夏休みの課題

山本昌志*

2007 年 8 月 8 日

概 要

C 言語を使った数値計算の学習を進めている．そこでは，テイラー展開と線形代数をつかうことになる．そのために，それらの復習を夏休みの課題として課す．さらに，ソーティングのプログラムの作成をとおして，C 言語のプログラムになれて欲しい．

1 数値計算の基礎

[**問 1**] テイラー展開の公式を導き出せ．そして，自分なりに理解せよ．公式を書くだけではだめである．3〜4 年生の数学の教科書に書かれているはずである．

[**問 2**] 問 1 の結果を利用して，関数 $f(x)$ を $x = a$ の周りでテイラー展開する式を示せ．

[**問 3**] 問 1 の結果を利用して，関数 $f(x)$ を $x = 0$ の周りでテイラー展開する式を示せ．これをマクローリン展開と言う．

[**問 4**] 問 1 の結果を利用して，関数 $f(x + \Delta x)$ を x の周りでテイラー展開する式を示せ．

[**問 5**] 次の 3 つの関数を，$x = 0$ の周りでテイラー展開，即ちマクローリン展開せよ．以下の 3 つの関係は，どうなっているか考察せよ．

$$e^x =$$

$$\sin x =$$

$$\cos x =$$

[**問 6**] テイラー展開の結果を利用して，$\theta = 31$[度] の時の $\sin\theta$ と $\cos\theta$ の値を小数点以下 6 桁の精度で求めよ．電卓を使っても良いが，三角関数の機能を使ってはならない．

[**問 7**] 以下の連立方程式を行列とベクトルで表現せよ．

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1N}x_N &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2N}x_N &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \cdots + a_{3N}x_N &= b_3 \\
&\vdots \\
a_{M1}x_1 + a_{M2}x_2 + a_{M3}x_3 + \cdots + a_{MN}x_N &= b_M
\end{aligned}$$

*独立行政法人　秋田工業高等専門学校　電気情報工学科

[**問 8**] 以下の連立方程式を行列とベクトルで表現せよ．

$$\begin{aligned} x_1 + 2x_2 - x_3 + 3x_4 &= 1 \\ -2x_1 - 3x_2 - 5x_4 &= 2 \\ 2x_1 + 2x_2 + 3x_3 + 5x_4 &= 3 \\ -x_1 + 3x_2 - 12x_3 &= 4 \end{aligned}$$

[**問 9**] 前問の連立方程式の解と係数行列の逆行列を求めよ．解と逆行列は掃き出し法を使うこと．

2 C 言語の練習

2.1 ソーティングとは

ソーティングとは，整列あるいは並び替えのことである[1]．プログラミングでは，数値を大きい順，あるいは小さい順に並び替える技法のことを言う．数値の並び替えは非常に重要な技法で，実際のプログラムではいたるところで使われる．ソーティングでもっと重要なことは，処理速度である．高速な処理を目指していろいろなアルゴリズムが考えられている．ここでは，ソーティングの簡単なアルゴリズムを示し，C 言語のプログラムの学習を進める．

2.2 単純挿入法

ソーティングの中でも，最も簡単な単純挿入法のプログラムを作成する．有名な C 言語の本「NUMERICAL RECIPES in C」によると，これは経験を積んだトランプ師が使う方法と同じであるということである．順序がバラバラのトランプを並び替える場合，

- まず，2 枚目のカードを拾い，1 枚目と順序関係が正しい位置に置く．
- 次に 3 枚目のカードを拾い，最初の 2 枚と順序関係の正しい位置にそれを挿入する．
- 同じことを繰り返す．即ち，i 枚目のカードを拾い，最初の $i-1$ 枚のカードの順序関係の正しい位置にそれを挿入する．
- 最後のカードを正しい位置に挿入したら，並び替えは完了である．

この単純挿入法を用いて，リスト 1 で作成された整数の配列 `a[0]`〜`a[1023]` を小さい順に並び替えよ．このリストの 19 行目以降に単純挿入法のプログラムを書く．**ヒント** 図 1 に単純挿入法のフローチャートを示す．

[1] ここでの説明は，NUMERICAL RECIPES in C を参考にしている．

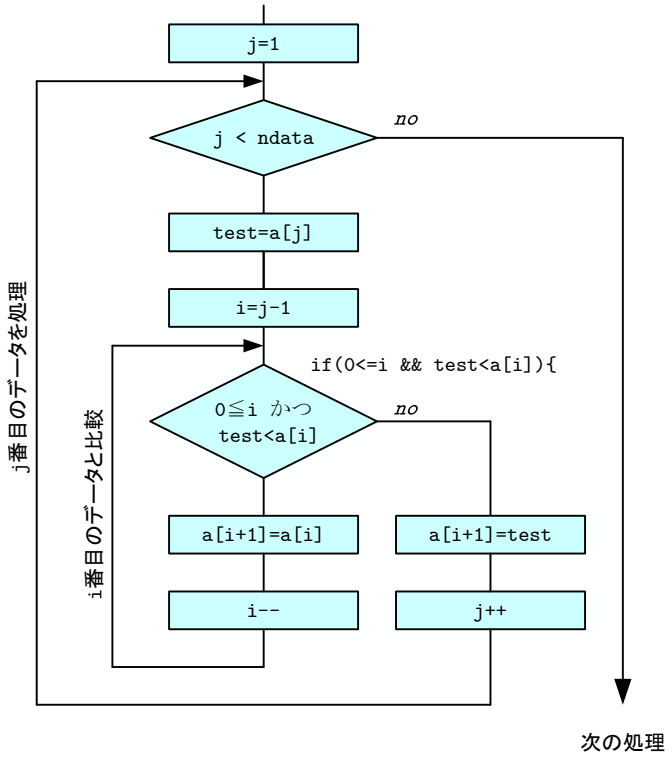

図 1: 単純挿入法のフローチャート．ndata はデータ数で，a[0]〜a[nadata-1] の配列に格納されている整数を小さい順 (昇順) に並べる．

リスト 1: 単純挿入法のプログラム

```
1  #include <stdio.h>
2  #include <stdlib.h>     /* 乱数発生のため       */
3  #include <time.h>       /* 時刻の関数を使うため */
4
5  int main(void){
6    int a[1024], i, j, ndata, test;
7
8    ndata=1024;
9
10   srand((unsigned int)time(NULL));      /* 起動毎に異なる乱数を発生させるため */
11
12   for(i=0; i<ndata; i++){
13     a[i]=rand();                        /* 配列a[i]に乱数の整数を設定 */
14   }
15
16
17
18   /*  これ以降に単純ソートと昇順に並んだ出力のプログラムを書く */
19
```

```
20
21
22
23
24
25
26
27    return 0;
28  }
```

2.3 shellソート

Shell ソート[2]は 1959 年に D.L.Shell が考案した方法で，単純挿入法を改良したものとなっている．バブルソート法は，隣同士を比較するが，Shell ソートでは，大きな h 飛ばしで比較する．ソートに時間のかかる大ききな数や小さな数は，一気に右や左に移動する．h 飛ばしで比較すると，

$$
\begin{aligned}
&a[1] \leqq a[h+1] \leqq a[2*h+1] \leqq a[3*h+1] \leqq a[4*h+1] \leqq a[5*h+1] \cdots \\
&a[2] \leqq a[h+2] \leqq a[2*h+2] \leqq a[3*h+2] \leqq a[4*h+2] \leqq a[5*h+2] \cdots \\
&a[3] \leqq a[h+3] \leqq a[2*h+3] \leqq a[3*h+3] \leqq a[4*h+3] \leqq a[5*h+3] \cdots \\
&\qquad\qquad\qquad\qquad \vdots \\
&a[h] \leqq a[h+h] \leqq a[2*h+h] \leqq a[3*h+h] \leqq a[4*h+h] \leqq a[5*h+h] \cdots
\end{aligned}
$$

と並び替える．この並び替えには単純挿入法をつかう．

そうして，とび幅 h をどんどん小さくし，最後は $h = 1$ にすると並び替えが完了となる．この h の選び方にこつがあって，小さいほうから $1, 4, 13, 40, 121, \cdots$ と

$$h_{i+1} = 3h_i + 1 \qquad\qquad h_1 = 1$$

とするのが良いらしい．良いというのは早いと言うことである．最初に実行する一番大きな h は，データの個数の半分以下にする．

Shell ソートの手順は，次の通りである．

1. 最初の飛び幅 h を決める．データの個数の半分以下で最大の h_i を最初の飛び幅とする．
2. $i = 1, 2, 3, \cdots, h$ に対して，`a[i],a[h+i],a[2*h+i],a[3*h+i],`$\cdots$ を並び替える．
3. 次の `i++`して，並び替える．
4. 次の `h=(h-1)/3` にして，再度，並び替えを実行する．

リスト 1 の 19 行目以降に shell ソートのプログラムを書き，配列を小さい順 (昇順) に並び替えよ．

[2]この辺の説明は，www.rkmath.rikkyo.ac.jp/ kida/shellsort.htm を参考にしている．

3　数値計算の練習

3.1　微分方程式

オイラー法を使って，微分方程式を計算するプログラムを作成せよ．以下，詳細に説明しているので，これを良く読めばプログラムの作成ができるはずである．

3.1.1　計算方法

次の微分方程式

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \cos x \tag{1}$$

をオイラー法により計算する．ただし，初期条件は $x = 0$ の時 $y = 0$ とする．当然，これは数値計算するまでもなく，解は

$$y = \sin x \tag{2}$$

と分かっている．分かっているが，ここではコンピュータープログラムにより計算する．ここでの内容を良く理解すれば，通常解けない微分方程式でも，コンピューターにより計算できることが分かるだろう．

コンピューターで微分方程式を計算する方法を示す．微分方程式の解を $y = f(x)$ とする．すると，微分—正しくは導関数—は，

$$\frac{\mathrm{d}y}{\mathrm{d}x} \simeq \frac{\Delta y}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3}$$

と近似することができる．右辺の $\Delta x \to 0$ の極限が微分の値となる．したがって，元の微分方程式は

$$\frac{f(x + \Delta x) - f(x)}{\Delta x} \simeq \cos x \tag{4}$$

と書くことができる．これと，式 (1) から，

$$f(x + \Delta x) \simeq f(x) + \Delta x \cos x \tag{5}$$

である．これがオイラー法で微分方程式を解くときの基本の式である．Δx の値を小さくすると，この近似の精度が上がる．初期条件を上手に使うと，微分方程式の解の値が芋づる式にわかる．仮りに，$\Delta x = 0.001$ とすると，次のような手順で計算する．

(1)	$f(0.000) = 0$	これは初期条件である．
(2)	$f(0.001) = f(0.000) + 0.001 \times \cos(0.000)$	右辺は (1) の結果を利用する
(3)	$f(0.002) = f(0.001) + 0.001 \times \cos(0.001)$	右辺は (2) の結果を利用する
(4)	$f(0.003) = f(0.002) + 0.001 \times \cos(0.002)$	右辺は (3) の結果を利用する
(5)	$f(0.004) = f(0.003) + 0.001 \times \cos(0.003)$	右辺は (4) の結果を利用する
	$\vdots$	これを繰り返す

(1) は初期条件なので計算するまでもない．そして，(1) の結果を利用すると，(2) の右辺が計算でき，その左辺の値を決めることができる．同様に (2) の結果を利用すると，(3) の右辺が計算でき，その左辺の値が決められる．これを繰り返すのである．すると，$x = 0$ の初期条件からはじめて，任意の x まで $f(x)$ の値を求めることができる．元の微分方程式 (1) の近似解が求まったことになる．$\Delta x = 0.001$ は計算のステップ幅と呼ばれ，これを小さくするとさらに計算時間は必要になるが，計算精度が上がる．

例えば，これを 4000 回この計算を繰り返すと，微分方程式 (1) の解が $y = f(0.000)$ から $f(4.000)$ まで計算できる．すなわち，0〜4[rad] (229.1[度]) までの値が 0.001rad(0.057[度]) きざみで分かるのである．4000 回の計算なんか，コンピューターにとっては大したことはない．私の PC では，計算と表示に用した時間は，0.15 秒である．コンピューターの計算速度には，本当に驚かされる．

3.1.2 計算アルゴリズム

計算の原理が分かったので，プログラミング方法を示す．計算のフローチャートは図 2 のようになる．これにしたがって，プログラムを書けば良い．難しいことは何もない．

まずは，計算のステップ幅 Δx を決めなくてはならない．C 言語では

```
dx=4.0/4000;
```

と書く．プログラムではギリシャ文字は使えないので，ステップ幅を dx としている．解となる計算結果は後で利用することを考えて，配列に格納する方が良い．配列の先頭には，初期条件を格納する．すなわち，

```
x[0]=0.0;
y[0]=0.0;;
```

とするのである．次に i の値を 1〜4000 まで変えて，

```
x[i]=i*dx;
y[i]=y[i-1]+dx*cos(x[i-1));
```

を繰り返し計算する．このように計算回数が決まっている繰り返しには，for 文を使うのが一般的である．

```
for(i=1;i<=4000;i++){

   ここに繰り返したい内容を書く．

}
```

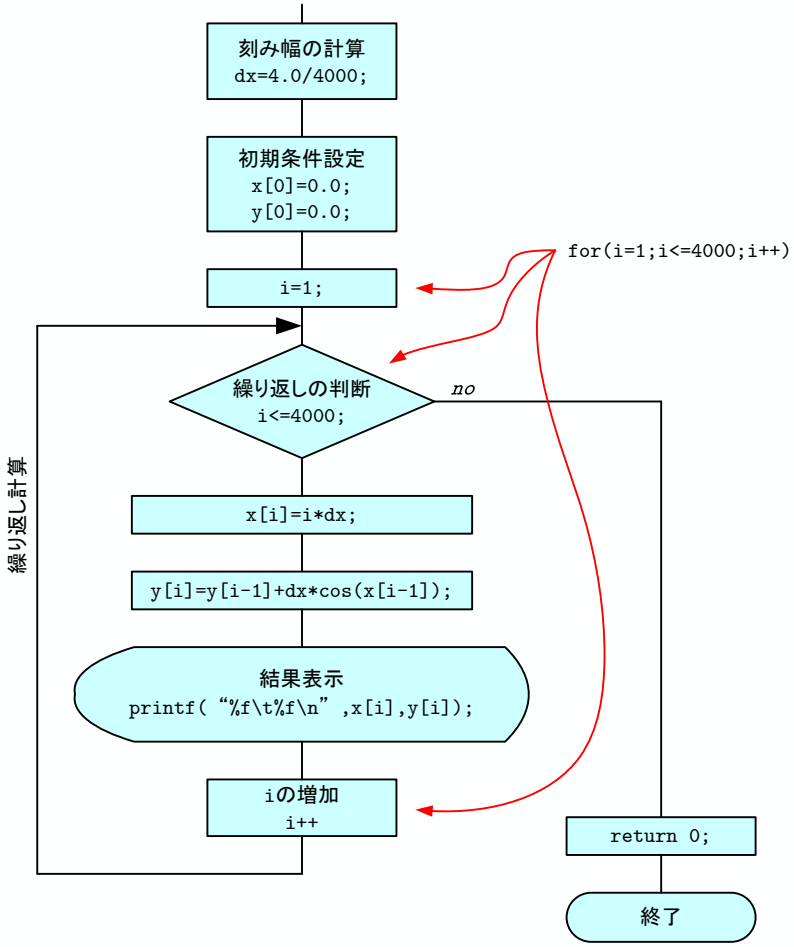

図 2: オイラー法のフローチャート.

3.2 連立方程式

次の連立方程式

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

を解くプログラムを作成する．解 (x_1, x_2, x_3) を求めろ—ということである．条件は，以下の通りとする．

- 9 個の係数の a_{11}〜a_{33} はキーボードから入力する．

- 3 個の非同次項 b_1〜b_3 もキーボードから入力する．
- 連立方程式の解 (x_1, x_2, x_3) を表示する．

3.3 課題提出要領

提出方法は，次の通りとする．

期限	10 月 4 日 (木) 20:00
用紙	A4
提出場所	山本研究室の入口のポスト
表紙	表紙を 1 枚つけて，以下の項目を分かりやすく記述すること． 授業科目名「計算機応用」 課題名「課題　夏休みの宿題」 5E　学籍番号　氏名 提出日
内容	ソースプログラムは，プリントアウト，手書き，いずれも OK とする

なお，これは後期の成績に加味する．