

プログラム作成時の注意 (2007/06/01)

・ヘッダの読み込みについて

コンパイル時に警告が出たり、コンパイルできないときは、ヘッダの読み込みが適切でないことがあります。ほとんどの課題では、

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

としておき、コンパイル時にオプション「-lm」をつけます。

・ヤコビ法・共役勾配法の初期値

ヤコビ法・共役勾配法の初期値は、プログラム内で与えるようにしてください。

・入力ファイルの配布

提出前に各自でプログラムの動作をチェックするための入力ファイルを

```
http://www.math.sci.hokudai.ac.jp/~s063039/cm1/test1.tar.gz
```

に置いておきます。ブラウザ (firefox) でダウンロードするか、次のようにしてください。

```
$ wget http://www.math.sci.hokudai.ac.jp/~s063039/cm1/test1.tar.gz
$ tar xvf test1.tar.gz
$ ls
```

最後に書いてある ls を実行して、test_data ディレクトリが存在することを確認してください。この test_data ディレクトリの中にテストファイルがあります。

・配列の範囲

```
double a[5];
```

と定義すると、a[0]、a[1]、a[2]、a[3]、a[4] を使用することができます。それ以上の a[5] や a[6] を参照している場合は、おかしいことが起こります。コンパイルは成功することがあるので、なぜかおかしい値が出る場合は、確認してください。

・変数の初期化

変数を定義したときにある値を代入しない限りは、さまざまな値を持っています(0とは限りません)。その変数を使って計算する前に、代入をしてください。

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

int main(){
    double a[10][10];

    int i,j;
    for(i=0; i<10; i++){
        for(j=0; j<10; j++){
            printf("a=%f\n", a[i][j]);
        }
    }

    return 0;
}
```

を実行すると、実感できるでしょう。

・計算するときの次元

計算が行われる空間の次元を

```
#define N 4
```

のようにしておくともプログラムが見やすくなります。たとえば、配列の定義で

```
double a[N], b[N][N];
```

として使います。おなじようでも

```
int i=4;
double a[i], b[i][i];
```

のように変数を持ちいるのは、トラブルの元なのでやめてください。