

texgraph コマンドリファレンス

for texgraph.sty ver. 1.0.0

2016 年 8 月 18 日 (木) 21:58 作成.

threegirls89.

概要

texgraph は \LaTeX 上で外部のテキストファイルに記述されたデータのグラフをプロットするためのスタイルファイル群です. 座標の計算に python を用いているため, この導入が必須です. バグ報告, 修正, その他のご要望は

threegirls89.asm@gmail.com

へお願いいたします.

目次

1	はじめに	2
2	使用方法	3
2.1	ファイル構成	3
2.2	組版方法	3
2.3	データファイル形式	3
2.4	参照軸	3
3	基本文例	5
4	コマンド詳説	6
4.1	texgraph 環境	6
4.2	プロットコマンド	6
4.3	軸設定コマンド	6
4.4	枠設定, 描画コマンド	7
4.5	軸タイトル描画コマンド	7
4.6	凡例描画コマンド	8
4.7	その他描画コマンド	8
5	使用例	10

1 はじめに

本パッケージ `texgraph` は \LaTeX 上で外部のテキストファイルに記述されたデータのグラフをプロットするためのスタイルファイル群である。

本パッケージの特徴を述べる。グラフ画像は `eps` 形式で生成される。 \LaTeX + `dvipdfmx`, `pdf\text{\LaTeX}` など `tex` ファイルから直接 `pdf` 文書を得ることが主流の現代では `eps` 形式は `obsolete` であると言えるが, `postscript` 言語の持つ (\TeX と比較して) 高い数値処理能力を利用するため, 加えて `postscript` 側で各種描画命令を関数化することにより \TeX プログラミングの負荷を削減するために採用している。グラフ上の文字は \TeX により埋め込んでいる。 `picture` 環境中で `eps` 画像を取り込み, `picture` 環境の機能で文字を指定座標に置くことでグラフ描画を実現する。この方法により \TeX のフォントをグラフ上で使用でき, グラフ中の文字サイズを本文に合わせる事が容易になる, 指数などの表記として \TeX の数式を使用できる利点がある。

また, 取りうる数値幅の大きい目盛数字を \TeX で扱うことは困難であるため, 目盛数字描画の計算を `python` に外注している。



以下蛇足: 開発の経緯について。

筆者は `sma4win`^{*1}なるグラフ描画ソフトウェアを愛用していた。このソフトウェアはグラフ描画までの手順が少なくデータファイルから手早くグラフを得られる, (一見画面表示は貧弱だが) 印刷したグラフが綺麗である利点を持つ一方, `windows` 専用である, `png` 或いは `wmf`(`windows metafile`) 形式のみの出力であるため, 仮想 `Postscript` プリンタを用いて \LaTeX で容易に利用可能なベクタグラフィクスである `eps` 画像を得るという煩雑な手順を経て利用していた。大学の実験レポートを書く際にこの手順が煩わしく感じられるようになり, \LaTeX にグラフ描画コマンドを定義すればグラフの描画を \LaTeX 上で完結することができ, かつプラットフォーム依存も解消し, レポート執筆の手間の短縮に寄与すると考え本スタイルファイルを製作した。

^{*1} `sma4 for windows` のページ <http://hp.vector.co.jp/authors/VA002995/soft/sma4win.htm> より入手可能。古いソフトウェアだが `windows 10` でも使用可能。

2 使用方法

本パッケージの使用方法を説明する.

2.1 ファイル構成

本パッケージの構成は以下の通りである.

- texgraph.sty
- tgepspreamble.sty

texgraph.sty が本体であり, tgepspreamble.sty は postscript で書かれたグラフ描画函数一式を eps ファイルに書き出すためのスタイルファイルである. 両者は同一階層に置かれなければならない.

2.2 組版方法

本パッケージは L^AT_EX 2_ε の使用を前提とする. 本パッケージを利用するには texgraph.sty を `\usepackage` する. また, postscript 画像を取り込むため graphicx.sty を `\usepackage` することが必須である. 適切な dvi ドライバを指定して `\usepackage` してほしい.

また, L^AT_EX のオプションに `--shell-escape` を付加しなければならない. これは計算処理を python スクリプトに行わせる関係で, `\write18` コマンドを実行する必要があるからである. この機能はシェルにおいて任意のコマンドを実行することを許可するため, 危険を伴う. 信頼の置けるコードに対してのみ実行すべきである.

更に, python を実行できる必要がある. UNIX, Mac OS X では標準配布なので既に実行できる状態にあるが, Windows では python をインストールし, 単独で実行できるよう環境変数 Path を通す必要がある. python の Version 2.x か 3.x かは問わない.

2.3 データファイル形式

本パッケージの読み込むデータファイルは以下に示す形式である必要がある. 基本的に sma4win のデータファイル形式に準ずる. 異なる部分は脚注にて補足する.

- (1) データは列毎に並ぶ. 1 列が 1 系列である.
- (2) 全ての列のデータ数は等しくなければならない. 不揃いの場合は不足分を `=` で埋める.*²
- (3) スペース or タブ区切りのテキストファイル (CSV は不可). 連続したスペース or タブは 1 つのスペースとして扱う.*³
- (4) 行頭が `#` の行はコメント行.
- (5) コメント行以外の行に数値, 区切り文字以外と見なされる文字 (列) があってはならない.*⁴
- (6) `=` はプロット線の分割にも使用できる. データファイル上 `=` より上にある点と下にある点の間に線は描画されない.

2.4 参照軸

本パッケージでは 2 種の横軸, 縦軸が利用可能である. 横軸は x 軸, u 軸の 2 種, 縦軸は y 軸, r 軸の 2 種が存在する.*⁵ 2 つの横軸, 縦軸には異なる数値範囲, 軸タイプ (線型 or 対数) を設定できる.

*² sma4win では各列の行数は不揃いで好い.

*³ sma4win では 1 つのスペース or タブが 1 列の区切りとなる. 2 つの連続したスペースは間に空文字列データがあるものとして扱われる.

*⁴ sma4win では数値, 区切り文字以外の文字 (列) は無視される.

*⁵ 軸の呼称は sma4win に倣っている.

グラフ上の目盛数字, 軸タイトルの位置は x 軸: 下, y 軸: 左, u 軸: 上, r 軸: 右となる. また, 枠タイプを方眼とした場合は x 軸, y 軸の目盛線のみ描画される.

3 基本文例

本パッケージを使用する際の文例を以下に示す.

仕様により, 軸の設定をしなければ枠の描画, データのプロットは行えず, また後に実行したコマンドによる描画が上に重なるため, 枠描画コマンドを後に書くと罫線がプロット点の上に重なるという制約が存在する. 軸の設定, 枠の設定, プロット, 凡例描画, その他の文字列はこの順に書くことを推奨する.

```
\documentclass[...]{...}
\usepackage[...]{graphicx}
\usepackage{texgraph}
...
\begin{document}
...
\begin{texgraph}<bottommargin=10pt,leftmargin=10pt,rightmargin=10pt,topmargin=10pt>%
    (10truecm,10truecm)
% 軸の設定
\xaxis<increment=0.5,division=5,plotttype=linear,format=f,presition=1><(-1,1)
\yaxis<increment=0.5,division=5,plotttype=linear,format=f,presition=1><(-1,1)
% 枠の設定, 描画
\frame{measurelength=5,measurethickness=0.5,submeasurethickness=0.1,%
    framethickness=1,frametype=section}
\xtitle{xxxx}
% 軸タイトル
\yttitle{yyyy}
% データファイルをプロット
\plot<pointtype=1,linetype=1,linethickness=0.5,pointsize=5>{hoge.txt}[x:1][y:2]{hoge}
\plot<pointtype=2,linetype=1,linethickness=0.5,pointsize=5>{hoge.txt}[x:1][y:3]{piyo}
% 凡例の描画
\legend<parsep=\baselineskip,plotlength=20pt,labelsep=1zw,hmargin=5pt,vmargin=5pt>%
    (1truecm,3truecm)
% その他文字列
\tgputstr<margin=1,align=t>[x,y](0.242,0){0.242}
% その他線
\tgline<linethickness=0.3>[x,y](0.242,0)(0.242,1)
\end{texgraph}
...
\end{document}
```

4 コマンド詳説

本パッケージを構成するコマンドを解説する。

keyval の設定項目の内、デフォルト値があるものは省略可能である。デフォルト値がないものは省略不可。

4.1 texgraph 環境

本スタイルファイルを使用して 1 枚のグラフを描画するための環境が texgraph 環境である。texgraph.sty の最も基本的なコマンドである。

書式 `\begin{texgraph}<#1>(<#2,<#3> ... \end{texgraph}`
 引数 #1 keyval (省略可).
 #2 グラフ幅 (寸法付き数字).
 #3 グラフ高さ (寸法付き数字).

keyval で設定する内容は以下の通りである。

topmargin 上方向余白 (寸法付き数字, デフォルトは 2mm).
 bottommargin 下方向余白 (デフォルトは 10mm).
 leftmargin 左方向余白 (デフォルトは 14mm).
 rightmargin 右方向余白 (デフォルトは 2mm).

4.2 プロットコマンド

データをプロットするコマンドには `\plot` がある。プロットコマンドは軸設定コマンドより後に書かれなければならない。また、仕様により後に実行したコマンドによる描画が上に重なるため、枠描画コマンドを後に書くと罫線がプロット点の上に重なる。

書式 `\plot<#1>{<#2>[<#3:<#4>][<#5:<#6>]<#7>`
 引数 #1 keyval (省略可).
 #2 データファイル名.
 #3 横参照軸 (x or u).
 #4 データファイル中の横軸の値が書かれた列 (先頭を第 1 列とする).
 #5 縦参照軸 (y or r).
 #6 データファイル中の縦軸の値が書かれた列.
 #7 凡例文字列. 凡例を表示しない時は空文字列 `{}` を指定する.

keyval で設定する内容は以下の通りである。

pointtype 点の種類. デフォルトは 0. 下記の注意をみよ.
 pointsize 点サイズ (数字のみ, pt 単位). デフォルトは 5.
 linetype 線種 (0: 線なし, 1: 実線, postscript 書式の破線の何れか). デフォルトは 1.
 linethickness 線の太さ (数字のみ, pt 単位). デフォルトは 1.

注意. pointtype は 0–32 の数字を指定する。点は 32 種類用意されている。0 は点なしを意味する。数字 1–32 に対応する点を図 1 に示す。

4.3 軸設定コマンド

軸の設定を行うコマンドには `\xaxis`, `\yaxis`, `\uaxis`, `\raxis` がある。

軸設定コマンドはプロットコマンド、枠描画コマンドに先んじて書かれなければならない。

数字	1	2	3	4	5	6	7	8	9	10	11	12
点	●	○	◎	⊙	■	□	▣	▤	◆	◇	◊	◈
	13	14	15	16	17	18	19	20	21	22	23	24
	▲	△	▴	▵	▼	▽	▾	▿	+	×	※	✱
	25	26	27	28	29	30	31	32				
	⬢	⬡	⬣	⬤	⬥	⬦	⬧	⬨				

図 1 プロット点見本.

x 軸の設定用 `\xaxis` コマンドについて解説するが, y 軸, u 軸, r 軸の設定についてはそれぞれ `\yaxis`, `\uaxis`, `\raxis` が対応する. 書式, 設定内容は同じである.

書式 `\xaxis<#1>(<#2,<#3>)`
 引数 #1 keyval.
 #2 グラフ左 (下) 端の値.
 #3 グラフ右 (上) 端の値.

keyval で設定する内容は以下の通りである.

increment 主目盛増分. デフォルトは 1. (線型プロットのみ有効. 対数では省略, 設定によらず 10 倍毎)
 division 主目盛間の副目盛の数. (対数プロットでは 1 以上を指定すると 9 本の副目盛)
 plottype linear (線型プロット) or log (対数プロット).
 format 目盛数字の書式 (e: 指数, f: 小数, l: 指数のみ)
 presition 目盛数字の仮数部小数点以下桁数. デフォルトは 0.

4.4 枠設定, 描画コマンド

枠の設定, 描画を行うコマンドには `\frame` がある.

枠描画コマンドに先んじて軸設定コマンドが書かれていなければならない.

書式 `\frame#1`
 引数 #1 keyval.

keyval で設定する内容は以下の通りである.

frametype 枠の種類 (frame: 枠と measurelength で指定した長さの目盛線, section: 方眼). デフォルトは frame.
 measurelength frametype に frame を設定した際の目盛線長さ (数字のみ, pt 単位). デフォルトは 3. section の時は省略可.
 measurethickness 主目盛線太さ (数字のみ, pt 単位). デフォルトは 0.4.
 submeasurethickness 副目盛線太さ (数字のみ, pt 単位). デフォルトは 0.1.
 framethickness 外枠の太さ (数字のみ, pt 単位). デフォルトは 1.

4.5 軸タイトル描画コマンド

軸タイトルの描画を行うコマンドには `\xtitle`, `\ytitle`, `\utitle`, `\rtitle` がある.

x 軸タイトルの設定用 `\xtitle` コマンドについて解説するが, y 軸, u 軸, r 軸の設定についてはそれぞれ `\ytitle`, `\utitle`, `\rtitle` が対応する. 書式, 設定内容は同じである.

書式 `\xtitle<#1>#2`
 引数 `#1` `keyval` (省略可).
 `#2` 軸タイトル文字列.

`keyval` で設定する内容は以下の通りである.

`xoffset` x 方向変位 (単位付き数字). デフォルトは x , u 軸 0pt, y 軸 -35 pt, r 軸 35pt.
`yoffset` y 方向変位 (単位付き数字). デフォルトは y , r 軸 0pt, x 軸 -25 pt, u 軸 25pt.

4.6 凡例描画コマンド

凡例の描画を行うコマンドには `\legend` がある.

凡例の描画を行うに先んじて `\plot` コマンドで全てのグラフをプロットしなければならない. また, `\frame` コマンドで枠を先に描画しなければならない.

書式 `\legend<#1>(<#2,<#3>)`
 引数 `#1` `keyval` (省略可).
 `#2` 凡例左上端の, グラフ左下端を基準とした x 座標 (単位付き数字).
 `#3` 凡例左上端の, グラフ左下端を基準とした y 座標 (単位付き数字).

`keyval` で設定する内容は以下の通りである.

`parsep` 凡例間の縦方向間隔. デフォルトは 14pt.
`plotlength` 凡例プロットの長さ. デフォルトは 20pt.
`labelsep` 凡例プロットと凡例文字列の間隔. デフォルトは 1em.
`hmargin` 凡例の下地となる白枠の横方向余白. デフォルトは 5pt.
`vmargin` 凡例の下地となる白枠の縦方向余白. デフォルトは 5pt.

4.7 その他描画コマンド

前述のコマンドの他, グラフを修飾するために文字列, 直線, 矢印を描画するコマンドが存在する.
 文字列描画コマンド.

書式 `\tgputstring<#1>[<#2,<#3>](<#4,<#5>)#6`
 引数 `#1` `keyval` (省略可能).
 `#2` 横参照軸 (x or u).
 `#3` 縦参照軸 (y or r).
 `#4` 文字列基準点 x 座標 (数字のみ, `#2` の座標).
 `#5` 文字列基準点 y 座標 (数字のみ, `#3` の座標).
 `#6` 文字列.

`keyval` で設定する内容は以下の通りである.

`align` 文字列を揃える方向. `picture` 環境に準ず. `r`, `l`, `t`, `b`, `c` とその複合. デフォルトは `c`.
`margin` 文字列の背景となる白四角の文字列に対する余白 (pt 単位). デフォルトは 2.

直線描画コマンド.

書式 `\tgline<#1>[#2,#3](#4,#5)(#6,#7)`
 引数 #1 keyval(省略可能).
 #2 横参照軸 (x or u).
 #3 縦参照軸 (y or r).
 (#4,#5) 始点座標 (指定した参照軸による座標数字).
 (#6,#7) 終点座標 (指定した参照軸による座標数字).

keyval で設定する内容は以下の通りである.

dash postscript 書式の破線. 省略可能であり, 省略時は実線.
 linethickness 線の太さ (数字のみ, pt 単位). デフォルトは 1.

矢印描画コマンド.

書式 `\tgline<#1>[#2,#3](#4,#5)(#6,#7)`
 引数 #1 keyval(省略可能).
 #2 横参照軸 (x or u).
 #3 縦参照軸 (y or r).
 (#4,#5) 始点座標 (指定した参照軸による座標数字).
 (#6,#7) 終点座標 (指定した参照軸による座標数字).

keyval で設定する内容は以下の通りである.

dash postscript 書式の破線. 省略可能であり, 省略時は実線.
 linethickness 線の太さ (数字のみ, pt 単位). デフォルトは 1.
 length 鋸の長さ (数字のみ, pt 単位). デフォルトは 14.
 angle 鋸の中心からの開き角度 ($^{\circ}$ 単位). デフォルトは 15.
 type 鋸のつき方 (0: なし, 1: 終点のみ, 2: 始点のみ, 3: 両方). デフォルトは 2.

5 使用例

本パッケージの使用例を以下に示す。組版結果を図 2 に示す。

```
\begin{texgraph}<leftmargin=45pt,rightmargin=45pt,bottommargin=35pt>(9cm,9cm)
\axis<division=1,plottype=log,format=l>(100,1e6)
\yaxis<increment=20,division=4,plottype=linear,format=f,presition=0>(-80,20)
\raxis<increment=45,division=3,plottype=linear,format=f,presition=0>(-190,10)
\frame{frametype=frame,measurelength=8,measurethickness=0.5,submeasurethickness=0.1,%
      framethickness=1}
\plot<pointtype=7,pointsize=8,linetype=1,linethickness=0.5>%
      {Butterworth.txt}[x:1][y:2]{\footnotesize Butterworth--Gain}
\plot<pointtype=8,pointsize=8,linetype=[2 2] 1,linethickness=0.5>%
      {Butterworth.txt}[x:1][r:3]{\footnotesize Butterworth--Phase}
\plot<pointtype=1,pointsize=6,linetype=1,linethickness=0.5>%
      {Bessel.txt}[x:1][y:2]{\footnotesize Bessel--Gain}
\plot<pointtype=2,pointsize=6,linetype=[2 2] 1,linethickness=0.5>%
      {Bessel.txt}[x:1][r:3]{\footnotesize Bessel--Phase}
\xtitle{f$ [Hz]}
\yttitle{|G(f)|$ [dB]}
\rtitle{$\arg G(f)$ $^\circ$}
\tgarrow[x,y](100,-3)(10000,-3)
\tgline<dash=[5 5] 0>[x,y](10000,-80)(10000,-3)
\tgputstring<align=r,margin=0>[x,y](83,-4){$-3$}
\legend<labelsep=0.5em,parsep=12pt,hmargin=0.5mm,vmargin=0.5mm>(43mm,78mm)
\end{texgraph}
```

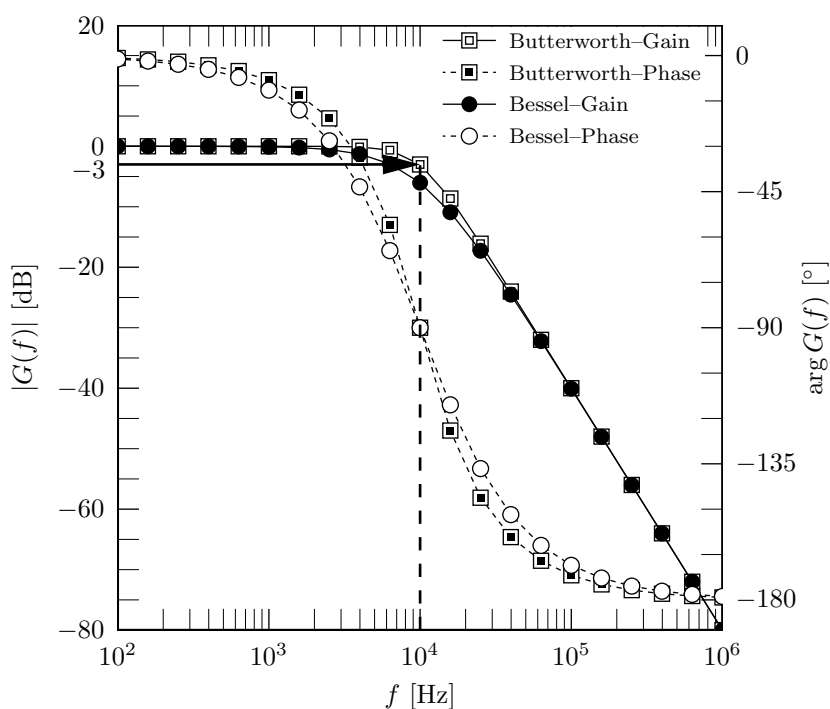


図 2 texgraph.sty を用いたグラフ描画の例.