

# 数値シミュレーション：1階の微分方程式 (シラバス8・9回目)

(教科書186ページの13・5「微分方程式」参照)

## 【1】微分方程式と物理現象

力学では、マニュアル書に従って「誰がやっても、過去の透視・未来の予言」ができます。時間を逆に辿ったり、先に進んだりします。これを使って透視や予言するのですが、道具は「水晶玉」等ではなくて、「時間についての微分方程式」になります。人間の能力では、実際の現象をあまりうまく予言できないので、スーパーコンピュータ等に予言してもらうことがしばしばです。台風の進路などの予言や、地球にぶつかりそうになる彗星の進路予言等、これですね。もっと身近には、12:00発の新幹線に東京で乗ると大阪に何時につくか、予言できちゃってますね。

さて、ここでは微分方程式をコンピュータにといてもらうことをやってみます。使用する方法は

### ●Euler法

と、それを改良した

### ●Runge-Kutta法

です。

## 【2】微分方程式を差分方程式へ

微分方程式は、

$$\frac{dx}{dt} = f(x, t)$$

である。 $dx$ は微小変化分という意味で、微分では

$$dx \rightarrow 0, dt \rightarrow 0$$

とする。そこで、 $dx \rightarrow 0, dt \rightarrow 0$ をやめて、適当な幅（微小幅）として近似する。また、

$$dx = f(x, t) \times dt$$

とかけるので、 $dt$ がわかれば $dx$ が計算できる。すなわち、最初の $x, t$ を与えて、次々に計算を進める。最初の値

$$x = x_0, t = t_0$$

を決めて $dx$ を計算する。 $dt$ を一定の値（0.01とか0.05とか0.1等）として、それを

$$\Delta t$$

とし、つぎの値を $x = x_1, t = t_1$ とすれば、

$$t_1 - t_0 (\equiv dt) = \Delta t, \quad x_1 - x_0 (\equiv \Delta x) = f(x_0, t_0) \times \Delta t$$

となる。以下、同様にして

$$t_2 - t_1 = \Delta t, \quad x_2 - x_1 = f(x_1, t_1) \times \Delta t$$

$$\vdots$$

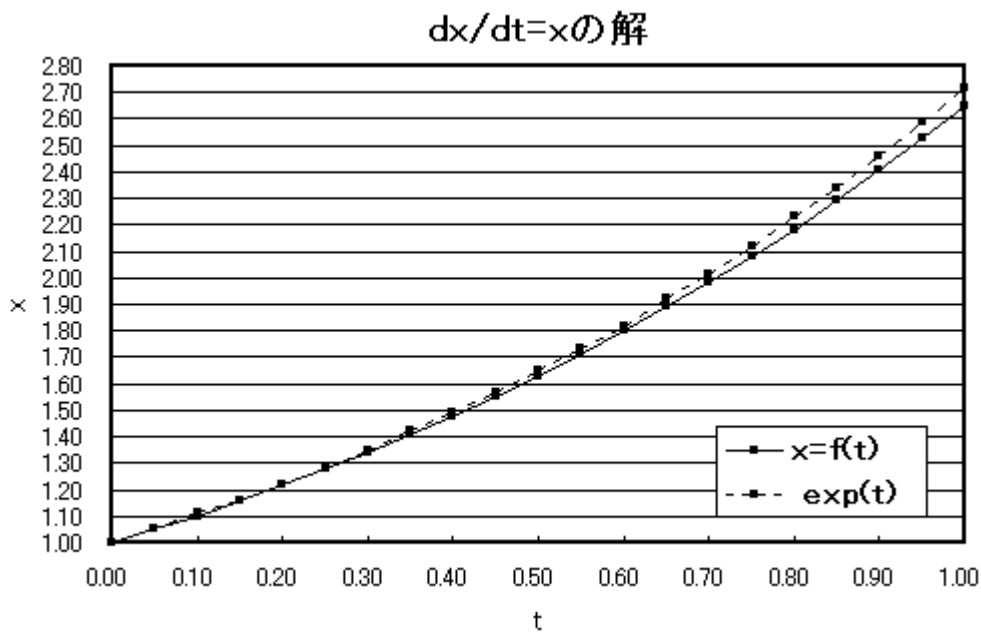
となる。そして、 $(x_0, x_1, x_2, x_3, \dots)$  と並べていけば、答えが得られる。これを

## Euler 法

という。ここで例として

$$dx/dt = x \Rightarrow x = e^t (= \exp(t))$$

を解かせてみる。数値計算と答え： $\exp(t)$ を比較すれば・・・



$$t_0 = 0.00, t_1 = 0.05, t_2 = 0.10, \dots, t_{19} = 0.95, t_{20} = 1.00$$

$$\Delta t = 0.05$$

このように、 $x$ が大きくなると誤差も少しずつ増えるが、大雑把にはあっている。

### 【3】C言語プログラム：Euler（オイラー）法

$dx/dt=x$  をプログラムする。

$$\frac{dx}{dt} = f(x, t)$$

において

$$f(x, t) = x$$

である。ここで、

$$\text{STEP} = \Delta t$$

として

$$\text{STEP}$$

を用いる。その結果、

$$\begin{array}{l}
 \boxed{\frac{dx}{dt} = f(x,t)} \left\{ \begin{array}{l} \Rightarrow \frac{\Delta x}{\Delta t} = f(x,t) \\ \Rightarrow \Delta t = \text{STEP} \\ \Rightarrow \Delta x = f(x_0, t_0) \times \Delta t \end{array} \right. \Rightarrow \left\{ \begin{array}{l} f(x_0, t_0) \text{が計算できる} \leftarrow \begin{array}{l} x_0 = \text{ある値 (例えば, 0)} \\ t_0 = \text{ある値 (例えば, 0)} \end{array} \\ \leftarrow \begin{cases} \Delta x = x_1 - x_0 = f(x_0, t_0) \times \Delta t \\ \Delta t = t_1 - t_0 \end{cases} \\ \boxed{x_1 = x_0 + f(x_0, t_0) \times \text{STEP}} \\ t_1 = t_0 + \text{STEP} \\ \leftarrow \begin{cases} \Delta x = x_2 - x_1 = f(x_1, t_1) \times \text{STEP} \\ \Delta t = t_2 - t_1 \end{cases} \\ \boxed{x_2 = x_1 + f(x_1, t_1) \times \text{STEP}} \\ t_2 = t_1 + \text{STEP} \end{array} \right.
 \end{array}$$

として、次々に時間を進めて ( $t_1 = t_0 + \Delta t$ ) 答え ( $x_1 = x_0 + f(x_0, t_0) \Delta t$ ) が計算できる。

$$\begin{array}{l}
 \boxed{x_1 = x_0 + f(x_0, t_0) \times \text{STEP}} \\
 \boxed{x_2 = x_1 + f(x_1, t_1) \times \text{STEP}} \\
 \vdots
 \end{array}
 \left. \vphantom{\begin{array}{l} \boxed{x_1 = x_0 + f(x_0, t_0) \times \text{STEP}} \\ \boxed{x_2 = x_1 + f(x_1, t_1) \times \text{STEP}} \\ \vdots \end{array}} \right\} \Rightarrow \begin{array}{l} \text{next\_x} = x + f(x, t) \times \text{STEP} \\ \text{next\_x} = \text{calculate\_nextx}(x, t) \end{array}$$

$$\Rightarrow \boxed{\begin{array}{l} \text{double calculate\_nextx}(\text{double } x, \text{double } t) \\ \{ \\ \quad \text{return } x + f(x, t) * \text{STEP}; \\ \} \end{array}}$$

として、プログラムで死傷する関数を作成する。

## プログラム (repoS.c)

```

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

#define STEP 0.05

// dx/dt=f(x)=x
double f(double x, double t)
{
    double fx;

    fx = x;

    return fx;
}

```

//記号は\* \*/と同じでコメントを表す。  
プログラミングとは無関係の文になる。

```

// next_x - x = f(x, t)*STEP
double calculate_nextx(double x, double t)
{
    double next_x;

    next_x = x + f(x, t)*STEP;

    return next_x;
}

int main(void)
{
    double x, x_next, t;

    x = 1.0;
    t = 0.0;

    printf("%s, %s, %s¥n", "t", "x=f(t)", "exp(t)");

    while(t < (1.0+STEP)) {
        printf("%f, %e, %e¥n", t, x, exp(t));
        x_next = calculate_nextx(x, t);
        x = x_next;
        t = t+STEP;
    }

    return 0;
}

```

になる。

#### 【4】C言語の枠組：Runge-Kutta (ルンゲ・クッタ) 法

Euler 法の改良版であり

$$x_{n+1} - x_n = f(x_n, t_n) \times \Delta t$$

の代わりに

$$x_{n+1} - x_n = \overbrace{f(x_n, t_n)}^{k_1} \times \Delta t$$

記号  $k_1$  を用いて

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

を用いる。ここで、

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

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

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

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

である。これを図示すれば、すばらしい結果となる。

```

// next_x = x + g(x, t)*STEP
double calculate_nextx(double x, double t)
{
    double next_x;
    double k1, k2, k3, k4;

    //  $k_1 = f(x_0, t_0)$ 

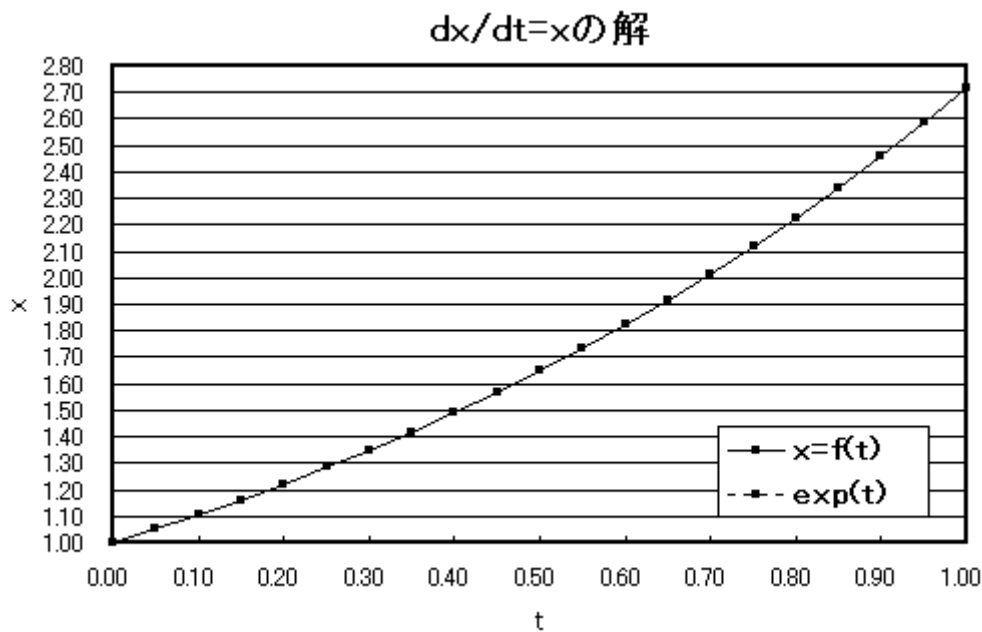
    k1 = f(x, t);
    //  $k_2 = f\left(x_0 + \frac{k_1 \times \Delta t}{2}, t_0 + \frac{\Delta t}{2}\right)$ 
    k2 = f(x+k1*step/2.0, t+STEP/2.0);
    //  $k_3 = f\left(x_0 + \frac{k_2 \times \Delta t}{2}, t_0 + \frac{\Delta t}{2}\right)$ 
    k3 = f(x+k2*step/2.0, t+STEP/2.0);

    //  $k_4 = f(x_0 + k_3 \times \Delta t, t_0 + \Delta t)$ 

    k4 = f(x+k3*step, t+STEP);
    //  $g(x_0, t_0) = \frac{k_1 + k_2 + 2k_3 + k_4}{6} \times \Delta t$ 
    next_x = x + (k1+2*k2+2*k3+k4)*step/6.0;

    return next_x;
}

```

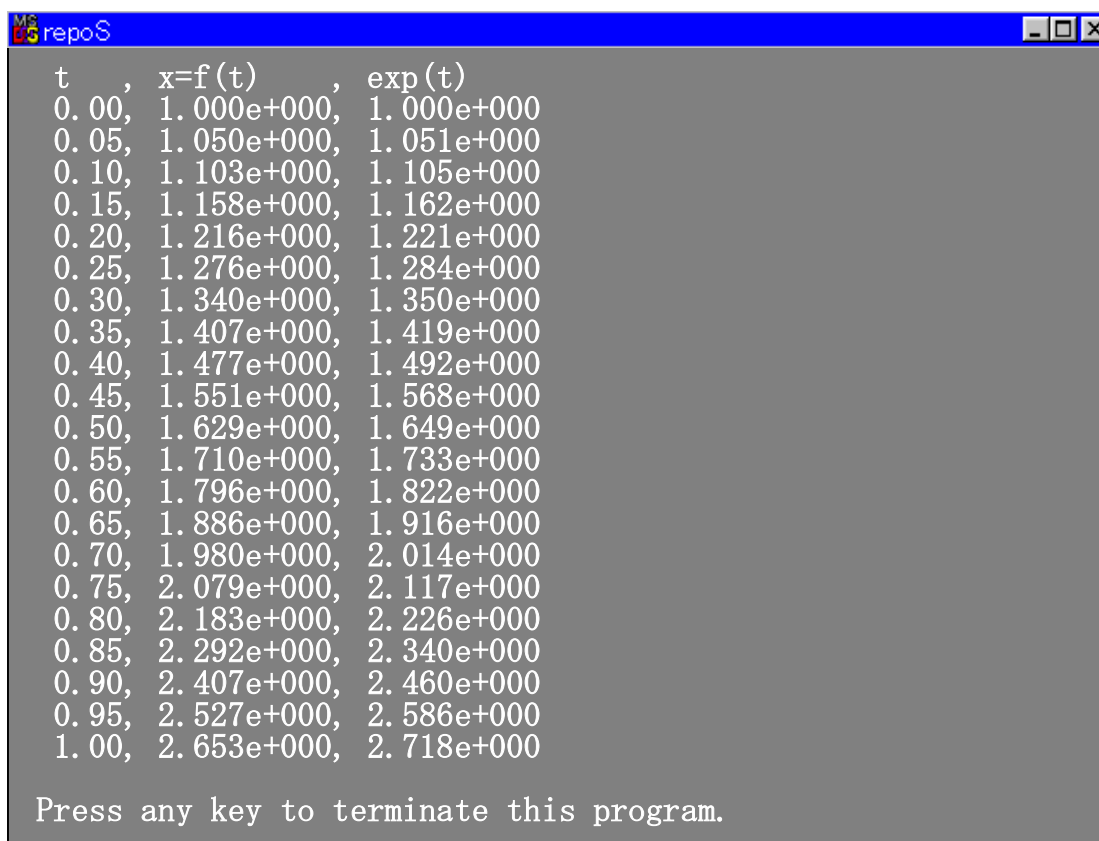


## 第5回目レポート

レポートの回数(5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) repoS.c の

- A) プログラム
- B) main() プログラム内で  $t$  は  $0 \sim 1$  の値を取らせている。そのためには「while( $t \leq 1.0$ )」でよさそうであるが、「while( $t < (1.0 + \text{step})$ )」としてあるのは理由がある。その理由を述べよ。
- C) 実行結果
- D) 以下の画面表示をするように変更したプログラム: 空白を使って調節しないこと。printf の書式設定を使用し「表題は左寄、数値は右寄」の設定にすること。実行結果のコピーは不要。



```
MS-DOS [repoS]
t      , x=f(t)      , exp(t)
0.00, 1.000e+000, 1.000e+000
0.05, 1.050e+000, 1.051e+000
0.10, 1.103e+000, 1.105e+000
0.15, 1.158e+000, 1.162e+000
0.20, 1.216e+000, 1.221e+000
0.25, 1.276e+000, 1.284e+000
0.30, 1.340e+000, 1.350e+000
0.35, 1.407e+000, 1.419e+000
0.40, 1.477e+000, 1.492e+000
0.45, 1.551e+000, 1.568e+000
0.50, 1.629e+000, 1.649e+000
0.55, 1.710e+000, 1.733e+000
0.60, 1.796e+000, 1.822e+000
0.65, 1.886e+000, 1.916e+000
0.70, 1.980e+000, 2.014e+000
0.75, 2.079e+000, 2.117e+000
0.80, 2.183e+000, 2.226e+000
0.85, 2.292e+000, 2.340e+000
0.90, 2.407e+000, 2.460e+000
0.95, 2.527e+000, 2.586e+000
1.00, 2.653e+000, 2.718e+000

Press any key to terminate this program.
```

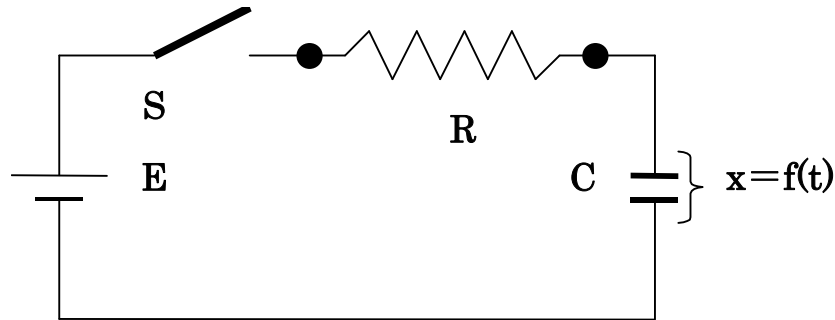
の4点を提出。

2) Runge-Kutta 法で1)のプログラムを作成せよ。【ヒント】オイラー法のプログラムの repoS.c を修正して作成するとよい。

- A) プログラム
- B) 実行結果

を提出。

3) 以下の図に示す RC 回路について。



- A) 【物理の問題】時刻  $t$  でスイッチを入れた後のコンデンサ  $C$  の電圧変化  $V$  (図では  $x$  のこと) の満たす微分方程式

$$\frac{dV}{dt} = \frac{E - V}{CR}$$

を導け (電磁気学の参考書: RC 回路などに記述されている)。

【ヒント】電圧の保存則より、電流をコンデンサーに蓄えられるて  
いる電荷 ( $Q$ ) の時間変化 ( $\frac{dQ}{dt}$ ) を用いて表す。

- B) 【プログラミングの問題】A)の微分方程式を Runge-Kutta 法を用いて解くこと。但し、 $C=1 \mu\text{F}$ 、 $R=1 \text{k}\Omega$ 、 $E=1 \text{V}$  とする。また、スイッチを入れた後の経過時間 ( $t$ ) を  $\text{ms}$  ( $=10^{-3}$ 秒) で計るとする (つまり、数値 0.010 は 10 としてプロットする)。従って、時間の刻みは、例えば、0.05ms くらいで、1 秒間計測すればよい。コンデンサ  $C$  の電圧は  $x=f(t)$  とする。

$C=1 \mu\text{F}$ 、 $R=1 \text{k}\Omega$ 、 $E=1 \text{V}$  の値は、STEP=0.05 秒の時に  
#define STEP 0.05

と使用したように#define を用いて記述できる。

属性	SI 単位	実用単位(例)
抵抗	$\Omega$	$\text{k}\Omega$
キャパシタンス	F	$\mu\text{F}$
インダクタンス	H	mH
時間	s	ms
電圧	V	V
電流	A	A

数値は、SI 単位に直してプログラムに使用する。

- A)の答え
- B)のプログラム
- B)の実行結果：問 1) repoS.c の D)の出力画面をまねること。但し「exp(t)」の列は必要ないので「 $t$  と  $x=f(t)$ 」の表でよい。
- B)の  $t-x$  軸でのグラフ：表計算ソフト (Excel など) を使用するとよい (散布図を選択)。「,」で区切った出力結果を、「VS2008 の使用法」

の「プログラム実行画面の取り込み」を参照して、csv ファイル（例えば、out.csv）で保存すると便利である。

の4点を提出。



提出例(使える表紙は次のページ)

<p><b>コンピュータ物理学演習Ⅱ</b></p> <p>第5回目</p> <p>月 日</p> <p>学籍番号</p> <p>氏名</p>
---

1 ページ目

<p><b>次ページの添付用</b></p>
------------------------

2 ページ目

<p>作成する解答など</p>
-----------------

3 ページ目以降

# コンピュータ物理学演習 II

## 第5回目

月 日提出

学籍番号

氏名

## 第5回目レポート（添付用）

1) repoS.c の

- A)プログラム
- B)main()プログラム内でtは0～1の値を取らせている。そのためには「while(t<=1.0)」でよさそうであるが、「while(t<(1.0+step))」としてあるのは理由がある。その理由を述べよ。
- C)実行結果
- D)以下の画面表示をするように変更したプログラム:空白を使って調節しないこと。printfの書式設定を使用し「表題は左寄せ、数値は右寄せ」の設定にすること。実行結果のコピーは不要。

2) Runge-Kutta法のプログラム:repoS.cをもとにプログラムせよ。

- A)プログラム
- B)実行結果

の2点を提出。

3) repoS.cの以下の図に示すRC回路について。

A)時刻tでスイッチを入れた後のコンデンサCの電圧変化V(図ではxのこと)の満たす微分方程式

$$\frac{dV}{dt} = \frac{E-V}{CR}$$

を導け(電磁気学の参考書:RC回路などに記述されている)。

【ヒント】電圧の保存則より、電流をコンデンサーに蓄えられるている電荷(Q)の時間変化( $\frac{dQ}{dt}$ )を用いて表す。

B) A)の微分方程式をRunge-Kutta法を用いて解くこと。但し、 $C=1 \mu\text{F}$ 、 $R=1 \text{k}\Omega$ 、 $E=1 \text{V}$ とする。また、スイッチを入れた後の経過時間(t)をms( $=10^{-3}$ 秒)で計るとする(つまり、数値0.010は10としてプロットする)。コンデンサCの電圧は $x=f(t)$ とする。

- A)の答え
- B)のプログラム
- B)の実行結果:問1) repoS.cのD)の出力画面をまねること。但し「exp(t)」の列は必要ないので「tと $x=f(t)$ 」の表でよい。
- B)のt-x軸でのグラフ:表計算ソフト(Excelなど)を使用するとよい。「,」で区切った出力結果を、「VS2010の使用法」の「プログラム実行画面の取り込み」を参照して、csvファイル(例えば、out.csv)で保存すると便利である。

の4点を提出。