1.4 グラフィックス表示（入門編その１）　　ver. 0.1

さて、これまで少し退屈を感じられたかもしれないので、コンピューターを使うことの大きな役割の１つであるグラフを書くという作業をしてみよう。おそらく、GAUSS がより身近に感じられるに違いない。

といっても、GAUSS のグラフィック表示、これがなかなか手ごわい。ふつうの統計計量のグラフィック表示の操作方法とは全く異なる方法でグラフィックスを表示させるGAUSS では、このやり方自体が厳密すぎて GAUSS を使う学者や学生を遠ざけている最大の要因となっていることは否めない。この章では、最低限のグラフを書く基本と、グラフィックス表示をするための仕組みを解説したい。

次にあげる３つの命令がグラフィックスを表示させるための最小構成となる。

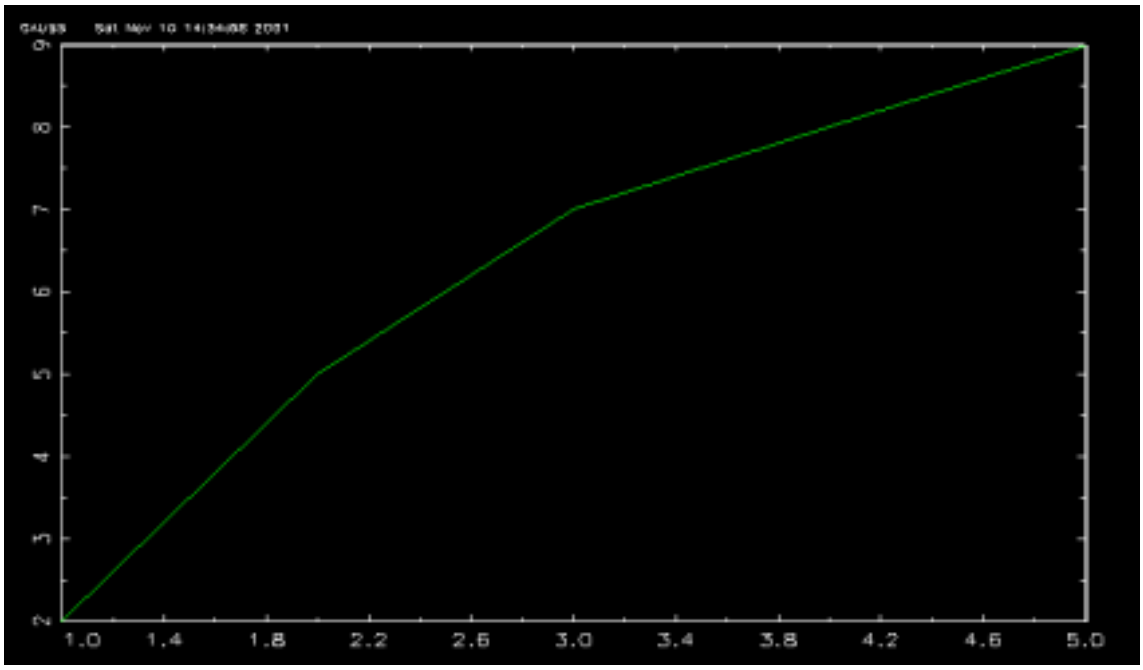
```
library pgraph;
graphset;
xy(x,y);
```

この３つをこの順序で（途中に計算などの命令がいくつか入る）ならべて、x と y の中に列ベクトルをそれぞれ代入するとグラフが描かれる。上の１行目は、pgraph というグラフを描くためのライブラリーを呼び出しアクティブにします。２行目は、そのライブラリーの中にあるグローバル変数と呼ばれる設定値をリセットします。これは、プログラムで行なわれるメモリーの初期化 new のような働きをして、もし直前のプログラムの実行でかわってしまっているのなら、そのグラフ設定変数についても規定値に戻します。最後に、ｘとｙの中に列ベクトルの値が格納されていれば、それをｘｙ空間のグラフで表します。

プログラム

```
new;  cls;
library pgraph;
graphset;
x={ 1,
    2,
    3,
    4,
    5 };
y={ 2,
    5,
    7,
    8,
    9 };
xy(x,y);
```

グラフ表示



上のように、横軸がｘ、縦軸がｙのグラフになります。グローバル変数の設定値というのは、この場合、グラフの各座標は線で結ばれ、かつ実線、緑色。そして、日付時刻が表示されるという GAUSS が勝手に描くデフォルト設定のことを言います。このままでは、都合の悪いこともあるでしょう。そこで、それぞれデフォルトになっているグローバル変数の設定値を変更することになります。ここで、グローバル変数と言っているのは、library や procedure を作ったときに、そのプログラム内部の変数は、ローカル変数、外部に引き渡されて、そのプログラムの外部から変更できるようになっている変数をグローバル変数と呼んでいます。

ポイント　ｘｙ空間の２次元のグラフを描く最低限のプログラム構成は、

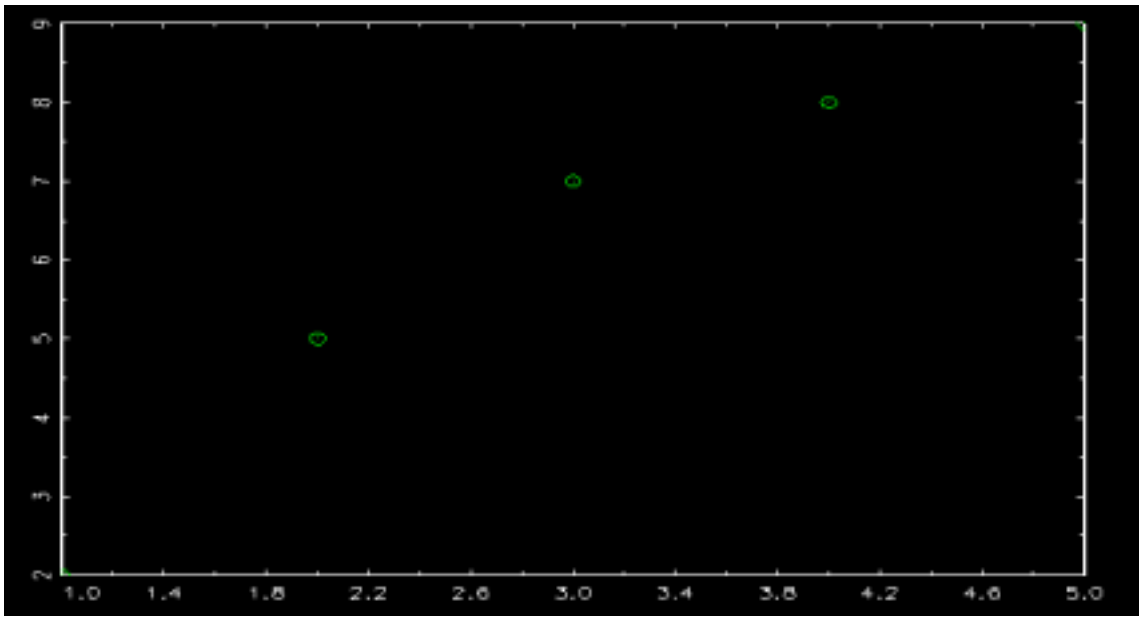
```
library pgraph;
graphset;
x=列ベクトル;　y=列ベクトル;
xy(x,y);
```

さて、このプログラムに、座標と座標を結ぶ線を消して、座標だけを表示するようにしてみましょう。あわせて、グローバル変数を変更することによって、日付時刻の表示を消してみましょう。これだけの操作で、とりあえず大抵の二次元グラフは描けるはずです。(,)の中のｘとｙはもちろん別の名前の変数でもかまいません。

プログラム

```
new;  cls;
library pgraph;
graphset;
x={ 1,
    2,
    3,
    4,
    5 };
y={ 2,
    5,
    7,
    8,
    9 };
_plctrl = -1;
_pdate ="";
xy(x,y);
```

グラフ表示



上のように、〇のマークで座標が表されて、ラインは消えました。同時に、日付時刻もなくなりました。_plctrl = -1 の設定は、pgraph という library でラインだけを引くという規定値０になっているグローバル変数を変更して、結ぶラインなしに座標のみをプロットする設定値である-1 にしたものです。Ｃ言語のように、GAUSS は構成されていて、このよう

に library などのグローバル変数を設定するときには、__のマークがついた変数にスケラー形式、文字列形式、または行列形式で数値を代入格納するようになっています。なお、何十もあるグローバル変数すべてを設定する必要はなくて、必要なものだけプログラムの中で変更します。そのほかは、デフォルト規定値のままでグラフが描かれます。_pdate=””は、いままで左上に表示されていた日付時刻を示す文字列の記号が” “の中に入っていたデフォルトを、何もなしの” “に置き換えて、表示を止めてしまったものです。ここでも pdate の前にＣ言語風に__のマークがついています。GAUSS は変数の前に__のマークがついているものは、プログラム内の普通の変数とは区別して、いま呼び出している library など（この場合は pgraph）のグローバル変数と解釈します。まだ見なれていない方には、少しとっつきにくい表示方法かもしれませんが、慣れていきましょう。

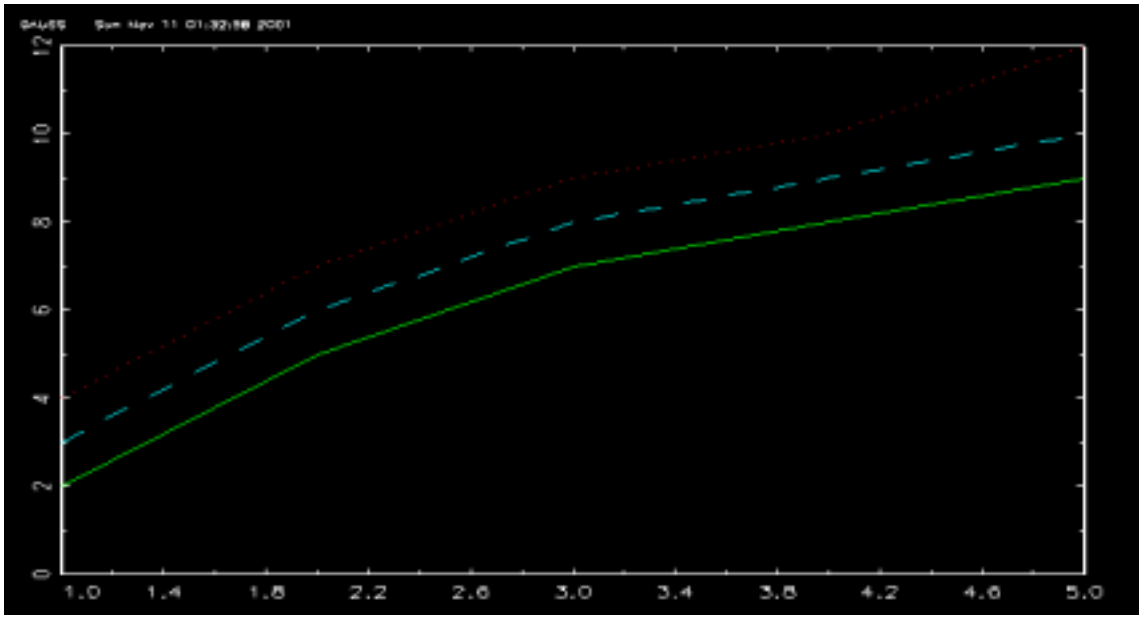
ポイント　日付時刻の表示を消すには、_pdate=””;
ポイント　ラインなしの座標プロットは、_plctrl = -1;
ポイント　グラフ描画のデフォルトを変更するには、_マークのついたグローバル変数に数値を代入して設定しなおす。

次に、もとのデフォルトのままのシンプルなグラフに戻って、今度は、ｘのところに３列ぐらいの数値をいれてやって、それを表示してやりましょう。

プログラム

```
new;  cls;
library pgraph;
graphset;
x={ 1,
    2,
    3,
    4,
    5 };
y={ 2 3 4,
    5 6 7,
    7 8 9,
    8 9 10,
    9 10 11};
xy(x,y);
```

グラフ表示



上のように、今度は、ｘの１列ごとの値が、それぞれのシリーズとして GAUSS 内部で自動的に色づけされて描かれます。X の最初の列は緑色の実線、2 番目の列は水色の破線、3 番目の列は赤色の点線などというふうにデフォルトでそう決まっているのです。もちろん、色や線の種類をかえることもできますし、先程やったように、ラインを消して座標だけにすることも可能です。graphset;というふうにプログラムのなかで pgraph という library のグローバル変数をリセットしているので、今回はデフォルトどおり日付時刻が左上に表示されています。

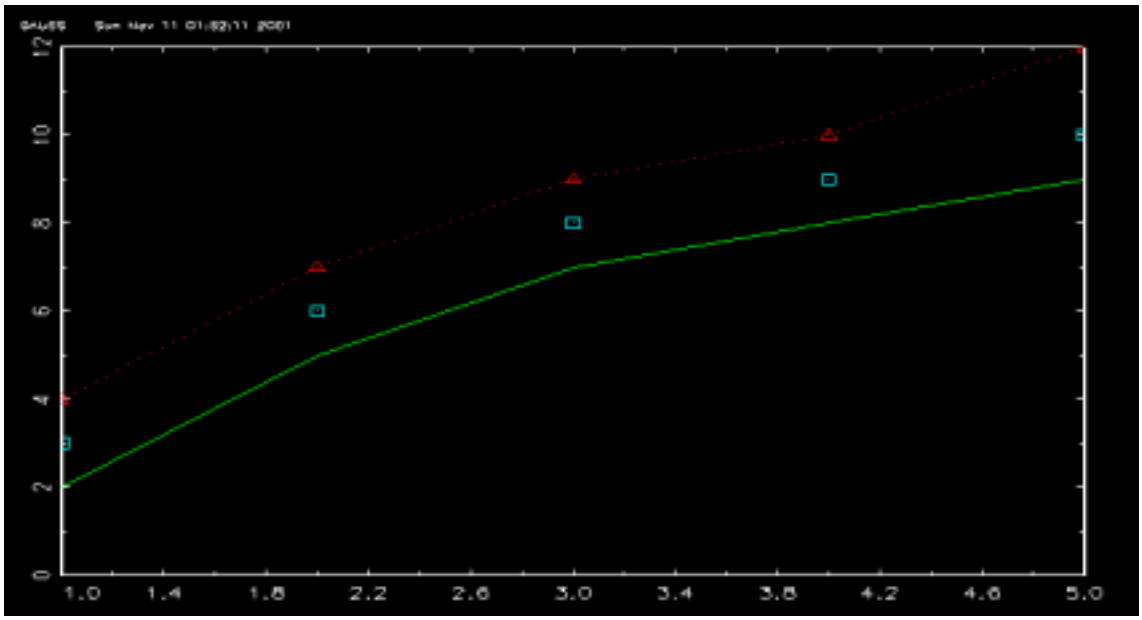
　ここで、先程のように_plctrl の規定値を変更して少し操作してみましょう。つぎのようなプログラムを実行してみてください。変更点は、_plctl の行だけです。

プログラム

```
new;  cls;
library pgraph;
graphset;
x={ 1,
    2,
    3,
    4,
    5 };
y={ 2 3 4,
    5 6 7,
```

```
    7 8 9,
    8 9 10,
    9 10 11};
_plctrl={ 0, -1, 1 };
xy(x,y);
```

グラフ表示



いままで_plctrl にスケラーを代入していたものが、今度は列行列を入れています。（GAUSS ではカンマまでが１行目、その後が２行目というふうに続いていくことを思い出してください。）ｘの最初の列には０の規定値のまま、2 番目の列には-1 を、そして第３列目には１を代入するという意味です。GAUSS では、このように x に相当する部分が何列かのデータのシリーズの集まりになると、グローバル変数の設定もカンマをつけた列ベクトルの数値の代入と似た形をとります。もちろん、グローバル変数によっては、行ベクトルの設定の仕方の場合もありますし、またこれら２つの組合せの結果、縦横何行何列の行列値をグローバル変数に代入することもないはけではありません。ちなみに、_plctrl の規定値は、ラインのみの０で、-1 はプロットのみ、１はラインつきプロットとなっています。

ポイント　ｘに複数のシリーズのデータを入れる場合、ラインとプロットの規定値変更は

_plctrl={ 整数,整数,整数….};

整数が　０：　ラインのみ（デフォルト値）

１：　ラインと（１点ずつの）プロット

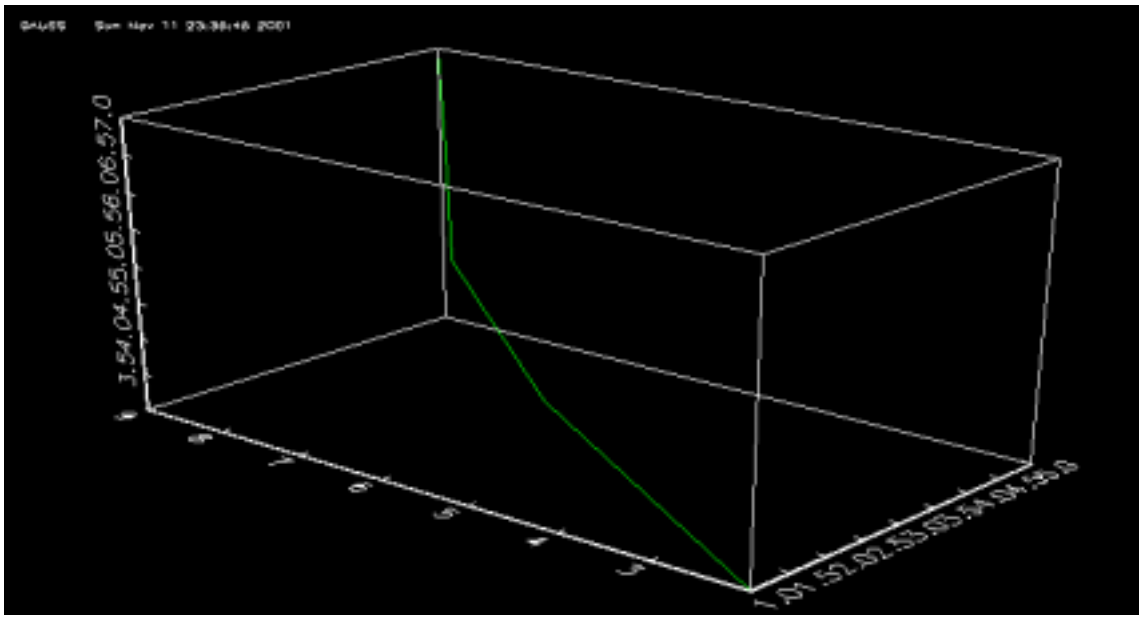
－１：　（１点ずつの）プロットのみ

これまでの例は、すべて二次元のグラフでしたが、三次元のグラフも簡単にできます。

プログラム

```
new;  cls;
library pgraph;
graphset;
x={ 1,
    2,
    3,
    4,
    5 };
y={ 2,
    5,
    7,
    8,
    9 };
z={ 3,
    4,
    5,
    6,
    7};
 xyz(x,y,z);
```

グラフ表示

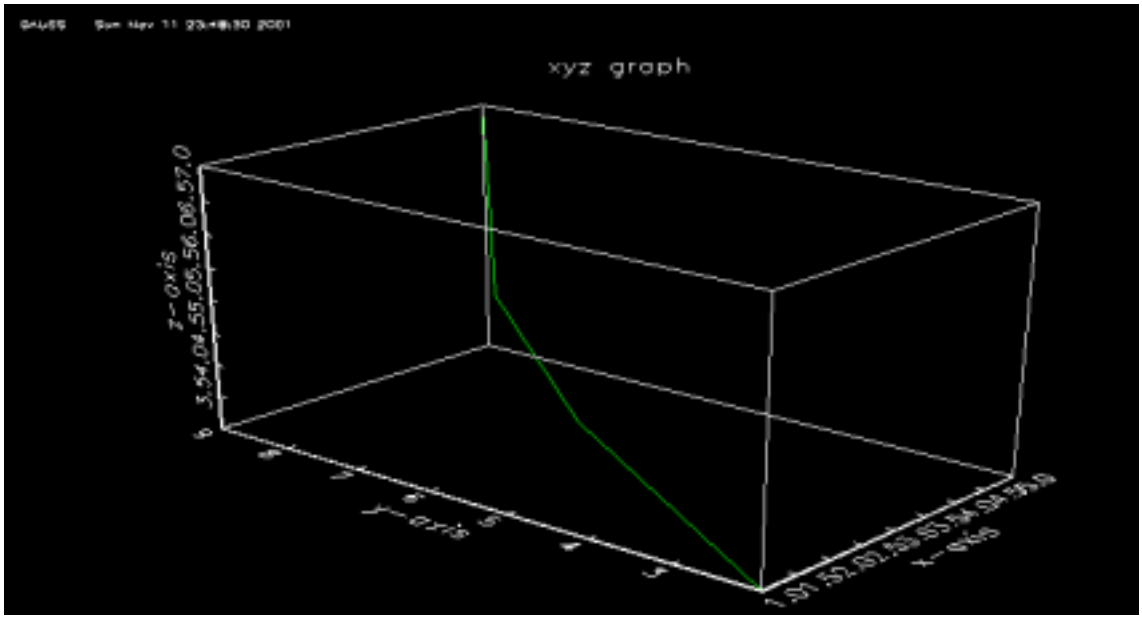


すべて、これまでと同じで、xy(x,y)のところを xyz(z,y,z)に置き換えてやるだけです。（もちろん、ｚの列データも与えてやらなくてはいけませんが。）さて、これではどれがｘ軸ｙ軸ｚ軸かさっぱりわからないので、グラフのタイトルとともに、ラベルを挿入してみます。

次のようなプログラムの追加を xyz(x,y,z);の行の前あたりに置いて実行してください。
追加プログラム

```
xlabel(“x-axis”);
ylabel(“y-axis”);
zlabel(“z-axis”);
title(“xyz graph”);
```

グラフ表示



上のように、細かい注文は除いて、ラベル付きのほぼ思い通りのグラフを描けるようになったと思います。xlabel,ylabel,xlabel のそれぞれの(　)の中には引用符“　”付きの任意の文字列(string)が入ります。お好きなように名前をつけてください。ただし、引用符は‘を２つつけても認識しませんので注意が必要です。引用符のキーは数字２の上にあって、シフトキーを押したままで２を押すと出てくる“のマークのことです。ラベルを xlabel と ylabel だけにすれば、２次元のグラフも同様にラベルがつけられますのでためしてみてください。もちろん”　“の中は英語です。

ポイント　ｘ軸のラベルつけは、xlabel(“文字列”);
ｙ軸のラベルつけは、ylabel(“文字列”);
ｚ軸のラベルつけは、zlabel(“文字列”);
グラフのタイトルつけは、title(“文字列”);