

# Ngraph で描いた図データの $\text{\LaTeX}$ 化

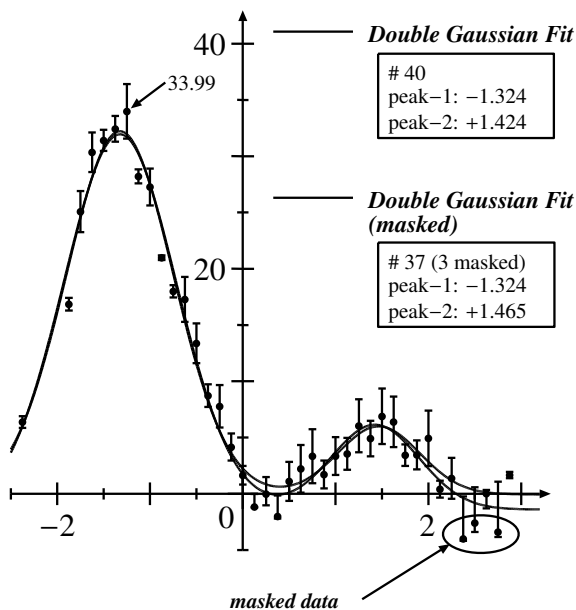
— TPIC special を利用した  $\text{\LaTeX}$  ファイルの生成 —

岩熊哲夫 (2014 年 10 月 12 日)

## 目次

1 概要	1	5.2 線の太さ	7
2 インストール	2	5.3 線の属性	7
3 三通りの実行の仕方	2	5.4 網掛けの濃さ	7
3.1 Windows エクスプローラ上で	2	5.5 輪郭線	7
3.2 コマンドプロンプトに対して	2	5.6 フォントサイズ指定	8
3.3 Ngraph の中から外部ドライバとして	3	5.7 文字列の位置と回転	8
4 何のために、そして $\text{\LaTeX}$ 文書にどうやって読み込むのか	3	5.8 特殊文字	9
5 いくつかの仕様・・・というか、手抜きの結果	5	5.9 旧 v.5 版のデータについて	9
5.1 スタイルファイル 'eepicsup.sty' の利用	5	6 多分どうでもいいだろうこと	10
		6.1 お詫びとご注意	10
		6.2 Times フォントの埋め込み	10
		6.3 カラーのグラフはスライドで使いましょう	10
		6.4 お問合せと最新版	13
		更新ログ	17

### Demo #2: LEAST SQUARE FIT



### Demo #4: MULTI GRAPH

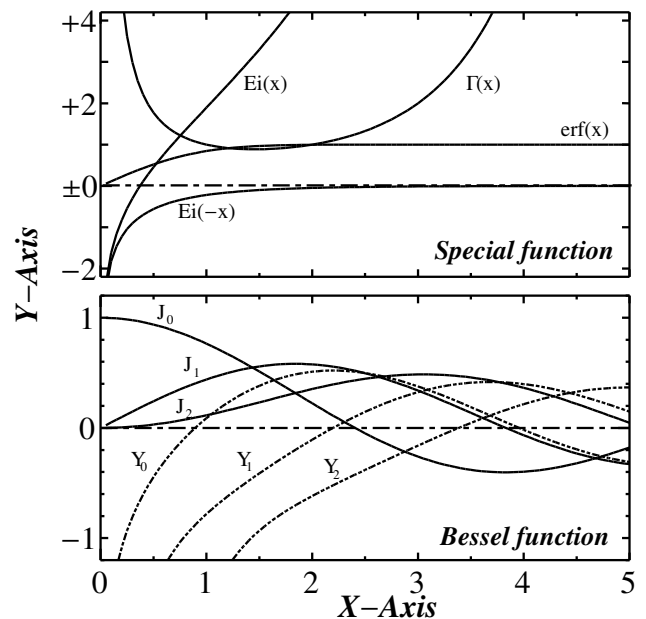


図 1 Ngraph の demo2 と demo4 を 50% 縮小した例；私の設定ではないので左右に少しだけはみ出します

## 1 概要

まず何をしたいのかについて、図 1 をご覧ください。この図は  $\text{\LaTeX}$  のソースファイル、つまり文字列等を自由に編集できる（右上図中の「プラスマイナス ( $\pm$ )」と「ガンマ ( $\Gamma$ )」が  $\text{\TeX}$  のフォントです）テキストファイルのデータで描いたものです。それからこの文書を Reader で読んでいる状態で、キーの Ctrl と D を同時に押してください。現れた小窓内の「フォント」タブをクリックし、使われているフォント名をごらんください。そこで、こりゃいいやぁとお思いになった場合には、以下を読み続けてみてください。これを開発した理由に

については第4節の最初の部分をお読みください。このプログラムは、Ngraph で描いたグラフデータを  $\text{\LaTeX}$  文書で扱えるソースファイルに変換します。ただし、TPIC special を組み込んだマクロが定義されたスタイルファイル `'eepic.sty'` および `'eepicsup.sty'` を利用することを前提としています。後者は拙作で、本アーカイブに同梱してあります。対応する Ngraph のバージョンは、v.5 と v.6 です。v.5 対応は以前私自身が作成したものと基本的には同じ<sup>1</sup>ですが、矢印が綺麗に尖るように改良しました。一方、新しい Ngraph Windows 版 v.6 については別途斉木先生のソフトウェア `gra2eepw` がありますが、何故か私のコンピュータ上では動かないので、斉木先生に承諾を得た上で、私のこの `Gra2Eepic` でも v.6 のファイル进行处理できるようにしました。この版も Ngraph v.6 の外部ドライバに登録すれば使い易くなります。

## 2 インストール

適当なディレクトリ、多分 Ngraph と同じディレクトリに実行ファイル `'Gra2Eepic.exe'` を置いて実行するだけです。ま、このマニュアルとサンプルデータもそこに置いてください。アンインストールするには、その実行ファイル等を削除してください。レジストリは何も触りません。なお、ウィルスチェック後に `'Gra2Eepic.exe'` のアイコンを右クリックして、プロパティの「全般」タブ一番下の「ブロックの解除」をしていただければ、使い易くなります。

## 3 三通りの実行の仕方

### 3.1 Windows エクスプローラ上で

まずはアイコンのクリックというやり方で実行していただき、現れた窓の右上で縮小率 Scale を % で指定します。通常私が 50% に縮小する<sup>2</sup>ので、その状態になっています。そして `gra` ファイルのアイコンをドラッグして、寝台特急「はやぶさ」「富士」のところにドロップするだけです。無事に変換できたかどうかは、`gra` ファイルのあるディレクトリに `tex` ファイルができたかどうかで判断してください。ん！そもそも `gra` ファイルはどうやって作るかって？それは Ngraph からの出力オプションの一つですね。ま、この方法を使う方は非常に限られていると推測します。なお、どのやり方を使っても、`tex` ファイルは常に重ね書きされ警告も出ませんので注意してください。

### 3.2 コマンドプロンプトに対して

あるいは、いわゆる DOS 窓の中で

```
prompt> Gra2Eepic -50 -o fig1.tex c:\mydir\theme25\data1.gra
```

としても同じです。もちろん、`'Gra2Eepic.exe'` には Path が通っているものとします。入力 `gra` ファイルは必ずフルパスで指定し、省略できません。それ以外はすべてオプションです。`'-50'` は縮小率を 50% にすることを意味し、出力 `tex` ファイル名は `'-o'` の次に書きます。省略した場合には、入力ファイルと同じファイル名の `tex` ファイルがその同じディレクトリに出力されます。この出力 `tex` ファイル名に上の例のようにディレクト

<sup>1</sup> 2014 年春までの Windows XP が利用できた間は Ngraph v.5 を使っていたため、この `Gra2Eepic` の拡張が不要でした。実はその後もオフラインのコンピュータおよび VMware Player 中で利用し続けていました。

<sup>2</sup> A4 サイズ 2 段組論文集原稿で、横に二つ並ぶものが 50% 縮小で作ることができるように、Ngraph 側のデフォルトの設定を v.5 のときに変更して使い続けていました。それに対応した v.6 の設定ファイル `'Ngraph.ngp'` と `'Ngraph.ini'` も同梱しますので、参考にしてください。デフォルトでないグラフの場合は、新規作成で X を 4000 にして幅を 12000 にし、Y の高を 12000 にします。そして X 軸と Y 軸の邦文フォントを明朝にしたものが、アーカイブ内の `'Ngraph.ngp'` の（私の）設定に相当します。

りが付いていない場合には、入力ファイルと同じディレクトリに出力されます。この方法を使う方もマイノリティだと想像されます。

### 3.3 Ngraph の中から外部ドライバとして

また、Ngraph v.6 の外部ドライバとして登録することもできます。これが最も標準的な使い方だと思います。前節のコマンドプロンプトに対する仕様は、これができるようにしたものです。「設定」「外部ドライバ」で「追加」を選びます。「名称」は例えば‘Gra2Epic’とでもします。ドライバには、多分「参照」ボタンで探すなどして‘Gra2Epic.exe’を設定してください。「オプション」は例えば縮小率‘-50’とでも書いておいてください。「拡張子」にはドットを忘れずに付けて‘.tex’としておきます。これで2回「OK」して、「設定」「初期状態としてセーブ（各種設定）」と進み、「外部ドライバ」をチェックして「OK」で設定終了です。これ以降は、描画後の「出力」で「外部ドライバ」の‘Gra2Epic’を選べばtex ファイルに変換されます。出力先ディレクトリとファイル名は変換時の「ファイル」窓で確かめてください。そのファイルを編集して数学記号や種々の変更をすれば、それを読み込む $\text{\LaTeX}$  ファイルと同じフォントで図が描けることになります。

## 4 何のために、そして $\text{\LaTeX}$ 文書にどうやって読み込むのか

まず、こんなソフトウェアは要らないと思う方がほとんどでしょう。Ngraph のグラフはps にもできますし、クリップボードにコピーして、他の図化ソフトウェアに読み込むこともできます。しかし私がv.5 を使っていたときに気になったのは、せっかく正確な数学記号が定義されているのに、図中の数学記号等が $\text{\TeX}$  のそれではないこと<sup>3</sup>でした。例えば式のある英文書を普通に $\text{\LaTeX}$  コンパイルしたとき、例えば‘cmex’, ‘cmmi’, ‘cmr’ のCM フォントが使われていたとします。これを‘times’ オプションでコンパイルすると、一部の文字が‘NimbusRomNo9L’になります。‘cmmi’等のCM フォントも残っていますが、これを‘txfonts’ オプションにすると、フォントは‘NimbusRomNo9L’, ‘rtxr’, ‘txex’になります。しかし、この文書に一般の図化ソフトウェアでps 化されたグラフを取り込むと、図中の数学記号はTimes や Helvetica 等のフォントで表現されてしまう<sup>4</sup>わけです。これが気に入らなかった<sup>5</sup>わけです。例えば‘txfonts’ オプションの‘ $\alpha$ ’はSymbol-Italic フォントなら‘ $\alpha$ ’になります。ちょっと違うでしょ。もう少し簡単な、しかし最もわかり易い例は、 $\text{\LaTeX}$  の種々の数学記号、例えば $\hbar$ 等とか、マクロを使った $\sigma_{ij}^{\nabla}$ 等を図中に表示できないことが不満だったわけです。

さて、このGra2Epic が出力したファイルを読み込む文書の方のプリアンブルで、パッケージオプションを

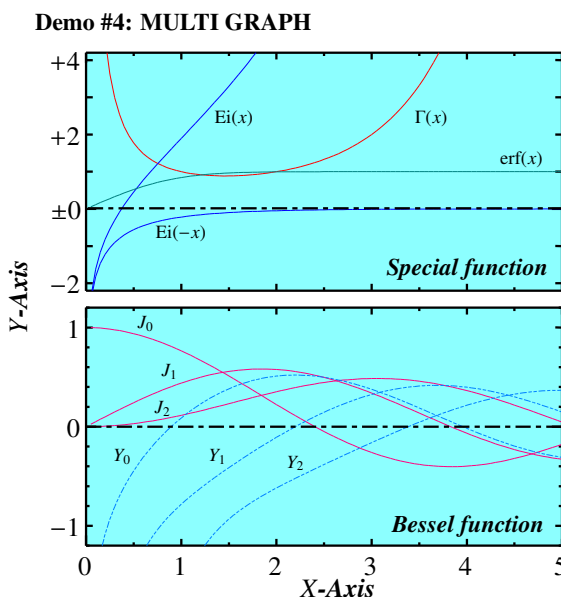


図2 demo4 のフォントを少し変更したもの

<sup>3</sup> 当時はPostScript の作り方使い方を知らなかったことと、有料のソフトウェアをほとんど（今でも）使っていなかったのも理由の一つです。今は $\text{\TeX}$  の数学記号フォントもグラフソフトウェア等で定義できているのかもしれませんが。

<sup>4</sup> 例えばNgraph のdemo4 をeps 化して $\text{\LaTeX}$  文書に読み込むと、‘Helvetica’ と ‘Symbol’, ‘Times’ 種の5個のフォントが含まれます。pdf にこの3種類は埋め込めますが、図中にある記号の形が $\text{\LaTeX}$  本文中のそれと微妙に異なるわけです。

<sup>5</sup> 実はそうならない方法もあるのかもしれませんが、私は知らないのです。

```

\usepackage{epic,eepic,eepicsup}
\usepackage{graphicx} % 横向き文字列が必要な場合のみ
\usepackage{txfonts} % あるいは \usepackage{times} % Times フォントにする

```

で指定してください。1 行目は必ずこの順番で指定してください。2 行目は、縦軸の文字列を横向きにすると  
きにだけ必要となります。3 行目の 2 種類のオプションは、英数字を Times Roman に近づけるためのもので  
す。後述のように、これを用いた方ができあがりが多いが多少は良く<sup>6</sup>なります。

そして、Gra2Epic が出力した tex ファイルが例えば ‘outputf1.tex’ だとすると、図環境中に

```

\begin{figure}
\mbox{\hfill\input{outputf1.tex}\hfill\mbox{}}
\vspace*{-3mm}
\caption{Gra2Epic で変換した Ngraph の図}
\label{fig:outfileno1}
\end{figure}

```

あるいは

```

\begin{figure}
\begin{center}
\input{outputf1.tex}
\end{center}
\vspace*{-5mm}
\caption{Gra2Epic で変換した Ngraph の図}
\label{fig:outfileno1}
\end{figure}

```

のように<sup>7</sup>読み込む手続きをしてください。Ngraph のサンプルを変換したものを図 1 に載せましたが、このサ  
ンプルに対しては一切加筆や変更はせず、50% 縮小だけをしました。また、描画のための Qfig で挿絵を作成  
して Ngraph v.6 と合成した例を図 3 と図 4 に載せました。図 3 の上の図では、数学記号にしなければならない  
ところを敢えてそのままにしてあります。最終的な tex ファイルの string を検索して適切な数学記号に変更し  
たのが下の図です。これがこの Gra2Epic を開発してやりたかったことです。

もし dvipdfmx で変換した pdf ファイル中で、横向きの文字列の図があるページにある他の図等にノイズのよ  
うな線が現れたときには、その図を別のページに移動<sup>8</sup>してください。それができないときは、あるいは

```

\usepackage[dvipdfmx]{graphicx}

```

のように、オプションの ‘[dvipdfmx]’ を付けてみてください。これでうまくいく場合もありますが、そうで  
ない場合もあり、原因はわかりませんでした。またこのときは dviout の表示上では文字列が横向きにはなりま  
せんが、pdf ファイルでは正しい図になります。

<sup>6</sup> 私は普通の文書でも前者に [varg] オプションを付けて使っています。

<sup>7</sup> ついでに書いておきます。キャプションコマンドはセンタリングの中には入れません。またセンタリング環境を使うと、多少上下ス  
ペースが空きますから、キャプションとの間に少し大きめの調整が必要になります。

<sup>8</sup> つまり、移動してもいいように、論文の本文の論理や書き方を変更していただきたいってことですね。本末転倒ですよ。うーむ。

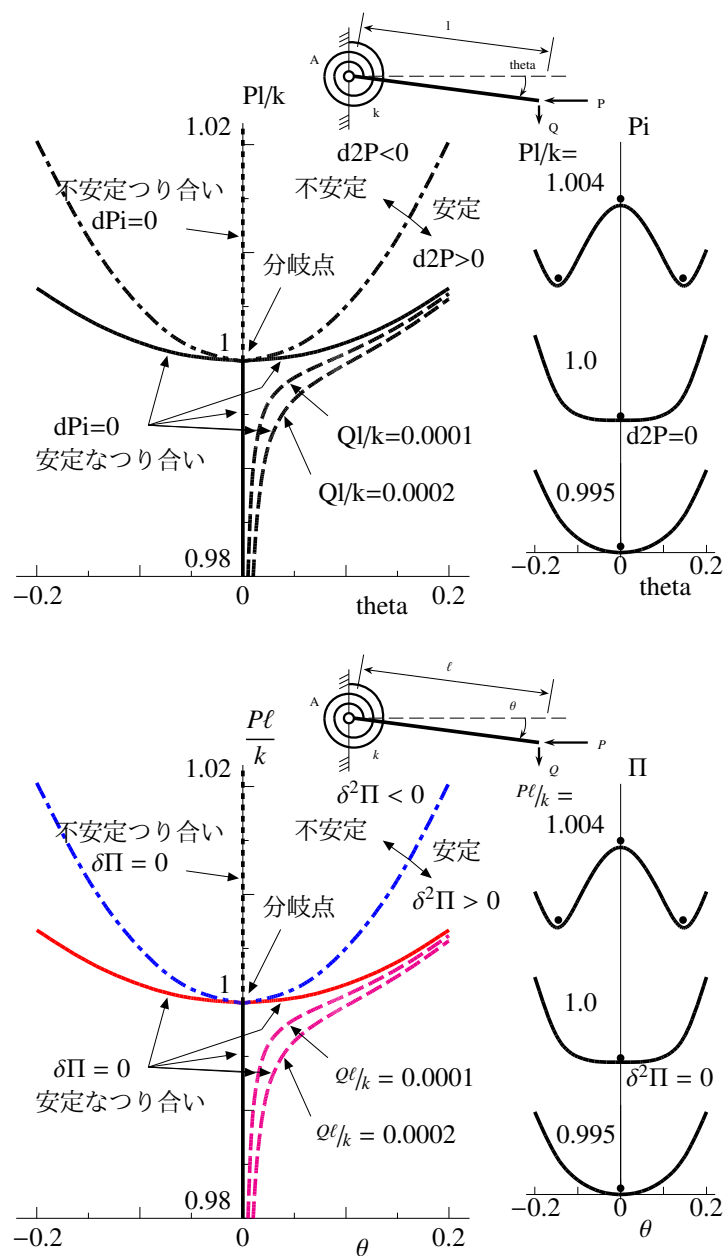


図3 Qfig で描いた挿絵（右上）を Ngraph で作ったグラフと合成した例；上は Gra2Eepic そのまま，下は文字列を数学記号に変更したもの；Qfig の挿絵に使う文字列は Qfig 側ではなく，Ngraph 側で入力した方がよいことがわかります

## 5 いくつかの仕様・・・というか，手抜きの結果

### 5.1 スタイルファイル ‘eepicsup.sty’ の利用

まず私のスタイルファイル ‘eepicsup.sty’ の何を使っているのかについて書いておくと

- 図形の（輪郭）線を消す新しいコマンド
- 網掛けの濃さをオプションとする拡張コマンド

の二つです。これはいずれも TPIC の portability は保持されています。もしこれを止めたい場合には

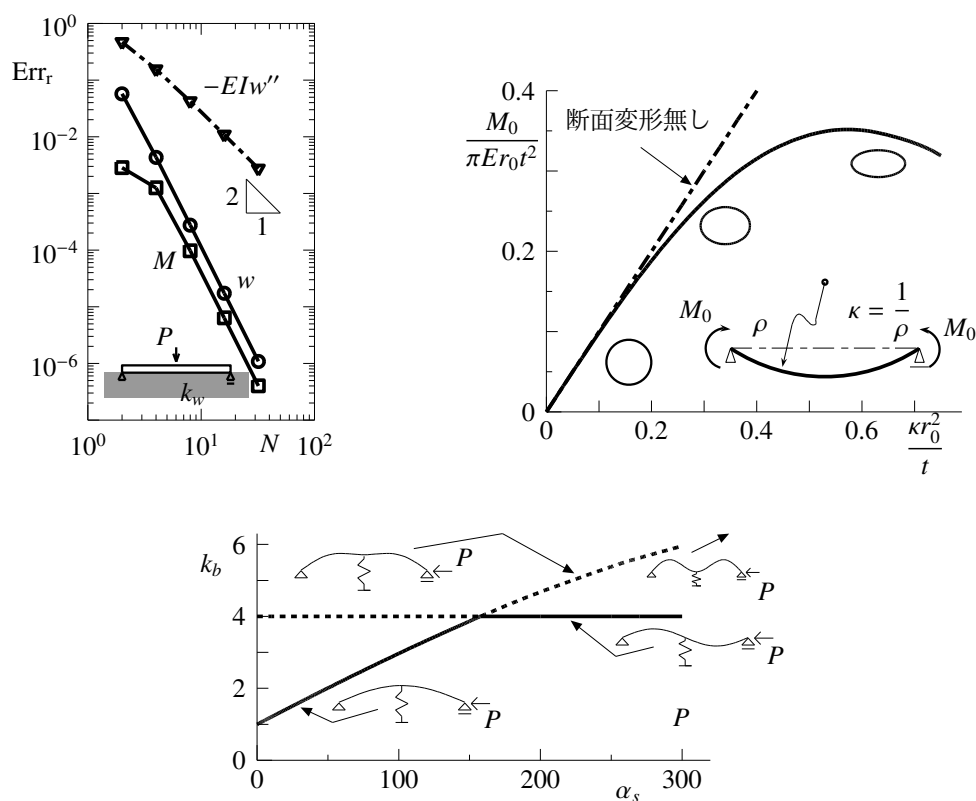


図 4 Qfig 挿絵を含んだ実例をいくつか・・・

- `\shadingwithoutoutline` と `\outlinedshading` の行の削除
- `\shade[0.1]` 等の数値オプション `[0.1]` 等の削除

を実施した上で、プリアンプルのパッケージ指定リストから `'eepicsup.sty'` を削除してください。輪郭線が出たり、網掛けが一種類しか使えなくなりますから、あまりお勧めしません。行の削除等は面倒でしょうから、そのときは文書のプリアンプルに

```
\let\shadingwithoutoutline\relax
\let\outlinedshading\relax
\def\shade[#1]{\special{sh}}
```

の 3 行を加えた上で、パッケージ指定リストから `'eepicsup.sty'` を削除してください。

また v.1.2 までの一点鎖線には、このスタイルファイルで定義した `\DashLine` コマンドを利用していました。これは dviout 上での TPIC 解釈で追加されたもので、portability がありませんでした。また `'eepic.sty'` の `\dashline` も、中間点が不規則に並ぶことが原因で綺麗な破線にはできなかったこともあり、v.1.3 からは、破線等を Ngraph での定義を真似て改善しました。

## 5.2 線の太さ

20, 40, 80 (0.2, 0.4, 0.8 mm) を使うことを前提<sup>9</sup>としています。したがって、0.3 mm 未満の線をコマンド `\thinlines` で、0.6 mm 未満の線を `\thicklines` で、それ以上を `\Thicklines` で指定して描くようになります。0.8 mm の線が論文中の最重要な結果の線になります。なお線の太さには縮小率は適用されません。

`\Thicklines` がもう少し太いといいと思いますが、通常は私も定義のままで使っています。もし太くしたい場合には、文書のプリアンブルで

```
\makeatletter
\def\Thicklines{\let\@linefont\tenlnw \let\@circlefont\tencircw
\@wholewidth\fontdimen8\tenlnw \@wholewidth 1.8\@wholewidth
\@halfwidth .5\@wholewidth
\@gphlinewidth\@wholewidth \divide\@gphlinewidth 4736\relax}
\makeatother
```

のように `\Thicklines` を再定義してください。定義の 2 行目右にある ‘1.8’ という数値が、オリジナルの定義では ‘1.5’ でした。‘2’ にするとちょっと太過ぎるかもしれません。ちなみに、この文書ではこの ‘1.8’ を使ってみました。

## 5.3 線の属性

Ngraph v.6 では、予め設定された破線パターン以外に、ユーザーが自由に定義できるようになっています。例えば図 1 の右図の 9 番目のデータの描画設定を見てください。線種に ‘200 50 100 50 100 50’ という数値による定義がなされています。これは、長短々の破線パターンの定義です。このような場合でも、Gra2Epic はこのパターンに縮小率を適用しますので、図 1 のような結果になります。

## 5.4 網掛けの濃さ

私の Qfig での設定との関係で、青とシアンが `\shade[0.1]` に、緑とマゼンタが `\shade[0.2]` に、赤と黄色が `\shade[0.4]` になります。もちろん黒は `\blacken` で白は `\whiten` です。それ以外の色は適当に近い色の設定に合わせました。Qfig をお使いでない方も、モノクロの出力に対しては薄い順に「シアン」と「マゼンタ」「黄色」をできるだけお使いください。

## 5.5 輪郭線

塗りつぶし図形の輪郭線は原則として描きません。Ngraph でそういう扱いになっていたと記憶します。ただし、v.6 の塗りつぶし箱には輪郭線が指定できるようです。これは Qfig 側でも対応はしてあります。つてのは要らない情報でしたか。

---

<sup>9</sup> 某スプレッドシートプログラムのデフォルトままの図をそのまま利用したと想像される手抜き論文をときどき査読します。卒論等ではとても多いですね。研究者・研修学生としてはもちろん失格ですね。これは、中田亨氏のウェブページ「ダメな論文を書く 14 の方法の第 4 項」<http://www015.upp.so-net.ne.jp/notgeld/sotsuron.html> にも書いてあることです。論文の書き方本（最近のは読んでいません、物理学会等の著名なもの）にも書かれているように、線は 3 種類、枠線等が一番細く（例えば 0.2 mm）、グラフの線はその倍々の 0.4, 0.8 mm で描き、重要な結果を最も太い 0.8 mm で描くというのが基本中の基本です。

## 5.6 フォントサイズ指定

これも私が講義ノート作成で必要としている設定を基本にしたため、10pt 文書の作成を前提としていて、縮小率を乗じたサイズが6.5pt 未満の場合には`\tiny`、7.5pt 未満が`\scriptsize`、8.5pt 未満が`\footnotesize`、9.5pt 未満が`\small`、11pt 未満が`\normalsize`、13pt 未満が`\large`、15pt 未満が`\Large`、18pt 未満くらいで`\LARGE`、22pt 未満が`\huge`、それ以上の場合が`\Huge`になります。これは前述のように、私が Ngraph v.5 を使っていたときに Ngraph の設定を調整し、できあがりの 50% 縮小したグラフが横幅約 8 cm 未満になって、普通の A4 文書の中で横に二つ綺麗に並ぶようにしたことに準じています。つまり、Ngraph 上の軸線上数値が 20pt なので、それが 50% 縮小された  $\text{\LaTeX}$  文書中で 10pt になるようにしたものです。つまりフォントサイズには縮小率がそのまま適用されます。

## 5.7 文字列の位置と回転

Gra2Epic では、文字ピッチや高さを平均的な長さで近似していることから、位置はかなりいいかげんになります。特に文字間の位置合わせを定義した場合に余計な空白が生じます。これは残念なことに、 $\text{\TeX}$  本来の CM フォントを用いたときにさらに顕著になってしまいます。そのためプリアンプで、オプションのフォントが宣言してあります。例えば数値の  $10^3$  等は 10 と上付きの 3 が PostScript の文字メトリックで計った距離離れた場所に Ngraph が配置し（それぞれ別々の文字列設定命令 '`H`' と '`S`' および位置合わせ命令 '`N`' で定義されていますが、Gra2Epic は厳密な文字メトリックを認識していないことから、3 の位置がずれます。例えば図 1 右の 0 次の第二種 Bessel 関数 ' $Y_0$ ' の ' $Y$ ' と下付き ' $0$ ' が微妙に近過ぎることからもわかると思います。例えば  $10^3$  のときは、出力された tex ファイルが

```
% strings
\put(3100,-6500){\normalsize 10}
% strings
\put(3420,-6305){\scriptsize 3}
```

となっていますから、ここを直接編集して

```
% strings
\put(3100,-6500){\normalsize  $10^3$ } % <=== 数学記号にする
% strings
% \put(3420,-6305){\scriptsize 3} % <=== コメントアウトしてしまう
```

にしてください。図 2 がそうやって図 1 の文字列を変更したものです。また図 4 の対数グラフにもその例を示しました。

さらに 90 度より大きく 270 度未満の回転した文字列の位置合わせも厳密な対応はできていません。例えば図 9 を Ngraph の出力と比較してみてください。そこでこれを改善するために、`rotating` パッケージを用いた場合のコマンドをコメント欄に

```
% \put(3421,-6839){\begin{rotate}{30} \tiny 0.05\end{rotate}}
\put(3349,-6839){\tiny \rotatebox[origin=l]{30}{0.05}}
```

のように定義しておきました。上の行をアンコメントし、下の行をコメントアウトした例を図 10 に示しましたが、こちらの方が元データに近いです。利用するには



表 1 一部の特殊記号の変換一覧；鉤括弧の中が実際の出力文字

<code>\\ [\\]</code>	<code>\$\backslash\$</code>	<code>\b</code>	<code>{\kern-1.3ex}</code> (バックスペース)
<code>\&amp;</code>	(タブには対応できないので削除)	<code>- [-]</code>	<code>\$-\$</code> (マイナスと解釈)
<code>\x27 [']</code>	(アポストロフィ)	<code>\xAD [-]</code>	<code>-</code> (ハイフン)
【以下 Symbol フォント】			
<code>\$ [∃]</code>	<code>\$\exists\$</code>	<code>\x27 か' [≡]</code>	<code>\$\ni\$</code>
<code>- [-]</code>	<code>--</code> (ダッシュ)	<code>^ [⊥]</code>	<code>\$\perp\$</code>
<code>' [⏟]</code>	<code>\$\overline{\mbox{\phantom{x}}}\$</code>	<code>@ [≅]</code>	<code>\$\cong\$</code>
<code>\ [⋅]</code>	<code>\$\setbox0\hbox{\$\cdot\$}\raise-0.2em\copy0\raise0.2em\copy0\raise-0.2em\box0\$</code>		

`\usepackage{rotating}`      あるいは      `\usepackage[dvipdfmx]{rotating}`

の指定が必要です。なお、パッケージオプションは `graphicx`, `color` のそれに合わせる必要がありますので注意してください。

## 5.8 特殊文字

少し特殊なものは表 1 に記しましたが、通常のキーボードにある文字は Symbol フォントも含め、 $\text{\LaTeX}$  コンパイルでエラーにならないように変換して<sup>10</sup>あります。16 進コード、例えば `\x27`<sup>11</sup> は、原則として BASIC で用意されている組込関数 `ChrW(Val("&H27"))` による文字列で置き換えますが、`\xA0` から `\xFF` までは適当な (いい加減という意味)  $\text{\LaTeX}$  記号に置き換えてあります。中には無理やりの置換もありますので、お気に召さない場合にはご自身で編集してください。図 5 にいくつか表示してみました。上がローマンで下が Symbol です。特に Symbol で括弧の部品等を含む場合には (これも適当ですが)

`\font\fontGRAMath=cmex10 scaled \magstep0`

というフォント名を定義した上で表示できるようにはなっています。ただしメトリックが `Ngraph` のものとは全く異なりますので、出来上がりは滅茶苦茶になります。 $\text{\LaTeX}$  ユーザーならこういった括弧の部品等は必要が無いので、あくまでもエラーを無くすために置換されているものと捉えてください。もしそれでもいい場合で、CM フォントを使いたくない場合には、他の適当なものに置き換えてください。例えば `txfonts` や `times` オプションを用いる場合には

`\font\fontGRAMath=txex scaled \magstep0`

とすればいいようです。なお、軸数値の負を明確にするためにハイフンを \$ で囲んでマイナスになるようにしていますので、もし `Ngraph` の文字列に「ハイフン」を入れたいときは明示的に `\xAD` を使ってください。

## 5.9 旧 v.5 版のデータについて

矢印: 先端を尖らせるために、矢印を輪郭無しで黒塗りつぶしし、線を矢印の重心までしか引きません。

多角形と閉曲線・楕円: `Qfig` 出力の 10 角形以上の多角形と閉曲線・楕円では、塗りつぶしができません。

<sup>10</sup>  $\Upsilon$  だけが無いようです。

<sup>11</sup> アポストロフィ記号とでも読むのだろうか。シングルクォートというのは間違いだろう。「"」もインチ記号であってダブルクォートじゃないから。あるいは分と秒の記号かな。

## 6 多分どうでもいいだろうこと

### 6.1 お詫びとご注意

斉木先生の gra2eepw と二重化してしまっています。個々の命令の変換則についてお互いに情報交換はしてありませんので、斉木先生の版が動く環境<sup>12</sup>の方におかれましては、それぞれの変換結果が異なることはご了承ください。斉木先生のお話ですと、私の v.5 用の旧版に準じた変換にしてあるとのことでした。また今回、新しい Gra2Eepic を開発するに当たって、v.1.3 からは一点鎖線等を、Ngraph の処理を真似て線の属性ではなく個々の線分を一つずつ定義するようにしたので、dvipdfmx できちんと処理できるようになりました。

なお、このソフトウェアを用いた変換エラーが原因で論文受理等でトラブルになることに対しては、申し訳ありませんが責任は持てません。旧版では変換ミスの経験は無く、国内外の論文集用原稿や講義ノートにも長い間使ってきました。しかしこの新版を用いた確認は、Ngraph の demo および私の講義ノート等の数個の図でやっただけですから、例えば文字列等で思いがけないことが起こるかもしれません。特に Ngraph v.6 は使い始めてからの日が浅いこともあって間違いがあるかもしれません。投稿前に元図と変換図をよく比較して間違いが無いことを必ず確認してください。もしバグと思われることがあれば、公開できる時期（当該論文が受理された後）になってからデータ等のファイルを付けて、その情報をいただけますと幸いです。

### 6.2 Times フォントの埋め込み

dvipdfmx を使う場合には 'GS' を利用しているそうです。最近は論文集等で、フォントを埋め込む指示が出ます。'GS' のデフォルトでは Times シリーズは埋め込まれないようになっているらしいので、'GS' の設定を変更する必要があります。

```
GS をインストールした先の...\gs\gs9.xx\Resource\Init\gs_pdfwr.ps
```

というファイルを探してください。その 30 行目くらいに

```
/Times-Roman /Times-Bold /Times-Italic /Times-BoldItalic
```

というのがありますので、これを

```
% /Times-Roman /Times-Bold /Times-Italic /Times-BoldItalic
```

のようにコメントアウトしてください。これで、Times シリーズフォントも pdf に埋め込まれるようになるそうです。Ngraph は Helvetica がデフォルトになっていますが、Symbol フォントも含め、同様の方法でフォント埋め込みができます。

### 6.3 カラーのグラフはスライドで使いましょう

それから、Ngraph で線をカラーにした場合であっても出力はモノクロに限定されています。応力等のコンター等（例えば Ngraph の demo1 等）を除けば、カラーである必要のあるグラフを今まで一度も見たことがありません。「暮しの手帖」方式とまでは言いませんが、一つの図で言いたいことをそんなにたくさんにはいけません。だとするとカラーを使わずに、線の属性と太さと吹き出しや凡例等だけで複数の結果を、『白地に鮮やかな黒線』で確実に区別できます。

ただスライドでは、白地の場合の黒以外に赤と青・マゼンタの 3 色に限定すれば効果的です。一応、Ngraph 中での指定色情報は tex ファイルに

<sup>12</sup> 我々の研究室の学生さんの一人は両方使っています。動的ライブラリの置き方等のシステムの微妙な状況で動作が異なるようです。

```
% Color (R, G, B) = (255, 0, 0)
%{\color[rgb]{1,0,0}% <===== 色指定のコメント行
\path(1023,-5686)(1118,-5748)
..... 線のデータが並ぶ・・・
\path(6435,-5714)(6478,-5686)
% Color (R, G, B) = (0, 0, 255)
%{\color[rgb]{0,0,1}% <===== 色指定のコメント行
```

といった RGB 情報コメントとして記入してありますので、それを利用して、Gra2Epic が出力した tex ファイル中にある当該「線」を探し出して

```
% Color (R, G, B) = (255, 0, 0)
{\color[rgb]{1,0,0}% <===== 色を指定するためにアンコメント
\path(1023,-5686)(1118,-5748)
..... 線のデータが並ぶ・・・
\path(6435,-5714)(6478,-5686)
% Color (R, G, B) = (0, 0, 255)
}%{\color[rgb]{0,0,1}% <===== ‘}’ の次の % を忘れないように !!!!!
```

のように、その当該線のみのカラー設定をアクティブにします。具体的な例を図3の下図に、赤と青・マゼンタの線で示しました。もちろん、これを読み込む文書のプリアンブルで、オプションの color パッケージを

```
\usepackage{color}           あるいは       \usepackage[dvipdfmx]{color}
```

のように設定しておく必要があります。ただし、graphicx パッケージに付けたオプションと同じオプションをつけなければいけません。

あるいはすべての \color 指定行をアクティブにする場合は、その行頭の % を削除し

```
% Color (R, G, B) = (255, 0, 0)
}{\color[rgb]{1,0,0}% <===== 色を指定するためにアンコメント
\path(1023,-5686)(1118,-5748)
..... 線のデータが並ぶ・・・
```

のようにすればカラー化できます。ただすぐにわかると思いますが、一番最初の \color 行頭にある「括弧閉じ ‘}’」と一番最後の \color 直前の「括弧始め ‘{’」を削除することは忘れないようにしてください。このようにした上で、もし、着色したい塗りつぶし領域が存在する場合には、例えばそこは

```
% Color (R, G, B) = (21, 21, 21)
}{\color[rgb]{0.08,0.08,0.08}%
% boxes
\shadingwithoutoutline%\shade[0.9]
\blacken\path(2541,-10835)(2541,.....
          あるいは
\shade[0.4]\path(2541,-10835)(2541,.....       網かけ濃さは [0.1] [0.2] もあります
          あるいは
\whiten\path(2541,-10835)(2541,.....
```

となっていますから、この一つの図形だけをカラーで塗りつぶすには、行頭の `\blacken` 等を `\shade[0.9]` に変更してください。あるいはすべての塗りつぶしをカラー化するなら、`\shade[0.9]` を検索キーに利用し、お使いのエディタで `\shadingwithoutoutline` コマンド行末からの

```
\shade[0.9] 【改行】 \blacken      (Qx エディタだと \\shade[0.9]\n\\blacken)
```

という 2 行にわたる文字列を (Qx エディタだと 【改行】 が ‘`\n`’ です)

```
【改行】 \shade[0.9]                (Qx エディタだと \n\\shade[0.9])
```

で置き換えます。同じことを `\whiten` と 3 種類の `\shade` に対しても行います。こうすることによってその部分は、例えば

```
% Color (R, G, B) = (21, 21, 21)
}{\color[rgb]{0.08,0.08,0.08}%
% boxes
\shadingwithoutoutline%
\shade[0.9]\path(2541,-10835)(2541,.....
```

になるはずです。一番濃い網掛けで着色しようというわけです。ちなみに、白塗り黒塗りの箇所には検索キーの「`%\shade[0.9]`」がありませんから、間違ってそこをカラー化することはありません。

例えば Ngraph の demo5b で各自試してみてください。まず Gra2Epic 変換のままの dvi を表示してみてください。モノクロのままでもそこそこの結果になると思います。次に、上述のようなカラー化をして dvi を表示してみてください。実はそのままだとページ当たりのカラー数が多過ぎて、全部は表示できません。しかし、pdf 化すれば Ngraph 出力とほぼ同じ結果になります。つまり dviout では、1 ページに使えるカラーの数は限定されています。図 6 には、左に Ngraph の demo1 を変換したものと、右には Ngraph の出力図をクリップボードを通して画像にしたものを並べておきました。RGB 値は小数点以下二桁にしてありますが、三桁にしてもほとんど違いは無いようです。

おっと!ところで、スライドを  $\text{\LaTeX}$  では作らないって。ま、そうでしょうねえ。でもね、‘jslides.cls’とか PowerDot もいいですよ。参考までに  $\text{\LaTeX}$  と jslides での設定と、 $\text{\LaTeX}$  と PowerDot についての例等については、私のウェブページ

<http://mechanics.civil.tohoku.ac.jp/%7Ebear/bear-collections/index-j.html>

にある「スタイルファイルの記述とリンク一覧」という行の下の説明文中の「使い方の例」を参考にしてください。dvipdfmx でフォントを埋め込んだ pdf ファイルにしておくと、講演会場がどんな環境でもほぼ問題無くスライド表示が可能です。言い換えると、日帰り出張時に自分のコンピュータを持っていく必要がなく、会場あるいは自分の前か後の講演者のコンピュータを借りる (USB メモリを差し込むだけって) ことができるわけです。これで講演者交代の時間のロスも無くなりますね。

なお非常に個人的にですが、論文中の図には極力カラーを使わないようにした方がいいと思っています。その代わりに、確実に線の属性や 3 種類の太さを使い分けて説得力のある図を作るように、頭の方を使っていたと考えています。Ngraph の石坂さんの例がカラーなのは存じ上げていますが、v.5 に出会って以来手放せなくなり<sup>13</sup>ながら、実はスライド以外でカラーを使ったことが無いのでした。基本は、読む人が一見して

<sup>13</sup> XP が使えなくなつてようやく v.6 への移行です。石坂さんにはたいへん失礼しております。

#### Demo #5D: CURVE

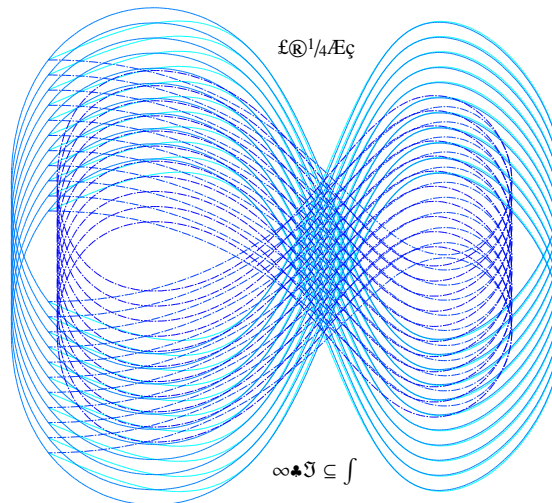


図5 demo5d をカラー化して特殊文字をいくつか加えた

「なめるほど」と思うようなモノクロの図です。「綺麗な配色だけどの線が重要だという主張<sup>14</sup>なのだろう」とか「モノクロ印刷したら<sup>15</sup>ほぼ真っ白のグラフだった」なんて図は作らないようにしてほしいなあと考えています。そのためもあって、カラー化についてチェックボックス等のオプションを敢えて設けず、必要最小限だけを手動で<sup>16</sup>設定できるようにしました。

#### 6.4 お問合せと最新版

メールはしばらくは [tetsuo.iwakuma.e7@tohoku.ac.jp](mailto:tetsuo.iwakuma.e7@tohoku.ac.jp) で通じると思います。また最新版は Qfig と同じ

<http://tedrockbear.s239.xrea.com/>

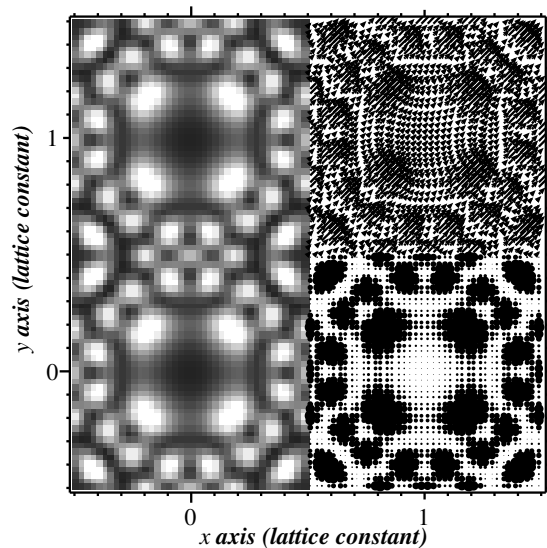
に置いておきます。ちなみにインターネットで公開している私の講義ノートは、2014 年夏の段階で 900 ページくらいありますがその中に、部分的なものや試作も含めて、Ngraph のグラフを 200 個以上、Qfig の絵を 500 個以上使っています。

<sup>14</sup> もちろん、カラーで上手なグラフを描く方もおられるとは思いますが、学生さんの論文と査読論文を見る限りは、そういったものにはいまだに会っていません。

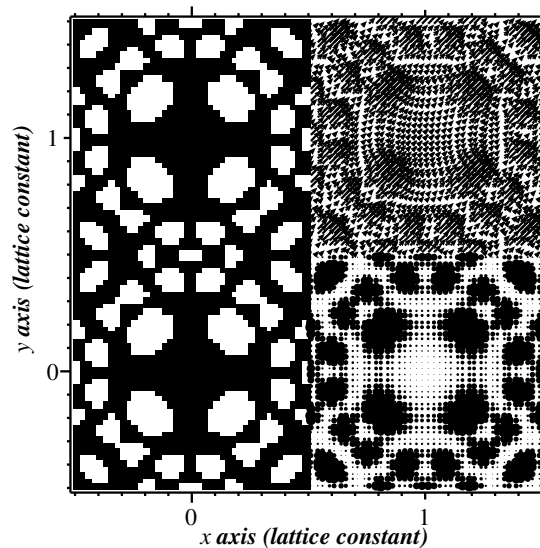
<sup>15</sup> 私は、どんな書類でも査読論文でも原則としてモノクロプリンタしか使わないのです。あるとき（最近うるさいので節約を兼ねて裏紙に）印刷した査読用論文にほぼ真っ白のグラフがあり、返却しようとメールしかけて、ひょっとしたらと思って画面で表示させたところ、白地に黄色や緑色の線だけのグラフでした。トホホでした。多分そのせいで、査読の点数は低かったと思います。ごめんなさい。

<sup>16</sup> もちろん第一の理由は手抜きのためですが。

Demo #1: COLOR(), MARKSIZE() FUNCTION



Demo #1: COLOR(), MARKSIZE() FUNCTION



Demo #1: COLOR(), MARKSIZE() FUNCTION

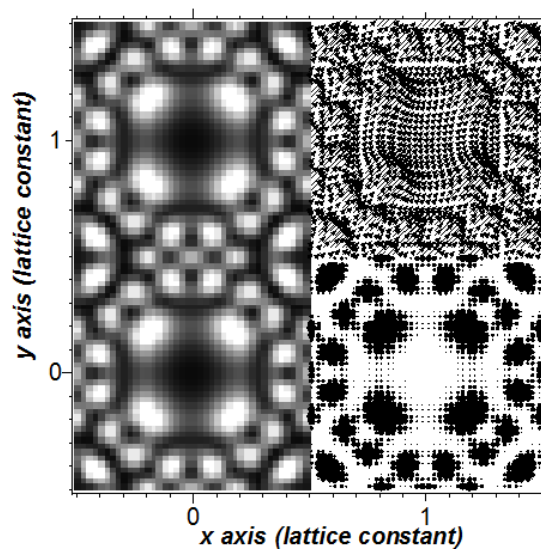
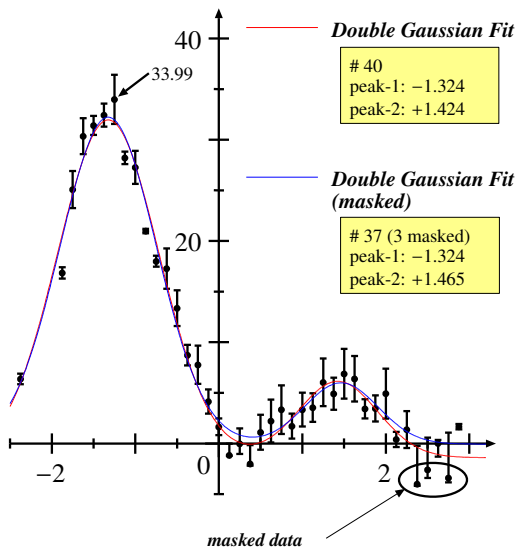


図 6 Ngraph の demo1 で、左上が Gra2Epic で 45% 縮小変換したカラーの  $\text{\LaTeX}$  ファイルの絵，右上はモノクロの絵，そして下にあるのが Ngraph 描画画面をクリップボードを通して IrfanView と Gimp に渡して作成した ps ファイル画像

## Demo #2: LEAST SQUARE FIT



## Demo #3: AXIS DIVISION

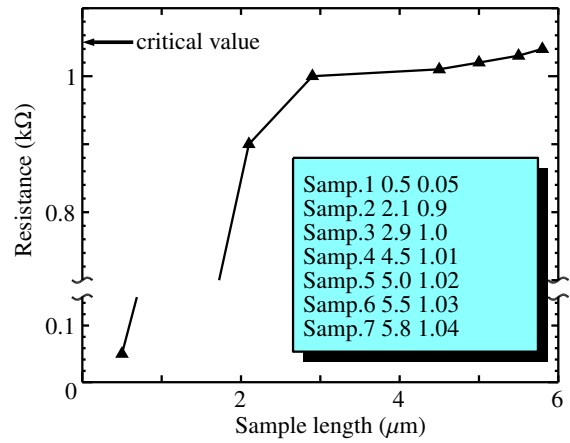
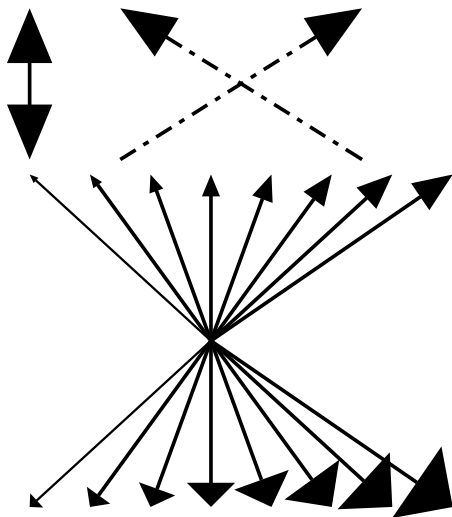


図 7 demo2 はハイフンを改善編集； demo3 は軸の文字列を改善編集

## Demo #5A: ARROW



## Demo #5B: MARK

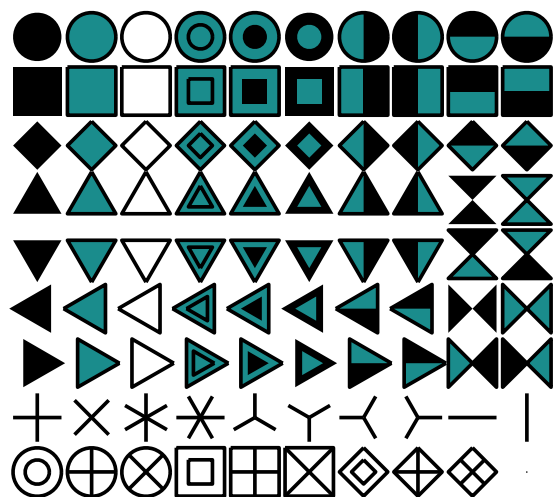
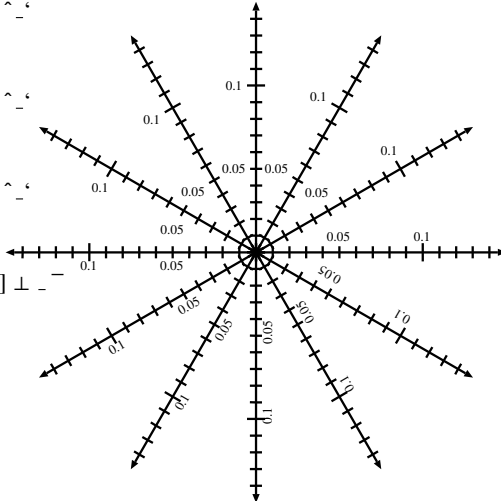


図 8 demo5a はあまりに太い線が出ない例； demo5b はカラー化の例

### Demo #5C: TEXT

Helvetica:	!''#\$%&'()*+,-./0123456789;:<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Times:	!''#\$%&'()*+,-./0123456789;:<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Courier:	!''#\$%&'()*+,-./0123456789;:<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Symbol:	!¥#%& ÷ () * +, - . / 0123456789 ; < = > ? ≅ ↕ ΑΒΧΔΕΦΓΗΘΚΛΜΝΟΠΡΣΤΥΖ Ω Ξ ΨΥΖ [: . ] ⊥ αβγδεφηκλμνπθρστυφωζυζ{ }~

## Demo #5E: AXIS



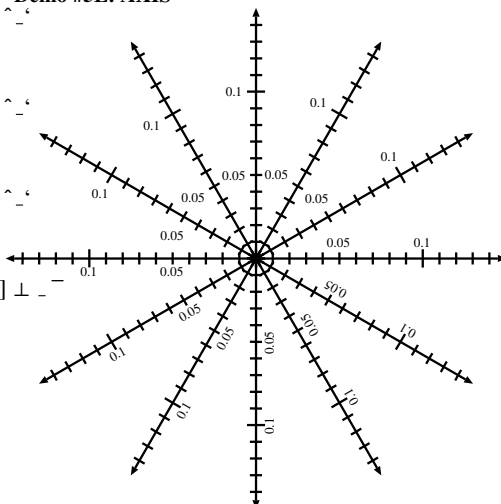
*date: Sat Aug 16 2014*  
*time: 09:11:54*  
*file: demo5c.ngp*

図 9 demo5c は特殊文字の確認と文字列の回転例； demo5e も文字列の回転例

## Demo #5C: TEXT

Helvetica:	!'#\$%&'()*+,-./0123456789:;<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Times:	!'#\$%&'()*+,-./0123456789:;<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Courier:	!'#\$%&'()*+,-./0123456789:;<=>?@ ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_` abcdefghijklmnopqrstuvwxyz{ }~
Symbol:	!\'#\%&'()*+,-./0123456789:;<=>?@ ABXΔΕΦΓΗΘΚΑΜΝΟΠΘΡΣΤΥΖΩΞΨΥΖ[:.] ⊥ αβγδεφγηηκλμνοπθρσττυπωεψζζ{ } ~

## Demo #5E: AXIS



*date: Sat Aug 16 2014*  
*time: 09:11:54*  
*file: demo5c.ngp*

図 10 demo5c と demo5e で rotating パッケージを用いた場合の回転例



## 更新ログ

### v.1.3.2: 2014/10/12

#### 昆虫採集:

1. Symbol 変換ミス。多分こうする人はいないと思うが、例えば '\x3C' で入力した '<' が 'i' になってしまうミス。

### v.1.3.1: 2014/8/17

#### 夏休みの昆虫採集:

1. 破線等の円と楕円や箱の対応忘れ。Ngraph v.6 をようやく使い始めたことからのミス。まだ見逃している機能がたくさんありそう。
2. 図の正しい枠領域のためには、`\end{picture}` の後ろに % 必要。
3. 命令 'P' は輪郭無しにしないと太過ぎることが判明。当たり前か。
4. RGB 値は 128 「以上」を 255 にした方がいい・・・のは、これも当たり前だったか。
5. 文字列は横向きだけとは限らなかったようだ。
6. 文字列の頭にスペースを置いてもいいことに対処。

#### 変更・追加・削除:

1. 輪郭線の扱いをできるだけ精密化。
2. 破線等に `\dashline` 等を使わないようにしたので、Dot-Dashed オプションを無くしてもすべての属性の線が出せるようになった。これで `dvipdfmx` が正しく変換できるようになった。ふうーっ！
3. スライド等での利用を念頭に置き、少しだけ容易にカラー化できるようにしてみた。
4. 折れ線の角での破線の連続性確保の改善努力と、図形端部での破線処理のさらなる改善の努力をしてみた。
5. 特殊文字にさらに対応してみた。しかし・・・

### v.1.2.3: 2014/7/27

#### 昆虫採集:

1. 楕円弧・扇の塗りつぶし対応忘れ。
2. レジェンドと合成ファイル図形の二点鎖線等の正常化。

#### 変更・追加・削除:

1. 黒の場合も塗りつぶしの輪郭線を無くした。
2. 特殊文字にもう少し対応してみた。

### v.1.2.2: 2014/7/23

#### 変更・追加・削除:

1. コマンドラインの引数をさらに変更して Ngraph の外部ドライバに登録できるようにした。

### v.1.2.1: 2014/7/21

#### 昆虫採集:

1. 塗りつぶし色が赤と緑・青の場合の対処忘れ。

#### 変更・追加・削除:

1. 当初追加を悩んでいた Ngraph v.6 対応を，斉木先生の承諾を得て追加。ようやく私も v.6 を使う気になってきたため。
2. 面倒で諦めていた特殊文字等のいつくかに対応。
3. 一点鎖線オプションを追加。
4. コマンドラインの引数を変更。

**v.1.1.1:** 2014/7/15 に学内事情対応のために公開。旧版が v.0.5 だったことから 1 に格上げ。