使って a を計算し、画面に表示しましょう。図5に示すように、test-2.c に3行追加し、「ファイル」→「別名で保存」で test-3.c という名前でも保存してください。



```
*test-3.c (~) - gedit
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)
新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け 検索 置換
*test.dat x *test.c x *test-2.c x *test-3.c x
#include<stdio.h>

int main()
{
    FILE *dat;
    float x, y;
    int i;
    float XiYi, Xi2;
    float a;

    dat = fopen("test.dat", "r");
    XiYi = 0;
    Xi2 = 0;
    for(i=1; i<=5; i++){
        fscanf(dat, "%f %f", &x, &y);
        printf("x = %f, y = %f \n", x, y);
        XiYi = XiYi + x*y;
        Xi2 = Xi2 + x*x;
    }
    printf("XiYi = %f, Xi2 = %f \n", XiYi, Xi2);
    a = XiYi/Xi2;
    printf("a = %f \n", a);
    fclose(dat);
    return 0;
}
C v タブの幅:: 6 v (26行, 1列) [挿入]
```

図5 : test-3.c

端末 (terminal) にてコンパイル、実行すると最後に傾き a の値が表示されます。このように少しずつ動作確認しながらプログラムを完成させてゆく方法が理解できたことと思います。

```

端末
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
vm-i7# cc test-3.c -o test-3
vm-i7# ./test-3
x = 2.000000, y = 9.500000
x = 4.000000, y = 18.000000
x = 6.000000, y = 33.000000
x = 8.000000, y = 45.000000
x = 10.000000, y = 47.000000
XiYi = 1119.000000, Xi2 = 220.000000
a = 5.086364
vm-i7#

```

図6：test-3.c をコンパイルし，test-3 を実行

練習問題

図7の回路を使って表2のような実験データが得られた。データファイル test.dat をこの表にあわせて書き直し，上で作成したプログラム test-3 を使って抵抗値 R を求めよ。

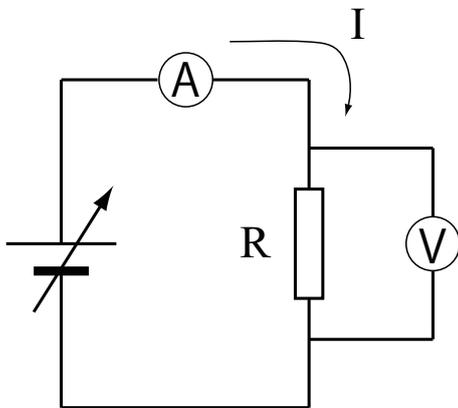


表1：実験結果

データ番号 i	(設定値) 電流 I_i [A]	(測定値) 電圧 V_i [V]
1	2.0	2.4
2	4.0	3.2
3	6.0	5.7
4	8.0	9.1
5	10.0	10.9

図7：オームの法則 $V = RI$

以上