

2021年度 コンピュータ演習

第6回：11月12日

**材料科学総合学科
大森 俊洋**

2021年度コンピュータ演習：第6回

1. 結晶の熱振動の統計熱力学（格子比熱）
2. 復習：小演習3 数値積分②（台形公式）
3. 練習：数値積分③（シンプソンの公式）
4. 小演習4：数値積分③（シンプソンの公式）
5. 課題の提出（11/16, 0時 締切り厳守）

FAQ ①

Q1. 『計算がうまくできません. 』

◆ 変数の “**型**” に気をつけましょう.

```
int n;          // 整数型の変数
```

```
double t;       // 浮動小数点型の変数
```

```
scanf("%d %lf", &n, &t);
```

```
printf("%d %f ¥n", n, t);
```

```
    // %d はint型, %lf はdouble型
```

```
printf(" %6.4f ¥n", &t);
```

```
    //全体6桁分の出力幅、小数部4桁
```

FAQ ②

Q2. 『実行時にエラーになります. 』

- ◆ 数学関数を利用する場合, コンパイル時に必ず “**-lm**” を付けて下さい.

例 > cc ソースファイル名 **-lm**

⇒ 実行ファイル “a.out” ができる.

- “**-lm**” は, 数学関数のヘッダ : math.h を include するときに入力する.
- “-o 実行ファイル名” を付けると, “実行ファイル名” を指定できる.

例 > cc -o jikkou pr1.c **-lm**

(ソースファイル : pr1.c, 実行ファイル : jikkou)

FAQ ③

Q3. 『“#define”の使い方は？』

“#define（マクロ定義）”の例

```
#define R 8.31447 // 気体定数
#define EPS 1.E-6 // 収束条件の閾値
#define samesign(x,y) ((x>0)==(y>0))
#define GNUPLOT "/usr/bin/gnuplot"
```

× #define R = 8.31447; // これは間違い

FAQ ④

Q4. 『数式の書き方は？』

(1) べき乗の書き方：例 $f(x) = e^x - x^2$

× $f(x) = e^x - x^2$

○ $f(x) = \exp(x) - x * x$

(2) 全角文字は利用できません.

※ 特に，空白に注意すること.

コンピュータ演習・本日の課題

1. 結晶の熱振動の統計熱力学（格子比熱）

- ・ デュロン・プティの法則 ($C_V \doteq 3R$)
- ・ アインシュタインモデル
- ・ **デバイモデル**

2. 練習：1/4円の面積の計算

3. 小演習3：数値積分（台形公式）

- ・ PAD図 \Rightarrow 数値積分のプログラム作成

4. **小演習4：数値積分（シンプソンの公式）**

- ・ PAD図 \Rightarrow 数値積分のプログラム作成

これまでの課題の流れ

小演習2：Cプログラムからgnuplotで作図
(練習 03_pr3.gpm, 03_pr4.cを参考)

小演習3：数値積分のCプログラミング (練習 04_pr1.c, 04_pr2.cを参考)

小演習4 (今回)：数値積分を行い、
gnuplotで作図 (小演習2と3を応用)
*一部の人は小演習3(c)で実施

本日の演習の準備（練習）

◆ 演習ホームページの第6回から以下のファイルをダウンロードして下さい.

① (06_pr1.c)

1. 結晶の熱振動の統計熱力学（大学院・相変態論）

復習

(3) 熱振動のエネルギー デバイ・モデル（振動数 ν を分布関数）

$$[C_{\text{振動}}]_V^D = 9R(T/\Theta_D)^3 \int_0^{\Theta_D/T} \frac{x^4 \exp(x)}{[\exp(x)-1]^2} dx$$

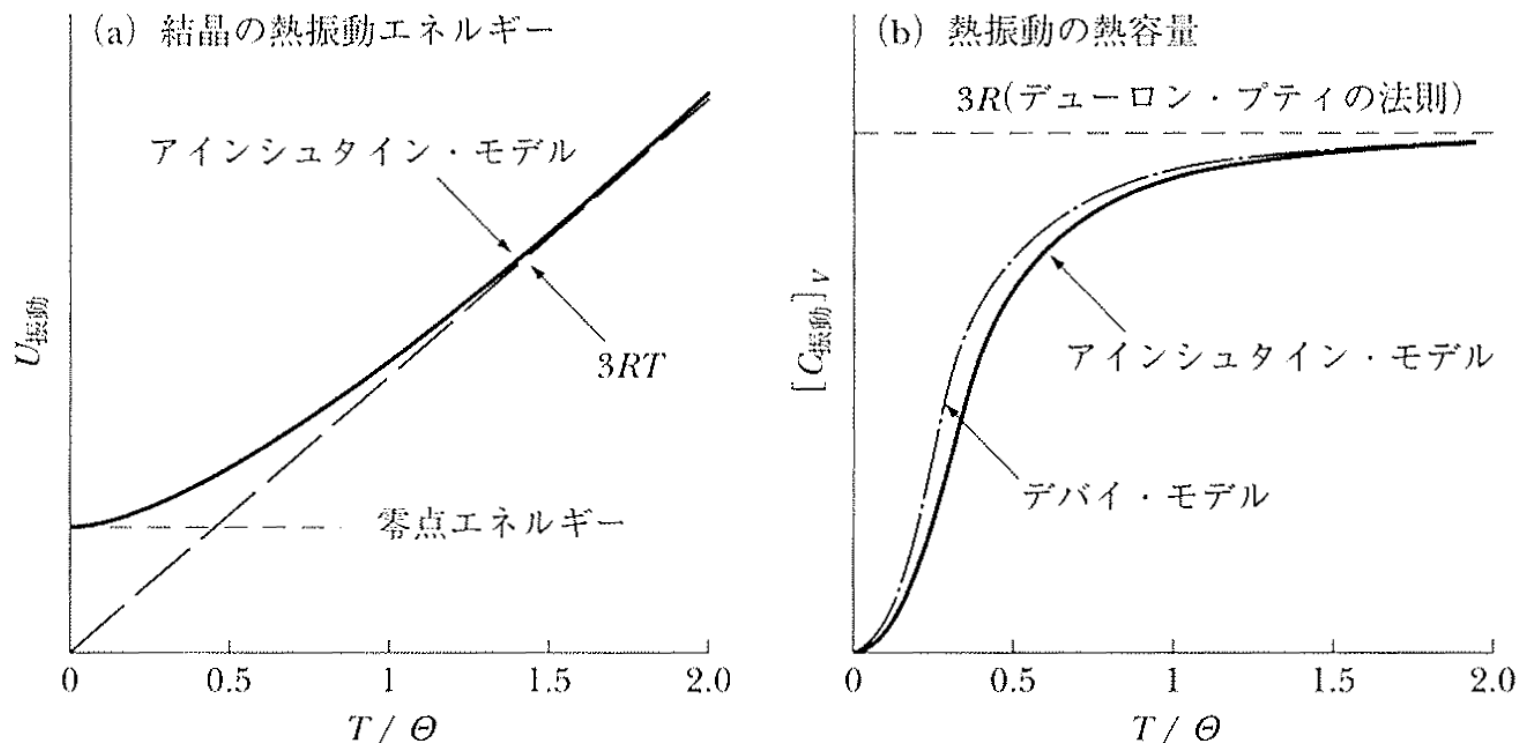


図2.12 結晶の熱振動のエネルギーと熱容量

1. 結晶の熱振動の統計熱力学（大学院・相変態論）

(3) 熱振動のエネルギー デバイ・モデル（振動数 ν を分布関数）**復習**

$$[C_{\text{振動}}]_V^D = 9R(T/\Theta_D)^3 \int_0^{\Theta_D/T} \frac{x^4 \exp(x)}{[\exp(x) - 1]^2} dx$$

表：デバイ温度, Θ_D （学籍番号の下1桁のNo.を選択）

No.	元素	Θ_D /K	No.	元素	Θ_D /K
0	Be	1160	5	Cu	343
1	Al	428	6	Mo	450
2	Si	640	7	Ag	226
3	V	360	8	W	400
4	Fe	467	9	Au	164

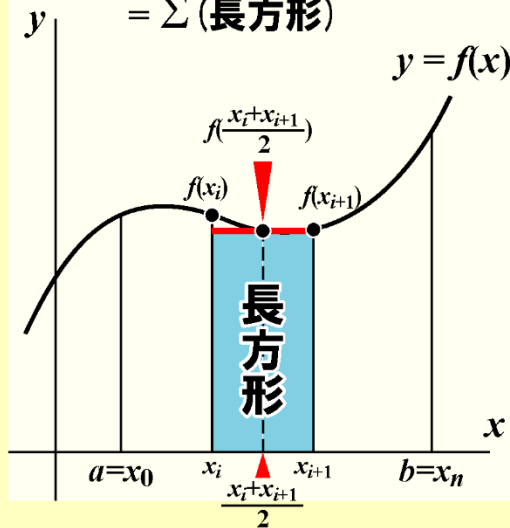
1. 練習：数値積分

復習

- 数値積分：定積分の値を数値的に求める方法。

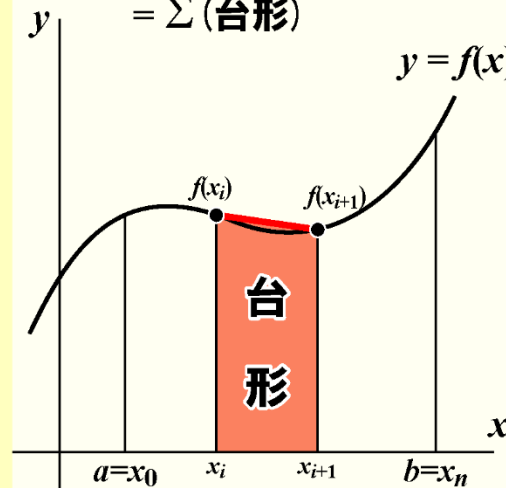
① 方形公式

$$S = \int_a^b f(x) dx$$
$$= \sum (\text{長方形})$$



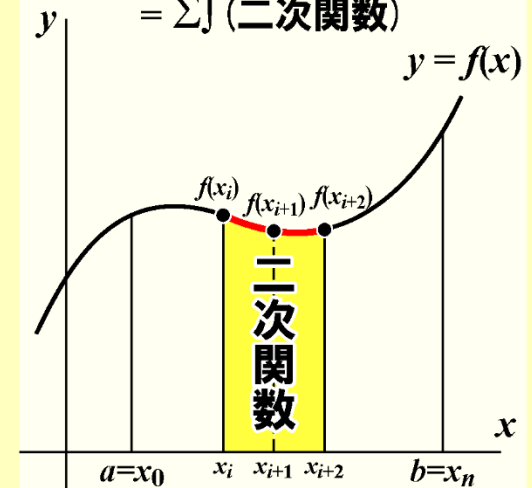
② 台形公式

$$S = \int_a^b f(x) dx$$
$$= \sum (\text{台形})$$



③ シンプソンの公式

$$S = \int_a^b f(x) dx$$
$$= \sum \int (\text{二次関数})$$



2. 前回：小演習3－数値積分（台形）－

【小演習3】数値積分①：格子比熱のデバイモデル

(a) $f(x) = x^4 e^x / (e^x - 1)^2$ ($0 < x \leq 20$) のとき,

$y = f(x)$ のグラフ ($0 \leq x \leq 20, 0 \leq y \leq 6$) をgnuplotを用いて作図し, png形式で出力したファイル（ファイル名：**学籍番号-3a.png**）を提出せよ.

(b) 温度 T [K] における元素 X の定積比熱 C_V （デバイモデル）を台形公式で計算するプログラムを作成し, ソースファイル（ファイル名：**学籍番号-3b.c**）を提出せよ. (n は 1000 とする)

2. 前回：小演習3－数値積分（台形）－

(a) 練習 03_pr2.gpmを使えば良い。

***注意**

$f(x) = f(x) = x * x * x * x * \exp(x) / ((\exp(x) - 1.) * (\exp(x) - 1.))$

gnuplotでは、 x の4乗を“ $x**4$ ”と書くことができるが、Cプログラムでは不可。（その代わり、`pow()`を使うことはできる。）

例) `pow(double x, double y)` x の y 乗

3. 小演習 3 (b) : 数値積分と定積比熱

変数

(i) i, n
(d) $t, xmax, cv,$
 a, b, fa, fb, s, h

(i)はint, (d)はdouble
を表す.

温度 t 入力

$xmax \leftarrow QD / t$

積分上限

$n \leftarrow 1000$

分割数 (台形の数)

$h \leftarrow xmax / (double)n$

台形の高さ

$a \leftarrow 0.$

$fa \leftarrow f(a+EPS)$

$f(0)$ は×

$s \leftarrow 0.$

積分値

for($i=0 ; i<n ; i++$)

繰返し
 n 回

$cv \leftarrow 9.*R*(t/QD)^3*s$

比熱

結果の出力 (画面)

"温度: $T = t,$ "

"比熱: $Cv = cv \backslash n$ "

定数

R

8.31447

気体定数

QD

###

EPS

1.0E-6

関数

$f(x)$

$\frac{x^4 e^x}{(e^x - 1)^2}$

$f(0)$ は×

###はデバイ温度

$b \leftarrow a + h$

$fb \leftarrow f(b)$

$s \leftarrow s + (fa + fb) * h / 2.$

台形の面積を加える

$a \leftarrow b$

右端 b を左端 a に更新

$fa \leftarrow fb$

上底 fb を底辺 fa に更新

2. 前回：小演習3－数値積分（台形）－

(b) 練習 05_pr2.cを使えば良い。

違うところは…

マクロ定義

```
#define R 8.31447
```

入力

```
scanf ("%d", &n);    nが整数の時 (int n;)  
scanf ("%lf", &t);   tが浮動小数点数の時  
                      (double t;)
```


2. 前回：小演習3－数値積分（台形）－

【小演習3(b)】繰返し

＊教科書第5章参照

$j = 1 \sim 1000$

繰返し

```
※①      ※②      ※③  
for (j=0 ; j<1000 ; j++) {  
    ..... ; (繰返す処理を{}間に記述する)  
}
```

※① : $j=0$: 最初だけ実行
※② : $j<1000$: 繰返しを継続する条件
※③ : $j++$: {}間処理後に毎回実行

2. 前回：小演習3－数値積分（台形）－

【小演習3(b)】繰返し

$j = 1 \sim 1000$

繰返し

```
for (i=0 ; i<n ; i++) {
```

```
    b = a + h;
```

右辺($a + h$)を左辺(b)に代入する
(イコール"=="ではない)

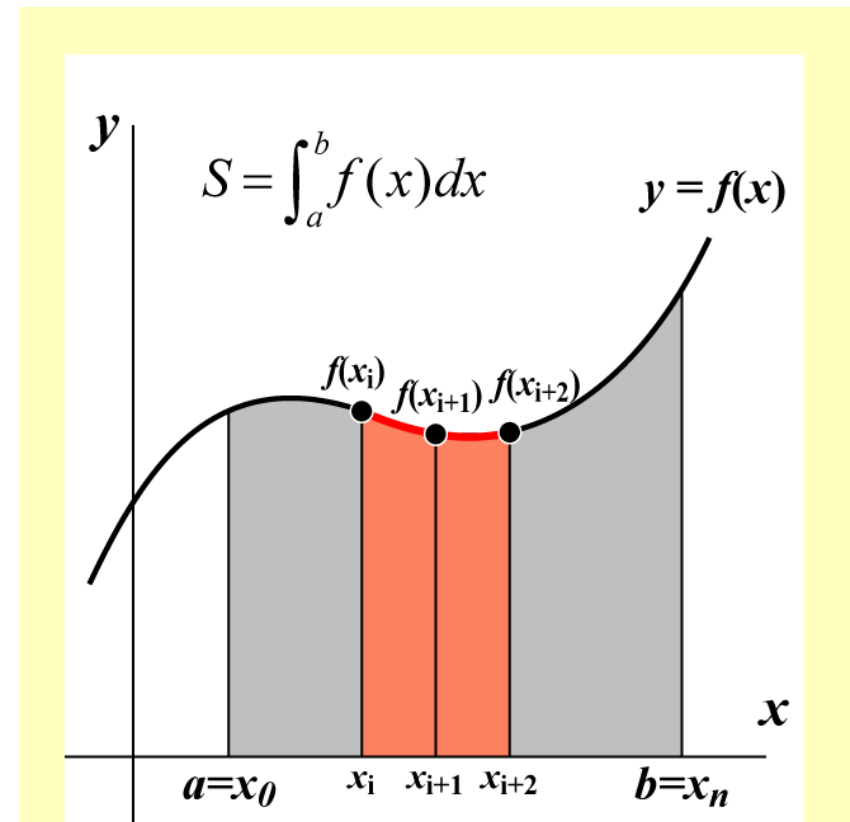
3. 練習：数値積分（シンプソン）

- ・ **シンプソンの積分公式**：数値積分法の一つで，被積分関数を2次式で近似した場合に得られる積分公式。

- (1) 全積分区間を偶数個の小区間に分割して，小区間2つ毎に被積分関数を2次式で近似する。
- (2) このとき，小区間 (x_i, x_{i+2}) での $f(x)$ の積分値は，

$$\int_{x_i}^{x_{i+2}} f(x) dx \cong \frac{h}{3} \{f(x_i) + 4f(x_{i+1}) + f(x_{i+2})\}$$

- (3) 小区間 (x_i, x_{i+2}) での積分値を全積分区間 (a, b) で足し合わせることにより，目的の積分値が計算できる。



図：シンプソンの公式による数値積分

3. 練習：数値積分（シンプソン）

1/4円の数値積分（06_pr1.c）

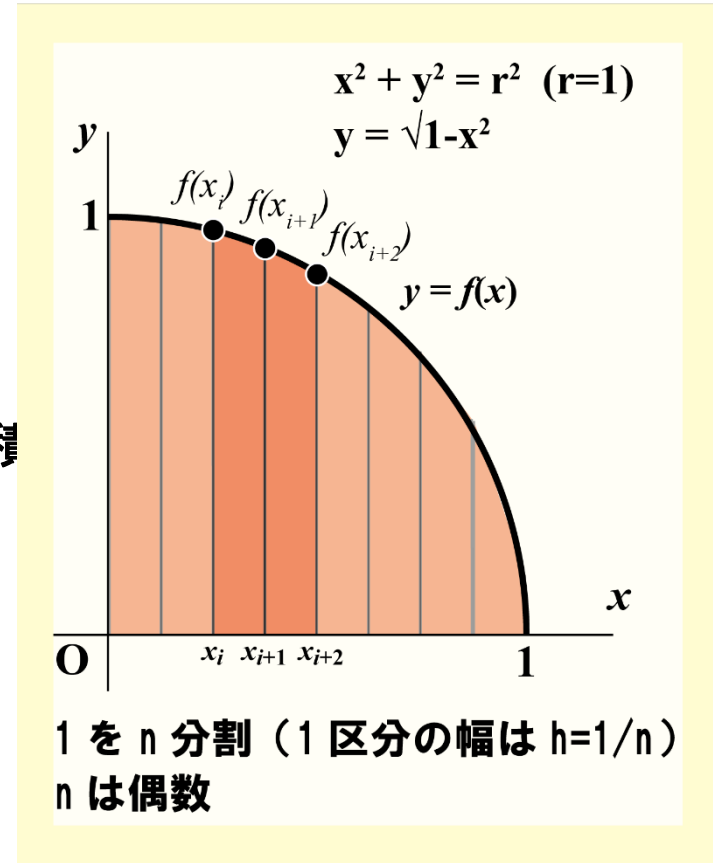
```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
```

```
double func(double x){ // 関数の宣言:被積分関数
    return sqrt(1.-x*x);
}
```

```
int main(){
    int n,i;
    double a,b,c,fa,fb,fc,s,h;
```

```
printf("Input an even number, n=? "); // nは偶数
scanf("%d",&n); // 積分区間の分割数
```

```
printf("¥nCalculating a quarter area of a circle.¥n¥n");
```



3. 練習：数値積分（シンプソン）

```
h = 1./ (double)n; // 分割の幅
a = 0.; // 区間の左端
fa = func(a);
s = 0.;
for(i=0;i<(n/2-1);i++){ // 一歩手前まで
    b = a + h; //区間の右端
    fb = func(b);
    c = b + h;
    fc = func(c);
    s = s + (fa+4.*fb+fc)*h/3.; // シンプソンの公式
    a = c; //右端⇒左端に更新
    fa = fc;
}
c = 1.; // 最後は円の右端
fc = func(c);
b = (a+c)/2.;
fb = func(b);
s = s + (fa+4.*fb+fc)*h/3.; // シンプソンの公式

printf("n= %d , s= %6.4f , pi= %10.8f ¥n",n,s,s*4.);

exit(0);
}
```

4. 小演習4：数値積分（シンプソン）

【小演習4】数値積分②：格子比熱のデバイモデル

元素X（P.11）の定積比熱 C_V （デバイモデル）を「シンプソンの積分公式」を用いて数値積分の計算を行い、

$1 \leq T \leq 1000$ （1 [K] 間隔）の各温度における定積比熱 C_V と温度 T の関係を表すグラフを作図するプログラムを作成せよ。

ソースファイル（ファイル名：学籍番号-4.c）とpng形式で出力したグラフのファイル（ファイル名：学籍番号-4.png）を提出せよ。

4. 小演習4：数値積分（シンプソン）

変数

(i) i, j, n
 (d) $t, xmax, cv,$
 $a, b, c, fa, fb, fc, s, h$
 (FILE) $*fp$

$fp \leftarrow \text{fopen}("$$$","w")$ ※

$n \leftarrow 1000$

$\text{for}(j=1 ; j \leq 1000 ; j++)$

$\text{fclose}(fp);$

繰返し

$t \leftarrow (\text{double})j$

$xmax \leftarrow QD / t$

$h \leftarrow xmax / (\text{double})n$

$a \leftarrow 0.$

$fa \leftarrow f(a+EPS)$

$s \leftarrow 0.$

$\text{for}(i=0 ; i < n/2 ; i++)$

$cv \leftarrow 9.*R*(t/QD)^3*s$

結果の出力（画面）

"温度： $T = t,$ "

"比熱： $Cv = cv \backslash n$ "

結果の出力（FILE）

" $t \quad cv \backslash n$ "

R

8.31447

QD

###

EPS

1.0E-6

$f(x)$

$$\frac{x^4 e^x}{(e^x - 1)^2}$$

はデバイ温度

$b \leftarrow a + h$

$fb \leftarrow f(b)$

$c \leftarrow b + h$

$fc \leftarrow f(c)$

$s \leftarrow s + (fa + 4.*fb + fc)*h/3.$

$a \leftarrow c$

$fa \leftarrow fc$

繰返し

※ \$\$\$ はファイル名
 "w" は書出し用に
 ファイルを開く

小演習4のヒント（ファイル出力）

```
FILE *fp; //ファイルポインタの宣言
```

```
fp = fopen("ファイル名.dat", "w");  
//ファイルのオープン(wは書き込み用)
```

```
fprintf(fp, "%6.4f", t);
```

```
fclose(fp);
```


小演習 4 のヒント（ファイル出力）

```
int main(){
    int n,i,j;
    double t,xmax,cv,a,b,c,fa,fb,fc,s,h;
    FILE *fp,*pipe;

    fp=fopen("ファイル名.dat","w");
    if(NULL==fp){
        printf("¥nERROR : Impossible to open¥n");
        exit(1);
    }
```

小演習3や06_pr1.cを応用

```
fprintf(fp,"      ", , );
}
```

fclose(fp); ➡ 次にgnuplotを動かす

小演習4のヒント（gnuplotを動かす）

Cプログラムからgnuplotで作図する方法

⇒ pipe 第3回 **03_pr4.c**を参考

```
// CのプログラムからGNUPLLOTを動かす方法①:  $f(x) = \exp(x) - x \cdot x$ 

#include <stdlib.h>
#include <stdio.h>
#include <math.h> // 数学関数を利用する

#define GNUPLOT "/usr/bin/gnuplot" // gnuplotのパス

int main(){
    FILE *pipe;

    // 関数をgnuplotで画面に表示し, pngファイルに出力する.
    pipe=popen(GNUPLOT " -persist", "w");
    if(NULL==pipe){
        printf("Cannot open the pipe to GNUPLOT.¥n");
        exit(1); // 異常終了
    }
}
```

小演習4のヒント（gnuplotを動かす）

第3回 03_pr4.cを参考

```
fprintf(pipe, "set title ¥\"y = exp(x) - x*x ¥\"¥n");  
                                                    //グラフのタイトル  
fprintf(pipe, "f(x) = exp(x) - x*x¥n");           //関数の定義
```

中略

```
fprintf(pipe, "plot f(x)¥n");                     //画面に表示  
fflush(pipe);                                     //バッファを掃き出す
```

この時は関数 $f(x)$ を描画した。今回は数値計算結果を出力する！

ヒント：

ファイル名は前後に ¥" を付けること

例) `fprintf(pipe, "plot ¥\"result.dat¥\" with lines¥n");`

出力に関するまとめ

出力

出力処理

- (1) 画面（標準出力）に出力

```
printf( "フォーマット式", 変数);
```

- (2) ファイルに出力 (*fp* : ファイル変数)

```
fp = fopen( "ファイル名", "w" ); // "w"は書出し用
```

```
fprintf( fp, "フォーマット式", 変数);
```

- (3) gnuplot にコマンドを送る (*pipe* : パイプ変数)

```
pipe = popen( gnuplotのパス, "w" ); // "w"は書出し用
```

```
fprintf( pipe, "gnuplotコマンド\n" );
```

```
fflush(pipe);
```

例 : `fprintf(fp, "%6.4lf %6.11lf\n", x2, t);`