コンピュータ実験レポート２

24cb062h 菅原明

すべてのファイルを https://github.com/meowwowcat/report2.git に置きました。 作成時に CHATGPT をところどころ使用してあります。

課題 17

方法（17.c）

課題 16 で作成下 u.dat を読み込む。ここでの誤差は 0.5。

func は自由落下の運動の式，calc_chi2 は χ ２乗。

```
#include<stdio.h>
#include<math.h>

#define ndat 20
#define range 0.5
#define vy0 6.0
#define y0 5.0
#define g 9.8

float func(float t){
        float y = y0 + vy0 * t - 0.5 * g * t * t ;
        return y;
}


float calc_chi2(float t[], float y[], float s[]){
        float chi2 = 0.0;
        int i;
        for(i = 0; i < ndat; i++){
                float diff = y[i] - func(t[i]);
                chi2 += diff * diff /(s[i] * s[i]);
                printf("%f\n",chi2);
        }
        return chi2;
}

int main() {
        FILE*fp;
        float t[ndat],x[ndat],y[ndat],exp_y[ndat],error_y,s[ndat];
        int ih;
        fp = fopen("u1.dat","r");
        for(ih = 0;ih < ndat;ih++){
                fscanf(fp,"%f, %f, %f, %f,
%f\n",&t[ih],&x[ih],&y[ih],&exp_y[ih],&s[ih]);
                s[ih] = range;
        }
        fclose(fp);

        float chi2;

        chi2 = calc_chi2(t, exp_y, s);
        printf("chi2 = %f\n",chi2);

        return 0;
}
```

結果(/17/./a.out)

```
2.930944
3.037220
3.178596
3.521991
3.828907
4.220784
4.318129
6.397493
8.707895
9.279430
9.717674
10.958673
11.042774
11.325799
11.772023
12.412023
14.656023
18.319424
19.062466
21.288530
chi2 = 21.288530
```

考察

χ 二乗の値が小さいので妥当

課題 18

方法 (18.C)

誤差付きのデータを用いて，ｖ と ｘ０を最小二乗法で求める．

ndat はデータポイントの数，データの誤差は 0.1

para_v,para_x0 は傾きと切片を収納．

```c
#include<stdio.h>

#define ndat 20
#define precision 0.1
int main(void)
{
     FILE*fp;
     float t[ndat],x[ndat],s[ndat],y[ndat],exp_y[ndat];
     float para_v,para_x0;
     int ih;

     if((fp=fopen("u.dat","r"))!=NULL){
             for(ih=0;ih<ndat;ih++){
                     fscanf(fp,"%f %f %f %f
%f",&t[ih],&x[ih],&y[ih],&exp_y[ih],&s[ih]);
                     s[ih]=precision;
                     }
              fclose(fp);

             }
             else{
                  printf("file open error!\n");
                  return -1;
```

```
                    }
                float SUM_T2=0,SUM_T=0,SUM_X=0,SUM_TX=0,SUM_1=0;
                int i;
                for(i=0;i<ndat;i++){
                            SUM_T2  += t[i] * t[i]/(s[i] * s[i]);
                            SUM_T   += t[i] / ( s[i] * s[i]);
                            SUM_X   += x[i] / ( s[i] * s[i]);
                            SUM_TX  += t[i] * x[i] / ( s[i] * s[i]);
                            SUM_1   += 1.0 / ( s[i] * s[i]);
                        }
                 para_v = (SUM_TX * SUM_1 - SUM_T * SUM_X) / (SUM_T2 * SUM_1 - SUM_T * SUM_T);
                para_x0 = (SUM_T2 * SUM_X - SUM_TX * SUM_T) / (SUM_T2 * SUM_1 - SUM_T * SUM_T);

                fp=fopen("output.txt","w");
                if( fp != NULL){
                        fprintf(fp,"v=%f , x0=%f\n",para_v,para_x0);
                        fclose(fp);
                        printf("書き込みお k\n");
                        }
                else{
                        printf("書き込みだめ\n");
                        return -1;
                       }

                return 0;
}
```

結果(output.txt)

```
v=3.999732 , x0=10.000156
```

考察

課題 16 で用いた初期値と誤差が少ない.したがって結果は妥当である.

課題 19

方法(re19.c)

func は速度vと初期位置x_0を用いて$v_x t + x_0$のあたい.

calc_chi2 は最小二乗法の χ 二乗計算を行う関数.

dcdv は χ 二乗の微分.

v_xを 2 8 まで 0.01 の間隔で変化,v_xに対する χ 二乗を計算.

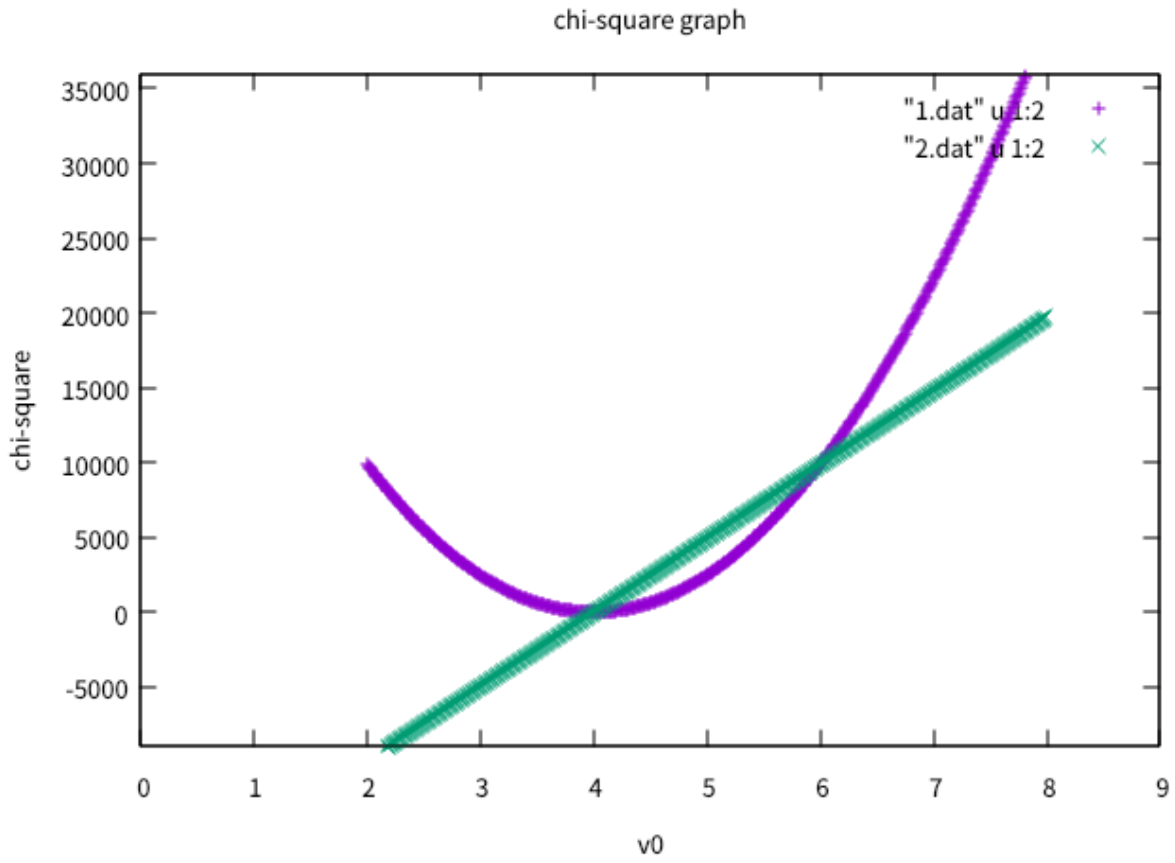
dat ファイルに書き出し,gnuplot でグラフを表示させる.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define ndat 20
#define precision 0.1
#define dv 0.001

float func(float t, float vx, float x0);
```

```c
float calc_chi2(float t[], float x[], float s[], float a, float b);
float dcdv(float t[], float x[], float s[], float a, float b);

int main(void) {
    FILE *fp, *fp2, *fp3;
    float t[ndat], model_x[ndat], model_y[ndat], error_y[ndat], exp_y[ndat], err[ndat];
    int ih;

    if ((fp = fopen("u.dat", "r")) != NULL) {
        for (ih = 0; ih < ndat; ih++) {
            if (fscanf(fp, "%f %f %f %f %f", &t[ih], &model_x[ih], &model_y[ih], &error_y[ih], &exp_y[ih]) != 5) {
                printf("Error reading data at line %d\n", ih + 1);
                fclose(fp);
                return 1;
            }
            err[ih] = precision;
        }
        fclose(fp);
    } else {
        printf("file open error!\n");
        return 1;
    }

    float x0 = 10, vx;
    float chi2, dcdvx;

    fp2 = fopen("1.dat", "w");
    fp3 = fopen("2.dat", "w");

    for (vx = 2; vx < 8; vx += 0.01) {
        chi2 = calc_chi2(t, model_x, err, vx, x0);
        dcdvx = dcdv(t, model_x, err, vx, x0);
        fprintf(fp2, "%f %f\n", vx, chi2);
        fprintf(fp3, "%f %f\n", vx, dcdvx);
    }
    fclose(fp2);
    fclose(fp3);

    double vx1 = 2, vx2 = 8, dcdv1, dcdv2, mid_vx, mid_dcdv;
    int step;

    for (step = 0; step < 100; step++) {
        dcdv1 = dcdv(t, model_x, err, vx1, x0);
        dcdv2 = dcdv(t, model_x, err, vx2, x0);
        mid_vx = (vx1 + vx2) / 2;
        mid_dcdv = dcdv(t, model_x, err, mid_vx, x0);
        if (mid_dcdv * dcdv1 > 0) {
            vx1 = mid_vx;
        } else {
            vx2 = mid_vx;
        }
        if (fabs(vx1 - vx2) < dv) break;
    }

    printf("Root vx: %f\n", mid_vx);
    return 0;
}

float calc_chi2(float t[], float x[], float s[], float a, float b) {
    float chi2 = 0;
    int i;

```

```
    for (i = 0; i < ndat; i++) {
        chi2 += (x[i] - func(t[i], a, b)) * (x[i] - func(t[i], a, b)) / (s[i] * s[i]);
    }
    return chi2;
}

float func(float t, float vx, float x0) {
    return vx * t + x0;
}

float dcdv(float t[], float x[], float s[], float a, float b) {
    return (calc_chi2(t, x, s, a + dv, b) - calc_chi2(t, x, s, a, b)) / dv;
}
```

結果



考察

課題 16 で用いた初期値と誤差が少ない.結果は妥当である.