

# 数値シミュレーション：2階の微分方程式 (シラバス10・11回目)

## 【1】2階の微分方程式と差分方程式

微分方程式を、

$$\frac{d^2x}{dt^2} + c \frac{dx}{dt} = f(x, t)$$

とする。これを2つの1階の微分方程式に変更する。

$$\frac{d^2x}{dt^2} + c \frac{dx}{dt} = f(x, t) \Rightarrow \frac{d^2x}{dt^2} = -c \frac{dx}{dt} + f(x, t) \Rightarrow \begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -cy + f(x, t) = F(x, y, t) \end{cases}$$

それぞれの1階の微分方程式に Runge-Kutta 法を適用する。それぞれ、差分に直して

$$\frac{dx}{dt} \Rightarrow \frac{x_{n+1} - x_n}{\Delta t}, \quad \frac{dy}{dt} \Rightarrow \frac{y_{n+1} - y_n}{\Delta t}$$

より

$$x_{n+1} - x_n = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \times \Delta t$$

$$y_{n+1} - y_n = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \times \Delta t$$

を用いる。ここで、

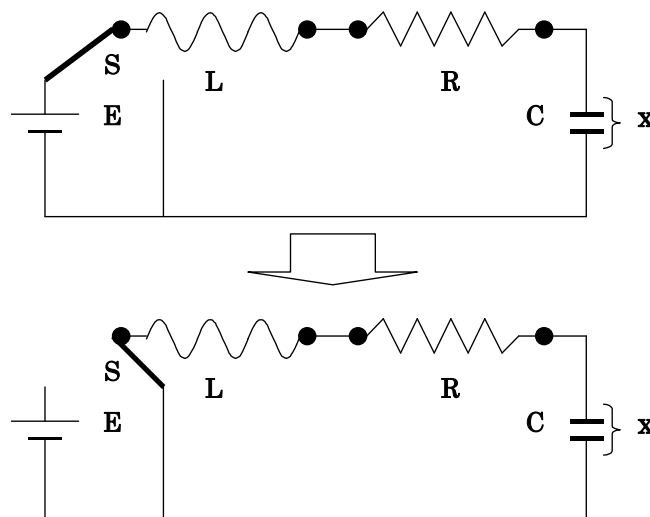
$$k_1 = f(x_n, y_n, t_n)$$

$$k_2 = f\left(x_n + \frac{k_1 \times \Delta t}{2}, y_n + \frac{k_1 \times \Delta t}{2}, t_n + \frac{\Delta t}{2}\right)$$

$$k_3 = f\left(x_n + \frac{k_2 \times \Delta t}{2}, y_n + \frac{k_2 \times \Delta t}{2}, t_n + \frac{\Delta t}{2}\right)$$

$$k_4 = f(x_n + k_3 \times \Delta t, y_n + k_3 \times \Delta t, t_n + \Delta t)$$

## 【2】RLC回路のシミュレーション



図のようなRC回路にコイル(L)が入っている。スイッチSで端子を電池(E)側から切り替えると、コンデンサに生じた電位が徐々に消滅していく。そのときのコンデンサの電圧変化  $x$  は、時間とともに変化する。電流の向きを、RからLの方向にとると、

$$LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0$$

となる。これを2つの1階の方程式にするには、

$$LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0 \Rightarrow \frac{d^2x}{dt^2} + \frac{R}{L} \frac{dx}{dt} = -\frac{x}{LC} \Rightarrow \begin{cases} c = \frac{R}{L} \\ f(x, t) = -\frac{x}{LC} \end{cases}$$

なので

$$\begin{aligned} \frac{dx}{dt} &= y = f_1(x, y) \\ \frac{dy}{dt} &= -cy + f(x, t) \Rightarrow \frac{dy}{dt} = -\frac{R}{L}y - \frac{1}{LC}x = -\frac{RCy + x}{LC} = f_2(x, y) \end{aligned}$$

になる。初期条件として

$$\begin{aligned} x(0) &= E \\ y(0) &= 0 \end{aligned}$$

にする。

### 【3】数値計算に必要な工夫

微分方程式は

$$\begin{cases} \frac{dx}{dt} = y \\ \frac{dy}{dt} = -\frac{RCy + x}{LC} \end{cases}$$

になる。ただし、 $\frac{dy}{dt} = -\frac{RCy + x}{LC}$  は、このまま数値計算すると、例えば、

- $R = 0.025$  (k $\Omega$ )
- $C = 0.25$  ( $\mu$ F)
- $L = 10$  (mH)

のとき、

$$\begin{aligned} \frac{dy}{dt} &= -\frac{RCy + x}{LC} = -\frac{0.025k\Omega \times 0.25\mu F \times y + x}{10mH \times 0.25\mu F} = -\frac{0.025 \times 10^3 \Omega \times 0.25 \times 10^{-6} F \times y + x}{10 \times 10^{-3} H \times 0.25 \times 10^{-6} F} \\ &= \frac{0.025 \times 0.25 \times 10^{-3} y + x}{2.5 \times 10^{-9}} = 0.0025 \times 10^6 y + \frac{10^9}{2.5} x = 0.25 \times 10^4 y + \frac{10^9}{2.5} x \end{aligned}$$

となり、**係数が非常に大きな値**となり、数値計算には不適である。このようなときには、時間変数  $t$  は秒で計測するのであるが、この

- 計測時間の単位を  $ms (= \frac{1}{1000} s)$  や  $\mu s (= \frac{1}{1000,000} s)$  のように短くする

とよい。このことを見るために、実際に数値を用いて

$$\frac{dy}{dt} = -\left(0.25 \times 10^4 y + \frac{10^9}{2.5} x\right) \Rightarrow \frac{d^2 x}{dt^2} = -\left(0.25 \times 10^4 \frac{dx}{dt} + \frac{10^9}{2.5} x\right)$$

なので、大きな数が現れる。そこで

●  $10^8$  (後で使うため  $(10^2)^4$  を見越している) で両辺を割る

と

$$\begin{aligned} \frac{d^2 x}{dt^2} = -\left(0.25 \times 10^4 \frac{dx}{dt} + \frac{10^9}{2.5} x\right) &\Rightarrow \frac{d^2 x}{\boxed{10^8} dt^2} = -\left(\frac{0.25 \times 10^4}{\boxed{10^8}} \frac{dx}{dt} + \frac{10^9}{\boxed{10^8} 2.5} x\right) \\ &= -\left(\frac{0.25}{10^4} \frac{dx}{dt} + \frac{10}{2.5} x\right) \end{aligned}$$

になる。分母の大きな数を、時間  $t$  に含めると

$$\frac{d^2 x}{10^8 dt^2} = -\left(\frac{0.25}{10^4} \frac{dx}{dt} + \frac{10}{2.5} x\right) \Rightarrow \frac{d^2 x}{d(\boxed{10^4} t)^2} = -\left(0.25 \frac{dx}{d(\boxed{10^4} t)} + \frac{1}{4} x\right)$$

そこで、新しい変数として、

$$T = 10^4 t$$

をとれば、もとの微分方程式は

$$\frac{d^2 x}{dT^2} = -\left(0.25 \frac{dx}{dT} + \frac{1}{4} x\right) \quad (T = 10^4 t)$$

になり、

● 大きな数は姿を消す

ことがわかる。その結果、プログラムでは、 $T = 10^4 t$  について、0 秒から 60 秒間計測するとすれば、 $t$  による実際の計測時間は

$$\left. \begin{array}{l} T = 0\text{s} \\ t = 0\text{s} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \text{プログラムでの経過時間: } T = \boxed{60\text{s}} \quad (T = 10^4 t) \\ \text{計測時間: } t = \frac{1}{10000} \times 60\text{s} = \frac{6}{1000} \text{s} = \boxed{6\text{ms}} \end{array} \right.$$

になる。従って、

● 計測時間は、**ms** で表示

で表示するとよい。

この変換を系統的に行うには、

$$\frac{dy}{dt} = -\frac{RCy+x}{LC} \Rightarrow \frac{d^2 x}{dt^2} = -\left(\frac{R}{L} \frac{dx}{dt} + \frac{1}{LC} x\right)$$

において  $LC = 0.3 \times 10^{-8}$  なので、 $10^8$  のかわりに、 $1/LC$  を用いて、

● 両辺を  $\frac{1}{LC}$  で割る = 両辺に  $LC$  を掛ける

と

$$\begin{aligned} \frac{d^2x}{dt^2} &= -\left(\frac{R}{L} \frac{dx}{dt} + \frac{1}{LC}x\right) \Rightarrow LC \frac{d^2x}{dt^2} = -\left(RC \frac{dx}{dt} + x\right) \\ &\Rightarrow \frac{d^2x}{d\left(\frac{t}{\sqrt{LC}}\right)^2} = -\left(\frac{RC}{\sqrt{LC}} \frac{dx}{d\frac{t}{\sqrt{LC}}} + x\right) = -\left(R\sqrt{\frac{C}{L}} \frac{dx}{d\frac{t}{\sqrt{LC}}} + x\right) \end{aligned}$$

そこで、新しい変数として、

$$T = \frac{t}{\sqrt{LC}}$$

をとれば、もとの微分方程式は

$$\boxed{\frac{d^2x}{dT^2} = -\left(R\sqrt{\frac{C}{L}} \frac{dx}{dT} + x\right) \left(T = \frac{t}{\sqrt{LC}} : \text{プログラムでの経過時間[s]}\right)} \Rightarrow \boxed{\text{計測時間 } t = \sqrt{LC}T \text{ ms で表示: } \sqrt{LC}T \times 10^3}$$

になる。従ってプログラムで用いる微分方程式は

$$\begin{aligned} \frac{dx}{dT} &= y = f_1(x, y) \\ \frac{dy}{dT} &= -\left(R\sqrt{\frac{C}{L}}y + x\right) = f_2(x, y) \end{aligned}$$

になる。

#### 【4】プログラム

$$1) \frac{dx}{dt} = y = f_1(x, y) \Rightarrow \frac{\Delta x}{\Delta t} = f_1(x, y) \Rightarrow \begin{cases} \Delta x = f_1(x, y) \Delta t \\ \underbrace{\Delta x}_{x_{\text{next}} - x} = f_1(x, y) \Delta t \end{cases} \Rightarrow x_{\text{next}} = x + f_1(x, y) \Delta t$$

プログラムでは

$$x_{\text{next}} = x + f_1(x, t) \Delta t = \text{calculate\_nextx} \left( x, y, \overset{\text{STEP}}{\Delta t} \right)$$

$$2) \frac{dy}{dt} = -\left(R\sqrt{\frac{C}{L}}y + x\right) = f_2(x, y) \Rightarrow \frac{\Delta y}{\Delta t} = f_2(x, y) \Rightarrow \begin{cases} \Delta y = f_2(x, y) \Delta t \\ \underbrace{\Delta y}_{y_{\text{next}} - y} = f_2(x, y) \Delta t \end{cases} \Rightarrow y_{\text{next}} = y + f_2(x, y) \Delta t$$

プログラムでは

$$y_{\text{next}} = y + f_2(x, y) \Delta t = \text{calculate\_nexty} \left( x, y, \overset{\text{STEP}}{\Delta t} \right)$$

#### repo1.c

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

#define R (0.025*1000.0)
#define C (0.25/1000000.0)
#define L (10.0/1000.0)
```

```
#define INITIAL_TIME 0.0
#define LAST_TIME 10.0
#define INITIAL_X 1.0
#define INITIAL_Y 0.0
#define STEP 0.04

double f1(double x, double y)
{
    double fy;

    fy = y;

    return fy;
}

double f2(double x, double y)
{
    double fx;

    fx = -(R*sqrt(C/L)*y+x);

    return fx;
}

double calclate_nextx(double x, double y, double step)
{
    double k1, k2, k3, k4;
    double next_x;

    k1 = f1(x, y);
    k2 = f1(x+k1*step/2.0, y+k1*step/2.0);
    k3 = f1(x+k2*step/2.0, y+k2*step/2.0);
    k4 = f1(x+k3*step, y+k3*step);

    next_x = x + (k1+2.0*k2+2.0*k3+k4) *step/6.0;

    return next_x;
}

double calclate_nexty (double x, double y, double step)
{
    double k1, k2, k3, k4;
    double next_y;

    k1 = f2(x, y);
    k2 = f2(x+k1*step/2.0, y+k1*step/2.0);
    k3 = f2(x+k2*step/2.0, y+k2*step/2.0);
    k4 = f2(x+k3*step, y+k3*step);

    next_y = y + (k1+2.0*k2+2.0*k3+k4)*step/6.0;

    return next_y;
}

void writedata(double t, double x)
{
    // msで表示するため、1.0E3を掛けておく
    printf("%7.4f,%10.3e\n", sqrt(L*C)*t*1.0E3, x);
}

int main(void)
{
```

```

double t, last_t;
double x, next_x, y, next_y;

x = INITIAL_X;
y = INITIAL_Y;
t = INITIAL_TIME;
last_t = LAST_TIME*(1.0+STEP);

while(t < last_t) {
    writedata(t, x);
    next_x = calculate_nextx(x, y, STEP);
    next_y = calculate_nexty(x, y, STEP);
    x = next_x;
    y = next_y;
    t = t+STEP;
}

return 0;
}

```

になる。

## 【5】表計算ソフトの自動立ち上げ (Windows用のみ)

計算結果を画面表示と同時にファイルを作成し、作成したファイルを自動的に表計算ソフトで表にして表示するようにする。

### 1) csv ファイルを作成: fopen 関数

fopen(番地, 決まった文字列);

fopen(csv ファイル名の格納番地, "wt");

FILE \* file\_open;

file\_open= fopen(filename, "wt");

file\_open の数値が零(NULL)だと作成に失敗

### 2) csv ファイルへの書き込み: fprintf 関数

fprintf(FILE \* 型番地, ...);

printf("%5.2f %10.3e\n", t, x);

fprintf(file\_open, "%5.2f, %10.3e\n", t, x);

### 3) 書き込み終了後 csv ファイルを閉じる: fclose 関数

fclose(FILE \* 型番地)

fclose(file\_open)

### 4) 表計算ソフトで csv ファイルを開く: ShellExecute 関数

ShellExecute((HWND)0, "open", 番地, NULL, NULL, SW\_SHOWNORMAL)

ShellExecute((HWND)0, "open", csv ファイル名の格納番地, NULL, NULL, SW\_SHOWNORMAL)

戻り値は特別で HWND 型になっている。戻り値が 32 以下だと csv ファイル用表計算ソフトの起動に失敗

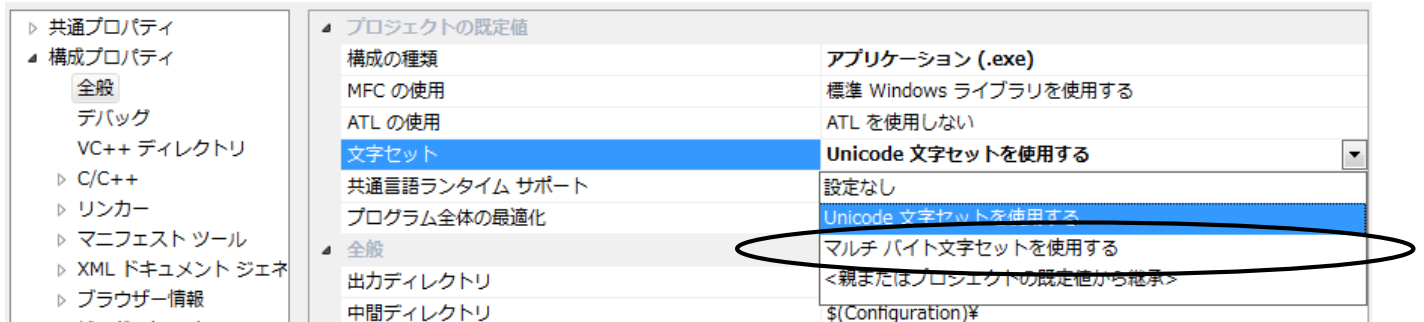
の関数を使用する。ShellExecute 関数を使用するために

windows.h を#include する

こと。

メニュー[プロジェクト]-[プロパティ]の

- [文字セット] マルチバイト文字セットを使用するを選択



プログラムは repo1.c に**太文字部**を追加や変更します。

repo2.c

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <windows.h>
```

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

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

```
#define R (0.025*1000.0)
```

```
#define C (0.25/1000000.0)
```

```
#define L (10.0/1000.0)
```

```
#define INITIAL_TIME 0.0
```

```
#define LAST_TIME 50.0
```

```
#define INITIAL_X 1.0
```

```
#define INITIAL_Y 0.0
```

```
#define STEP 0.0025
```

```
double f1(double x, double y)
```

```
{
    double fy;
```

```
    fy = y;
```

```
    return fy;
```

```
}
```

```
double f2(double x, double y)
```

```
{
    double fx;
```

```
    fx = -(R*sqrt(C/L)*y+x);
```

```
    return fx;
```

```
}
```

```
double calculate_nextx(double x, double y, double step)
```

```
{
    double k1, k2, k3, k4;
    double next_x;
```

```
k1 = f1(x, y);
k2 = f1(x+k1*step/2.0, y+k1*step/2.0);
k3 = f1(x+k2*step/2.0, y+k2*step/2.0);
k4 = f1(x+k3*step, y+k3*step);

next_x = x + (k1+2.0*k2+2.0*k3+k4)*step/6.0;

return next_x;
}

double calculate_nextx (double x, double y, double step)
{
    double k1, k2, k3, k4;
    double next_x;

    k1 = f1(x, y);
    k2 = f1(x+k1*step/2.0, y+k1*step/2.0);
    k3 = f1(x+k2*step/2.0, y+k2*step/2.0);
    k4 = f1(x+k3*step, y+k3*step);

    next_x = x + (k1+2.0*k2+2.0*k3+k4)*step/6.0;

    return next_x;
}

double calculate_nexty (double x, double y, double step)
{
    double k1, k2, k3, k4;
    double next_y;

    k1 = f2(x, y);
    k2 = f2(x+k1*step/2.0, y+k1*step/2.0);
    k3 = f2(x+k2*step/2.0, y+k2*step/2.0);
    k4 = f2(x+k3*step, y+k3*step);

    next_y = y + (k1+2.0*k2+2.0*k3+k4)*step/6.0;

    return next_y;
}

void writedata(FILE *file_open, double t, double x)
{
    // msで表示するため、1.0E3を掛けておく
    printf("%7.4f,%10.3e\n", sqrt(L*C)*t*1.0E3, x);
    if(file_open != NULL) {
        fprintf(file_open, "%7.4f, %10.3e\n", sqrt(L*C)*t*1.0E3, x);
    }
}

int main(void)
{
    // 作成ファイル名はexample2.csv
    char filename[] = "example2.csv";
    // 作成ファイル追尾用変数
    FILE * file_open = NULL;
    double t, last_t;
    double x, next_x, y, next_y;

    x = INITIAL_X;
    y = INITIAL_Y;
    t = INITIAL_TIME;
    last_t = LAST_TIME*(1.0+STEP);

    // ファイルの作成に失敗すると数値(file_open=NULL)を返す
    file_open = fopen(filename, "wt");

    writedata(file_open, t, x);

    while(t < last_t) {
        next_x = calculate_nextx(x, y, STEP);
        next_y = calculate_nexty(x, y, STEP);
        x = next_x;
        y = next_y;
        t = t+STEP;
        writedata(file_open, t, x);
    }

    // NULLのときにファイル作成失敗
```



```

if(file_open == NULL) {
    printf("%n====>ファイル(%s)の作成に失敗しています。%n====>既にファイルが開か
    れているかもしれません。", filename);

    getchar();
}
else{
    HINSTANCE hinst;

    fclose(file_open);

    hinst = ShellExecute((HWND)0, "open", filename, NULL, NULL, SW_SHOWNORMAL);

    if(hinst <= (HINSTANCE)32) {
        printf("%n====>表計算ソフトの自動起動に失敗しました。");

        getchar();
    }
}

return 0;
}

```

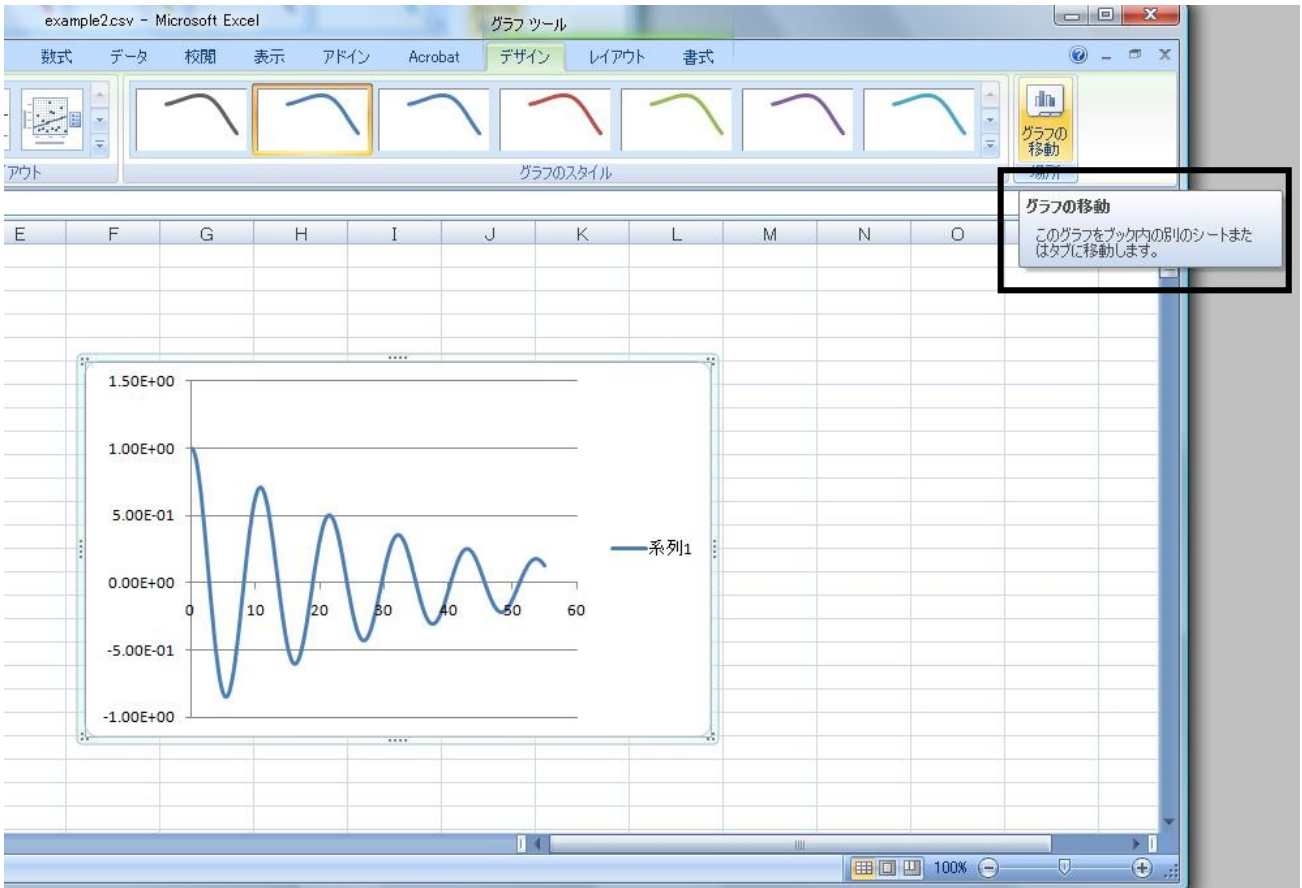
エクセルからグラフ作成方法

### 散布図－散布図（平滑線）

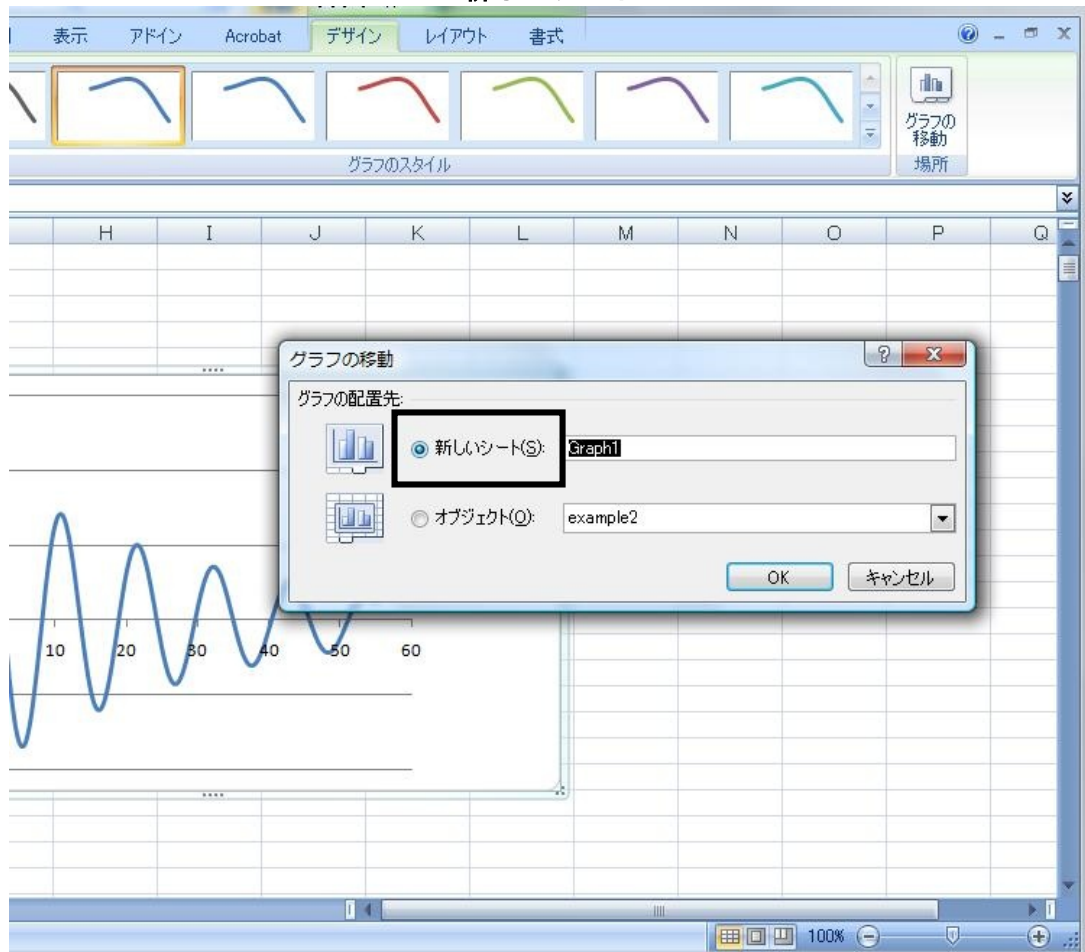
The screenshot shows Microsoft Excel with a scatter plot of data from columns A and B. The x-axis (A) ranges from 0 to 2.4, and the y-axis (B) ranges from 1.00E+00 to 2.29E-01. A tooltip for 'Scatter Plot (Smooth Lines)' is displayed, stating: '値の組を比較します。×軸上のデータ要素が多く、データが開散を表す場合に使用します。' (Compares groups of values. Used when there are many data elements on the x-axis and the data is scattered).

A	B
0	1.00E+00
0.1	1.00E+00
0.2	9.97E-01
0.3	9.90E-01
0.4	9.80E-01
0.5	9.66E-01
0.6	9.49E-01
0.7	9.30E-01
0.8	9.07E-01
0.9	8.81E-01
1	8.52E-01
1.1	8.21E-01
1.2	7.86E-01
1.3	7.50E-01
1.4	7.11E-01
1.5	6.70E-01
1.6	6.27E-01
1.7	5.82E-01
1.8	5.35E-01
1.9	4.87E-01
2	4.37E-01
2.1	3.86E-01
2.2	3.34E-01
2.3	2.82E-01
2.4	2.29E-01

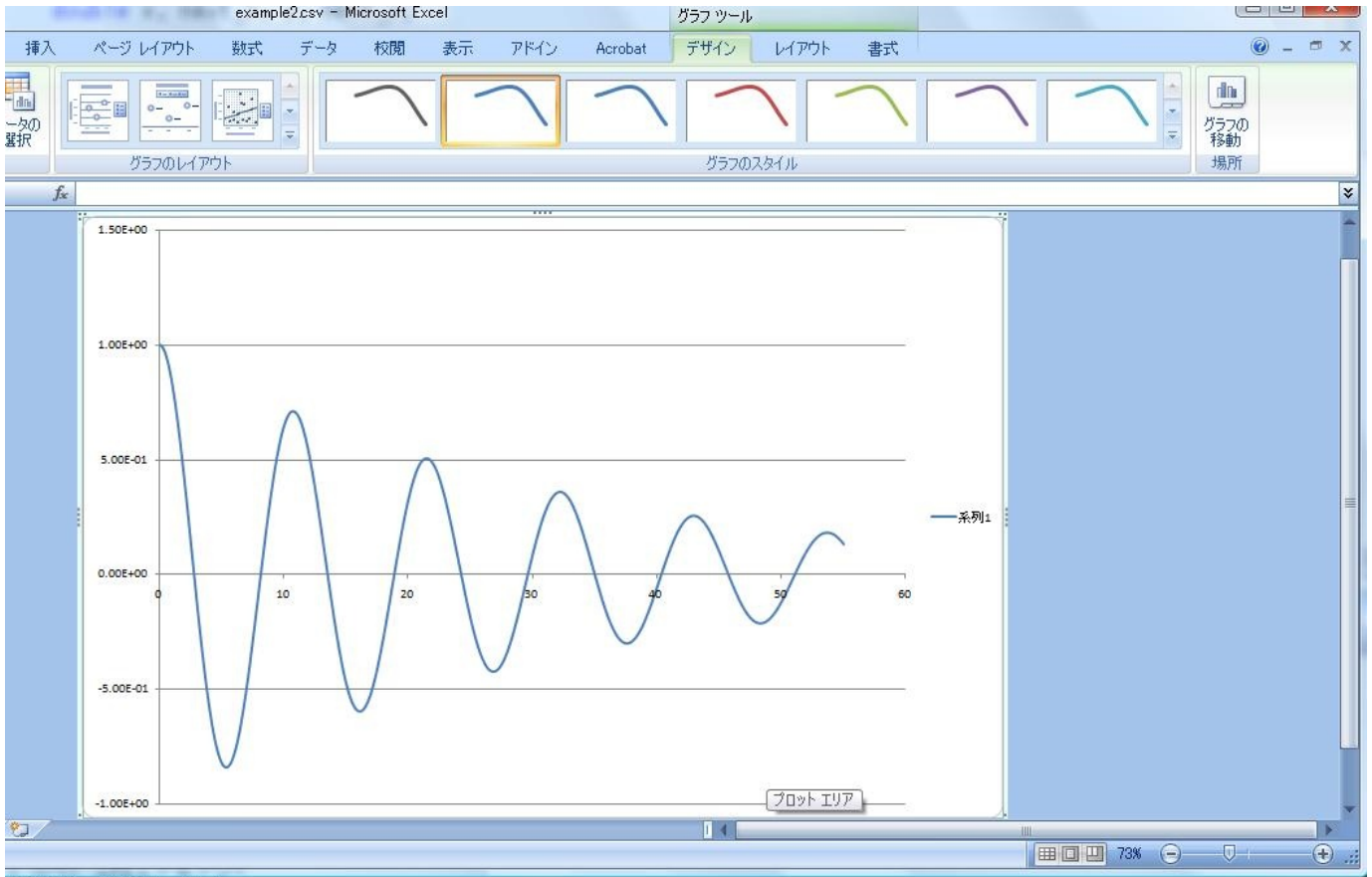
### グラフの移動



### 新しいシート



完了



## 第6回目レポート

レポートの回数(6回目)、学生証番号と氏名を明記すること  
必ず表紙を付け、最後のページ(の添付用)を表紙の次に入れる

(A4レポート用紙使用のこと)

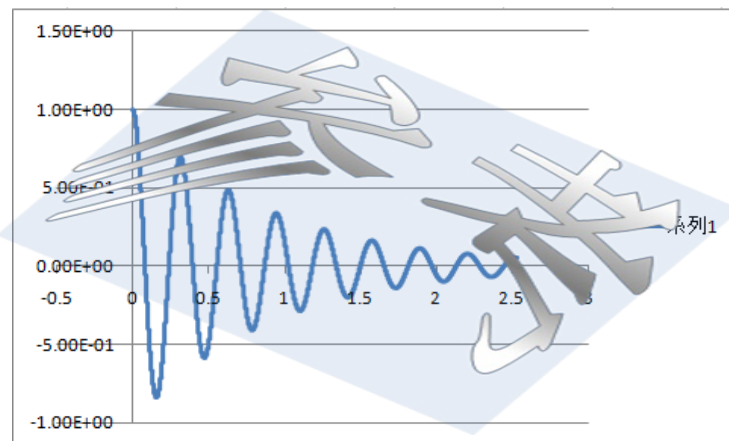
1) RLC 回路において、

- コンデンサー電圧の満たす方程式  $LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0$  を導け。
  - キルヒホッフの第II法則を用いる
  - コンデンサーに蓄えられる電荷を  $Q$  とすると・・・
  - コンデンサの電圧  $V(=x)$  は・・・  $Q$  を使って書ける
  - 電流は・・・  $Q$  を使って書ける
  - $Q$  の微分方程式をコンデンサの電圧  $V(=x)$  の微分方程式に変えて・・・
- 初期条件  $y(0) = 0$  は何を意味するか。物理の言葉で説明せよ。

の2点を提出。

2) **repo. 2 のプログラム**を作成し、 $t$ - $x$  グラフを作成せよ。

- プログラム
- $t$ - $x$  グラフ(散布図を選択)ができるが、数値などの見栄えを美しくする事)



● グラフが示すように振動しながら減衰してゆくが、方程式を元にして何故かの説明の3点を提出。

3)  $m = 0.1 \text{ kg}$  の質点を先端にもつバネの減衰振動の方程式

$$m \frac{d^2x}{dt^2} = -kx - b \frac{dx}{dt}$$

を、 $LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0$  のプログラムを参考にして作成し、数値

$$A) \begin{cases} k = 0.1 \\ b = 0.01 \end{cases}$$

$$B) \begin{cases} k = 0.1 \\ b = 0.4 \end{cases}$$

を用いて、t-x グラフを作成せよ。

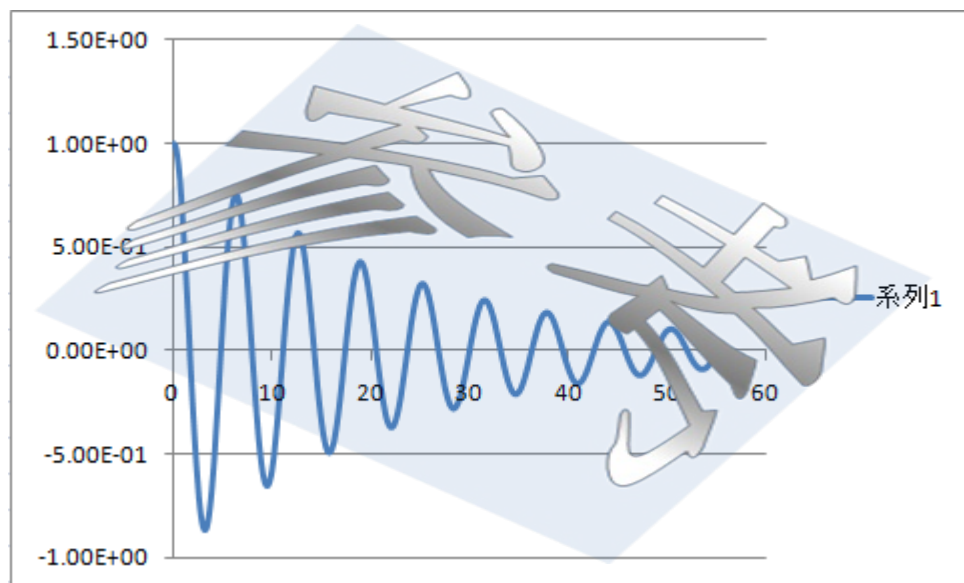
●プログラム:A)とB)共通で1点

- 時間間隔は「0.0025 秒 = 5ms」に  
#define STEP 0.0025
- ただし、データの表示時間は、t だけでよい:  
printf("%5.2f,%10.3e\n", t, x);

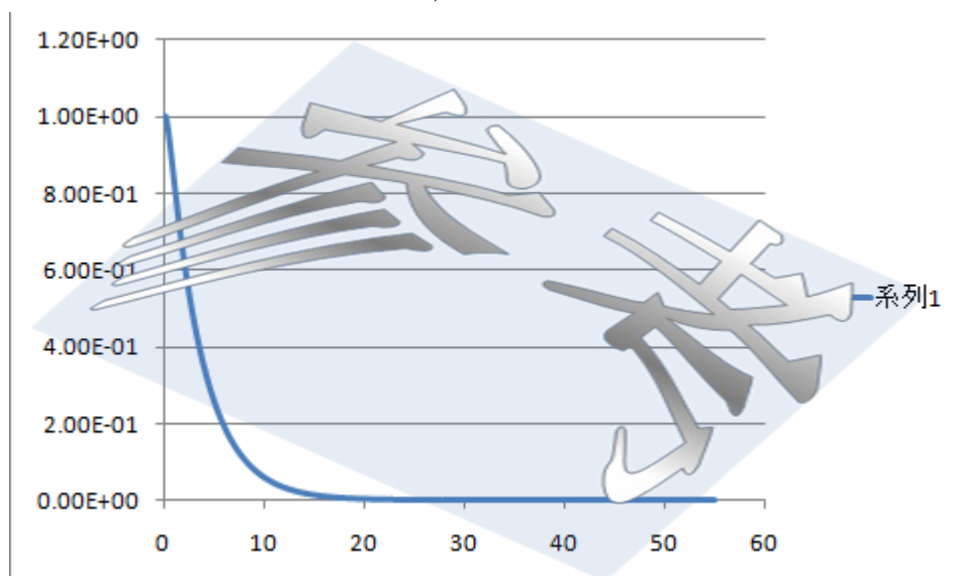
●t-x グラフ:A)とB)の2点

の3点を提出。

A)のグラフ



B)のグラフ



提出例(使える表紙は次のページ)

**コンピュータ物理学演習Ⅱ**  
第6回目  
月 日  
学籍番号  
氏名

1 ページ目

**次ページの添付用**

2 ページ目

作成する解答など

3 ページ目以降

# コンピュータ物理学演習 II

## 第6回目

月 日提出

学籍番号

氏名

## 第6回目レポート（添付用）

- 1) RLC回路のコンデンサー電圧の満たす方程式  $LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0$  を導け。
- 2) repo.1c/repo.2 のどちらかのプログラムを作成し、t-x グラフを作成せよ。
- プログラム
  - 実行結果
  - t-x グラフ
  - グラフが示すように振動しながら減衰してゆくが、方程式を元にして何故かの説明

の4点を提出。

- 3)  $m = 0.1 \text{ kg}$  の質点を先端にもつバネの減衰振動の方程式

$$m \frac{d^2x}{dt^2} = -kx - b \frac{dx}{dt}$$

を、 $LC \frac{d^2x}{dt^2} + RC \frac{dx}{dt} + x = 0$  のプログラムを参考にして作成し、数値

A)  $\begin{cases} k = 0.05 \\ b = 0.02 \end{cases}$

B)  $\begin{cases} k = 0.1 \\ b = 0.4 \end{cases}$

を用いて、t-x グラフを作成せよ。#define STEP 0.1 とし、50 秒間計測する。

- プログラム：A)とB)共通で1点
- 実行結果：A)とB)の2点
- t-x グラフ：A)とB)の2点

の5点を提出。