

## FAQ(1) グラフィックス関連

### 【質問 1】

簡単なプログラムに `x y (x,y);` グラフなどのグラフィックスのコマンドを加えたのですが、グラフが全く表示されません。

### 【解答】

GAUSSでは、計量の他のパッケージプログラムと違って、`xy(x,y);`や`hist(x,9);`などすることによって、そのままではGAUSS内部から `x y` グラフや 9 本のバーからなるヒストグラムを作成することはできません。なぜならば、描画関係のコマンドは、GAUSS本体とは別の、`pgraph`というライブラリに納められているからです。グラフを描きたければ、必ず `pgraph`という名前のライブラリを呼び出して、GAUSS上でそこに入っているコマンド類が使用可能にしておく必要があります。

### 【対処法】

`library pgraph;` という 1 行を加えて、グラフ関連のコマンドを使う前に、`pgraph`というライブラリを呼び出す。さらに、これに付随して、`graphset;`というグラフィックス関連のグローバル変数(各種設定パラメータ)を初期化しておくことも癖をつけてください。これは、ちょうどGAUSS本体の`new;`に相当するものだと考えてください。

```
library pgraph;  
graphset;
```

上の 2 行をプログラム全体の冒頭あたり、または、描画を必要とする `procedure` の内部に置きます。なお、Light 版にもこの `pgraph` のライブラリは付属しています。正式 Full 版と機能的に全く同じものです。同じように、別売最適化ライブラリである `cml` や `maxlik` も、

```
library cml;                library maxlik;  
cmlset;                     maxset;
```

などとして呼び出します。つまり、`pgraph` もこれらのライブラリと同じ仲間です。

#### 【質問 2】

GAUSS にグラフを描かせたのですが、サンプルのようなたくさんのグラフ例の画面は出てくるのですが、自分がプログラムしたグラフを描くことができません。

#### 【解答】

これは、pgraph が呼び出すべきファイル群と現在のディレクトリー構造にズレが生じてしまって、pgraph のみがうまく機能していないプロテクトされた状態にあることが考えられます。考えられる原因としては、もともとあった場所から GAUSS のフォルダーを移動させてしまったり、正式ではない方法で GAUSS をインストールしたり、または一時的にリムーバブルなどの外部記憶装置にインストールしてあるものをハードディスクにコピーしてきたりした場合に起こります。GAUSS のフォルダーの階層や位置はさわらないようにしましょう。

#### 【対処方法】

- 1 ) 正式な方法でもう一度 GAUSS 全体をインストールし直す。
- 2 ) 外部記憶装置に GAUSS があるのであれば、コピーイメージではなくて、そのものを起動させる。または、外部記憶装置そのものを 1 ドライブとしてマウントする。
- 3 ) 不具合のバージョンの GAUSS を開いたままで、他のバージョンの GAUSS ( Light 版も可 ) を開いて、2 つが動作している状態でグラフを描く作業を試してみる。

上のうちいずれかを状況に応じて試みてください。

【質問 3】

グラフを描くと GAUSS のマークと作成日時が表示されてしまって邪魔です。どうすればよいでしょうか？

【解答】

これは簡単に取り除けます。グローバル変数 `_pdate` に空の値を与えれば、次回 `graphset;` でグローバル変数を初期化するまではグラフにマークと日時は入らなくなります。

【対処方法】

以下のようなグローバル変数の設定を、`graphset;`の後、描画するコマンド、例えば `xy(x,y)` の前のどこかに

```
_pdate="";
```

の 1 行を置きます。ここで、変数の下線の部分はグローバル変数であることを表す方法です。次の `p` は `pgraph` または正式名 `PQG` ( `Publication Quality Graphics` ) を表す一連のグローバル変数の最初の文字です。すなわち、ここでは `PQG` のグローバル変数 `_pdate` という特定の予め定められた変数に `"` という引用符 2 つの間に何も入っていない空の値を与えることをしています。これにより、`GAUSS` のマークと日時の表示は止まります。

#### 【質問 4】

凡例を出すグローバル変数や線の色を変えるグローバル変数をいろいろと設定したのですが、一部だけ自分の意図したように変わってくれません。

#### 【解答】

グローバル変数は、予め定められたデフォルト値が入っている変数です。通常の設定をそのまま用いてグラフを描くには、設定する必要がないものです。つまり、プログラム上には書かれてはいないけれども、pgraph の内部でそれぞれのグローバル変数にデフォルトの数値が入っているのです。そのなかの 1 つまたは数個のデフォルト設定を外部から数値を与えてやって、デフォルトのグラフ描画とは違う設定にするものが、グローバル変数の設定です。もし、`_pdate=""`;とすべきところを`__pdate=""`;などと全角の下線や 2 つ分の下線のついた変数を用いてしまったり、`_pdata=""`;などというように変数の名前自体が違ってある場合には、エラーが全くでない状態で、間違った名前の変数に何かの数値、または、この場合、空を表すものを代入していることになります。綴りが間違えていても、エラーが全く出ませんので注意が必要です。

#### 【対処方法】

グローバル変数を用いた場合には、その綴りが正しいのかを入念に一字一句確認をする作業を怠らないこと。下線が 1 つなのか 2 つなのか、それにちゃんと半角で入力されているのかを確認し、必要とあれば、一度消してもう一度その部分だけを注意深く入力し直してみる。また、`graphset`;の命令がない場合も、挙動が不安定になったり、前の設定のままになっていたりする場合があるので、必ずつけるようにする。ただし、グローバル変数を複数のグラフごとに変更するのでなければ、グローバル変数の設定の後に `graphset`;の文を入れるとその設定はデフォルトに戻ってしまうことは当然のことである。

【質問 5】

GAUSS のような背景が真っ黒で、グラフが緑であるのもよいのですが、Matlab や R のように背景を白にしたいのですが。

【解答】

GAUSS のグラフは、グラフィックモニターとラインモニタが別であった 80 年代に生まれた名残で、そのころのグラフィックモニターが背景が黒いところに緑色のグラフを描くものの伝統をそのまま現在にも残しているものです。黒地に緑は GAUSS の特徴でもあります。なるべくならば、そのままの設定で使いましょう。見てくれる人も一目で GAUSS で描画したのだなとわかるはずです。それでも、背景を白にする必要がある場合には、そうする簡単な方法があります。

【対処方法】

一度なにがしかのグラフを描きます。グラフが描かれている状態では TFK Viewer が開かれている状態になっているはずです。その TFK Viewer の中から View Options のところの

Use Color in TKF File	White
Don't convert colors	Convert all colors to black

の 2 箇所を変更して OK を押すと、それ以降あらたに設定を変更するまでは白黒でグラフが描かれ続けます。

#### 【質問 6】

グラフを保存するために `_ptek="sample1";` としたのですが、どこにもグラフィックファイルが保存されていません。

#### 【解答】

GAUSS では、通常、特にディレクトリや階層を指定しないでファイル名だけでファイルを保存しようとする、そのファイルは Working Directory に保存されます。そのデフォルトはそのバージョンの GAUSS のフォルダーの中になります。この保存ファイル名をしていする作業も、`_ptek=` が下線と p から始まっていることからわかるように、グローバル変数設定にあたります。ただし、描画のメインの命令である `xy` や `hist` などの後にこのファイル保存をするというグローバル変数設定を置いたとしても、プログラムの流れの順番から言って、もうすでに描画が行なわれた後なので、`_ptek` の指定は無効になります。

#### 【対処方法】

GAUSS のメニューの中にある File から Change Working Directory を選択して適当な場所に変更すると、次のファイルの保存は選択されたディレクトリのその階層に変更できます。そうでなければ、

```
_ptek="sample1.tkf";          _ptek="d:sample1.tkf";
```

などというふうにご自分のお使いのディレクトリをファイル名の冒頭につけてください。また、ファイル名の拡張子を `.tkf` としておけば Windows からクリックするだけでグラフを見ることができます。拡張子をつけなくても保存はできていますが、そのままではグラフを見ることはできません。その場合には、Windows 上からファイル名に `.tkf` の拡張子をファイル名に追加するとグラフのマークが出てきてクリックするだけで認識できるようになります。グラフィックファイルの保存の `_ptek=` をはじめとするグローバル変数は描画コマンドの前に置くことにも注意してください。後ろに置いても無視されます。

【質問 7】

複数の何枚かのグラフを描いてうまく表示はされるのですが、順に表示はされるのですが次々にグラフが表示されてしまって、最後に描いたグラフしか画面上には残りません。

【解答】

GAUSS のデフォルト設定では、グラフは一枚だけのウインドウに順に描いていく設定なっています。多数のグラフィックウインドウを開いて、そのそれぞれにグラフを描くには多ウインドウの設定をしなくてはなりません。また、GAUSS では命令を一時的に止めて、何らかのキーを Command Window 上で押すことによって再び GAUSS の命令を実行していく方法もあります。

【対処方法】

多ウインドウに GAUSS の設定を変更するには、描画コマンドよりも前のどこかに

```
pqgwin many; または pqgwin auto;
```

とします。またもとの設定に戻すには、pqgwin one; または pqgwin manual;とします。また、GAUSS の命令そのものをグラフごとに一時的に止めるには、

```
waitc;
```

という 1 行を 2 つの描画コマンドの間のどこかに挿入します。これにより、GAUSS のプログラムの実行が、プログラムのその時点で一時停止します。Command Window 上で (このウインドウを一番上にして) 何かのキーを押すと、再び GAUSS のプログラムの実行が、waitc;の命令の後の行から再開されます。複数のグラフの描画の場合には、これら 2 つの方法のいずれか、または、その組み合わせによって解決されます。

【質問 8】

複数のグラフを別々の画面に描いたのですが、スケールがグラフごとに違ってしまいます。スケールを同じにする方法ないのでしょうか？

【解答】

各座標の最小値と最大値を正確に指定することが GAUSS では可能です。また、各座標の大小の目盛を指定することまでも可能です。

【対処方法】

やり方は 2 通りあります。一番単純な方法は、その名も `scale` という関数にそれぞれの座標の最小最大を与える方法です。例えば、2 つの  $x$   $y$  グラフで  $x$  座標を最小 0 最大 10 に固定して、 $y$  座標を最小 0 最大 100 に固定するとすると、下のように

```
xscale={0,10}; yscale={0,100 };  
scale(xscale,yscale);  
xy(x,y1);  
xy(x,y2);
```

`scale` というグラフィックス専用の関数の第 1 要素に  $x$  座標の最小最大を、第 2 要素に  $y$  座標の最小最大をそれぞれ  $2 \times 1$  のベクトルで与えて代入すれば、その後に描かれるグラフはすべてこの最小最大の範囲内だけが描かれて、スケールはきっちり一致します。もう 1 つの方法は、大小の目盛まで指定できる方法です。

```
xtics(0,10,1,10);  
ytics(0,100,10,5);
```

というふうに、(最小値,最大値,ステップの値, 1 ステップ内の小目盛数)の順に  $x$  および  $y$  座標に数値を設定すれば `scale` の命令と同じことプラス大小目盛の微調整ができます。こちらは、各要素ベクトルではなくて数値で指定します。



【質問 9】

ギリシャ文字や特殊数学文字を入力しようと思ってマニュアルのとおりやってみたのですが、英語のアルファベットのままです。

【解答】

特殊フォントをロードする命令が足りないのであろうと思われます。

【対処方法】

例えば、simple と simgrma の 2 つのをこの順番で GAUSS 上にロードするには、

```
fonts("simplex simgrma");
```

とします。この場合、simplex が 201 番、simgrma が 202 番になります。また、すべてのフォントを下のような順番で

```
fonts("simplex,microb,complex,simgrma");
```

ロードした場合、simgrma は 4 つ目なので 204 番になります。フォントロード命令 fonts がなければ、GAUSS はギリシャ文字や特殊数学文字が入った simgrma を認識することはできません。また、201 番とか制御コードとして使う部分の数字は、上のようにロードする順番に依存するのあって、それぞれのフォント番号が固定されているわけではないので注意が必要です。

【質問 10】

通常の  $x$   $y$  グラフを用いて折れ線グラフをいくつか重ねて表示したいのですが、どのように重ねればよいのでしょうか？

【解答】

グラフを重ねるというイメージではなくて、GAUSS では  $x$  軸は共通で  $y$  軸方向の値を「複数列」指定すると考えてください。GAUSS はグラフでも数値を「列ごとに」扱う言語なのです。

【対処方法】

例えば、 $x$  軸側は共通で 0 から 0.01 ステップで 100 だけのシーケンスをとるものとします。 $y$  軸側には、そのシーケンスを  $x$  とおいた時の  $\sin(x)$  および  $\cos(x)$  および  $\tan(x)$  の 3 つの系列のグラフを書きたいとします。その場合には、library pgraph; graphset; の後に

```
x=seqa(0,0.01,100);  
y=sin(x)~cos(x)~tan(x);  
xy(x,y);
```

とすることによって、変数  $y$  には 3 つの縦ベクトルを水平方向に ~ でマージしたもの、すなわち、 $100 \times 3$  の行列が作成され、これと  $x$  軸側の  $100 \times 1$  の縦ベクトルとともに  $x$   $y$  グラフの関数の中に代入すれば、3 つの系列のグラフが自動的に色分けされて同時に表示されます。この場合も、GAUSS の根本的な考え方である「列ごとに」操作する仕様が一貫してあらわれています。

【質問 11】

x y グラフのゼロのところ、あるいは任意の数値のところに基準線をいれたいのですが、どうすればよいのでしょうか？

【解答】

これは、y の系列にもう 1 列ゼロ列ベクトル、あるいは任意の数値の列ベクトルをマージすることによって実現できます。

【対処方法】

例えば、1 つ前の質問の例題に y 座標が 1 である基準線を入れるのには、手続きの後に、

```
x=seqa(0,0.01,100);  
y=sin(x)~cos(x)~tan(x)~ones(100,1);  
xy(x,y);
```

という具合に、y の他の系列の個数（行の数）と同じ個数からなる 1 のベクトルを水平方向にマージしてもう 1 つの系列を加えます。もし、0.5 やゼロの基準線がほしければ、

```
~0.5*ones(100,1);  
~zeros(100,1);
```

などの列ベクトルを y 軸側の変数にマージして加えられます。1 の列ベクトルの何倍にするかということで、ゼロや 1 以外の基準線は作成できます。GAUSS では、シークエンスは縦（列）ベクトルを作成するものなので注意が必要です。y 軸方向の系列の個数（行数）は基準線も含めてすべて一致していないとグラフを描くことはできません。

【質問 12】

ヒストグラムを描いた時に、0 が中心の分布であるのに、どうも中心にこないのですが？

【解答】

ヒストグラムの第 2 パラメータである区切りの数がよくないと思われます。

【対処方法】

例えば、標準正規乱数を作成して、その列ベクトルを 10 の区切りのヒストグラムにすると  
して、グラフィックスの手続きの後に、

```
x=rndn(10000,1);  
hist(x,10);
```

とすると 0 のところにバーがきません。例えば、この場合、

```
hist(x,19);
```

というふうに、もう少し区切りの数を増やして、さらにその数を奇数にすればうまくいきます。データの数に応じて区切りの数（バーの数）を自分で調節してください。GAUSS では自動調節にはなっていません。

**【質問 13】**

Contour グラフの場合、プログラムのにはおかしいところはないのですが、エラーがでます。どうすればよいのでしょうか？

**【解答】**

Contour グラフのみ、その内部計算アルゴリズムから、 $x$  軸と  $y$  軸に与えられる格子の数には制限があります。

**【対処方法】**

$x$  軸および  $y$  軸の要素の数を奇数に変更してください。数少ない GAUSS のスペシャルルールです。なお、surface グラフや  $x y z$  グラフにはそのような制限は全くありません。

【質問 14】

Surface グラフと Contour グラフの x 軸 y 軸の設定と x y z グラフの x 軸 y 軸の設定の関係がよくわかりません。同じようなグラフを描くのにはどうすればよいですか？

【解答】

1.6 章のグラフの表示(3)で説明しましたように、GAUSS には列ベクトルと行ベクトルを足し合わせると行列ができるようになっています。そのことを利用して、x 軸の行ベクトルにはゼロ列ベクトルの足し合わせることによって、座標を表す同じ行がたくさんできます。同様にして、y 軸の列ベクトルにゼロ行ベクトルを足し合わせることによって、座標を表す同じ列がたくさんできます。これらを用いることができます。

【対処方法】

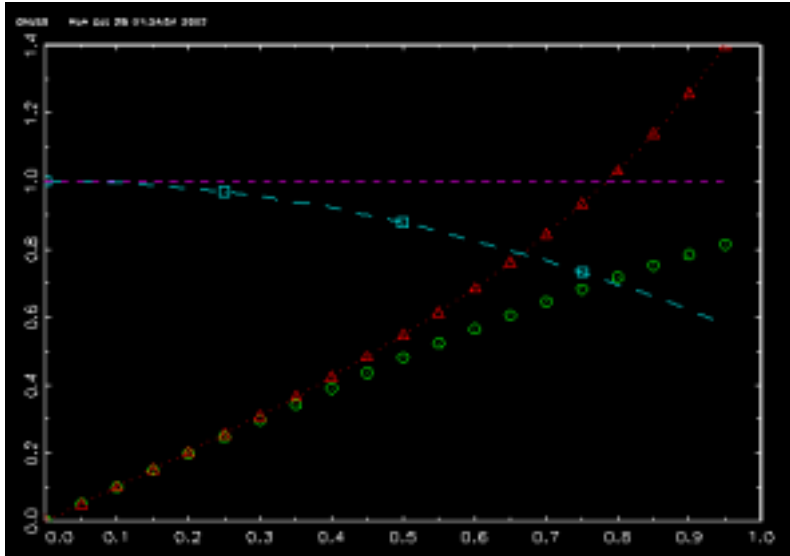
次のようなプログラムによって、どういうものなのかの原理がわかると思います。

```
new; cls;  
library pgraph;  
graphset;  
pqgwin many;  
x=seqa(-2,0.04,100)'; y=seqa(-2,0.04,100); z=cos(x)+sin(y);  
surface(x,y,z);  
x=x+zeros(100,1); y=y+zeros(1,100);  
xyz(x,y,z);
```

上のように、Surface グラフは x 軸には行ベクトル、y 軸には列ベクトルを必要とします。一方、x y z グラフは、(x,y)で 1 つの座標面になるように設定しなければならないので、相反するゼロベクトルを足し合わせます。これにより、全く同じ Surface グラフと x y z グラフを描くことができます。もう一度、1.6 章の冒頭部分を見て確認してください。

### 【質問 15】

GAUSS の x y グラフでは、軌跡の線を結ぶことがデフォルトのようですが、ドットだけを描いたり、ドットと線の両方を描く設定がもう 1 つよくわかりません。わかりやすく解説してください。



### 【解答】

グローバル変数 `_plctrl` を列ベクトルで設定します。イメージ的には、デフォルトの 0 はドットがないという意味で 0。正の整数の場合には、線付きでその数ごとのドットを描く。負の場合には、線なしでその数ごとのドットを描くということで、正は線付き、負は線なしというイメージです。

### 【対処方法】

上のようなグラフを描くのに、

```
new; cls;  
library pgraph;  
graphset;  
x=sega(0,0.05,20);  
y=sin(x)~cos(x)~tan(x)~ones(20,1);  
_plctrl={-1,5,1,0};  
xy(x,y);
```

というふうに、y 軸方向には基準線を含めて 4 つの系列がありますから、 $4 \times 1$  のベクトルで `_plctrl` を設定します。通常の列ベクトルの設定の仕方と同じように、`{ }` 括弧で 1 行ごとにカンマをつけます。-1 は最初の系列は線なし 1 つずつドット、5 は線付き 5 つずつドット、1 は線付き 1 つずつドット、最後の 0 はドットなしという意味です。

【質問 16】

通常のドットの散布図に最小 2 乗推定線を入りたいのですが、どうすればできるのでしょうか？

【解答】

GAUSS では残念ながら自動では最小 2 乗推定線を入れることはできません。その代わりに最小 2 乗推定値のベクトルを求めて、それを通常どおり y 軸方向に付け加えて表示させます。一方、実質データは\_plctrl を 1 に設定して、線なしで表示させます。

【対処方法】

例えば、簡単なプログラムで上のことをやらせてみるのには、

```
new; cls;
library pgraph;
graphset;
y={1,0,4,3,2,6,5,7,10,7,10,10,14}; x={0,1,2,3,4,5,6,7,8,9,10,11,12}; x1=ones(13,1)~x;
yhat=x1*(inv(x1'x1)*x1'y);
y=y~yhat;
_plctrl={-1,0};
xy(x,y);
```

上では x は 0 から 12 までの数列で、y には、データ y そのものと yhat を水平方向にマージしたものが入っています。これを x y グラフで、最初の系列は線なし、2 番目の系列はデフォルトの 0 で線だけで表示するように、グローバル変数\_plctrl を設定しています。x には 1 のベクトル付きの x1 を入れてしまってはいけません。x 軸側は列ベクトルです。



**【質問 17】**

バージョン 3.x のシリーズには playw というグラフィックユーティリティーが付属しているのですが、何ですか？

**【解答】**

現在の GAUSS は VWR というユーティリティーによってグラフィックスは管理されています。その昔は、playw でグラフィックを見たり印刷したりコンバートしたりをしていましたが、現在は GAUSS のデフォルトのままです。

**【対処方法】**

古いバージョンの playw には Postscript 形式にコンバートする時にエラーになることが報告されています。現在の GAUSS を使っている方は、わざわざ playw を使う必要はありません。具体的な使い方や、VWR との関連性を詳しく見たい方は、何かグラフを描かせて、TKF File Viewer の Help のところの View readme file をご覧ください。

【質問 18】

グラフを Word の文章の挿入するにはどうしたらよいですか？また、この文章のように pdf ファイルに挿入するのにはどうしたらよいのですか？

【解答】

一番早い方法は、グラフを描いて、TKF Viewer の Convert のところから Windows Bitmap で保存して、それを GAUSS の外部から開いて、それを Copy して、Word の文章上に Paste すればよいだけです。Word に Acrobat が組み込まれていれば、Word 上の Acrobat ボタンを押して、図入りの文章を pdf に変換保存するだけです。

【対処方法】

ただし、上の方法は簡易的な方法であって、図はかなり粗いものになります。高品質なグラフを利用するには、マニュアルでやる場合には、何かのグラフを描いて TKF Viewer の Convert のところから今度は EPS 方式で保存します。これを TeX 対応のソフトウェアまたは TeX そのもので呼び出して文章に組み込めば高品位な画像が得られるはずですが、なお、Acrobat で高品位な画像を得るには、デフォルトの解像度よりももっと高い解像度に変更する必要があります。その場合、容量が大きくなります。そのため、本書では低品位のグラフを便宜上表示しています。TKF Viewer 上では、実際には、もう少しきれいに見えます。なお、1.8 章にはマニュアルではなくてプログラムとしてのファイルコンバートについて説明していますので、参照してください。

**【質問 19】**

グラフの枠の大きさが、いじっていたら変わってしまいました。今までの大きさにするにはどうしたらよいでしょう？また、新しいバージョンの GAUSS にしてグラフを描いたところグラフが大きくなりすぎました。どうしたらよいでしょうか？

**【解答】**

これは単純な問題です。TKF Viewer の右下隅をつまんでグラフのデフォルトの大きさを大小に自由に変更できます。次回からは、その枠の大きさが記憶されていて、その大きさでグラフは描かれます。

**【対処方法】**

Word の文章に張りつける目的でグラフを描いているのであれば、どの大きさが一番最適なのかを細かく思考錯誤で見つけましょう。一度決まったら、右下隅をつまんでグラフの大きさを変更しないことです。

【質問 20】

グラフィックファイルを保存したり、ファイルをコンバートした後に、指定したディレクトリの特定の階層に保存できないのですが。

【解答】

階層を区切る記号はバックスラッシュのアジアフォントの見え方である¥¥を用います。ただし、文字列として引用符” “の中でバックスラッシュを用いる場合には、いかなる場合にもダブルにします。これが GAUSS でのルールです。

【対処方法】

x y グラフをある特定の場所、例えば、d:ドライブの中の graph という階層がもうすでに存在して、その階層の中に sample.tkf というグラフィックファイルを保存したいのならば

```
_ptek="d:¥¥graph¥¥sample.tkf";  
xy(x,y);
```

というふうにします。前述のように、\_ptek のグローバル変数設定は、描画命令の前で行ないます。また、上の¥¥と見えている部分は半角のバックスラッシュが2つあるものです。このように、引用符の中に文字列として扱う場合には、どのような用途においても、階層の区切りのバックスラッシュはダブルにします。ファイルコンバートやコンパイル時の指定などの時にもこのことはあてはまります。