gnuplot FAQ

この文書は、**gnuplot** バージョン 5 についての記述です。
FAQ バージョン: 2019 4 月

目 次

0 メタな質問 **4**

0.1 このドキュメントはどこで手に入れられますか 4

0.2 このドキュメントに関するコメントはどこに送れば良いですか 4

1 一般的な質問 **4**

1.1 **gnuplot** とは何ですか 4

1.2 それはどこから来て、そしてなぜ **gnuplot** と呼ばれているのですか 4

1.3 **gnuplot** は何を提供してくれますか 5

1.4 **gnuplot** はスクリプトに適していますか 5

1.5 **gnuplot** は私のコンピュータで動かすことができますか 5

1.6 ライセンスについて 5

1.7 **gnuplot** は FSF や GNU プロジェクトと何か関係があるのですか 6

1.8 更なる情報はどこで手に入りますか 6

2 インストール **6**

2.1 **gnuplot** の最新バージョンは 6

2.2 **gnuplot** はどこで手に入れられますか 6

2.3 **gnuplot** の最新の開発版はどこで手に入れられますか 6

2.4 **gnuplot** は私のシステムではどうコンパイルすれば良いですか 6

2.5 ドキュメントはどこですか、それはどうすれば手に入れられますか 7

2.6 動作するサンプル 7

2.7 私の **gnuplot** にどのオプションがコンパイルされているかを知るには 7

3 動作 **7**

3.1 ヘルプはどうすれば手に入れられますか 7

3.2 どうすればグラフをプリントアウトできますか 8

3.3 どうすればグラフを私のワープロソフトに取り込めますか . . . 9
3.4 どうすれば **gnuplot** のグラフを編集や後処理できますか . . . 9
3.5 どうすれば現在の設定を保存/再利用できますか . . . 10
3.6 一つのファイルに gnuplot のコマンドとデータの両方を書けますか . . . 10
3.7 どうすれば描画前にデータにフィルタをかけられますか . . . 10
3.8 私のプログラムで **gnuplot** のルーチンを利用できますか . . . 10

4 グラフの見た目のカスタマイズ **11**
4.1 色や線や点記号のデフォルト属性を調べたり変更するには . . . 11
4.2 陰線処理、隠面処理 . . . 11
4.3 グラフの境界をページの特定の位置に丁度合わせるには . . . 11
4.4 複数のグラフをひとつのページ内で隣同士に配置するには . . . 11
4.5 縦と横の軸の目盛りを 1:1 にできますか . . . 11
4.6 カラーとモノクロの両方の出力で機能するようなパレットは . . . 12
4.7 どうしたらデータ点をスキップできますか . . . 12
4.8 どうしたら n 点毎にデータを扱えますか . . . 12
4.9 どうしたら鉛直線を書けますか . . . 12

5 良く質問されるグラフの形式 **12**
5.1 アニメーション . . . 12
5.2 陰関数のグラフ . . . 13
5.3 2 つの関数の間の領域を塗りつぶすには . . . 13
5.4 3 次元データの 2 次元射影を描画するには . . . 13
5.5 pm3d 色地図/曲面上に dots/points の散布図を重ねるには . . . 13
5.6 ラベルつきの等高線を描画するには . . . 13
5.7 **gnuplot** は棒グラフ/ヒストグラム/箱型グラフが書けますか . . . 14
5.8 **gnuplot** は円グラフが書けますか、四半期グラフはどうですか . . . 14
5.9 **gnuplot** は一つのグラフで複数の y 軸を使えますか . . . 14
5.10 小 3 角面で構成される物体を描画するには . . . 14
5.11 どうしたら 2 つの関数を重ならないように書けますか . . . 15
5.12 どうしたら splot で格子でない曲線を書けますか . . . 15
5.13 どうしたら x-y 平面上で他の関数に制限されている f(x,y) のグラフを書けますか . . . 15

6 文字列の整形と特別な記号 **15**
6.1 "enhanced text" モードによる文字列のマークアップ . . . 15
6.2 文字列のマークアップをオンに/オフにするには . . . 16
6.3 特殊文字に関するすべて問題の回答は UTF-8 にすることですか . . . 16

6.4 Planck 定数 (h に横線) が必要なときはどうしたらいいですか . . . 16
6.5 太陽質量記号 (丸に点) が必要なときはどうしたらいいですか . . . 16
6.6 ギリシャ文字や他の特別な記号を使うには . . . 17
6.7 どうしたらアクセント付きの文字を出力できますか . . . 17
6.8 異なるサイズのテキストを一つのグラフで出力できますか . . . 17

7 その他 **17**
7.1 **gnuplot** には画像形式 XXX をサポートするドライバはありますか . . . 17
7.2 バグを見つけたのですがどうしたら良いですか . . . 17
7.3 どう追加や拡張がなされていますか、それはどう手に入れられますか . . . 18
7.4 "heavy-duty" データを **gnuplot** で使えますか、または **gnuplot** に代わるものはないですか 18
7.5 gnuplot が持たない特殊関数が必要な場合は . . . 18
7.6 私の対話型ターミナルでホットキーを使うには . . . 19
7.7 **gnuplot** の次期バージョンの開発について知りたいのですがどうしたらいいですか . . . 19

8 共通の問題 **19**
8.1 助けて! 私のフォントが使えない . . . 19
8.2 qt 出力形式のセッションでの最初のグラフが失敗したり変なレイアウトになります . . . 20
8.3 pm3d によるデータファイルの splot で何も表示されません . . . 20
8.4 なぜ **gnuplot** はとても小さい数字のデータを無視するのですか . . . 20
8.5 multiplot のグラフを *replot* すると一かけらしか出ません . . . 21
8.6 書いた式 (1/3 など) が意味のない結果を出力してしまいます . . . 21
8.7 私の出力が不完全です . . . 21
8.8 パイプやスクリプトから gnuplot を呼び出すとスクリーンにグラフがでません . . . 21

9 帰属 **22**

0 メタな質問

0.1 このドキュメントはどこで手に入れられますか

このドキュメントの最新版は、Web 上の http://www.gnuplot.info/faq/ にあります。

0.2 このドキュメントに関するコメントはどこに送れば良いですか

コメント、提案等は開発者向けメーリングリスト mailto://gnuplot-beta@lists.sourceforge.net に送ってください。

1 一般的な質問

1.1 gnuplot とは何ですか

gnuplot はコマンド駆動型の描画プログラムです。関数の描画、およびデータ点の描画を、2 次元、3 次元の両方で、そして多くの異なるスタイル、多くの異なる出力形式で行うことができます。また、**gnuplot** は自動的にグラフを生成するスクリプト言語としても使用することができます。これは主に科学的なデータを画面表示するものとして設計されています。**gnuplot** の著作権は保護されていますが、自由に配布でき、代金を払う必要はありません。ソースコードのダウンロードも歓迎します。

1.2 それはどこから来て、そしてなぜ gnuplot と呼ばれているのですか

gnuplot の作者は Thomas Williams, Colin Kelley, Russell Lang, Dave Kotz, John Campbell, Gershon Elber, Alexander Woo、そして多くの協力者です。

以下は Thomas Williams によります:

> 私は微分方程式の講義を、そして Colin は電磁気学の講義を取っていて、2 人ともそれらに関する数学が視覚化できればなあと考えていました。私達はある EE (Engineering Electronics) VLSI 研のシステム管理者として働いていて、そして、グラフィック端末とコーディングを行なう時間がありました。その投稿は我々の期待以上に良く受け入れられ、そして、不完全でしたが、ファイルデータに関するいくつかのサポートに我々を駆り立てました。
>
> GNUplot という紹介はどんなものも正しくはありません。このプログラムの本当の名前は "gnuplot" です。ずいぶんと "Gnuplot" と書かれているのを見るでしょうが、それは我々の大半が、固有名詞やタイトルであっても、文章を小文字で開始することに嫌悪を持っているからです。gnuplot は GNU プロジェクトや FSF とは、極わずかの意味を除けば少しも関係ありません。我々のソフトウェアは完全に独立にデザインされたもので、そして "gnuplot" という名前は実は妥協によるものです。私は "llamaplot" と呼びたかったし、Colin は "nplot" と呼びたかったのです。そして、我々は "newplot" がいいだろうということで合意しましたが、しかしそのときその名前を持つ、絶対的におぞましい Pascal のプログラムを計算機科学部でたまに使っていることを知りました。それで私は "gnuplot" がいいごろ合わせになるとして、後で Colin もそれに同意したのです。

1.3 gnuplot は何を提供してくれますか

- 2 次元での関数やデータの、点、折れ線、誤差表示、塗り潰した図形、ラベル文字列、ベクトルなどの沢山の異なる要素を組み合わせた描画
- 極座標軸、対数軸、より一般の非線形軸の割り当てや媒介変数座標
- 温度分布、ビースウォームグラフ、バイオリングラフ、ヒストグラムなどのデータ表現
- 3 次元でのデータの、点、折れ線、曲面の、たくさんのスタイル (等高線描画、網かけ) による描画
- 整数、実数、複素数での代数演算
- Marquardt-Levenberg 最適法を用いたデータ駆動型の曲線回帰
- 多くのオペレーティングシステム、多くのグラフィックファイル形式出力、そして多くの出力装置のサポート
- 広範囲にわたるオンラインヘルプ
- ラベル、タイトル、軸、データ点への TEX ライクな書式による見出し付け
- 対話型入力形式のエディット機能とヒストリ (履歴) 機能のサポート

1.4 gnuplot はスクリプトに適していますか

ええ。gnuplot は対話型実行中に追加コマンドを含むファイルを読み込むことができますし、既に存在するファイルや標準入力からのコマンド列をパイプを使ってバッチモードでそれを処理することもできます。gnuplot は、Octave のような高水準な数学パッケージの背後で実行するグラフィックドライバとして使われていますし、cgi スクリプトでラップすることで容易に Web 起動型の描画生成ツールとして使うこともできます。gnuplot は、なじみのある命令 *if else continue break while for* を用いた文脈駆動型、データ駆動型の処理制御、および繰り返しをサポートしています。

1.5 gnuplot は私のコンピュータで動かすことができますか

gnuplot は、MS Windows, linux, unix, OSX などの多くのプラットホーム上で、広く利用できます。現在のソースコードは、VMS, Ultrix, OS/2, MS-DOS などの古いシステムのサポートも保持していますが、16 ビットプラットホームはもうサポートしていません。

gnuplot のソースは、それなりに標準的 (ANSI/ISO C, POSIX 等) な環境から多少外れたものの上でもコンパイルできるでしょう。

1.6 ライセンスについて

gnuplot はボランティアの集団によって作られたものです。彼らは **gnuplot** 、およびその使用の許可、不許可に関する法的な文書を作成できません。また何の保証もありません。自分自身の責任で使用してください。

gnuplot は、無償であると言う意味でフリーウェアです。gnuplot をあなたが望むように使用し、改変することができますが、改変したものを配布することについてはある種の制限がつきます。*Copyright* というファイルの modification (改変) と redistribution (再配布) の項目を読んでそれに従ってください。

1.7 gnuplot は FSF や GNU プロジェクトと何か関係があるのですか

gnuplot は FSF が作ったのでも FSF がメンテナンスしているのでもありません。一時期 FSF から配布されていたこともありますが、しかし現在はこれは正しくありません。**gnuplot** は、GNU General Public License (GPL) に全く保護されていません。

1.8 更なる情報はどこで手に入りますか

gnuplot Web ページ http://www.gnuplot.info を見てください。

英語以外の他の言語での文書や入門書もあります。最新のリストについては、http://gnuplot.sourceforge.net/help.html の "Localized learning pages about gnuplot" の節をご覧ください。

2 インストール

2.1 gnuplot の最新バージョンは

gnuplot の最新の安定版は、2017 年 9 月に最初にリリースされた 5.2 です。更新バージョン (パッチレベル 1, 2, ...) は、だいたい 6 ヶ月毎にリリースされます。**gnuplot** の開発版は、現在は 5.3 となっています。

2.2 gnuplot はどこで手に入れられますか

初めに最も適切な場所は http://www.gnuplot.info です。ここから、例えば SourceForge 上の開発プロジェクトサイト http://sourceforge.net/projects/gnuplot などの、色々なサイトの場所を見つけられるでしょう。

ソース配布物 ("gnuplot-5.2.6.tar.gz" または同様の名前) は公式の配布サイト http://sourceforge.net/projects/gnuplot にあります。

2.3 gnuplot の最新の開発版はどこで手に入れられますか

gnuplot の開発版は、git リポジトリに保持されていて、以下に示すようにクローンを手元で作成し、ソースを調べたり、実行プログラムをソースから作成したりできます。

```
git clone https://git.code.sf.net/p/gnuplot/gnuplot-main gnuplot
```

開発版に対する質問は必ず mailto://gnuplot-beta@lists.sourceforge.net に送ってください。

2.4 gnuplot は私のシステムではどうコンパイルすれば良いですか

リリースノートと *README*, *INSTALL* のファイルを読んでください。C と C++ コンパイラが必要で、様々なサポートライブラリがあらかじめインストールされている必要もありますが、それはどのような設定オプションを選択するかによりますし、作成する実行ファイルにどの出力形式をサポートさせたいかにもよります。

- linux 上でリリース版からコンパイルするには、*./configure* (一人の一般ユーザ用にインストールする場合は、*./configure --prefix=$HOME/usr*) とし、次に *make* として、最後に *make install* です。*configure* スクリプトの出力には十分注意を払ってください (そう、それはとても長いのだけれど)。そこには、どのサポートライブラリが必要なのか、どの追加オプションが望ましいのか、といったことに対するヒントが現れています。一般には、各サポートライブラリの「開発版」パッケージをインストールしておく必要があるでしょう。
- linux 上で開発版ソースからコンパイルするには、*./configure* を実行させる前に、別のスクリプト *./prepare* を実行する必要があります。
- Windows では、makefile は *config/mingw*, *config/msvc*, *config/watcom*, *config/cygwin* 内にあります。makefile の先頭に書かれているオプションを修正し、その makefile が置かれているのと同じディレクトリ内で適当な *make* ツールを実行してください。追加説明も、makefile に書かれています。
- 他のプラットフォームでは、適切な makefile (例えば OS/2 なら *makefile.os2*) を *config/* ディレクトリから *src/* にコピーして、makefile のヘッダにあるオプションを必要なら更新して、*src* にディレクトリを移動して、*make* を実行してください。

2.5 ドキュメントはどこですか、それはどうすれば手に入れられますか

ソース配布物に完全なドキュメントが PDF ファイル形式で含まれています。個々のセクションは、gnuplot の作業中に *help* キーワード を打つことで見ることができます。ドキュメントの他の形式も、*docs* サブディレクトリ内でソースからコンパイルできます。

http://gnuplot.sourceforge.net/documentation.html に、英語版と日本語版のコピーが置いてあります。

2.6 動作するサンプル

ソース配布物には動作するサンプルのディレクトリが含まれています。これらサンプル、およびその結果の描画グラフは http://gnuplot.sourceforge.net/demo/ でも見ることができます。

2.7 私の **gnuplot** にどのオプションがコンパイルされているかを知るには

コンパイル済みの **gnuplot** が与えられている場合、*show* コマンドで、それがコンパイルされたときに使用されたコンフィグレーションとビルドオプションの一覧を表示できます。あなたの gnuplot に組み込まれている出力形式 ("terminal" とも言います) は、*set terminal* で表示できます。

```
gnuplot> show version long
gnuplot> set terminal
```

3 動作

3.1 ヘルプはどうすれば手に入れられますか

最初のプロンプトで *help* コマンドを実行してください。その後キーワードをたどっていってください。*help plot* や *help set* から始めるのがいいでしょう。

- もし、持っているならマニュアルを読んでください
- *demo* サブディレクトリのデモを実行するか、ネット上にあるそのコピーを見てください。ヒントを与えてくれるでしょう。http://gnuplot.info/demo
- システム管理者、または **gnuplot** を設定した同僚に尋ねてください。
- gnuplot に関する質問を議論する Usenet グループ news://comp.graphics.apps.gnuplot が古くからありますが、そこの住民はほとんど前世紀の人々です。
- よりブラウザ指向のヘルプフォーラムが StackOverflow にあります。http://stackoverflow.com/questions/tagged/gnuplot
- それらにすべて失敗した場合、質問をニュースグループ news://comp.graphics.apps.gnuplot に投書するか、またはメーリングリスト mailto://gnuplot-info@lists.sourceforge.net にメールを送ってください。そのメーリングリストが圧倒的な量の spam を別に受けてしまうのを避けるため、投稿の前にそのメーリングリストに参加 (subscribe) しなければいけないことに注意してください。参加の方法は、http://lists.sourceforge.net/lists/listinfo/gnuplot-info にあります。

質問をする際は、あなたが使用している **gnuplot** のバージョンとオペレーティングシステムについても言及する必要があります。あなたのやり方で期待したようにはうまくできなかったグラフについて質問したいときは、その問題を持つグラフを生成するような最小の gnuplot コマンド列を示すようにしてください。

3.2 どうすればグラフをプリントアウトできますか

plot コマンドが生成するものの出力形式は、その前の *set terminal* によって決定します。対話型でない出力に対しては、ファイル名を指定するためにこれと *set output* コマンドを組にして実行します。

以下の例の作業は、まず sin(x) のグラフをスクリーン上に描画し、同じグラフを PDF 出力ファイルとして再描画します。注意: PDF 出力は、スクリーン上に表示されるグラフと完全には同じにならない可能性があります。

```
gnuplot> plot sin(x)
gnuplot> set terminal pdf
Terminal type is now 'pdfcairo'
Options are ' transparent enhanced fontscale 0.5 size 5.00in, 3.00in '
gnuplot> set output "sin.pdf"
gnuplot> replot
gnuplot> unset output        # 出力ファイルを閉じる （でないと開いたまま）
gnuplot> unset terminal      # デフォルト対話型出力形式に戻す
gnuplot>
```

出発点が、デフォルト対話型出力形式でない場合は、同じことを *push* と *pop* で達成できます。

```
gnuplot> set terminal push    # 現在の出力形式を保存 （デフォルでなくてもよい）
gnuplot> set terminal pdf
gnuplot> set out 'a.pdf'
gnuplot> replot
gnuplot> unset out
gnuplot> set term pop         # 保存した出力形式を復帰
```

対話型出力形式の中にはそのツールバー上にプリンタアイコンを提供するものもあります (*win, wxt, qt* 等)。それは、他の出力形式を使用するのではなく、システムに用意されている一般的なシステムツールを使い、現在のグラフを印刷するか、またはファイルに保存します。すなわち、印刷メニューで "save to png" を選択して得られるファイルは、*set term png; replot;* によって得られるファイルとは違うものになります。一般に、この方法で保存したグラフは、コマンドラインから出力形式を変更して作成したものよりも、よりスクリーン画像に近いものを生成するでしょう。

3.3 どうすればグラフを私のワープロソフトに取り込めますか

基本的に、グラフをそのワープロソフトが理解できる形式のファイルに保存すれば、そのワープロソフトで読み込むことができるでしょう。後で正しいサイズに変換もできるという点で、ベクトル形式のフォーマット (PostScript, emf, svg, pdf, TEX, LATEX) の方がいいでしょう。

使用できるファイル形式の一覧を見るには *set term* としてください。

多くのワープロソフトは Encapsulated PostScript (*.eps) のグラフを取り込めます。**gnuplot** では、*set terminal postscript eps* または *set terminal epscairo* を使うことで eps 出力を選択できます。**gnuplot** は、出力する EPS ファイルにプレビュー用のビットマップ画像を埋め込みませんので、ワープロソフトに対応させるために、そのファイルをワープロソフトに取り込む前に、外部ツールを使って自分自身でプレビュー画像を追加しないといけない場合もあるかもしれません。

LibreOffice や Microsoft Office 等のアプリケーションには、EMF 形式のベクトル画像を処理できるものがあります。そのような画像ファイルは emf 出力形式でも作れますし、windows 出力形式の描画ウィンドウのツールバーから 'Save as EMF...' を選択することでも作れます。

LibreOffice は SVG 形式も、AutoCAD の dxf 形式も読みこむことができます。

gnuplot を使って TEX や LATEX 文書に取り込むためのグラフを作る方法は何通りもあります。取り込む *.tex ファイルの断片を直接生成する出力形式もありますし、\includegraphics コマンドを使って取り込むための *.eps, *.pdf, *.png 出力を生成するものもあります。epslatex と cairolatex 出力形式は、画像ファイル (*.eps か *.pdf) とそれを参照する *.tex 文書ファイルの両方を生成します。tikz 出力形式は、テキストとグラフを生成し、その出力を pdflatex で処理すれば PDF ファイルが作成できます。

大抵のワープロソフトはビットマップ画像 (png, pbm 等) を取り込むことができますが、この方法の短所は、そのグラフが **gnuplot** によって生成されたときのサイズによってその解像度が制限されてしまうことです。一般に、文書が最終的にプリントアウトされる解像度よりもその解像度はかなり低くなります。

3.4 どうすれば **gnuplot** のグラフを編集や後処理できますか

これはあなたが使う出力形式に依存します。

- **svg** 出力形式 (scalable vector graphics) の出力は **Inkscape** (http://www.inkscape.org), **Skencil** (http://www.skencil.org), **Dia** (http://projects.gnome.org/dia/), のような svg エディタで更なる編集も行なえますし、**OpenOffice.org** (OO.o Draw プリミティブへの変換によって) の中に取り込むこともできます。
- PostScript や PDF 出力は、Adobe Illustrator や Acrobat のようなツールで直接編集できますし、**pstoedit** パッケージで他の色々な編集可能なベクトル形式に変換することも可能です。pstoedit は http://www.pstoedit.net にあります。
- DXF 出力形式は AutoCAD の書式で、他のいくつかのアプリケーションでも編集できます。

- ビットマップ形式の画像 (png, jpeg, pbm 等) は ImageMagick や Gimp の様なツールで編集することができます。一般に、ベクトルグラフィック形式の後処理を行なうにはベクトルグラフィックプログラム (ドロー系ツール) を、ピクセル画像にはピクセル画像系のプログラム (ペイント系ツール) を使う必要があります。

3.5 どうすれば現在の設定を保存/再利用できますか

それには *save "filename"* と *load "filename"* のコマンドを使ってください。

3.6 一つのファイルに **gnuplot** のコマンドとデータの両方を書けますか

gnuplot のバージョン 5 では、「ヒアドキュメント」形式での名前付きデータブロックをサポートしています:

```
gnuplot> $DATABLOCK << EOD
  cats 4 2
  dogs 1 4
EOD
gnuplot> plot $DATABLOCK using 2:3:1 with labels
```

名前付きデータブロックは一度定義すれば、それを何度でも使うことができます。

データは、plot コマンド上で疑似ファイル指定 `plot "-"` を行ってインライン形式で与えることもできます。この場合は、データは一度しか使えません。

```
gnuplot> plot "-"
1 1
2 4
3 9
e
```

3.7 どうすれば描画前にデータにフィルタをかけられますか

あなたの使用する OS が popen() 関数をサポートしているなら、他のプログラムやシステムユーティリティを *plot* コマンドの一部分として使用して入力データをフィルタリングすることができます。

```
gnuplot> plot "< sort +2 file.in"  # 2 列目のデータで事前にソート
```

この仕組みは、Unix 由来のコマンドラインツール *awk*, *sort*, *grep* などと組み合わせて使用するととても強力です。

3.8 私のプログラムで **gnuplot** のルーチンを利用できますか

パイプをサポートしているシステムでは、他のプログラムからパイプ経由で **gnuplot** にコマンドを渡すことができます。Octave (http://www.octave.org) のように、gnuplot をグラフィックエンジンとして使用している多くのアプリケーションがこの方法を使用しています。これは、form ベースの Web ページから **gnuplot** を起動する cgi スクリプトでも利用できます。

4 グラフの見た目のカスタマイズ

4.1 色や線や点記号のデフォルト属性を調べたり変更するには

複数の要素に対する *plot* や *splot* コマンドを実行した場合、**gnuplot** は、デフォルトでは色と線種の集合を巡回的に使用します。これは、その plot コマンド に特定の色や線種属性を指定することで変更できますし、デフォルトの列を変更することもできます。以下のコマンドは、それぞれ追加パラメータをたくさん指定できます。

test は、現在の出力形式に対する有効な色、線、点の属性などを表示します。

set color と *set monochrome* は、定義済みの列を選択します。

set linetype は、存在する線種の属性を変更するか、新しいものを追加します。

set palette は、温度地図のような pm3d モードで使用されるカラーパレットを変更します。

set pointsize は、追加の倍率としてすべての点のサイズをスケールします。

4.2 陰線処理、隠面処理

2 つの関連するコマンドがあります。*set hidden3d* は、3 次元描画スタイル *splot ... with lines* で描画する曲面に影響を与えます。それは、他の 3 次元描画スタイルで生成され、それらの曲面で隠される線分をクリッピングします。しかし、*pm3d* モードで生成されるグラフは処理しません。*with pm3d*, *with zerror*, *with boxes* のスタイル、および *set pm3d* が有効な最中に描かれた様々な描画要素についても同様です。これらのグラフに対する隠面処理は、見る側からの距離の順番でそれらを描画することによって代用的に達成されていますが、その順番は、*set pm3d depthorder* で制御できます。

4.3 グラフの境界をページの特定の位置に丁度合わせるには

上、下、左、右の境界位置を、ページ内の位置として比率で指定してください:

```
set lmargin at screen 0.05
set bmargin at screen 0.05
set rmargin at screen 0.95
set tmargin at screen 0.95
```

4.4 複数のグラフをひとつのページ内で隣同士に配置するには

あなたが必要とするコマンドは *set multiplot* です。gnuplot は指定した数のグラフを規則的な升目状に配置します (*set mulplot layout <rows>, <columns> ...*)。または、それらを *set origin* と *set size* を使ってひとつずつ位置を決めることもできます。

4.5 縦と横の軸の目盛りを 1:1 にできますか

set size square か *set view equal xy* としてみてください。

4.6 カラーとモノクロの両方の出力で機能するようなパレットは

set palette cubehelix を試してください。

4.7 どうしたらデータ点をスキップできますか

以下のようにデータ値として *?* を指定すればできます。

```
1 2
2 3
3 ?
4 5
```

set missing も参照してください。データファイルのコメント文字を指定するには、*set datafile commentschars* を参照してください。

4.8 どうしたら n 点毎にデータを扱えますか

これは、例えば *plot 'a.dat' every 2* のように、*plot* コマンドに色々なオプションを指定することでできます。すべての点を通る線を引き、しかし点の記号は n 点毎に書きたい場合は、
plot 'a.dat' with linespoints pointinterval n としてみてください。

4.9 どうしたら鉛直線を書けますか

状況によりますが、主な方法は以下の通りです:

- *set arrow .... nohead* この場合、矢の正確な開始点と終了点の位置を計算しておく必要があります。
- データ点を作って (インラインでの指定も可) それを描画

5 良く質問されるグラフの形式

5.1 アニメーション

gnuplot 出力形式で直接アニメーションファイルを出力できるのは以下のもの (gif) だけです:

```
set terminal gif animate {delay <time>} {loop <N>} {optimize}
```

デモの中の http://gnuplot.sourceforge.net/demo/animate.html も参照してください。

5.2 陰関数のグラフ

gnuplot では陰関数のグラフや曲線は直接は描画できません。しかし、次善の策はあります。

```
gnuplot> # 例: 次の行を適当な定義に置き換えてください。
gnuplot> f(x,y) = y - x**2 / tan(y)
gnuplot> set contour base
gnuplot> set cntrparam levels discrete 0.0
gnuplot> unset surface
gnuplot> set table $TEMP
gnuplot> splot f(x,y)
gnuplot> unset table
gnuplot> plot $TEMP w l
```

その仕掛けは、曲面 z=f(x,y) の 1 本の等高線 z=0 を描き、その等高線のデータを一時ファイルやデータブロックにセーブすることにあります。

5.3 2 つの関数の間の領域を塗りつぶすには

2 つの関数 f(x) と g(x) の間を塗りつぶすグラフは、疑似ファイル '+' と *filledcurves* を使えば簡単にできます。

```
f(x)=cos(x); g(x)=sin(x)
set xrange [0:pi]
plot '+' using 1:(f($1)):(g($1)) with filledcurves
```

注意してもらいたいのですが、上の例のスクリプトは 2 つの曲線の間の領域を塗りつぶすものであって、どちらが他方の上であるかを考慮はしません。もし、g(x)<f(x) か、または f(x)<g(x) を満たす領域のみを塗り潰したいなら、*filledcurves* の後ろに *above* か *below* のキーワードを追加してください。

5.4 3 次元データの 2 次元射影を描画するには

コマンド *set view map* は、視角とスケールを調整し、それに続く *splot* による 3 次元グラフを、*plot* による 2 次元グラフとほぼ同じレイアウトで表示してくれます。

5.5 pm3d 色地図/曲面上に dots/points の散布図を重ねるには

pm3d スタイルの *explicit* オプションを使ってください:

```
gnuplot> set pm3d explicit
gnuplot> splot x with pm3d, x*y with points
```

5.6 ラベルつきの等高線を描画するには

個々の等高線にラベルをつけたグラフは、**gnuplot** バージョン 4 ではある種の裏技と余分な処理手順が必要でしたが、バージョン 5 ではその手順はとても簡単です。等高線を、1 回目は "with lines" で、2 回目は "with labels" で 2 回描画するだけです。目立つラベルを作成したい場合は、以下のようにするといいでしょう。

```
set style textbox opaque noborder
set contours
splot 'DATA' with lines, 'DATA' with labels boxed
```

5.7 gnuplot は棒グラフ/ヒストグラム/箱型グラフが書けますか

gnuplot は、あらかじめ表形式にされたデータを表示するための、さまざまな横並び型 (clustered) と積み上げ型 (stacked) のヒストグラムスタイルをサポートしています。また、生のデータを場所毎に累積するためのいくつかのオプションも提供していて、それを切り替えて棒グラフとして表示させることも可能です。マニュアルの *bins* と *smooth frequency* の部分を参照してください。

5.8 gnuplot は円グラフが書けますか、四半期グラフはどうですか

円グラフは **gnuplot** ではちょっと面倒ですが、以下を見てください。http://gnuplot.sourceforge.net/demo/circles.html または、以下を参照してください。http://gnuplot-tricks.blogspot.com/2009/08/pie-charts-entirely-in-gnuplot.html

デモの中には単純なガントチャートの例があります。

5.9 gnuplot は一つのグラフで複数の y 軸を使えますか

はい。2 次元グラフでは別々に x 軸を下 (x1) と上 (x2) に持つことができますし、別々に y 軸を左 (y1) と右 (y2) に持つことができます。バージョン 5 では、*with parallelaxes* という描画モードで y 軸をさらに追加することもできます。

5.10 小 3 角面で構成される物体を描画するには

3 次元物体の面に色をつけたものをファイルに記述するには、以下のような形式にすれば可能です。

```
# 三角形 1
x0 y0 z0 <c0>
x1 y1 z1 <c1>

x2 y2 z2 <c2>
x2 y2 z2 <c2>


# 三角形 2
x y z
...
```

1 行の空行と 2 行の空行に注意してください。また、三角形は実際には、4 つの頂点を記述し、そのうちの 2 つを同一とすることにも注意してください。<*cN*> はオプションの色指定です。
これを以下のようにして描画します (どちらかの splot で):

```
set pm3d
set style data pm3d
set pm3d depthorder
splot 'facets.dat'
splot 'facets_with_color.dat' using 1:2:3:4
```

gnuplot は 3 次元モデリングソフトではありませんので、depthorder による描画では、面同士の貫通は処理されません。本当の 3 次元描画のためには、多分レイトレーシングソフトを使う方がいいでしょう。

5.11 どうしたら 2 つの関数を重ならないように書けますか

描画する各関数の直前に描かせたい範囲を置いてください。例えば、実験データと、領域の別の部分をカバーする 2 つの異なる関数モデル f1, f2 を描画するには:

```
gnuplot> set autoscale x  # x の範囲をデータから取得
gnuplot> plot 'data', [*:0] f1(x), [0:*] f2(x)
```

5.12 どうしたら splot で格子でない曲線を書けますか

splot への入力データが、各行が同じデータ点数 (通常は空行で分離されている) を持つように揃っていれば、splot はデフォルトではそのデータを曲面を記述するものとして扱います。そうでなく個々の曲線を描画したければ、*unset surface*, *set surface explicit*, *plot ... nosurface* などを組み合わせてみてください。

5.13 どうしたら x-y 平面上で他の関数に制限されている f(x,y) のグラフを書けますか

以下は一つの方法です:

```
gnuplot> f(x,y) = x**2 + y **2
gnuplot> x(u) = 3*u
gnuplot> yu(x) = x**2
gnuplot> yl(x) = -x**2
gnuplot> set parametric
gnuplot> set cont
gnuplot> splot [0:1] [0:1] u,yl(x(u))+(yu(x(u)) - yl(x(u)))*v,\
       >                   f(x(u), (yu(x(u)) - yl(x(u)))*v)
```

6 文字列の整形と特別な記号

6.1 "enhanced text" モードによる文字列のマークアップ

gnuplot バージョン 5 では拡張文字列モード (enhanced text) がデフォルトで、それは特殊文字を文字列内に埋めこむことによって文字列のマークアップを示します。

拡張文字処理制御記号			
制御記号	例	結果	説明
^	a^x	a^x	上付き文字
_	a_x	a_x	下付き文字
@	a@^b_{cd}	a^b_{cd}	空ボックス (幅がない)
&	d&{space}b	d␣␣␣␣␣b	指定した長さのスペースを挿入
~	~a{.8-}	$\tilde{a}$	'a' の上に '-' を、現在のフォントサイズの .8 倍持ち上げた位置に重ね書き
	{/Times abc}	abc	Times フォント、今のサイズで abc を出力
	{/Times*2 abc}	abc	Times フォント、今の倍のサイズで abc
	{/Times:Italic abc}	*abc*	Times フォント、イタリック体で abc
	{/Arial:Bold=20 abc}	**abc**	Arial フォント、太字、サイズ 20 で abc

6.2 文字列のマークアップをオンに/オフにするには

この処理モードから、特定のテキスト文字列を除外するには、キーワード *noenhanced* を使用してください。例えば、ファイル名を下付き文字と解釈させないためには:

```
set title 'Compare file_1.dat and file_2.dat' noenhanced
```

6.3 特殊文字に関するすべて問題の回答は **UTF-8** にすることですか

ええ。

不運にも、特に PostScript 出力の生成時など、使用するにはかなり厄介な環境がいくつかあります。UTF-8 のコンピュータ環境で作業をしている場合、gnuplot でそれを使用する場合には、多分何か特別なことをする必要はないでしょう。しかしそうでない場合は、出力する際には gnuplot に UTF-8 を使用することを以下のように伝える必要があります: *set encoding utf8*

UTF-8 文字列をあなたのキーボードで入力できない場合は、gnuplot の外でそれを解決するか、またはその 8 進のエスケープ文字列を 1 バイト毎入力してください。あなたのキーボードで UTF-8 を生成できるがどのキーストロークで特定の文字列を生成できるかを知らない場合は、多分適当な文字選択アプリケーションがあなたのデスクトップについているでしょう (例えば KDE なら *kcharselect*、GNOME なら *Character Map*)。

6.4 **Planck** 定数 (**h** に横線) が必要なときはどうしたらいいですか

最も自然な方法は、UTF-8 フォントを使い、文字 ℏ (Unicode コードポイント #x210F) を直接入力することです。

PostScript: PostScript では utf8 の処理は面倒なので、拡張文字列処理 (enhanced text モード) のマークアップと、多分特別な Symbol フォントを使った擬似的なことをやる必要があります:

`@{/=56 -} {/=24 h}` または `{/=8 @{/Symbol=24 -} _{/=14 h}}` 後者は 24pt (ポイント) の "-" (/Symbol フォントの中の長いハイフン) をスペースなしで使います。14pt の "h" は 8pt のスペース ("_" の前にあるスペース) だけずらして置きますが、それは下付き文字として書くので小さい文字になります。しかしそれらは通常我々が使用するプランク定数の記号とは違っていて、横線は斜線ではなく水平線になります。もう一つの方法は `{/=14 @^{/Symbol=10 -}{/=14 h}}` とすることです。

AMS-LaTeX の PostScript フォント (http://www.ams.org/tex/amsfonts.html にあり、そして多くの LaTeX 配布にも含まれています) を使えば、とても簡単に小さい Planck 定数をセットすることができますが、**gnuplot** (`help fontpath` を参照) と、PostScript インタープリタ (通常 Ghostscript) がファイル `msbm10.pfb` (または `msbm10.pfa`) がどこにあるかを知る必要があります。"h" に傾斜のついた横棒を重ねた `\hslash` を生成するには `{/MSBM10 \175}` を使ってください。通常の `\hbar` (水平の横棒のもの) は 8 進数の 176 です。ただし、水平の横棒はイタリック体としてしか存在しないことに注意してください。

6.5 太陽質量記号 (丸に点) が必要なときはどうしたらいいですか

Planck 定数と同様、最も自然な方法は UTF-8 フォントを使い、文字 ⊙ (Unicode コードポイント #x2299 ; "circled dot operator") を直接入力することです。非常によく似たグリフがコードポイント #x2609 にもあります。これは "sun" で、よりいいかもしれませんが、それを提供するフォントは多くはありません。

6.6 ギリシャ文字や他の特別な記号を使うには

古いやり方として、特別なフォント、例えば Adobe の "Symbol" フォントに切り替えるために拡張文字処理モード (enhanced text モード) を使用する方法があります。それは、あなたが出力したい文字に通常の ASCII 文字を割り当てています。これは、現在でも PostScript 出力には必要でしょう。しかし、より簡単な方法は、UTF-8 エンコーディングを選択し、特殊文字を他の文字列と同様にそのまま入力することです。この方法では、フォントの変更は必要なく、また CJK 文字集合を含むすべての Unicode コードポイントへのアクセスが可能になります。しかし **gnuplot** によって作られたファイルを実際に印刷、表示するには、あなたのコンピュータ、あるいは出力デバイスに適切なフォントがインストールされている必要があります。**gnuplot** 自身にはフォントは含まれていません。

いくつかの LaTeX 系出力形式 (*latex, epslatex, tikz, context, cairolatex*) は、生成した文字列を LaTeX に渡します。この場合は、`'\alpha_{3}'` のような、通常の LaTeX 風の記述が利用できます。

6.7 どうしたらアクセント付きの文字を出力できますか

utf8 ではないエンコーディングを使い続けるのであれば、ü (u ウムラウト; u の上に点 2 つのアクセント) や ñ のような文字をラベルに含めるには、適切なエンコードオプションを設定し、8 ビット文字コードを使用しなければいけません。エスケープシーケンスを使って 8 ビットコードを表現することも可能です。例えば次のようにします。

```
gnuplot> set encoding iso_8859_1
gnuplot> set title "M\374nchner Bierverbrauch \374ber die Jahre"
gnuplot> plot "bier.dat" u 1:2
```

しかし、そうでない人は UTF-8 を使うべきで、そこではそれらは普通の文字になります。

6.8 異なるサイズのテキストを一つのグラフで出力できますか

たいていの出力形式では、デフォルトフォントの名前とサイズを指定できます。「拡張文字処理」モード ("enhanced text") では、フォント、および文字サイズやボールド、イタリックの書体を一つのグラフ内で変更できます。

7 その他

7.1 gnuplot には画像形式 XXX をサポートするドライバはありますか

インストール済の **gnuplot** が使用できるグラフィックドライバの一覧を見るには、*set term* とタイプしてください。

通常の配布物には含まれていても、デフォルトでは使えないようになっているグラフィックドライバもあります。それらが使いたければ、ソースからコンパイルし直す必要があります。

7.2 バグを見つけたのですがどうしたら良いですか

最初に、それが本当にバグなのか、それともある set–command で翻すことができる特徴なのかを見極めてください。

次に、あなたが使っている **gnuplot** が古い版でないか確認してください。もしそうなら、新しい版でそのバグは取り除かれている可能性があります。

開発版は、現在のリリース版以後に報告されたバグの修正を既に持っている可能性があります。レポートを送る前に、質問のバグが既に修正されていないかチェックしてみてください。

これらをチェックした後でなおそれがバグであると確信したなら、http://sourceforge.net/p/gnuplot/bugs の bug-tracker で報告してください。その際、**gnuplot** のバージョン (パッチレベルも含む) とそれが動いているオペレーティングシステムも書いてください。特にそのエラーを再現する簡単なスクリプトを提供してもらえると大いに助かります。

souceforge の tracker は、バグのレポート、および後のリリースに反映するためのバグの修正の収集のためのものです。**gnuplot** に関する問題に対する回避策を見つける、あるいはそれを実際に解決するには、インターネット上に色々存在するフォーラムの方がよい場所でしょう。

7.3 どう追加や拡張がなされていますか、それはどう手に入れられますか

一般的に、現在の開発版は、最終安定リリース版の gnuplot にはまだ入っていない機能も入っています。2019 年 4 月現在、開発版のバージョン 5.3 は、以下を特にサポートしています:

- 64 ビット整数計算
- 3 次元ボクセル格子 (voxel grid)
- 3 次元描画スタイル *isosurface, circles, boxes*
- コマンド *set walls* で 3 次元グラフに xy, xz, yz 平面を表示
- Unicode エスケープシーケンス \U+hhhh
- 第 1 種、第 2 種の N 次ベッセル関数
- 変形ベッセル関数
- 複数の textbox スタイル

7.4 "heavy-duty" データを gnuplot で使えますか、または gnuplot に代わるものはないですか

gnuplot それ自身は、しんどい数値計算にはあまり向いていません。しかし、かなり大きなデータ集合 (100 万データ点程度) は簡単に処理できます。簡単ではない関数の評価のための gnuplot の速度が制限を与えてしまうような特定のアプリケーションを使用したい場合は、その関数を C か C++ でコーディングして作り、それをプラグインとして gnuplot で呼び出す、という方法もあります。

さらなる数値計算を必要とする場合は、以下の 2 つのソフトを調べてみてください:

octave (http://www.octave.org) は、高水準言語で、主に数値計算用です。Octave は GPL ライセンスに従い、原則的にフリーな Matlab クローンです。線形や非線形の問題を数値的に解くのに便利なコマンドラインインターフェースを備えています。なお、octave は **gnuplot** を描画エンジンとして使いますので、**gnuplot** の上位のデータ処理ソフトを手に入れることにもなります。

scilab (http://www.scilab.org) は、別の、**matlab** に代わるオープンソースソフトです。

7.5 gnuplot が持たない特殊関数が必要な場合は

上を見てください。他のソフトを使うか、カスタムプラグインを書くか、のいずれかが必要です。

7.6 私の対話型ターミナルでホットキーを使うには

ほとんどの対話型出力形式では、再描画、各要素の描画のトグル、軸のスケールの変更などを行うために、あらかじめ設定されたホットキー、およびユーザ定義可能なホットキーの両方をサポートしています。現在有効な gnuplot の描画ウインドウ上で *h* を打つと、ホットキーの一覧が表示されます。さらなる情報については、*help mouse* と *help bind* を参照してください。

7.7 gnuplot の次期バージョンの開発について知りたいのですがどうしたらいいですか

gnuplot のベータテストメーリングリストに入ってください。メール本文 (メールタイトルではなく) に `subscribe gnuplot-beta` と書いてメールを mailto://Majordomo@lists.sourceforge.net に送ってください。

8 共通の問題

8.1 助けて! 私のフォントが使えない

gnuplot は、それ自身ではフォント処理を行いませんので、それは必然的に個々のデバイスをサポートするライブラリに任せることになります。残念ながらこれは、フォントの検出においては、異なる出力形式には異なる説明が必要だということを意味しますので、ここでは、簡単なヒントをいくつか上げておきます。詳しい情報については、あなたが問題にしている出力形式用の gnuplot のドキュメントを参照してください。

png/jpeg/gif これらの出力形式は、libgd というサポートライブラリを使用し、これは環境変数 GDFONTPATH で与えられるディレクトリ内でフォントを探します。そのように libgd のフォントパスを設定すると、今度は gnuplot のデフォルトフォントを設定したいかもしれません。例:

```
setenv GNUPLOT_DEFAULT_GDFONT verdana
```

post PostScript フォント名は、その文書が印刷されるまで参照されませんから、あなたのプリンタでどのフォントが有効であるかを gnuplot は知りません。よって、どんなフォント名でも指定することができます。しかし、gnuplot の出力にフォントを埋め込むことは可能です。それについては、gnuplot の内部コマンド “help set term post fontfile” で得られる指示を参照してください。

svg フォント処理は、ビューワ依存です。

x11 x11 出力形式は、通常の x11 フォントサーバ機構を使用しますが、少しだけトリッキーな点があり、マルチバイトフォントを使用するためには、明示的に以下のように指定する必要があります:

```
set term x11 font "mbfont:sazanami mincho,vera,20"
```

win ツールバーの "Options" のプルダウンメニューから "Choose font..." を選択します。

wxt, qt Linux システムでは、これらの出力形式はシステムの fontconfig ユーティリティが提供するフォントを使用します。

8.2 qt 出力形式のセッションでの最初のグラフが失敗したり変なレイアウトになります

"slow font initialization" というエラーメッセージも出たかもしれません。これは、qt が共有システムフォントキャッシュに依存しているからです。もし、他の誰も使っていないフォントを要求すると、そのキャッシュの更新に時間がかかります。これは、Windows や OSX ではよく起こるのですが、それらのシステムでは他のソフトが異なるフォントの仕組みを使用する傾向にあるため、関連するフォントキャッシュが空になるからです。gnuplot の呼び出し時に、コマンドラインオプション *--slow* をつけてみてください。

8.3 pm3d によるデータファイルの splot で何も表示されません

set pm3d; splot 'a.dat' とやっても、カラーボックスは表示されるもののグラフは表示されないとしたら、もしかしたら 2 つのつながっている線 (孤立線) を分離する空行がないのでは ? 空行を入れなさい ! これが何を意味するのかに興味があるなら、gnuplot の demo ディレクトリにある *demo/glass.dat* や *demo/triangle.dat* などのファイルをじっくり見てください。
以下の awk スクリプト (例えば *addblanks.awk* と呼ぶことにします) は有用で、データファイルに、第 1 列目の数が変更したときに空行を追加してくれます。

```
/^[[:blank:]]*#/ {next} # コメント行 (# で始まる行) は無視
NF < 3 {next} # 3 列未満しかない行も無視
$1 != prev {printf "\n"; prev=$1} # 空行を出力
{print} # その行自体を出力
```

これにより、データファイルをコマンド *awk -f addblanks.awk <a.dat* によって前処理するか、unix 互換のプラットフォーム上でデータファイルを
gnuplot> splot "<awk -f addblanks.awk a.dat" のように描画するか、のどちらかを行ってください。

8.4 なぜ gnuplot はとても小さい数字のデータを無視するのですか

gnuplot は、ある目的のため 1e-08 より小さい数を 0 として扱います。よってとても小さい数の集まりを描画しようとするとそれらは 0 として描画されてしまいます。対数軸で描画する場合は、もっとひどいことにそれらは目盛りからはずれてしまいます。また、全ての数が "0" である場合、範囲は空であるとみなされます:

```
gnuplot> plot 'test1'
Warning: empty y range [4.047e-19:3e-11], adjusting to [-1:1]
gnuplot> set yrange [4e-19:3e-11]
gnuplot> plot 'test1'
               ^
y range is less than `zero`
```

その対策は **gnuplot** の "zero" の概念を変更することです。

```
gnuplot> set zero 1e-20
```

より詳しい情報については *help set zero* を参照してください。

8.5 multiplot のグラフを *replot* すると一かけらしか出ません

gnuplot は、直近の *plot* や *splot* コマンドを再生成するために十分な情報を保持するだけなので、multiplot に対しては、それを生成する完全なコマンド列をスクリプトファイルに保存する必要があります。そして、*load* することでその内容を **gnuplot** に取り込めば、あなたが必要とする回数だけ異なる出力形式、異なる出力ファイルに対して再描画をすることができます。

8.6 書いた式 (1/3 など) が意味のない結果を出力してしまいます

gnuplot は整数の式の場合、実数演算ではなく整数演算を行ないます。例えば式 1/3 は 0 と評価されます。もし実数の値が欲しいならば、その数の後にドット "." をつけてください。例:

```
gnuplot> print 1/3
                0
gnuplot> print 1./3.
                0.333333
```

整数の式をこのようにして評価するやり方は C や Fortran でも行なわれています。

8.7 私の出力が不完全です

set output か *unset output* によって、たまっている出力を吐き出させ、そして出力をクローズする必要があるかも知れません。

出力形式によっては、一つの出力ファイルに複数のページが含まれています (postscript, pdf, latex, svg, ...)。これらの出力モードでは、gnuplot はあなたが描画の追加ができるように各描画後にファイルをオープンしたままにしています。よって異なる出力形式を選択する (*set term*) か、または gnuplot を終了するまでは、そのファイルは完全ではなく、外部のアプリケーションでは使用できるようにはなっていません。

8.8 パイプやスクリプトから **gnuplot** を呼び出すとスクリーンにグラフがでません

よくある原因の一つが、**gnuplot** がグラフの描画直後に終了してしまうことで、終了時に描画ウィンドウが閉じてしまうため、あなたがグラフを見れないぐらいにすべての表示があっという間に終わってしまう、という状態です。これには、いくつかの解決法があります。

- gnuplot のコマンドラインオプション *-persist* を使用すること。これは、gnuplot 自身の終了後も描画ウィンドウを残します。
- オプション *set term* で *persist* を受け付ける対話型出力形式を使用すること。
- パイプの終了前に gnuplot にコマンド *pause mouse close* を送信すること。これは、描画ウィンドウが他の動作で終了するまで gnuplot を終了させないように指示します。
- あなたが gnuplot、および描画ウィンドウ等々を本当に必要としなくなった場合にのみ、スクリプトにパイプを閉じさせること。

以下は、これらの修正のうち 2 つを示す Perl スクリプトです:

```
#!/usr/local/bin/perl -w
open (GP, "|/usr/local/bin/gnuplot -persist") or die "no gnuplot";
# 各出力後に強制的にバッファを掃き出す
use FileHandle;
GP->autoflush(1);
print GP,"set term x11; plot sin(x) with lines\n";
print GP,"pause mouse close\n";
close GP
```

9 帰属

この FAQ は、Russell Lang, John Campbell, David Kotz, Rob Cunningham, Daniel Lewart, Alex Woo の寄与を John Fletcher が最初にまとめたものです。そして、Alex Woo の草案により、Alex Woo, John Campbell, Russell Lang, David Kotz らの修正と追加、Daniel Lewart による多くの修正が Thomas Koenig によりまとめ直されました。そして、**gnuplot** 3.7 用に再び Lars Hecking, Hans-Bernhard Broeker およびその他の人々による修正が Alexander Mai と Juergen v.Hagen によりまとめ直されました。**gnuplot** バージョン 4 用の修正は、Petr Mikulík と Ethan Merritt により行なわれました。**gnuplot** バージョン 5 用の修正は、Ethan Merritt により行なわれました。

(訳注: 日本語訳は Shigeharu TAKENO (mailto://shige@iee.niit.ac.jp) により行なわれました。)