Cプログラミング

第3回
連立一次方程式の解法(1)

青柳滋己
aoyagi@aoni.waseda.jp

本日の講義内容

- ファイル入出力
- 連立一次方程式の解法(1)
  - ガウス消去法
- 演習

前回の演習の解答例(1)

kadai2_1.c

```
#include <stdio.h>

void swap(int *a, int *b)
{
  int tmp;

  tmp=*a; *a=*b; *b=tmp;
}

int main(void)
{
  int x=100, y=200;

  printf("x=%d, y=%d¥n", x, y);
  swap(&x, &y);
  printf("Swap: x=%d, y=%d¥n", x, y);
  return 0;
}
```

複合代入演算子(復習)

- 変数 = 変数 演算子 式; のときに
  変数 演算子= 式; とかける.

例:

```
int a,b;

a = a + b;      ->    a += b;
a = a - b;      ->    a -= b;
a = a * b;      ->    a *= b;
a = a / b;      ->    a /= b;
a = a % b;      ->    a %= b;
```

+=
-=
*=
/=
%=

インクリメント・デクリメント演算子(復習)

- 変数aの値を1増やす(減らす)

```
a = a + 1;
```

```
a = a - 1;
```

これを次のように書ける.

```
++a;
a++;
```

```
--a;
a--;
```

  - 前置型と後置型

```
++a;      前置型
a++;      後置型
```

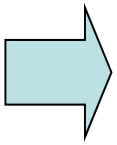
```
--a;      前置型
a--;      後置型
```

インクリメント・デクリメント演算子(復習)

前置型

```
int a, b;

a = 3;
b = ++a;
```

a=4, b=4　となる.

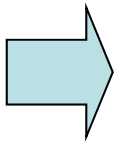


```
int a, b;

a = 3;
a = a + 1;
b = a;
```

後置型

```
int a, b;

a = 3;
b = a++;
```

a=4, b=3　となる.



```
int a, b;

a = 3;
b = a;
a = a + 1;
```

※　−−も同様.

前回の演習の解答例(2)

kadai2_2.c

```
#include <stdio.h>

int main(void)
{
  int data[]={10,8,6,4,2};
  int *p=data, i;

① printf("%d¥n", (*p)++);
② printf("%d¥n", *p++);
③ printf("%d¥n", --(*p));
  for (i=0; i<5; i++)
       printf("%d ", data[i]);
  printf("¥n");
  return 0;
}
```

```
% gcc -o kadai2_2 kadai2_2.c
% ./kadai2_2
10     …答え
11     …答え
7
11 7 6 4 2
%
```

```
p=&data[0];    /* 初期値 */
①の処理を分解すると
printf("", *p);   /* data[0]=10; */
(*p)++;           /* *p = *p + 1; */
                  /* → data[0]++; */
                  /*  data[0]=11 */

②の処理
printf("", *p);   /* data[0]=11 */
p++;              /* p++ → p = p + 1; */
                  /*   ⇒ p = &data[1]; */


③の処理           /* data[1]=8 */
--(*p)            /* *p = *p - 1; */
                  /* → data[1]-- */
                  /*  →  data[1]=7 */
printf("", *p);   /* data[1]=7 */
```

前回の演習の解答例(3)

kadai2_3.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLINE 256

void getsn(char *str, int len, FILE *fp)
{
  int i;

  fgets(str, len, fp);
  i=strlen(str);
  if (i==0)return;
  if (str[i-1]=='¥n') str[i-1]='¥0';
}  /* 改行があった場合削除('¥n'を'¥0'にする) */

int reverse(char *org, char *new)
{
  int i, len;

  len=strlen(org);
  if (len == 0) return 0;
/* 右へ続く */
```

kadai2_3.c

```
  for (i=0; i<len ;i++) new[i]=org[len-1-i];
  new[i]='¥0';
  return len;
}

int main(void)
{
  char str[MAXLINE], str1[MAXLINE];
  int i, len;
  FILE *fp;

  printf("Input?: ");
  getsn(str, MAXLINE, stdin);
  if (reverse(str, str1)) {
    if ((fp=fopen("data.txt","w")) == NULL) {
      fprintf(stderr, "cannot open file¥n");
      exit(1);
    }
    fputs(str1, fp); fputs("¥n", fp);
    fclose(fp);
  }
  return 0;
}
```

ファイル(復習)

- ファイルにはテキスト形式とバイナリ形式の2種類が存在
  - テキスト…文字のみが書かれているファイル．cat等のプログラムで画面に表示しても問題ない.
  - バイナリ…a.outファイルなど，文字以外の情報が入ったファイル．cat等で画面表示させようとすると端末がおかしくなる.
- C言語におけるファイルの扱いは上のテキストとバイナリで異なる.

ファイル入出力の手続き(復習)

- Cでのファイル入出力関数には高レベル入出力関数と低レベル入出力関数の2種類が存在.
- 本講義では高レベル入出力関数しか扱わない
- Cでのファイル入出力は次の手順で行う.

1. ファイル名を指定してファイルをオープンする.
   fopen 関数を用いる
2. ファイルの読み書きを行う.

   fscanf, fprintf, fgetc, fputc など
   fread, fwrite(バイナリ)
3. ファイルをクローズする.
   fclose関数

- fopen(),fclose()は必ず行うこと.

ファイルのオープン・クローズ(復習)

- ファイル入出力はFILE構造体へのポインタであるファイルポインタを使う.
- ファイルオープンするにはfopenを使う

```
fopen(ファイル名, モード);
```

- 使い方の例

```
#include <stdio.h>
.....
FILE *fp;  /* ファイルポインタ */

fp=fopen("testdata.txt", "r");
if (fp == NULL) { /* オープン失敗時の処理 */
......
}
```

```
/* ファイル読み込み */
while
FILE *fp;  /* ファイルポインタ */

fp=fopen("testdata.txt", "r");
if (fp == NULL) { /* オープン失敗時の処理 */
......
```

- ファイル名やモードは""で囲むこと.
- 一つのファイルポインタで同時に複数のファイルをオープンすることはできない.

ファイルのオープン・クローズ(復習)

- fopenの主なモードは右の通り.

モード	意味
"r"	読み出し専用
"w"	書き込み専用. ファイルが存在しなければ作成される. ファイルが存在すると前のファイルの中身はなくなる
"a"	追加書き込み専用. ファイルが存在しなければ作成される. ファイルが存在すると, 前の中身の後に追加される.

- ファイルのクローズは, fcloseの引数にファイルポインタを指定する.

```
fclose(fp);
```

fopenが失敗したとき(復習)

- 存在しないファイルを読み込もうとした時や，ファイルが他プロセスによって既に開いている時は，fopenが失敗する．
- fopenが失敗すると，NULLポインタが返されるので，エラーチェックを行う．exit();はそこでプログラムの実行が終了する．引数の値がプログラムの実行値になる．

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

FILE *fp;
.....
fp=fopen("testdata.txt", "r");
if (fp == NULL) {
  printf("fopen error¥n")
  exit(1);
}
```

```
/* 左のプログラムと同じ */
#include <stdio.h>
#include <stdlib.h>

FILE *fp;
.....
if ((fp=fopen("testdata.txt", "r")) == NULL) {
  printf("fopen error¥n")
  exit(1);
}
```

テキストファイルの読み書き

- fprintf，fscanfを用いた例.
- test.txtファイルに0,1^2,2^2,...10^2 を書き込んだ後，その数値を読み込み，画面表示する.

sample.c

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

int main(void)
{
  FILE *fp;
  int i;
   /* test.txtを書き込みモードでオープン */
  if ((fp=fopen("test.txt", "w"))
        == NULL) {
    printf("Cannot open file.¥n");
    exit(1);
  }
  for (i=0; i<100; i++)
    fprintf(fp, "%d¥n", i*i); /* 書き込み */
  fclose(fp); /* ファイルクローズ */
```

sample.c

```
/* 左からの続き */
  /* 読み込みモードでオープン */
  if ((fp=fopen("test.txt", "r"))
        == NULL) {
    printf("Cannot open file.¥n");
    exit(1);
  }
  while (fscanf(fp, "%d", &i)   /* 読み込み */
         != EOF) { /* End Of File */
    printf("%d¥n", i);
  }
  fclose(fp); /* ファイルクローズ */
  return 0;
}
```

fprintfとfscanf

- fprintfはprintfの第一引数にファイルポインタが指定できるようになったもの.
- fscanfはscanfの第一引数にファイルポインタが指定できるようになったもの.
- EOFはEnd Of Fileのことで，ファイルの終了に達すると，fscanfはこのEOFの値を返す.
- stdio.hには標準入出力 stdin，stdout も定義されていて，

  fprintf(stdout，...)；

  fscanf(stdin，...)；

はそれぞれ printf，scanf と同じ標準入出力に対する操作となる. (stdin,stdoutの他，stderr(標準エラー出力)もある)

fgetcとfputcの例

- 一文字単位のファイル読み書きを行うコピープログラム

samp2.c

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

int main(int argc, char *argv[])
{
  FILE *fp, *fp1;
  int ch;

  if (argc != 3) {
    printf("Usage: %s <source> <destination>¥n", argv[0]);
    exit(1);
  }
  if ((fp=fopen(argv[1], "r")) == NULL) {
    printf("Cannot open %s¥n", argv[1]);
    exit(1);
  }
  if ((fp1=fopen(argv[2], "w")) == NULL) {
    printf("Cannot open %s¥n", argv[2]);
    exit(1);
  }
```

samp2.c

```
/* 左からの続き */
  while ((ch=fgetc(fp)) != EOF) {
    fputc(ch, fp1);
  }
  fclose(fp);
  fclose(fp1);
}
```

バイナリファイルの読み書き(参考)

- fopen関数で第二引数に"rb"や"wb"などを指定する.
- バイナリファイルを読み書きする場合，バッファを用意して，fread，fwriteを使う.

```
int fread(char *buffer, int size, int count, FILE *stream);
int fwrite(char *buffer, int size, int count, FILE *stream);
```

buffer: データの読み書きに使う領域.

size:　読み書きするブロックの単位をバイト数で指定

count: 読み書きするブロックの個数

stream:　読み書きするファイルポインタ

ランダムアクセス(参考)

- freadやfscanfなどを使って読み込みを行う時には，先頭から順次読み込まれていく．これをシーケンシャルアクセスと呼ぶ．
- バイナリファイル等の場合，任意の場所から読み込みを開始したい時がある．これをランダムアクセスと呼ぶ．読み込み開始位置を変更するにはfseekを用いる．

```
int fseek(FILE *stream, long offset, int whence);
```

- whenceに指定できるのは右表の3つ
- 現在の読み込み位置を知るには，ftellを使う．

```
long ftell(FILE *stream);
```

SEEK_SET	ファイルの先頭
SEEK_CUR	現在位置
SEEK_END	ファイルの末尾

例：ファイルサイズを調べる方法の概要：

```
fp=fopen("test.dat", "rb")
fseek(fp, 0L, SEEK_END);
filesize=ftell(fp);
fseek(fp, 0L, SEEK_SET);
```

バイナリファイル読み書きの例(参考)

samp3.c

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

#define BUFSIZE 512

int main(int argc, char *argv[])
{
  FILE *fp, *fp1;
  char buf[BUFSIZE];
  int size;

  if (argc != 3) {
    printf("Usage: %s <source> <destination>¥n", argv[0]);
    exit(1);
  }
  if ((fp=fopen(argv[1], "rb")) == NULL) {
    printf("Cannot open %s¥n", argv[1]);
    exit(1);
  }
  if ((fp1=fopen(argv[2], "wb")) == NULL) {
    printf("Cannot open %s¥n", argv[2]);
    exit(1);
  }
```

samp3.c

```
/* 左からの続き */
  while ((size=fread(buf, 1, BUFSIZE, fp)) > 0) {
    fwrite(buf, 1, size, fp1);
  }
  fclose(fp);
  fclose(fp1);
  return 0;
}
```

ガウス消去法

$$\begin{cases} 3x_1 + x_2 + 2x_3 = 13 \\ 5x_1 + x_2 + 3x_3 = 20 \\ 4x_1 + 2x_2 + x_3 = 13 \end{cases}$$

という連立方程式を考える．これを行列を使うと以下のように$Ax=b$ の形式で表すことができる．

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 1 & 3 \\ 4 & 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ 20 \\ 13 \end{pmatrix}$$

ガウス消去法では，前進消去，後退代入の順に行って解を求める．

前進消去(1/2)

- 第1列の2行目以降を0にする
  - 第1行×(-5/3)＋第2行

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 4 & 2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ -\frac{5}{3} \\ 13 \end{pmatrix}$$

  - 第1行×(-4/3)＋第3行

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & \frac{2}{3} & -\frac{5}{3} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ -\frac{5}{3} \\ -\frac{13}{3} \end{pmatrix}$$

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 1 & 3 \\ 4 & 2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ 20 \\ 13 \end{pmatrix}$$

前進消去(2/2)

- 第2列の2行目以降を0にする
  - 第2行×1＋第3行

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ -\frac{5}{3} \\ -6 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & \frac{2}{3} & -\frac{5}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ -\frac{5}{3} \\ -\frac{13}{3} \end{pmatrix}$$

後退代入

- x_3, x_2, x_1 の順に代入して答えを求める
  - 第3行目より

$$x_3 = \frac{-6}{-2} = 3$$

  - 第2行目より

$$x_2 = -\frac{3}{2}\left(-\frac{5}{3} + \frac{1}{3} \times 3\right) = 1$$

  - 第1行目より

$$x_1 = \frac{1}{3}(13 - 1 \times 1 - 2 \times 3) = 2$$

$$\begin{pmatrix} 3 & 1 & 2 \\ 0 & -\frac{2}{3} & -\frac{1}{3} \\ 0 & 0 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 13 \\ -\frac{5}{3} \\ -6 \end{pmatrix}$$

ガウス法手順まとめ(1/3)

*n*元一次連立方程式に一般化する

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

(1)前進消去

$$\alpha_{i1} = -\frac{a_{i1}}{a_{11}},\ (a_{11} \neq 0,\ i = 2,3,\ldots,n)$$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix}$$

$$a_{ij}^{(1)} = a_{ij} + \alpha_{i1} a_{1j},$$

$$b_i^{(1)} = b_i + \alpha_{i1} b_1,\quad i = 2,3,\ldots,n$$

となる．上付き文字（1）は値が1回変更されたことを示す．

ガウス法手順まとめ(2/3)

これを2行目,3行目…と行っていく. k回目では.

$$\alpha_{ik} = -\frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}},\ (a_{kk}^{(k-1)} \neq 0,\ i = k+1, k+2, \ldots, n)$$

を用いて, $a_{ij}^{(k)} = a_{ij}^{(k-1)} + \alpha_{ik} a_{kj}^{(k-1)}$,

$b_i^{(k)} = b_i^{(k-1)} + \alpha_{ik} b_k^{(k-1)},\ \ i, j = k+1, k+2, \ldots, n$ を計算する.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1k} & \cdots & a_{1,n-1} & a_{1n} \\ & a_{22}^{(1)} & \cdots & a_{2k}^{(1)} & \cdots & a_{2,n-1}^{(1)} & a_{2n}^{(1)} \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & a_{kk}^{(k-1)} & \cdots & a_{k,n-1}^{(k-1)} & a_{kn}^{(k-1)} \\ & & & & \ddots & \vdots & \vdots \\ & & & & & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\ & & & & & & a_{n,n}^{(n-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ \\ b_k^{(k-1)} \\ \\ b_{n-1}^{(n-2)} \\ b_n^{(n-1)} \end{pmatrix}$$

$a_{ij}^{(k)}$の上付き文字(k)は値が1回変更されたことを示す.

ガウス法手順まとめ(3/3)

(2)後退代入

一番最後の式から, $x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}$ となる. x_nが決まると x_{n-1}, x_{n-2}…も順に決まる.

$$x_{n-1} = \frac{b_{n-1}^{(n-2)} - a_{n-1,n}^{(n-2)} x_n}{a_{n-1,n-1}^{(n-2)}}, \quad x_{n-2} = \frac{b_{n-2}^{(n-3)} - a_{n-2,n-1}^{(n-3)} x_{n-1} - a_{n-2,n}^{(n-3)} x_n}{a_{n-2,n-2}^{(n-3)}}$$

x_kは次の式で求められる.

$$x_k = \frac{b_k^{(k-1)} - \sum_{j=k+1}^{n} a_{kj}^{(k-1)} x_j}{a_{kk}^{(k-1)}}$$

本日の演習

- ガウス消去法プログラムを次ページを参考に完成させ，次の連立方程式を解け(kadai3_1.c)

$$\begin{cases} x_1 + 2x_2 + x_3 + x_4 = -1 \\ 4x_1 + 5x_2 - 2x_3 + 4x_4 = -7 \\ 4x_1 + 3x_2 - 3x_3 + x_4 = -12 \\ 2x_1 + x_2 + x_3 + 3x_4 = 2 \end{cases}$$

- １～６の整数乱数を100個ファイルに出力するサイコロプログラムを作成せよ(kadai3_2.c). (出力するファイル名は適当で良い)
- １～６の数字が100個入ったファイルを読み込み，各数字の個数を画面に出力するプログラムを作成せよ. (kadai3_3.c)

※ Course N@viにてプログラムを提出(締め切り：本日中)

参考

kadai3_1.c

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

#define N 4

void gauss(double a[][N+1], double b[]);

double mat1[N+1][N+1]={
  {0.0, 0.0, 0.0, 0.0, 0.0},
  {0.0, 1.0, 2.0, 1.0, 1.0},
  {0.0, 4.0, 5.0, -2.0, 4.0},
  {0.0, 4.0, 3.0, -3.0, 1.0},
  {0.0, 2.0, 1.0, 1.0, 3.0}};
double mat2[N+1]={0.0, -1.0, -7.0, -12.0, 2.0};

int main(void)
{
  int i;

/* 右へ続く */
```

kadai3_1.c

```
  gauss(mat1, mat2);

  printf("¥nAnswer:¥n"); /* 解の表示 */
  for (i=1; i<=N; i++) printf("%f ", mat2[i]);
  printf("¥n");

  return 0;
}

void gauss(double a[][N+1], double b[])
{
  int i,j,k;
  double alpha, tmp;

/* 以下を完成させる */
  /* 前進消去 */

  /* 後退代入 */

}
```