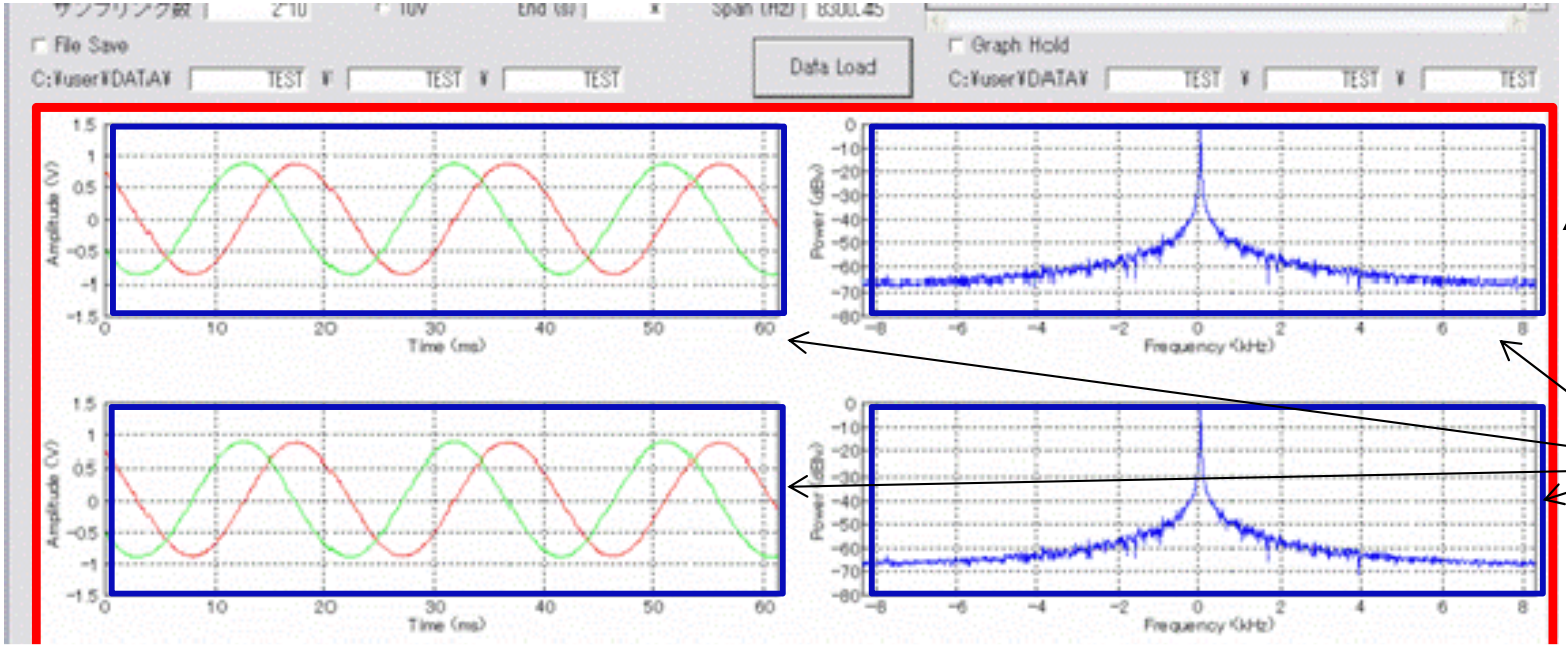
４回目 グラフ作成ライブラリ

１次元プロット関数関連

Clf, Set_figure, Aspect_ratio
Plot1d, Plot1d_int, Plotxy
Axis_xcap, Axis_ycap,
Grid_on, Legend
Text_draw

グラフィックウインドウの設定



グラフィック
ウインドウ

フィギュア
ウインドウ

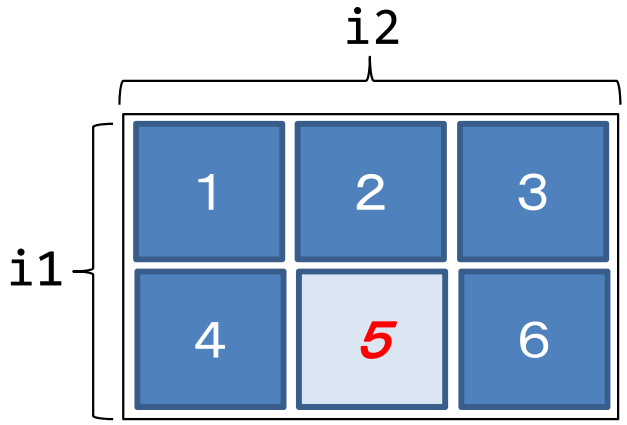
・グラフィックウインドウは図が描画されるウインドウ
・サイズはdef.h内、定数MAINWIN_W と MAINWIN_H を変更すればよい
　最終行付近のsGW構造体を変更してもよい

・Set_figure()関数により、複数のフィギュアウインドウを配置可能である
・各フィギュアウインドウの左、右、上、下マージンはdef.h 190行目付近
MERGIN_L, MERGIN_R, MERGIN_T, MERGIN_Bで変更できる。
・特にy軸の数値ラベル幅が広いときなど、適宜MERGIN_Lを大きくする必要がある
・Plot_1d(), Plot_2d()関数等でグラフを作成することができる

フィギュアウインドウの生成

フィギュアウインドウ　グラフィックウインドウ内にあるグラフ作成用の仮想ウインドウで一つのフィギュアウインドウにはある軸の値を持ったグラフを描くことができる。
Plot関数等は指定したフィギュアウインドウに表示される

```
void Set_figure(int i1,int i2,int n)
```



例　Set_figure(2,3,5);

- グラフィックウインドウをi1（縦の枠数）,i2（横の枠数）で構成されるマトリクス内のn番目の位置にフィギュアウインドウを作成する。
- i1,i2を変更するときは、Clf(2)で全消去する
- Set_figure(…)により図の位置に固有にのフィギュアIDが割り当てられる。
- カレントのフィギュアウインドウ番号はglobal変数のiID_Cfwが保持
- 初めて指定したときはフィギュアウインドウの最大最小値がクリア
- いったん設定すれば、最大値最小値の値は記憶。二回目以降に指定したときは、前に指定したフィギュアウインドウの軸設定値を使用
- 一度指定したフィギュアウインドウに違う軸を持つ図を描画する場合はClf(1)で一旦、図や軸の情報を消去して、再指定する必要

フィギュアウインドウの設定

```
void Aspect_ratio(double ax, double ay)
```

例） `Aspect_ratio(1,1);`

フィギュアウインドウのサイズを(幅:高さ)＝(ax : ay)で表される比率に変更

プロット関数より前に指定する必要

標準ではグラフィックウインドウのサイズはSet_figure関数で決まるウインドウ個数に従って均等に割り振り

この関数より、標準サイズの縦横サイズを超えない範囲で指定した比率にサイズを定める。

```
void Clf(int clflug)
```

例） `Clf(1);`

フィギュアウインドウの消去を行う

clflug

-1	現在のフィギュアウインドウの枠だけ描画（消去はしない）
0	現在のフィギュアウインドウの枠を描画し、枠内を消去 、軸情報は残される。
1	現在のフィギュアウインドウを軸のラベルも含めて消去、最大最小の軸情報もクリア
2	グラフィックウインドウ全体を消去、全てのフィギュアウインドウの最大最小の軸情報もクリア

プロット関数（１）

```
void Plot1d(double *yn ,int n)
```

例） **Plot1d(x,100);**

指定されているフィギュアウインドウにdouble型配列 yn[ ] のグラフを描画

yn[ ]: double型の配列変数名(c言語内で配列変数名ynは&yn[0]と同じ)
（100番目の要素から表示する場合 → &yn[99] とする）
n： 表示するデータの個数

Axis_ycap(…)で軸の最大最小が指定されていないときは、自動的に配列内の最大値、最小値を探しy軸を設定、数値ラベルの描画

色、線の種類はPlot_pen( )で指定

表示したいデータの個数が定義された要素数を超えると、無意味なデータが表示されるため注意が必要。

nが定義した配列要素数よりも小さければ問題ない

Clf(1)もしくはClf(2)関数で図のクリアをしなければ、同じウインドウに複数(<20)のグラフを描画可

プロット関数（2）

```
void Plot1d_int(int *yn, int n)
```

例） **Plot1d_int(x,100);**

現在Set_figure(…)で指定されているフィギュアウインドウに**整数型配列** yn[ ]のグラフを描画。

yn[ ]: **整数型**の変数名（100番目の要素から表示 → &yn[99] 等）
n : 表示するデータの個数
Axis_ycap(…)で軸の最大最小が指定されていないときは、自動的に配列内の最大値最小値を探しY軸を設定、描画

色、線の種類はPlot_pen(…)で指定

表示したい配列の要素数が定義された要素数を超えると、無意味なデータが表示されるため注意が必要

nが定義よりも小さければ問題ない

Clf(1)もしくはClf(2)関数で図のクリアをしなければ、同じウインドウに複数(<20)のグラフを描画可

プロット関数(3)

```
void Plotxy(double *xn , double *yn ,int n)
```

例） `Plotxy(x,y,100);`

指定されているフィギュアウインドウにdouble型配列 xn[ ] をx軸、double型配列 yn[ ] をy軸に対応させたグラフを描画。対数グラフ等に利用

xn[ ], yn[ ] :double型の配列変数名
（ 100番目の要素から表示 → &xn[99] , &yn[99] 等）
n : 表示するデータの個数
Axis_ycap(…), Axis_xcap(…)で軸の最大最小が指定されていないときは、自動的に配列内の最大値最小値を探しy軸を設定、描画

色、線の種類はPlot_pen()で指定

表示したい配列の要素数が定義された要素数を超えると、無意味なデータが表示されるため注意が必要, nが定義よりも小さければ問題ない

Clf(1)もしくはClf(2)関数で図のクリアをしなければ、同じウインドウに複数(<20)のグラフを描画可

グラフィックウインドウの描画色指定

```
void Plot_pen(int pf, int pw, int pc)
```

例） `Plot_pen(0,0,1);`

グラフィックウインドウに描画する色，線種を指定

図形描画だけでなく、グラフのプロット関数の線色、線種指定にも使う

pf :
0 - 実線
1 - 破線
2 - 点線
3 - １点鎖線
4 - ２点鎖線
5 - 描画しない

pw:
ペンの太さ。
ピクセル単位
で指定。
0は1ピクセル

pc:
0-黒
1-赤
2-緑
3-青
4-黄色
5-マゼンダ（水色）
6-シアン（ピンク）
7-白

pcに8以上を指定していれば、Plot()関数等のグラフ描画関数を呼び出す毎に自動的に色を0～6まで巡回させる。

色を指定するときは、LINE関数、RECT関数等より前にこの関数で指定する。
1回指定すれば、その色、線種の状態は保持される。

フィギュアウインドウの縦軸設定

```
void Axis_ycap(double min, double max, char *label)
```

例) Axis_ycap(-1,1,“Time [s]");

y軸の表示範囲（最小、最大値）の設定、数値ラベル、キャプションを表示

min ： y軸の最小値　　max ： y軸の最大値

label : y軸キャプションの文字列（100文字以内）、及び文字変数の先頭アドレス

プロット関数実行時に**y軸数値ラベルはデータの最大最小値から自動的に生成**

Y軸描画範囲があらかじめ決まっているときはプロット関数より先にAxis_ycap関数で、範囲を指定すると、表示範囲を任意に設定可

数値ラベルの間隔はminからmaxの間の適当な数値を自動的に計算して描画

フィギュアウインドウの横軸設定

```
void Axis_xcap(double min, double max, char *label)
```

例) `Axis_xcap(-5,5,"x");`

x軸の最小、最大値の設定、数値ラベル、キャプションを表示

min ： x軸の最小値　　max ： x軸の最大値

label : x軸キャプションの文字列（100文字以内）、及び文字変数の先頭アドレス

Axis_xcap関数では、数値ラベルやキャプションを描画するのみ。

x軸の数値ラベルはPlot1d関数等で自動生成されないため、Axis_xcap関数でx軸の数値ラベルを必ず指定する必要

数値ラベルの間隔はminからmaxの間の適当な数値を自動的に計算して描画

x軸描画範囲はmin,maxを変更しても、表示範囲は変更されないことに注意、すなわち、 Plot1d関数等で表示するx軸データの1番目と最後のデータに対応した数値ラベルを自分で考えて設定する必要

未定のときは min=max=0 を指定すれば、キャプションのみを描画

グラフ装飾関数（1）

```
void Grid_on(int grflug)
```

例） `Grid_on(3);`

図にグリッド線を描画

数値ラベルの値のある座標に直線を引く

軸の最大最小を決めた（Plot()関数、Axis_ycap(), Axis_xcap()）後に指定

grflug
- 0 ： y軸の0レベルのみを引く
- 1 ： y軸のみグリッドライン描画
- 2 ： x軸のみグリッドライン描画
- 3 ： x軸、y軸両方にグリッドライン描画

```
void Legend(char* text, int posflug)
```

例） `Legend("a|b|c",4);`

Plot()関数で重ね書きされたグラフの数だけ凡例を表示

凡例textは"line1|line2|line3"のように表記する。線の名前の区切り文字は'|'

posflug
- 0 ： 枠外右下に表示
- 1 ： 枠内左上に表示
- 2 ： 枠内右上に表示
- 3 ： 枠内左下に表示
- 4 ： 枠内右下に表示

グラフ装飾関数(2)

```
void Text_draw(double x , double y, char* text);
```

textで指定された文字を表示する。　　例） `Text_draw(10,200,"sin");`

表示位置の座標(x,y)はフィギュアウインドウの各軸の座標が基準

グラフのタイトル、凡例に表記できない情報等の記述に使用できる。

フォントの変更

（def.h内に定義されたglobal変数を変更する。これらの変数はmain関数内で変更しても以後のフィギュアウインドウの表示に反映される）

フォントの指定　　TCHAR型文字列　Used_Font

例　`lstrcpy(Used_Font,TEXT("MS GOTHIC"));`　ゴシック体に

使用可能フォント　　TIMES, MS GOTHIC, MS PGOTHIC, MS MINCHO, MS PMINCHO, TAHOMA, SYLFAEN, CENTURY, RAGE ITALIC, SCRIPT MT BOLD

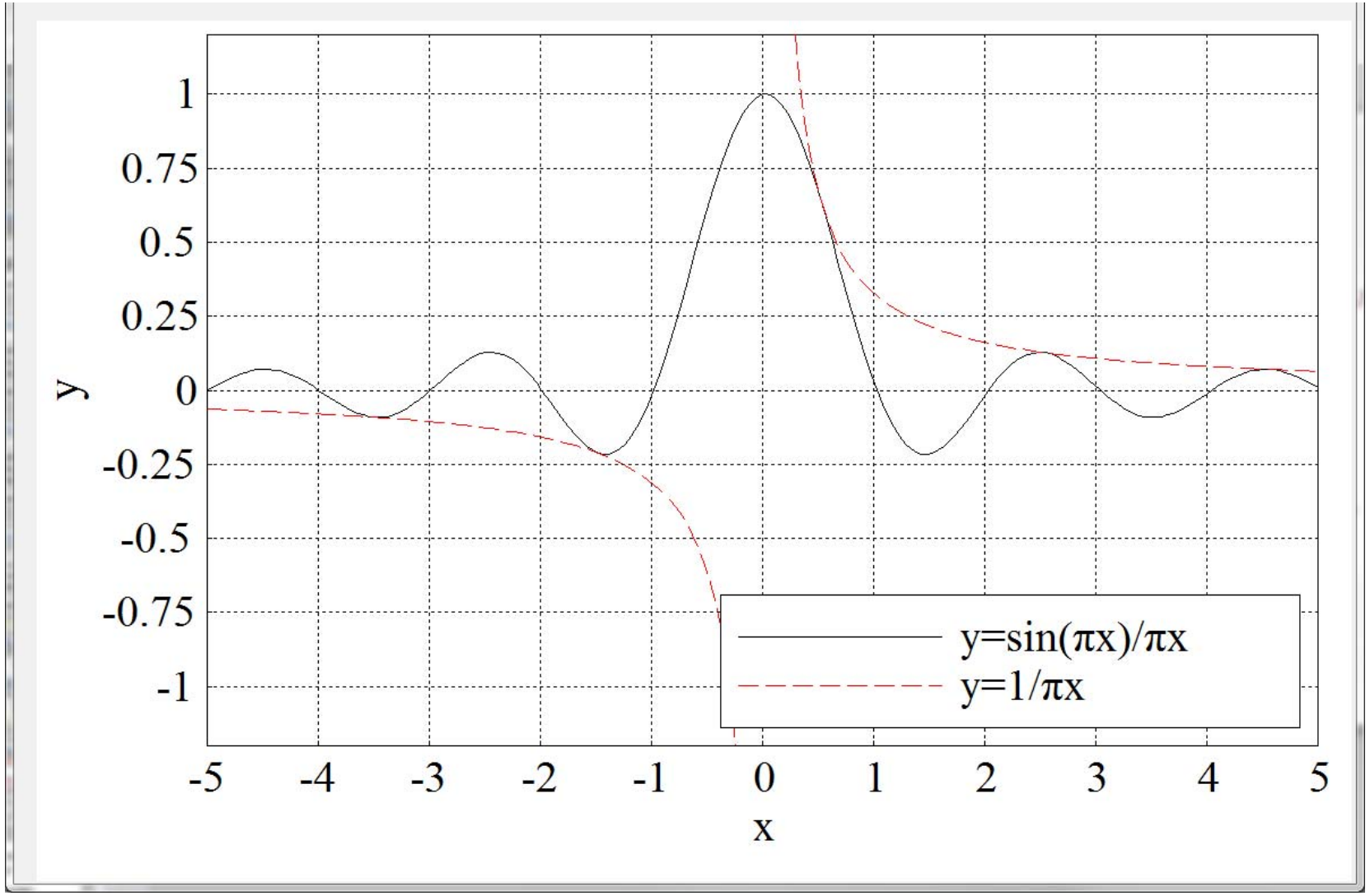
フォントサイズの指定　　LONG型変数　　Used_Font_Size

イタリック　　BYTE型変数　Italic_Font_Flug　を1にする。　0で普通体

例題

Sinc関数 $y = \sin(\pi x)\,/\pi x$ を$-5 \le x \le 5$ の範囲で表示せよ。X軸、y軸のラベルも表示すること

加えて赤色，点線で $y = 1/\pi x$ を重ね書きし、関数名を凡例で示せ



例題のプログラム

```
#define N 100    定数の設定
void main (int Number){
   int i;
   double pi=3.1415926535;
   double x[N],y[N];

   for(i=0;i<N;i++){
      x[i]=10.0/N*i-5.0;   配列変数xに、中心0,幅10,ポイント数 N の数値データを設定
      y[i]=sin(pi*x[i])/(pi*x[i]);     y軸のSINC関数をsin関数使って計算
   }                                   (math.hは既にインクルードされている)
   y[N/2]=1;   注意 sin(x) /xは計算機内では x = 0 で Inf. なので、x = 0 のとき y = 1 とする
   Set_figure(1,1,1); フィギュアウインドウが1個であってもSet_figure(1,1,1)で必ず設定
   Used_Font_Size=45; 図のフォントサイズ変更
   Axis_xcap(-5,5,"x");  x 軸の範囲、ラベルを設定
   Axis_ycap(-1.2,1.2,"y");  y 軸の範囲、ラベルを設定
   Plot1d(y,N); 配列変数 y の数値データを先頭からN個分波形として表示

   for(i=0;i<N;i++){
     y[i]=1.0/(pi*x[i]); 配列変数 y に 1/x の値を代入
   }
   Plot_pen(1,1,1);     図の線を、赤、点線、サイズ1に設定
   Plot1d(y,N);      配列変数 y の数値データを先頭からN個分波形として重ね書き表示
   Grid_on(3);   グリッド線を入れる
   Legend("y=sin(πx)/πx|y=1/πx",4);     凡例表示
}
```

注意 フォントを大きくした方が見やすいが、
大きくすると図の左マージンが取れないので
def.h のmergin_L を大きくする

演習４

４－１．２×２のフィギュアウインドウを作成し、4番目に$y = \sin(\pi t)/\pi t$ を$1 \le t \le 4$ の範囲で横軸 を t 、縦軸を y として表示せよ。

４－２．さらに、1番目に $x = \cos(\pi t)/\pi t$を $1 \le t \le 4$ の範囲で縦軸をt 、横軸を x として表示せよ。

４－３．さらに、3番に縦軸を y 、横軸を x の関数としてグリッド線とともに表示せよ。ただし、アスペクト比を1：1とする。