

第10回:フーリエ変換ー2

fft.hについて

- /* 関数のプロトタイプ */

(1) int calc_power_of_two(int number);

(2) void make_initial_data(double *re_part, double *im_part, int num_of_data, int power);

(3) void FFT1(double *re_part, double *im_part, int num_of_data, int flag);

- 1次元のFFT

(4) void FFT2(int flag);

- 2次元のFFT

FFT1

```
void FFT1( double *re_part, double *im_part, int  
num_of_data, int flag )
```

```
/* データの FFT (flag = 1), 逆 FFT (flag = -1) を行う関数      */
```

```
/* double *re_part, *im_part; 元データ(実数部, 虚数部)の先頭アドレス */
```

```
/* int num_of_data, flag;   データ総数, FFT・逆FFT を決めるフラグ */
```

FFT2

```
void FFT2( int flag )
{
    int i, j; /* ループ変数 */
    static double re[FFT_MAX], im[FFT_MAX]; /* 作業変数 */
    for (i=0; i<num_of_data; i++){
        for (j=0; j<num_of_data; j++){
            re[j]=data[i][j]; im[j]=jdata[i][j];
        }
        FFT1( re, im, num_of_data, flag );
        for (j=0; j<num_of_data; j++){
            data[i][j]=re[j]; jdata[i][j]=im[j];
        }
    }
}
```

```
for (i=0; i<num_of_data; i++){
    for (j=0; j<num_of_data; j++){
        re[j]=data[j][i]; im[j]=jdata[j][i];
    }
    FFT1( re, im, num_of_data, flag );
    for (j=0; j<num_of_data; j++){
        data[j][i]=re[j]; jdata[j][i]=im[j];
    }
}
}
```

演習内容

- 正弦波の振幅, 周波数をキーボードから入力させ, その信号に対してフーリエ変換を行うプログラムを作成せよ

実行例

サンプリング周波数を入力してください.

1000

生成する正弦波の周波数を入力してください.

10

振幅を入力してください.

1

データ数を入力してください.

100

実行例

サンプリング周波数を入力してください.

1000

生成する正弦波の周波数を入力してください.

50

振幅を入力してください.

1

データ数を入力してください.

2048

プログラムの雛型

ヘッダファイルのインクルード

```
#include "math.h"
```

```
#include "pgmlib.h"
```

```
#include "fft.h"
```

プロトタイプ宣言と帯域変数の宣言

```
void make_sin(void);
```

```
void save_spectrum(double *re, double *im);
```

```
double fs; // サンプリング周波数
```

main関数の定義

```
make_sin();
```

```
FFT1(data[0],jdata[0],num_of_data,1);
```

```
save_spectrum(data[0],jdata[0]);
```

実数データと虚数データを引数とする。
(画像処理を念頭に2次元配列のデータを利用している)

```
void make_sin(void)
{
    double f,A;
    printf("サンプリング周波数を入力してください。 ¥n");
    scanf("%lf",&fs);
    printf("生成する正弦波の周波数を入力してください。 ¥n");
    scanf("%lf",&f);
    printf("振幅を入力してください。 ¥n");
    scanf("%lf",&A);
    printf("データ数を入力してください。 ¥n");
    scanf("%d",&num_of_data);
    for(int i=0;i<num_of_data;i++){
        data[0][i] = A*sin(2*PI*f*(double)i/fs);
        //data[0][i] = A*sin(2*PI*f*(double)i/fs) +
        //A/2*sin(2*PI*(3*f)*(double)i/fs);
        jdata[0][i] = 0.0;
    }
}
```

save_spectrum関数

```
void save_spectrum(void)
{
    FILE *fp;
    double power;
    if((fp = fopen("spectrum.csv","w")) == NULL){
        printf("file not open");
        exit(0);
    }
    for(int i=0;i<num_of_data;i++){
        power = sqrt(pow(data[0][i],2)+pow(jdata[0][i],2));
        fprintf(fp,"%lf,%lf¥n",i*fs/num_of_data, power);
    }
}
```