

# LaTeX のための描画ソフトウェア

— TPIC special を利用した LaTeX ファイルと PostScript ファイル・Ngraph ファイルの生成 —

岩熊哲夫 (2014 年 8 月 12 日)

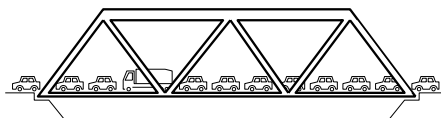
## 目次

<b>1</b>	<b>概要</b>	<b>2</b>	10.4 閉曲線の変換 (Modification and inclined Ellipses) . . . . .	13
<b>2</b>	<b>インストール</b>	<b>2</b>	10.5 楕円と円の変換 (Modification and Regularization) . . . . .	13
<b>3</b>	<b>窓が二つ?</b>	<b>3</b>	10.6 円弧の変更 (Modification) . . . . .	14
<b>4</b>	<b>マウスとキーボード</b>	<b>3</b>	10.7 箱の変換 (Modification) . . . . .	14
4.1	マウス操作モード . . . . .	3	10.8 文字列の変更 (Modification) . . . . .	14
4.2	キーボード操作モード . . . . .	4	10.9 やり直し (Undo) . . . . .	14
<b>5</b>	<b>作業種の指定と解除</b>	<b>5</b>	<b>11 簡易マニュアル (HELP)</b>	<b>14</b>
<b>6</b>	<b>描画範囲</b>	<b>5</b>	<b>12 ファイル入出力 (File I/O)</b>	<b>15</b>
<b>7</b>	<b>描画の仕方</b>	<b>5</b>	12.1 保存 (Save) . . . . .	15
7.1	折れ線 (Lines) . . . . .	5	12.2 読み込みとマージ (Load and Merge) . . . . .	15
7.2	多角形 (Polygons) . . . . .	6	12.3 LaTeX ファイルへの変換 (Export) . . . . .	16
7.3	曲線 (Curves) . . . . .	6	12.4 PostScript ファイルへの変換 (Export) . . . . .	16
7.4	閉曲線 (Closed Curves) . . . . .	6	12.5 Ngraph 用 gra ファイルへの変換 (Export) . . . . .	17
7.5	円と楕円 (Circles and Ellipses) . . . . .	6	<b>13 作業終了!</b>	<b>17</b>
7.6	円弧 (Arcs) . . . . .	6	<b>14 Qfig 生データそのものの編集</b>	<b>18</b>
7.7	箱つまり矩形 (Boxes; i.e. Rectangles) . . . . .	7	<b>15 LaTeX ファイルそのものの編集と利用</b>	<b>19</b>
7.8	矢印 (Arrows) . . . . .	7	15.1 必要なパッケージ . . . . .	19
7.9	文字列 (Strings) . . . . .	7	15.2 もう少し太くしたい . . . . .	19
<b>8</b>	<b>線の属性指定等や節点マーカー表示について</b>	<b>8</b>	15.3 横向きの文字列にしたい . . . . .	20
8.1	線の太さと属性 (thickness and property) . . . . .	8	15.4 図がちょっと小さすぎたので拡大したい, あるいはその逆 . . . . .	20
8.2	網掛けや塗りつぶし (shading) . . . . .	9	15.5 文字列に数学記号を入れたい . . . . .	20
8.3	節点のマーカー表示 (Marking) . . . . .	9	15.6 オプションパッケージを使わない方法 . . . . .	20
<b>9</b>	<b>編集の仕方</b>	<b>9</b>	15.7 Times フォントの埋め込み . . . . .	22
9.1	編集対象の選び方 . . . . .	9	<b>16 PostScript ファイルそのものの編集と利用</b>	<b>22</b>
9.2	コピー (Copy) . . . . .	10	16.1 必要なパッケージ . . . . .	22
9.3	移動 (Move) . . . . .	10	16.2 横向きの文字列にしたい . . . . .	22
9.4	節点の移動と削除 (Node Modification) . . . . .	10	16.3 文字列に数学記号を入れたい . . . . .	23
9.5	回転 (Rotate) . . . . .	10	<b>17 多分どうでもいいこと</b>	<b>26</b>
9.6	対称コピー (Symmetric Copy) . . . . .	11	17.1 たくさんの図形が必要 . . . . .	26
9.7	図形のグループ化 (Grouping) . . . . .	11	17.2 描画の途中誤って Qfig を終了してしまった . . . . .	26
9.8	図形番号と節点の確認 (Object Number Display and Marking) . . . . .	11	17.3 ワードプロセッサ等に挿入したい . . . . .	26
9.9	再描画 (Redraw) . . . . .	12	17.4 旧版にデータを渡したい . . . . .	26
9.10	図形を消す (Kill) . . . . .	12	17.5 Qfig 生データ仕様 . . . . .	27
9.11	すべて消去 (All Clear) . . . . .	12	<b>18 お問合せと最新版</b>	<b>28</b>
<b>10</b>	<b>少し細かい編集 (EDIT)</b>	<b>12</b>	ヒントになるような項目一覧	<b>30</b>
10.1	直線の変換 (Lines $\leftrightarrow$ Curves) . . . . .	13	使用例	<b>31</b>
10.2	多角形の変換 (Modification and inclined Regular Polygons) . . . . .	13	更新ログ	<b>33</b>
10.3	曲線の変換 (Curves $\leftrightarrow$ Lines and inclined Half Ellipses) . . . . .	13		

一応全ページを斜めにお読みください。また万一のときのために旧版データは別に保存しておいてください。

## 1 概要

右図は私の講義ノートにある説明図の一つです。L<sup>A</sup>T<sub>E</sub>X 文中に挿入する図や、Ngraph で描いたグラフ中に挿入する図を描くためのソフトウェアです。こりゃいいと思った方は、読み続けてみてください。さて、L<sup>A</sup>T<sub>E</sub>X 文書（あるいは多分 T<sub>E</sub>X 文書）で使いたい図を描き、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’<sup>1</sup>と‘eepicsup.sty’<sup>1</sup>を利用する L<sup>A</sup>T<sub>E</sub>X ソースファイルに、あるいは PostScript ファイルに、さらに Ngraph に合成できる Gra ファイルに変換する描画ソフトウェアです。必要は発明の母と言われる通り、MS-DOS の時代<sup>2</sup>から開発して自分で使っていたもの<sup>3</sup>で、種々の既製ソフトウェアには（多分）無い、斜めの楕円や斜めの半分の楕円、斜めの正多角形、頭が黒くて三角形ではない矢印、輪郭の無い網掛け領域<sup>4</sup>等を入力できるようにしたかったのが元々の目的でした。また Windows になったあとも、マウスを使わずにできるだけ正確な図<sup>5</sup>を描くことも目的でした。旧 (VBasic4) 版は XP では使えたのですが、その後駄目らしい<sup>6</sup>ので‘Visual BASIC 2010’で書き直すことにしました。ある程度は旧版とのデータ互換<sup>7</sup>性を保った積もりです。残念なことに‘Visual BASIC 2010’では、裏返した色による描画が直線と矩形しかないこと等が原因で、キーボード使用の際の使い方には不便が増している可能性があります、どうぞ悪しからず。



## 2 インストール

適当なディレクトリに実行ファイル‘Qfig.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、そのディレクトリごと削除してください。レジストリは何も触りませんが、ライブラリ‘.NET Framework’が必須です。これはオプションライブラリ（無料）なので

<http://www.microsoft.com/ja-jp/download/details.aspx?id=17851>

にアクセスしてダウンロードしインストールしてください。ちなみに、‘Visual BASIC 2010’本体等も無料ダウンロードできます。なお、ウィルスチェック後に‘Qfig.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。

ボタンの機能やオプションは英語（のようなもの）で表示してある一方で、各種補助説明等はデフォルトでは私の母国語（皆さん、癖があるとおっしゃいますので、九州弁が含まれているのかもしれませんが）で表示さ

<sup>1</sup> このスタイルファイルは私が作ったものなので、Qfig と一緒に配布。

<sup>2</sup> 東北大にネットワークが構築されたときはまだ MS-DOS の時代。その中で少々独特な、学生さんと教員を交えた一種ボランティアによる電子メール・掲示板システム TAINS bbms が開設されました。そしてあの JaWaT<sub>E</sub>X が渡邊さんによって開発公開され、掲示板でも T<sub>E</sub>X カテゴリーができ、理科系ばかりではなく経済学分野の人も交えた情報交換が盛んに行われました。そのころに開発し始めたものです。

<sup>3</sup> XP のときは、DOS 窓（コマンドプロンプト）でのもっと古い版の方が便利でした。その後 Win7 以降の DOS 窓で全画面表示ができなくなったので、早く改訂しようと思っていたのですが、オフラインの XP マシンで稼働してたこともあり、また諸々のこともあってなかなか着手できませんでした。

<sup>4</sup> 旧版では輪郭無しの網掛けを画面上で実現できなかったこともあり、拡張したスタイルファイル eepicsup.sty を用いた上で、L<sup>A</sup>T<sub>E</sub>X ファイルを編集することによって対処していました。

<sup>5</sup> 大学入試問題で使った図等の場合は、一旦描いたあとに生データを円周率等を使った正確な座標等に基づき書き直して用いたことがあります。生データ形式は DOS 版に入れておきましたが、今回それも明記しておくことにしました。

<sup>6</sup> 実行できる場合もありますが。



<sup>7</sup> ま、私以外でこれを使っている方はおられないでしょうが。

れます。留学生さん等で英語の方がいいなあという場合には、1 回以上 Qfig を実行した場合に（プログラムを置いたディレクトリに）できている ‘Qfig.ini’ ファイルの 3 行目を

```
Language = English
```


にして（日本語以外の ‘Klingon’ とか ‘Bajoran’ や ‘Romulan’ 等好きな言語を選んでいただき）立ち上げ直してください。英語には確実に訛りがあります。元に戻すには、ここを ‘Japanese’ に（スペルは間違えないように）するか、ini ファイルそのものを（万一のことを考えて改名する等して）一旦削除してください。ただし、プログラムの基本フォントを「メイリオ（明瞭?）」にしていますので、もしかしたらうまくいかないのかもしれませんが、周囲に日本語が使えないパーソナルコンピュータが無いので確認していません。そのため、フォントを「MS UI Gothic」にしたものも公開してありますが、残念ながら英語のマニュアルは永久に存在しません。

### 3 窓が二つ？

キーボード操作による描画のためにかなり長い間の紆余曲折がありまして、最終的にはコマンド窓（描画選択ボタン等が並んでいる）と描画窓（文字通り描いている図が表示される）および補助入力小窓（回転角設定等）が別々に<sup>8</sup>なっています。これによってようやく、キーボードによるスムーズな描画が実現<sup>9</sup>できました。またコマンド窓は、立ち上げたときには全面が表示されます。描画中はファイル操作が必要ありませんので、横に広がり過ぎると感じた場合には、窓中央にある  ボタン（じゃないです<sup>10</sup>が）を押すと右半分を隠すことができます。同じ場所に出現する  ボタンが右半分を復活します。

## 4 マウスとキーボード

### 4.1 マウス操作モード

立ち上げたときのデフォルトとしてはマウスの動きと左右クリックによって、作業種を指定したり、節点を定義します。以下その点を図形の定義点と呼ぶことがあります。描画窓での位置決めについては、通常は 1 mm ピッチ（画面の 4 ドットピッチ）での定義にしてありますが、 キーを押したままにすれば 0.25 mm ピッチ（画面の 1 ドットピッチ）で定義できるようになります。描画窓の左上に、カーソル位置座標が mm 単位で表示されているので参考にしてください。

描画中には、例えばあの「ペイント」や Libre Office の描画ソフトウェア等のように、クリックしたままドラッグするような操作の場面はありません。一旦クリックしたら指を離すことを前提にしてあります。そのため旧版でも、標準的な操作ではないというコメントをいただいたこともあります。私はマウス世代ではないのでこうなっています。また、メニューバーが無いのも標準的ではありませんが、メニュー方式があまり好きではないこと<sup>11</sup>はともかく、これは私の勉強不足が主な原因です。それから、マウスの左クリックと右クリックの意味が描画種や作業種によってバラバラ、つまり統一されていません。行き当たりばったりで作った旧版の操作をほぼそのまま受け継いであります。

<sup>8</sup> Windows という OS を使っている学生さんが何故全画面モードでいろいろな仕事をするのが理解できないでいます。

<sup>9</sup> 私はほぼ 100% キーボード利用です。

<sup>10</sup> どうやら矢印キーが効かなくなるのは、ボタンが原因らしいのです。それがたとえ disable されていてもなかなか動いてくれない。したがって描画窓にはボタンがなく、ラベルでボタンのような振りをしています。

<sup>11</sup> 多くのソフトウェアでも、ショートカットキーを定義したり、下のバーにアイコンを並べたりしてるじゃないですか。

## 4.2 キーボード操作モード

さて、キーボード世代<sup>12</sup>で、かつ私のような老人（還暦を既に迎えてしまいました）は手が震えて、マウスを用いた正確な節点定義ができないため、描画窓の中だけは、キーボードでもカーソル移動ができるようにしました。と言うより、それが本来の目的。描画窓右上の **K** ボタン<sup>13</sup>を押すと、描画窓 (25 mm, 25 mm) の位置にクロスカーソルが現れます。それ以降は、キーボードの上下左右のキーでカーソルが動くようになります。これも **Shift** キーを押せば 0.25 mm ピッチになりますが、マウスよりも移動が遅いため、より速く移動したいときには **Ctrl** キーを押したままにすれば 10 mm ピッチでカーソルが移動します。

旧版同様、マウスの左クリックは **[space]** キーに、右クリックは **Enter** キーに割り当ててありますので、このマニュアルの以下の説明に出てくる「左クリック」と「右クリック」という表現

左クリック	右クリック	<b>KILL!</b>	<b>Redraw</b>
<b>[space]</b>	<b>Enter</b>	<b>Delete</b>	<b>Ctrl+R</b>

は、**[space]** キーと **Enter** キーの押し下げに読み替えてください。キーボードを使っている間は、描画窓右上にある **K** ボタンが **M** ボタンになっているので、それを押せば通常のマウス操作状態に戻ります。種々の仕様のためか、開発途中（それが幻の Version 2 です、呵呵）カーソルが動かなくなる現象が頻発しました。その経緯もあって、キーの **H** と **J**, **K**, **L** もそれぞれ「左」「下」「上」「右」移動に使えます。昔のボックススペースが **Ctrl+H** であったことや、**T<sub>E</sub>X** で **^J** が改行であることから「左」と「下」は覚えられます。**L** は一番右にあるから? 呵呵。

さてキーボードで操作をしていると、キーボードからいちいち手を離してマウスを触るのが煩わしくなります。そこで旧版とほぼ同じように、ファンクションキーでも大半の作業ができるようにしました。キーボード操作モードにすると 12 個のファンクションキーの配置に機能を書き込んだ小窓が現れ<sup>14</sup>ます。この小窓はドラッグして描画窓の中だけは動かしますが、その小

左	下	上	右
<b>←</b>	<b>↓</b>	<b>↑</b>	<b>→</b>
<b>H</b>	<b>J</b>	<b>K</b>	<b>L</b>

窓が通過した場所にあった図形は一時的に消えますので再描画してください。残念ながら「輪郭無し」にするチェックボックスは含められていませんので、マウスを使ってください。またちょっと心配だったのと、外部の方が改善した旧版のような 3 行割り当てにするのが煩わしかったので、メモリのクリアも割り当ててありません。ただし **KILL!** は **Delete** キーに、**Redraw** は **Ctrl+R** に割り当てました。後者は旧版と同じです。

割り当てた作業種は上下 2 行になっていますが、一番左に赤い記号 **>** のある行の作業だけが実施可能です。上下の切り替えは、これも旧版と同じく **Esc** キーで行います。キーを押すたびに **>** が上下することでわかります。すべての作業状況はコマンド窓上と連動します。また、線の太さと属性および塗りつぶしの割り当てが上下両方にありますが、これは描画するときと編集するときの両方で使うことがあるために敢えて重複させてあります。旧版ではときどき **Esc** キーを押さないといけないことがあって、ちょっと不便でした。マウスの方がいいという方は、描画窓の右下の **F** ボタンを押すと小窓が消えてファンクションキーは使えなくなります。もう一度 **F** ボタンを押すとまた使えるようになります。

まだフローにフローがある<sup>15</sup>らしく、作業を続けているうちに仮に囲んだ矩形・移動中の図形やクロスカーソルが消えずに画面が汚れることがあるので、ときどき **Redraw** ボタンで再描画をしてください。それから、描画したあとに **Alt** キーあるいは **F10** キーを押すと、不思議なことに描いた図が消えてしまうこと<sup>16</sup>があり

<sup>12</sup> がしかし、年季が浅く、上下左右のカーソル移動に **H** や **J**, **K**, **L** を使うほどではありませんでしたが、長いことメールも vi で書いていました。そのため、Windows になっても DOS 窓作業や FD は手放せませんでした。って意味不明ですかね、もはや。

<sup>13</sup> 前述のように描画画面には各種ボタンを配置できないので、ラベルでボタンの振りをさせています。

<sup>14</sup> 何故かは不明ですが、このとき図形が消えることがあります。再現性が一定しないので未解決まま。

<sup>15</sup> なんだからねえ、ボタンを押した途端に裏返し色で描いたカーソルが消えたりするのは何故。グリッドの Paint もよくわからない。

<sup>16</sup> Visual BASIC の仕様でしょうか。この現象は一度起きたあとは再発しないようです。

ます。そのときも再描画してください。また、他のプログラムの窓や入力小窓をこの描画窓の上に置いたときも同様に消えることがありますので、そのときも再描画で対処してください。なお文字列入力等の各種小窓上では、キーボード操作はできません。

## 5 作業種の指定と解除

例えば **Lines**<sup>17</sup> を押して折れ線を描き始めると、必要なものを除くと、このボタン以外は使えなくなります。不便ですかねえ。前の版を作っていたときに、いつでも作業種を変更できる方がいいというご要望をいただいて変更しましたが、プログラムを作っていて訳がわからなくなった<sup>18</sup>ので、一つの作業を終わらないと次の作業には移れないようになっていました。よって、折れ線を必要なだけ描いて再度 **Lines** を押せば、他のボタンが押せるようになって次の作業に移ることができるようになります。これはキーボード操作モードでファンクションキーを使っている状況でも同じで、アクティブになっているファンクションキーを押してその作業を解除してください。また、細かい設定のための入力小窓が現れている間は、コマンド窓も描画窓もアクセスできなくなっています。なお、作業種や簡単な説明・メッセージは描画窓上方の細長い枠内に表示されます。

## 6 描画範囲

最終的には図のある領域だけがファイル出力されますので、図形の配置は相対的なものです。また、描画窓右端中央にある **U** ボタン（コマンド窓の **U** ボタンではないので注意）を押せば、描画している「用紙」を「上に」三分の一だけシフトできます。上下 2 画面分しかありませんが、これで、A4 サイズ（マージンを除いた縦 24 cm と横 20 cm で横が少し広め）を対象にできています。ただし、コピーや移動等で描画範囲からはみ出しても図にはなりませんから、いくらでも大きい図は作れますが、旧版の **Preview** はやめたために事前確認はできないので注意してください。なお、コピー・移動の作業中に「用紙範囲のシフト」はできませんので、二段階に分けてコピー・移動するか、グループ指定をした上で「用紙範囲のシフト」をし、そのコピー・移動を右クリックで実行してください。また **U** ボタンを押すと **D** ボタンに変わっていますので、再度それを押すと描画範囲が「下りて」きます。また、キーボードモードのときには **PgUp** キーと **PgDn** キーも使えます。何となく向きが逆でしょうか。

## 7 描画の仕方

### 7.1 折れ線 (Lines)

**Lines** で直線入力モードになります。なお、例えばあの「ペイント」や Libre Office の描画ソフトウェア等のように一線分を描くのではなく、複数の角点を持つ折れ線の定義操作になっています。以下の曲線等も同様です。まず左クリック（複数）で節点を定義していき、最後の節点を右クリックで定義すれば、一つの折れ線が定義できます。左のボタン群の下中ほどのところで（後述）線の太さと属性を設定できます。なお後述するように、節点はデフォルトでは 64 個までになっています。残り少なくなると頻繁にメッセージ窓が出ますので、一旦折れ線を確定して続きは別の折れ線にしてください。これは多角形や曲線・閉曲線も同様です。

<sup>17</sup> 単数の発音は [láin] です。決して [laín], つまりスマートフォンで使えるあのアプリケーション名のような発音をしてはいけません。

<sup>18</sup> 学生さんにはフローチャートや変数表を作ってからプログラミングするように指導しながら、自分は山勘でやっているので、プログラムがとても汚い。今回も、旧版のコードの意味がわからなくなり、そのまま Copy & Paste したルーチンがありました。トホホ

## 7.2 多角形 (Polygons)

**Poly** ボタンを押して折れ線と同じように節点を定義すれば、閉じた線になります。ただし、最後の右クリックで指定した節点が自動的に最初の節点と結ばれることになりますので、自分自身で最初の節点位置に戻ってから右クリックする必要はありません。2 点しか定義しなかった場合にはいい加減な場所にもう 1 点追加して定義されますので、あとで編集はできますが注意してください。線の太さと属性以外に、多角形の中の網掛けや塗りつぶしの選択（後述）ができます。

## 7.3 曲線 (Curves)

**Curves** ボタンで折れ線と同じように節点を定義すれば、その節点を「通る」ような、3 点ずつのパラメトリックな 2 次曲線で結ばれます（あまり綺麗ではありませんが、節点を必ず通るような他の計算法<sup>19</sup>を思いつきませんでした）。2 点しか定義しなかった場合にはいい加減な場所にもう 1 点追加して定義されますので注意してください。曲線でも線の太さと属性を選択できます。また **draw** 系の描画なので前述のように、マウスの左クリックドラッグによる曲線定義等はできません。

## 7.4 閉曲線 (Closed Curves)

**Clsd C** ボタンで多角形と同じようにして定義できます。これも 2 点しか定義しなかった場合にはいい加減な場所にもう 1 点追加して定義されるので注意してください。線の太さと属性および閉曲線内部の塗りつぶし設定もできます。

## 7.5 円と楕円 (Circles and Ellipses)

**Ellipses** ボタンを押して、左クリックで始めた場合は、そこと次の左クリックの場所とで定義される正方形に内接する円を定義できます。一方、2 回連続の右クリックで定義された長方形の中には、それに接する楕円を定義できます。線の太さと内部の塗りつぶしを選択できますが、属性は実線のみにしました。

## 7.6 円弧 (Arcs)

**Arcs** ボタンを押して定義を始めます。最初の左クリックで円弧の始点を、次の左クリックで終点を定義しますが、完成すると始点から終点までに時計回りの円弧が定義されますので注意して定義してください。2 回目の左クリックで、定義した端の 2 点が線で結ばれますが、小さい四角が付いている方が始点になっています。

この段階で、この 2 点の midpoint に下ろした垂線が表示され、マウスをあちこち<sup>20</sup>移動することによって垂線の長さが変化すると思います。この垂線の端が完成する円弧の中心点です。この状態における描画窓左上の座標値はこの中心点の座標になっています。この 2 本の線の配置と座標値を見て、最終的な円弧の状況を想像<sup>21</sup>しながら垂線の端（中心点）が欲しい位置に来たところで右クリックすると、円弧が確定されて描画されます。これも実線のみですが、太さは選択できます。ただし、**Ngraph** のような円弧や扇形の塗りつぶし<sup>22</sup>は単純にはできません。

<sup>19</sup> ‘**eeptic.sty**’ ではスプラインが定義されていますが、‘**Visual BASIC**’ のベジエ等での結果と異なるような気がしましたし、節点を通らないのでやめました。なかなかうまくいかないですねえ、世の中。

<sup>20</sup> どこに動かすと中心がどこに行くかについては、いろいろお試しあれ。遠くすると中心が遠くなるようでもないようですねえ。作った本人がよくわかってない。呵呵

<sup>21</sup> この版の **Visual BASIC** では円弧を裏返した色で描画する機能が無くなっているので、こんなことになっています。

<sup>22</sup> 塗りつぶし円や楕円を描いてから扇形等にするために白塗りの多角形を重ねれば、塗りつぶした円弧や扇形は描けます。

円弧の位置決めはとても難しいので、あまり正確な図は描けないと思ってください。当たり前ですが、中心が2点の midpoint にあるときの円弧は半円になります。精度を上げるためには、後述のように、あとで Qfig 生データファイルを直接編集してください。中心位置だけは後述の編集作業で変更できます。

## 7.7 箱つまり矩形 (Boxes; i.e. Rectangles)

**Boxes** ボタンを押すと、水平線と鉛直線でできた四角形を定義できます。続けた2回の左クリックが矩形の対角点になります。2回目は右クリック<sup>23</sup>でも同じです。線の太さと属性および塗りつぶしが選択できます。

## 7.8 矢印 (Arrows)

**Arrows** ボタンで、折れ線・曲線・円弧の端点にだけは矢印を付けることができます。左クリックで指定した端点に矢印が定義されることにはなりますが、もし既にそこに矢印が定義されている場合にはその矢印が消去されます。矢印が設定可能な節点を明示したい場合には、節点のマーカー表示機能（後述）が使えますが、動作が重くなることは我慢してください。さらに出力オプションにおいて、三角形あるいは2本の線で矢印を描くこともできます。ところで、直線や曲線の端の矢印が変な方向を向いたりしていませんか。これは、その端に複数の定義点が重なっていることが原因だと思われます。節点移動を何度か試してみてください。

ここでお詫びです。データの整数化をしてしまっていたことから、円弧端の矢印の位置がずれることがありました。生データそのものが整数化されていますので、気になる方は v.3.3.2 以降を用いて、当該円弧の対称コピーを利用して元円弧の消去と再配置をしたあと、矢印を付け直してください。すみませんでした。

なお旧版では、 $\text{\LaTeX}$  出力ファイルを用いて描いた太い線の矢印の矢の頭が尖っていませんでした。これは、矢があっても線を矢の先端まで引いたことから、太い線の端っこが矢印の矢をつぶしてしまっていたからです。この版では、矢印側の線の節点を矢印の重心点で置き換えてありますから、今までよりも綺麗に尖った矢になっているはずです。是非とも旧版のデータを読み込んで変更し直してください。

## 7.9 文字列 (Strings)

**Strings** ボタンを押す、場所を定義すれば文字列を挿入できます。描画窓で、配置したい文字列の一文字目のすぐ左上の場所にカーソルを合わせて左クリックすると、文字列入力小窓が表示されます。フォントの大きさと種類を選んで配置する文字列を入力したあと、**Set it!** ボタンで定義が完了します。あるいは **Enter** キーを押すことによっても確定されます。それで小窓は一旦消えますが、続けて別の場所で左クリックすれば、また小窓が現れて文字列が定義できます。文字を一文字も入力しなければ、そこには何も定義されません。また、入力小窓の下に位置していた図形が消えてしまうこともありますので、再描画してください。小窓はドラッグすれば移動でき、この小窓だけはその位置を覚えています。

この版の画面表示では、全角は ‘MS 明朝・ゴシック’ を使っています。半角は ‘Times’ に一番似ていた ‘PT Serif’ というフォントを指定してありますが、ボールドイタリックだけは、FontStyle 設定にそれが無かったので、MS ゴシックの英数字を使っています。もしかすると Windows でののは、ユーザー毎にインストールされているフォント種が異なっていて、代替フォントによる表示になってしまうのかもしれませんが、制御の方法がわからないので、以上のようにしてあります。‘Times’ の指定も試みたのですが、なんとなく変だったのでやめています。

ところで、文字の長さ（横幅）や高さについては、 $\text{\LaTeX}$  や PostScript の場合には一文字毎にその寸法が異なるといいます。そしてそれは、Windows のフォントのそれとは絶対に一致していないでしょうね。したがっ

<sup>23</sup> 最初の版では旧版に合わせるため、右クリックでは塗りつぶし選択になっていましたが、やめました。

て、描画窓に表示された文字列の配置と、最終的に  $\LaTeX$  化したときの配置や、PostScript ファイルでの配置とも微妙に（否、かなり）違ってきます。旧版では少なくとも英数字の横幅を Ngraph 用と  $\TeX$  用におおまかに定義していましたが、それでも表示は Windows のそのままにしていたから、描画窓があまり参考にはなりません。この版でも手抜きをしたため、文字の配置場所について場合によっては試行錯誤<sup>24</sup>が必要になるかもしれません。また全半角混じりの文字列を全角のみと半角のみを別々に入力すると、場合によっては重なったり離れすぎたりしますので、全半角が混じったまま定義した方がむしろいいと思います。 $\LaTeX$  の場合はそれで問題はありますが、PostScript の場合は全半角混じり文字列中の半角文字が全角文字のフォント種で表示されることにはなってしまいます。気になるようでしたら、別々の定義にした上で試行錯誤で位置決めしてください。

特に、描画面面上では文字列の左上の座標が定義点になるのに対し、 $\LaTeX$  と PostScript では左下の座標が定義点になります。この違いも、配置を正確にはできていない理由の一つです。ただ修正する場合に、例えば  $\LaTeX$  ファイルの `\put(x,y)` の数値を変更するのではなく、面倒でしょうが Qfig の生データ側で変更して再保存しておいた方が後々問題が生じない可能性が大です。面倒臭がらずに元ファイルを編集しましょう。

なお入力窓での文字の大きさ指定では '10pt' と '12pt' と表示されていますが、 $\LaTeX$  ファイルに変換したときは、前者は `\normalsize` になり、後者には `\large` が付きます。PostScript ファイルのときは明示的にサイズ指定してあります。したがって、10pt ではない  $\LaTeX$  文書で用いる場合には、指定文字サイズになっていないことには注意してください。どうしても指定サイズにしたいときは、 $\LaTeX$  ファイル出力時のオプションを利用してください。

また、数式や添え字等の  $\LaTeX$  のコマンドやマクロを入力しても Qfig はそれを処理しませんので、描画窓には入力した文字列がそのまま表示されます。もちろんそれを  $\LaTeX$  ファイルに変換すれば数式になりますが、画面上での位置合わせはできないわけです。私のやり方等は後述します。PostScript ファイルに出力したら、もう何がなんだか・・・となります。

## 8 線の属性指定等や節点マーカ表示について

### 8.1 線の太さと属性 (thickness and property)

コマンド窓左の  というボタンを押すと、線の太さを 3 種類で変更できます。'eepic.sty' で定義された `\thinlines`, `\thicklines`, `\Thicklines` に変換されます。PostScript の場合は、それなりの太さになっています。

またその下の 'Line Properties' のラジオボタンで、線の属性として実線・破線・一点鎖線・点線の選択ができます。一点鎖線はファイル変換のときのオプションで長い破線に変更できます。画面には実線・破線・一点鎖線・点線で表示<sup>25</sup>される（はずな）のですが、特に細い線ではこの区別がつかなくなるようなので、ラジオボタンのところに表示されている色で線を描いてあります。例えば青で示された線は最終的には一点鎖線（あるいは出力オプションで長い破線）になる、ということです。ちょっと不便ですかね。長い破線の利用については、ファイル出力のときのオプションで対処してください。v.3.3 からは 'eepic.sty' の `\dashline` 等を使わないようにしたので、一点鎖線等も `dvipdfmx` で問題無く pdf ファイルに変換できます。

<sup>24</sup> Try & Error と言う人が多いのは何故?

<sup>25</sup> 'Visual BASIC' のマニュアル（オンラインのみ）はとても探し難く、また例題も不親切なので、この破線で描くなんてことはなかなかわかりませんでした。本屋にもいいものがみつからないなあ。



## 8.2 網掛けや塗りつぶし (shading)

コマンド窓左の‘Shadings’が、最初は TransParent (透明) になっていますが、ここをクリックすることによって、薄い網掛け・中くらいの網掛け・濃い網掛け・黒塗りつぶし・白塗りつぶしのどれかを選択できます。これも‘eepicsup.sty’で定義された `\shade[0.x]` や `\blacken`, `\whiten` に変換されます。PostScript の場合もほぼ同様の濃さになります。 Shift キーを押しながらクリックすると、逆順に選択肢が現れます。オプションの‘No Outline’にチェックを入れると輪郭線の無い網掛け図形になり、このとき当然ですが線の属性には意味がありません。もし、 $\text{\LaTeX}$  出力オプションで‘eepicsup.sty’を使わない選択をした場合には、元の‘eepic.sty’で定義された濃さ指定の無い `\shade` が使われますので、網掛けの濃さは一種類になります。またその場合、輪郭線を無くすることもできなくなります。

なお、白黒塗りつぶし以外の「網掛け」を画面上で実現する方法がわからなかったのも、この選択窓に表示されている色で塗りつぶすことによって区別してあります。それから、 $\text{\LaTeX}$  出力の dviout での dvi ファイルの表示と dvipdfmx が変換した pdf ファイルの表現とで、TPIC の解釈が異なるらしく、前者では塗りつぶした領域の下の方の線や文字が見えたりするので注意してください。この描画窓に見えているものは最終的な pdf ファイルおよび PostScript ファイルの結果をほぼ表していると捉えていただいて結構だと思います。

## 8.3 節点のマーカー表示 (Marking)

文字列や曲線等の場合は定義点の位置がわからないかもしれません。そのときは、描画窓上部の m ボタンを押すと、現在指定してある作業種において選ぶことができる節点だけにマークが付きます。逆に言うと、作業を指定しない状態ではマークは付きません。マーキングは3種類あり、矢印のためのマーカー、節点の移動等ができる図形のためのマーカー、すべての図形へのマーキングです。

またマーカー表示はただでさえ重いのに、しょっちゅう書き直すようになっていきますので、あまり使わない方がいいでしょう。定義点がどこにあるかは慣れて覚えるか、後述の Obj # ボタンの右クリックの機能を利用してください。なんらかの手順の（私が予測できていない）作業をしたときに、マーカーが消えているにもかかわらず m ボタンの色が赤であるときには、もう一度ボタンを押して黒い文字にしておいてください。

# 9 編集の仕方

## 9.1 編集対象の選び方

この節では、一旦定義した図形の形や線の属性の変更、さらにコピーや回転等の作業を説明します。そのとき、編集対象図形の選択は、その図形の定義点をマウスでクリックすることで判断されます。円と楕円は中心と右端が定義点（旧版データを読み込んだときは中心と上端）で、箱は左上と右下が定義点です。文字列はその左上端です。定義点がわからないときは、マーカー機能を使うか、後述の Obj # ボタンの右クリック機能を使うと便利です。

その定義点を選び易いように、定義点からちょっと広めに離れた5ドット(1.25 mm)未満の距離でクリックしたときに、その対象図形を選択したものとみなしています。したがって、2.5 mm 未満離れたところに定義点を持つ異なる図形が複数あるときにその中間でクリックしてしまうと、複数の図形が選択候補になりますので、どの図形が編集対象なのかを選択するための確認メッセージが出ます。そのメッセージには図形番号（これを見てもわからないと思いますが）と図形の種類が表示された上で、該当する図形の輪郭や文字が赤くなります。それを見ながら、編集対象を一つ選んでください。選択距離を小さくして頻繁にはメッセージが出ない

ようにすることも考えましたが、マーカー表示がとても重いので、大き目の距離にしています。線分等のように定義点が明確な場合には、できるだけそのすぐそばでクリックしてください。

## 9.2 コピー (Copy)

**Copy** ボタンを押すと、図形のコピーを作れます。ある図形を定義している定義点あるいは節点を左クリックしてマウスをちょっと動かすと、細い線で描かれた図形が動くことを確認できます。ここの動作も重いですが我慢してください。その細い線をコピー先まで移動して左クリックすれば、コピーが作られます。閉曲線等で節点の位置がわからない場合には、マーカー表示を使うといいでしょう。このときは矢印もコピーされます。

また、そばに並んでいる複数の図形を同時にコピーするには、2回の右クリックでその図形群を囲むような矩形を指定すれば可能です。そうすると、その領域を表す矩形を移動することができるようになるので、その矩形をコピー先に移動したあと、また右クリックすると複数図形の同時コピーができます。一つの図形をコピーするときと同様、できればコピーしたい複数の図形の骨格くらいを表示しながらコピー先の位置を決めたいのですが、この版の **Visual BASIC** ではその裏返した色の複数の図形の再描画という動作がとても重くなるので、囲んだ矩形だけを移動させています。そのため、微妙な・正確なコピー先の位置設定はできないかもしれません。ヒント：矩形領域の左上と右下を定義するときに、目印となるような図形の節点ギリギリの場所を選ぶと、コピー先の位置を少しだけ決め易くなります。場合によっては、コピーしたくない図形もその領域に含まれてしまうこともあるので、あまり効率的だとは限りませんが、後述の例にある自動車の図形のように、たくさんの部品を集めて作った図のコピーには便利かもしれません。

さらに、グループが存在するときの右クリックは、グループのコピーとしてのみ機能し、右クリックによる領域コピーはできません。また、グループに指定されている図形の中の一つを左クリックでコピーすることもできません。どうしても必要なときはグループを一旦解散してください。

## 9.3 移動 (Move)

**Move** ボタンを押すと、コピーと同じ操作で図形の移動ができます。当たり前ですが、コピーとは異なり、元の位置の図形は無くなります。複数図形の同時移動やグループ図形の移動もコピーと同様です。

## 9.4 節点の移動と削除 (Node Modification)

**Node M** ボタンを押すと、直線と多角形・曲線・閉曲線の節点を個々に移動あるいは削除できます。移動したい場合には、その節点を左クリックで指定し、それを移動先まで動かして左クリックすれば再定義されます。削除したい場合には、その節点を右クリックすれば終わりです。ただ図形を保つための最小節点数は下回れません。後述の編集作業で、直線と曲線を入れ替えたりしているうちに、節点の移動や削除が必要になる場合があるので、この機能がついています。

## 9.5 回転 (Rotate)

**Rotate** ボタンを押し、直線と多角形・曲線・閉曲線の節点を選択すれば、その図形の時計回りを正とした回転コピーができます。回転中心となる節点を左クリックで選ぶと、角度設定小窓が現れます。±180度の範囲で回転できるので、10度ずつでよければ上下ボタンを押せば図を回転させながら、設定したい角度にまで回転できます。角度を手入力して **Enter** キーを押すこともできます。角度を決めたら **Set it!** ボタンで確定です。元の図形は残したまま、つまりコピーができますが、矢印はコピーされません。角度設定小窓を出したあと途中でやめるには **Cancel** ボタンを押してください。

ただし、回転させたい図形が多く節点を持つ場合（例えば後述の、楕円を閉曲線で近似した図形は 37 個の節点からできている）には、動作が重くなるので、もし目的の角度がしっかりとわかっている場合には手入力望ましいでしょう。あるいは上下ボタンで望ましい向きまで近づけてから、10 度未満の角の修正を手入力するという方法もあります。

円が回転できないのは、ま、いいとして、呵呵、楕円を回転させたい人はいるでしょう。そのときは後述の編集機能を使って楕円を一旦閉曲線に変換すれば回転できるようになります。箱の回転? うーむ。矩形を多角形として描けば回転できます。

## 9.6 対称コピー (Symmetric Copy)

**Sym C** ボタンを押したあとは、選択した節点を通る軸に対して対称に、直線と多角形・曲線・閉曲線および円弧をコピーできます。コピーしたい図形の任意の節点にカーソルを合わせて、左クリックすると上下反転のコピーが、右クリックすれば左右反転のコピーができます。この場合も矢印はコピーされません。

## 9.7 図形のグループ化 (Grouping)

離れた位置にある複数の図形をコピー・移動・削除をするときに使えると考えました。**Group** ボタンを押します。そのあと、図形の定義点を左クリックしていき、最後の図形の定義点を右クリックすることによって一つだけグループが形成されます。その都度、どの図形をグループ化したかのメッセージが出て当該図形が一時的に赤になりますが、最終的には描画面の右上に ‘One group is set!’ というメッセージが、解除されるまで表示され続けることで確認できます。なお、異なる図形のほぼ重なった定義点をクリックした場合には、両方がグルーピングされますので注意が必要です。その後、もう一度 **Group** ボタンを押すとグループは解散させられます。グループは一組しか定義できません。

グループが存在する状態で、そのグループ以外の一つの図形をコピー・移動・削除することはできますが、グループ化された図形のうちの一つをコピー・移動・削除することはできません。どうしてもそうしたい場合には、グループを一旦解散してください。また、グループが存在する状態で、領域指定のコピー・移動・削除はできません。同じ変数を使っているの、呵呵。

## 9.8 図形番号と節点の確認 (Object Number Display and Marking)

**Obj #** ボタンを押し、描画窓内で左クリックすると、そこに近い図形の番号と図形の種類が、近い順に三つだけ表示されます。あるいは右クリックをすると、そこから 2 cm 以内にある節点に 5 秒間だけマーキングします。後者は節点位置がわからないときの補助で、ちょうど短距離センサーみたいなものです、呵呵。節点が表示されている 5 秒間に注目したい図形（複数可）の節点をクリックすると、その図形が赤になって節点も赤い四角形になります。それを移動する等したい場合には、一旦 **Obj #** ボタンを押してこの機能を終了してから、移動なりコピーをしてください。画面が汚れるかもしれませんが、再描画で対処してください。

さて、左クリックによる図形番号表記の機能は、**L<sup>A</sup>T<sub>E</sub>X** ファイルや **PostScript** ファイルを直接編集したい（どちらのファイルにも図形番号と図形種類がコメントとして書かれています）とき等、どの図形が何番の図形番号を持つかを確かめたいときに使います。ただ、注意が必要です。

実は編集作業中に消去したはずの図形は、メモリからは消しておらず、その図形がスキップされて描画されないだけになって<sup>26</sup>います。その方が作業が楽だったからですが、消去した（はずの）図形は、**Qfig** 生データ

<sup>26</sup> ということは、同じ矢印位置で何度も左クリックし続けてしまうと、どんどんメモリの領域が取られてしまうので、どこかで破綻するかもしれません。

ファイルに保存したときだけに削除されますので、再読み込みしないとメモリ内の「整理」はできません。そこで、**LaTeX** あるいは **PostScript** ファイルに変換するときには、一旦生データの（プログラムがあるディレクトリの '**Qfigtemp.qfg**' ファイルに）保存・読み込みを実行するようにしました。したがって、そのあとに図形番号確認を行うと間違いがありません。

## 9.9 再描画 (Redraw)

**Redraw** ボタンを押せば、すべての図形が描き直され、描画領域の範囲の再計算が行われます。もしそのとき **Shift** キーが押されていればグリッドを消して再描画しますので、グリッドと重なった線等の確認ができます。元に戻すには、**Shift** キーを押さずに再描画します。ただし、キーボード操作中のグリッド非表示時、ファンクションキー機能窓を動かすと、何故かその下にグリッドが現れます。どうやら、立ち上げ時にグリッドを描くために **Paint** イベントを用いていることが原因らしいのですが、対処法も何もわからないので、気になるときは再描画を。

## 9.10 図形を消す (Kill)

**KILL** ボタンを押した上で、消したい図形の定義点を左クリックするだけで消すことができます。心配なので、1 度だけ確認メッセージが出る<sup>27</sup>ようにしました。メッセージには図形番号と図形種類が表示され、該当図形が赤い線や文字で表示されますので、それで確認してください。

コピーや移動と同じように、右クリックで指定した領域内の消去もできます。さらに、グループが存在するときには右クリックで消去できます。ということなので、グループに指定されている図形一個を左クリックで消去することはできませんし、領域消去もできません。

## 9.11 すべて消去 (All Clear)

**CLEAR** ボタンで完了です。一応、2 度確認メッセージが出ますが。

## 10 少し細かい編集 (EDIT)

細長い **EDIT** ボタンを押すと、以上の機能ではできなかったような図形の属性や寸法の変更に加えて、図形種そのものの変更ができます。以下、**EDIT** ボタンを押したあとの作業を、選択した図形毎に説明をします。変更以外の主な例を図 1 に示しました。

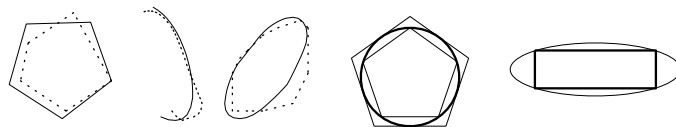


図 1 細かい編集例 — 点線や太い線が元の図形で、右クリック編集で変換された図形が細い実線；通常は元の図形は消えるので、この例では元の図形を同じ位置にコピーした上で点線で描いてあります

<sup>27</sup> 旧版の DOS 版では、私のソースコードをある大学やある会社の方に差し上げて実行可能なファイルに変換してもらったりしました。そのとき、オリジナルには無かった確認が増えていました。何故他の方をお願いしたかというと、私どもは国内シェアが一位のコンピュータを使っていたが、そこでコンパイルした実行ファイルは、実は同じ OS の当時 **DOS-V** と呼ばれた（世界共通の）マシンでは実行できなかったのです。そこで、多くの人に使っていただけるよう、**DOS-V** マシンでも実行できるようにコンパイルし直していただいたのでした。もうかなり昔の話です。

### 10.1 直線の変換 (Lines ↔ Curves)

直線の節点を左クリックすると、その時点で指定してある線の太さと属性を持つ曲線に変換されます。もし直線が2点でしか定義されていない場合には、中点に節点が追加されます。ここはわざと中点にしたので、見た目は直線のままですが、中点を動かせば曲線であることは確かめられます。こうしておかないと、また直線に戻るのが面倒になるので中点を選びました。つまり、もう一度左クリックすると、線の太さと属性を変更した元と同じ配置の直線になります。ということで、線の太さ等の変更にも使える機能です。右クリックでは何も起こりません。

### 10.2 多角形の変換 (Modification and inclined Regular Polygons)

多角形の節点を左クリックすると、その時点で指定してある線の太さと属性および塗りつぶしを持つ閉曲線に変換されます。もう一度左クリックすればまた多角形に戻ります。

右クリックすると、適当（いい加減）に傾いた正多角形に変換されます。確か、一番長い辺を持ち、元の多角形の角の数の正多角形に変換されるのではないのでしょうか。ソースを見てもわからず、もう忘れましたが、呵呵。ただこの版では、円を、それに内接する正多角形へと変換する機能を追加したので、それを回転させるという手もあります。そちらの方が大きさのある程度の精度で制御できるため、綺麗だろうと思います。

### 10.3 曲線の変換 (Curves ↔ Lines and inclined Half Ellipses)

曲線の定義点を左クリックすると、その時点で指定してある線の太さと属性を持つ折れ線に変換されます。もう一度左クリックすると、太さと属性を変更した曲線に戻ります。

右クリックすると、適当に傾いた楕円の半分に変換されます。「円筒」のようなものを描画するときに、これと対称コピーを使うと便利なので付けた機能です。この機能は他のソフトウェアには無いだろうなあと思っていますが、自己満足。

### 10.4 閉曲線の変換 (Modification and inclined Ellipses)

閉曲線の定義点を左クリックすると、その時点で指定してある線の太さと属性および塗りつぶしを持つ多角形に変換されます。もう一度左クリックすれば閉曲線に戻ります。

右クリックすると、適当に傾いた楕円に変換されます。ただこの版では、楕円を閉曲線で近似する機能を追加したので、それを回転させるという手もあり、きっとそちらの方が綺麗になるでしょう。

### 10.5 楕円と円の変換 (Modification and Regularization)

楕円と円の中心あるいは右端（旧版データでは上端）の定義点を左クリックすると、その時点で指定してある線の太さと塗りつぶしを持つ楕円と円になります。もし **Shift** キーを押したまま左クリックした場合は、線の太さと塗りつぶしを変更したあと半径の変更ができます。左クリックのあとのマウス移動で、破線矩形の形を少しずつ変更しながら、左上の点を固定したものとしての半径の変更をしてください。円は次の左クリックで、楕円は右クリックで確定されます。

図形選択のときに右クリックすると、円の場合はそれに内接か外接する、下辺が水平な正多角形に変換されます。このとき、内接・外接の選択をして角数を指定するための小窓が現れるので、そこで指定して **Set it!** ボタンで確定してください。三角形から 19 角形までが可能です。何故 19? もし右クリック時に **Shift** キーが押

されていた場合には、元の円を残したまま正多角形が追加<sup>28</sup>されます。楕円の場合は、近似した閉曲線に変換されます。どちらも回転できる図形種になるので、適切な方向に傾いた正多角形や楕円を作ることができます。

## 10.6 円弧の変更 (Modification)

円弧の端点を左クリックすると、その時点で指定してある線の太さを持った円弧に変更されます。また端点を右クリックすると、中心位置変更のための小窓が現れます。そこにある数値は元の円弧の半径に対する増減比です。それを変更していくと現在の中心点と変更先が表示されますので、また出来上りを想像しながら新しい位置を **Set it!** ボタンで決定してください。中心点の座標は、描画窓左上の座標欄にその都度表示されます。比は 2% から 300% までの範囲で設定できますが、何故か円弧の弦を越えた位置にまで移動できるので、そのときは必ずしもその比の半径にはなりません。どうしてでしょうねえ。呵呵。

## 10.7 箱の変換 (Modification)

箱の左上か右下点で左クリックすると、その時点で指定してある線の太さと属性および塗りつぶしを持つ箱になります。また円・楕円と同様、**Shift** キーを押したままの左クリックでは、続けて大きさの変更ができます。左上の点を固定したものとして、二度目の左クリックで大きさが確定します。

右クリックした場合には、箱の辺長比と同じ半径比を持ちそれに外接する楕円に変換されます。もしそのとき **Shift** キー<sup>29</sup> が押されていれば、元の箱も温存されます。

## 10.8 文字列の変更 (Modification)

文字列の左上の定義点を左クリックすると、文字列そのものとフォント種等が変更できます。ただし、この機能で文字列をヌルにした場合には、削除されず前のままになります。削除には削除機能を使ってください。右クリックでは何も起こりません。

## 10.9 やり直し (Undo)

旧版公開中に、途中の編集を覚えていてやり直しができるようにして欲しいとの要望をいただいたことがあります。確か、隣の研究室の学生さんだったかなあ。うーむ。どうしたらいいんだろう。なかなか難しいので、最後に行った編集の直前までは戻せるようにファイル一時保存をしておくことにしました。**EDIT** ボタンを押して編集している最中に **U** ボタンを押すと、編集を始めた状態に戻るようにはしておきました。プログラムがあるディレクトリの `'Qfigundo.qfg'` ファイル<sup>30</sup> に保存しただけですが、ちょっとだけ前進。

ただし、一旦 **EDIT** ボタンを押して編集を終えたあとは、一回だけ **U** ボタンを押すことができます。一応確認のメッセージは出ますが、編集前に戻しても戻さなくても、一度だけの機会になります。当然ですが、最後に行った編集すべてをしなかった状態に戻ります。

## 11 簡易マニュアル (HELP)

**HELP** ボタンを押すと、別の窓が現れます。左に並んだボタンを押せば、それに対応した作業の説明文が右の窓に表示されます。**EDIT** ボタンを押した場合にも同様の説明文が、さらにボタンで選択した図形毎に表示

<sup>28</sup> 塗りつぶしされている可能性もあるので、内接多角形の場合は多角形が、外接の場合は円が、大きな図形番号を持ちます。

<sup>29</sup> 最初の版ではこれで外接円になるのですが、やめました。また円の外接多角形と同様、箱の図形番号の方が大きくなります。

<sup>30</sup> これで `'temp'`, `'-bak'`, `'undo'` と、同じバックアップファイルが別々にできています。何かあったときにはどれかが役立つかも、と別々にしたわけではないのですが。

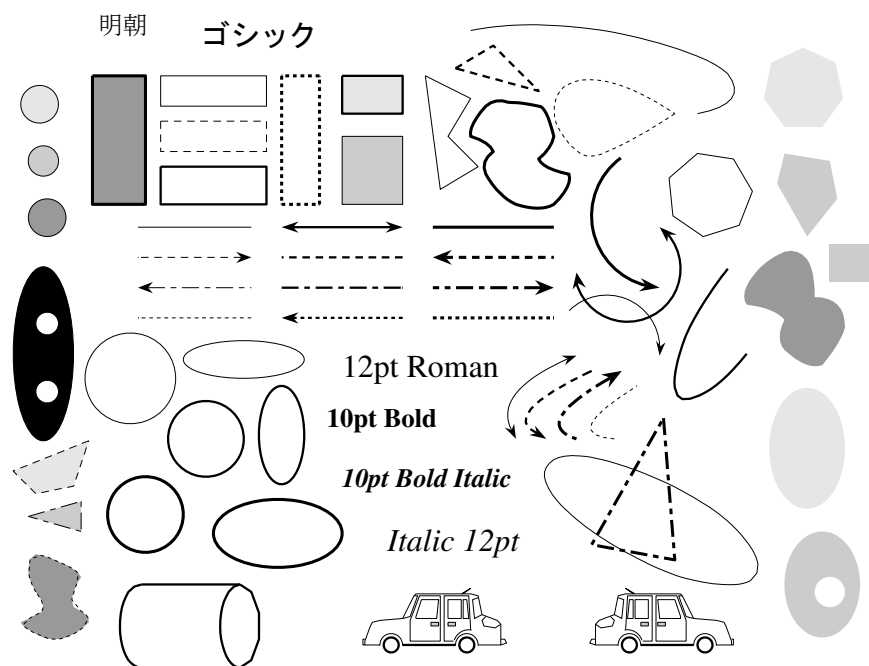


図 2 デフォルトの一点鎖線を選んだ場合のサンプル図

されます。元に戻るには再度 **EDIT** ボタンを押してください。ヘルプ窓そのものは右上の **×** で閉じることができます。

## 12 ファイル入出力 (File I/O)

### 12.1 保存 (Save)

Qfig 生データの保存は **Save** ボタンで実行されます。データの仕様については後述します。オプションとして縮小率を設定できますが、通常は縮小しなくてもいいような描画をそもそも心掛けるのが望ましいです。縮小率の上下ボタンでは数値は 5% ずつしか変化しませんが、小窓に直接数値を（正の整数で 100 以下の値を選んで **Enter** キーを押して）入力することもできます。ただし拡大はできません。

さらに、 $\text{\LaTeX}$  のファイルを読み込んで Qfig で描画する機能はありませんので、必ず生データでの保存をしてください。後述の 'Export' は一方通行です。生データの保存を忘れてしまった場合には、最初からやり直しになります（チャンスが全く無いわけではありませんが、後述）ので、十分注意してください。いわゆる、手抜きです。

### 12.2 読み込みとマージ (Load and Merge)

Qfig 生データの読み込みは **Load** ボタンで実行されます。旧版の Qfig 生データもそのまま読み込めますが、破線パターンが変更されますが、他にも微妙な違い<sup>31</sup>があるかもしれません。描画中の図形に追加したいときは **Merge** ボタンです。マージの場合にはマージ位置（元図の左上の原点をどこに移動するか）を設定してください。マージ位置については、いろいろ試してみてください。

保存時に縮小できるようにしたことと、マージができることは対になっています。例えば、ある図形部品を

<sup>31</sup> 旧の DOS 版のデータを持っているのは私くらいでしょうが、それとは文字列の互換性がありません。エラーにはならないので修正を。

少し丁寧に書く必要がある場合には、大き目のサイズで作図をしたものをそのまま保存しておくと同時に、それを縮小したものをマージ用に別途保存しておきます。前者はその後の修正等に対処するための予防措置です。後者を別の図にマージする場合には、その位置をうまく制御しないと既に描いた図形に重なってしまいます。それに対処するために、マージ位置を設定できるようにしてあります。試行錯誤が必要かもしれません。図2の下方の自動車の図がこのような縮小とマージを使って重ねた例です。自動車の作図では幅を10cm程度にして作業して、そのあと縮小したものをマージしました。自動車だけで20部品くらいだったでしょうか。

なお、もう一つ面白いオプションを追加しておきました。ただしマージではない場合にのみ可能です。それは、既に保存した図形群のデータを、その鏡像として読み込むものです。ラジオボタンで鏡像を選んでみてください。例えば、図2の下方の二台の自動車画像がそうやって作られています。ただし、文字列はその開始位置のみが鏡像位置に動き、文字列そのものは鏡像にはしていませんから、あれっ何だこれっ?になります。

さらに、この新しい版のQfig生データであれば、コマンド窓にファイルアイコンをドラッグ&ドロップすることによっても読み込むことができます。ただし、マージはできません。また旧版データも読み込めません。さらに、描画窓へのドラッグ&ドロップには対応していません。手抜きしました。

### 12.3 $\text{\LaTeX}$ ファイルへの変換 (Export)

$\text{\LaTeX}$  ボタンで、スタイルファイルの‘`eepicsup.sty`’を用いた $\text{\LaTeX}$ のソースファイルに変換・出力されます。図2には、旧版で配布していたサンプルに全角の文字列を追加したものを示しておきました。ラジオボタンで「一点鎖線」か「長い破線」かを選べますが、通常は下のボタンの「一点鎖線」を選べばいいでしょう。

$\text{\LaTeX}$  ボタンを押すときに $\text{\text{Shift}}$ キーを押したままにすると、次のようなオプションを選択できます。

- ‘`eepicsup.sty`’を使わないようにできます。このときは、網掛けの濃さは一定で塗りつぶし等には輪郭がつきます。したがって、太めの線の塗りつぶし矢印の先は尖りません。
- 図3のように、矢印を三角形か線のみにすることができます。
- 10pt ではない文書に 10pt, 12pt で文字列を表示することができます。

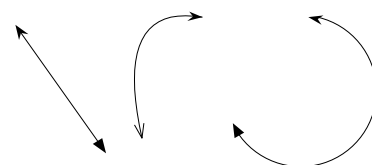


図3 矢印三様

このオプションを別窓で指定して $\text{\text{Make it so!}}$  ボタンを押すと、 $\text{\LaTeX}$  ボタンが赤文字になっていますので、再度このボタンを押して保存してください。

### 12.4 PostScript ファイルへの変換 (Export)

$\text{\text{PostScript}}$  ボタンで一応はPostScript言語で書いたファイルに変換・出力されます。インターネット情報を見ながら初めて触った言語なので、ひょっとするとうまくいかないことがあるかもしれません。とりあえずはサンプルの変換には成功していて、図4がそれです。漢字コードをもっとうまくやろうと思いましたが、どうせWindows上でのソフトウェアなのでShift-JISのままにしました。ま、その方が、直接PostScriptファイルの編集も楽ですからね。もちろんこのファイルは単なるテキスト形式（エディタで読める）です。

文字列のうち英数字はTimesのシリーズを使うようにしています。このフォントをpdfファイルに埋め込むことについては後述します。また、数学記号はイタリックで書かざるを得ないですが、積分記号等の数式を入力する泥臭い方法についても後述します。画面表示では荒い表示に見えますが、印刷するとそこそこの結果になるようです。



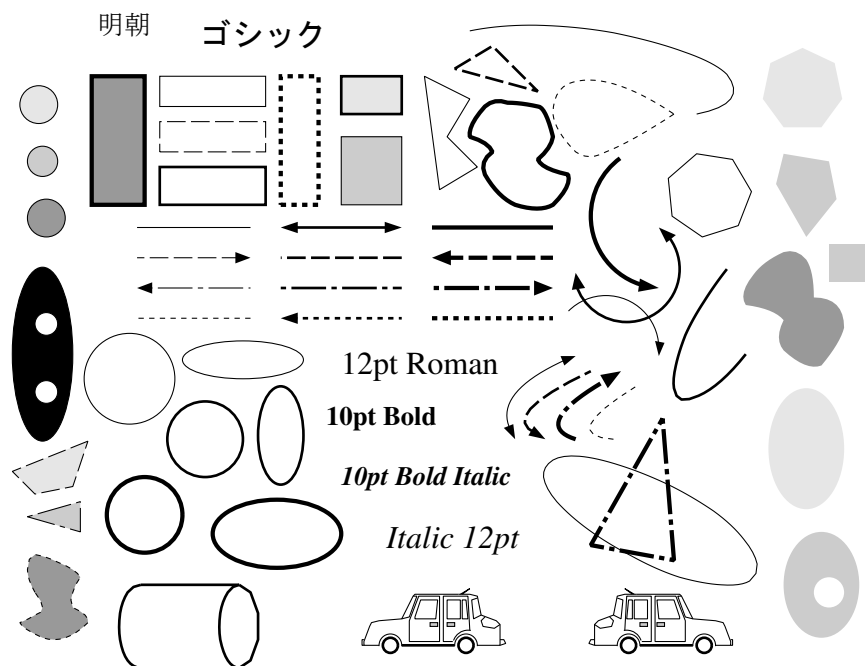


図 4 PostScript 出力の例 — 一点鎖線と三角矢印を選んだ場合

また、**PostScript** ボタンを押すときに **Shift** キーを押したままにすると、今のところ一つだけオプションを選択できます。

- 矢印を三角形にするか、線のみにすることができます。

このオプションを別窓で指定して **Engage!** ボタンを押すと、**PostScript** ボタンが赤字になっていますので、再度このボタンを押して保存してください。

## 12.5 Ngraph 用 gra ファイルへの変換 (Export)

**Ngraph** ボタンで、Ngraph for Windows (v.6) の gra 形式ファイルに変換出力されます。もし **Shift** キーも同時に押し下げてあれば DOS 版 v.5 の Ngraph データになります。このとき

- v.5 版の場合で塗りつぶしができる多角形は 10 角形までで、閉じた曲線や楕円は塗りつぶせません。
- Ngraph 上で、塗りつぶした図形の輪郭は無くなります。ただし、v.6 の箱だけは輪郭が付けられます。
- 保存時の一点鎖線・長破線オプションは反映されます。

例を図 5 に示しました。

## 13 作業終了！

ボタンがありません。コマンド窓右上の **×** で終わってください。確認はありませんので、自己責任ということで（後述）よろしくお願いします。また、複数窓を開いていることもあって、最小化ボタンも消していました。

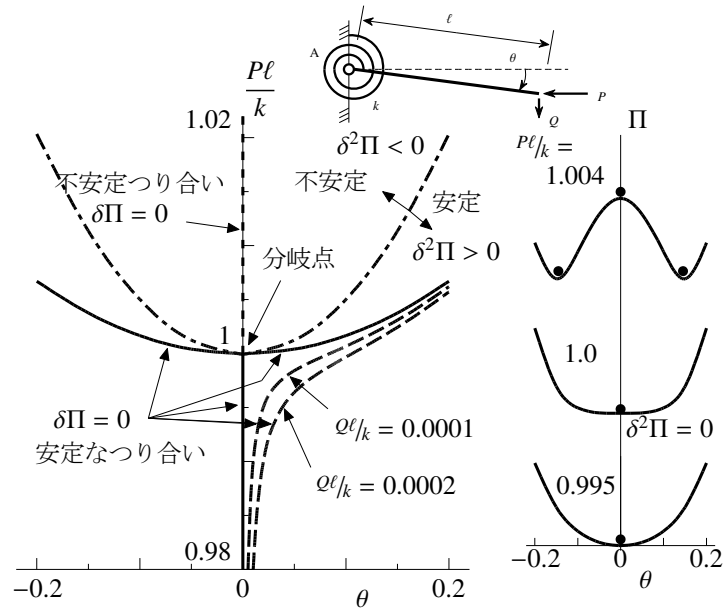


図 5 Qfig で挿絵を描いて Ngraph でグラフと合成して Gra2Epic で変換した図

## 14 Qfig 生データそのものの編集

データ仕様は後述してありますので、例えば周期パターンで同じ図形をたくさん並べる場合等は、別途 FORTRAN 等で Qfig 生データを作ることも可能ですし、その方が速いはずです。一つだけ図形を Qfig で作成して、その生データを眺めれば、データ仕様の意味がすぐにわかると思います。さてここでは、データを少し精密にすることだけを書いておきましょう。

例として「円弧」を描いてみてください。x 軸は水平右向きで、y 軸は鉛直下向きなので、例えば mm 単位で、(80, 50) の位置から (100, 70) の位置に (80, 70) を中心とする半径 20 mm の円弧を描くことにしましょう。キーボード操作モードを使えば端点は非常に精度よく定義できますが、円弧の中心点はさほど正確には定義できません（マウスカーソル位置をプログラム側から設定できないので）。したがって、生データは

```
6 4 2 -1 0 0 0
320 200
319 281
400 280
81 4.724734
0 6.270841
```

のようになってしまいます。このデータ仕様については後述しますが、この例では画面のドット単位で、中心が (319, 281) で、半径 81, 4.724737 rad の角度から 6.270841 rad の角度までの円弧になっています。つまり、水平軸から時計回りに 270.71 度の位置から 359.29 度の位置に、中心 (79.75 mm, 70.25 mm) の半径 20.25 mm の円弧を描こうとしたことになっていて、中心座標と角度・半径が正確ではありません。そこで生データをエディタ等で直接編集して（図形番号を確認すれば編集対象はすぐにわかります）

```
6 4 2 -1 0 0 0
320 200
320 280      <-- 円弧の中心を正確にし
```

```

400 280
80 4.712389 <-- 半径を正確にした上で、270 度も正確にし
0 6.283185 <-- 360 度も正確にした

```

のように修正すると、正確な円弧になります。もちろん、画面上や  $\text{\LaTeX}$ ・PostScript ファイルの出力では、この二つの図形はほぼ重なりますが、入試問題等の場合にはできるだけ修正した方が望ましいので、手間を惜しまないようにしてください。そんな大事な仕事にこんな自作のフリーソフトウェアを使うなんて!? 私は使いました。ごめんなさい。

## 15 $\text{\LaTeX}$ ファイルそのものの編集と利用

### 15.1 必要なパッケージ

出力したファイルを使う場合には、オプションの標準スタイルファイル ‘`epic.sty`’ と ‘`eepic.sty`’ に加えて、私が作った ‘`eepicsup.sty`’ が必要になります。輪郭無しの網掛けと `\shade` の濃さの選択を諦めれば、これは不要です。例えば ‘`QfigData.tex`’ に図形データが出力されている場合には

```

\documentclass{jarticle}
\usepackage{epic,eepic,eepicsup}
\usepackage{graphicx} % 横向きの文字列にしたいときのみ必要
.....
\begin{document}
さて・・・
\begin{figure}
\mbox{\hfill\input{QfigData.tex}\hfill\mbox{}}
\caption{僕が描いた絵}
\label{fig:mypicture}
\end{figure}
.....
\end{document}

```

のようにします。私のような NTT  $\text{\LaTeX}$  派（少数派）の方<sup>32</sup>は、あそこにハイフンが入ります。p.31 に、講義ノートで使っている例をいくつか載せておきました。

### 15.2 もう少し太くしたい

ところで、`\Thicklines` がもう少し太いといいと思いませんか。通常は私も元の定義のままで使っていますが、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

```

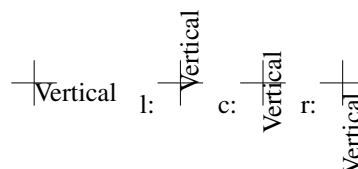
<sup>32</sup> このマニュアルも NTT  $\text{\LaTeX}$  でコンパイルされていますので、 $\text{\LaTeX}$  派の多くの方には違和感を持たれてしまうかもしれませんが、私にはこの方が読み易いのです。単なる慣れでしょうけどね。

```
\@gphlinewidth\@wholewidth \divide\@gphlinewidth 4736\relax}
\makeatother
```

のように`\Thicklines`を再定義してみてください。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。ただしこの文書ではオリジナルままの太さです。

### 15.3 横向きの文字列にしたい

例えば図面の縦向きの寸法線には横向きの数値を書くことになっているかもしれません。このような文字列の回転は、‘`graphicx`’パッケージの回転を使うと実現できます。例えば



```
\put(340,200){\normalsize Vertical}}
```

といった部分を

```
\put(340,200){\normalsize \rotatebox[origin=x]{90}{Vertical}}}
```

とするだけです。ここに`x`を`l`, `c`, `r`としたのが上の例です。クロスが`Qfig`上の文字列の定義点（この例の場合は座標(340,200)）なので、これを参考にして位置合わせをしてください。ただし、‘`graphicx`’パッケージでは正しいドライバオプションを選ばないといけません。例えば`dvipdfmx`を用いている場合

```
\usepackage[dvipdfmx]{graphicx}
```

としなければならないかもしれません。これが適切でないと変な線が発生することがありますので、そういう状況になったときはこのオプション指定を思い出してください。

### 15.4 図がちょっと小さすぎたので拡大したい、あるいはその逆

これも直接`LaTeX`ファイルの頭にある`\unitlength`を変更すれば、いくらでも拡大縮小できますが、お勧めしません。でも、ある図をある図の中の説明に使う、なんてときには、この方法が楽でしょうね。どうやって重ねるかって? うーむ。ただし、文字列との配置のニュアンスは変わってくるので、文字サイズも変更しないとイケなくなるかもしれません。

### 15.5 文字列に数学記号を入れたい

`Qfig`作成画面上では数式を表示できないので、文字列に数式を入れても画面では確認できません。ただしファイルにはそのままの文字列が出力されるので、例えば`\alpha`等と入力してあれば、`LaTeX`に変換・保存したあとでコンパイルすると‘ $\alpha$ ’になっちゃいます。私の場合は、文字幅のことを念頭に置いた適当な文字（例えば‘`al`’とか）をとりあえず入力しておいてファイル出力し、あとでその`LaTeX`ファイルを直接編集して数式（例えばこの`al`を`\alpha`）に書き直しています。こうすれば、普段`LaTeX`文書を作成しているときのフォント種のための図にできますから、本文中のフォントと同じものが図にも現れます。

### 15.6 オプションパッケージを使わない方法

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

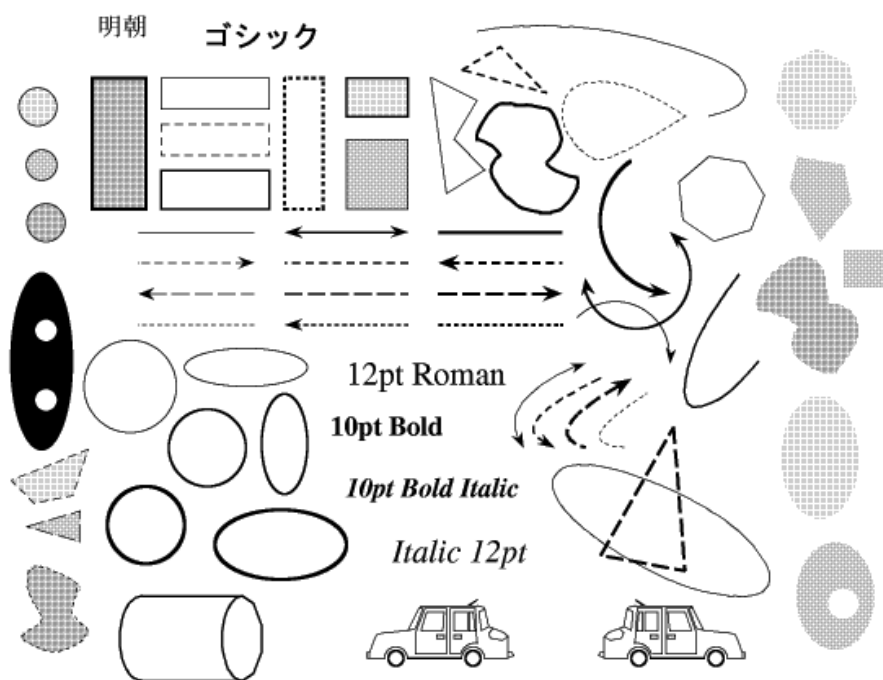


図 6 長い破線を選んだ dvi から dviout で ps 画像を取り出した例

- 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。しかし、もしこれを止めたい場合には文書のプリアンブルに

```
\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。網掛けが一種類しか使えず輪郭線も出ますが、これでオプションパッケージは排除できます。

さて、Elsevier のオンライン投稿システム等ではオプション・スタイルファイルの利用は禁止されています。多分 ‘eepic.sty’ までは標準だとは思いますが、試したことはありません。実は私は、図を単純な PostScript 画像に変換して使っています。まず前述のような方法で数学記号を入れた図を含んだ dvi ファイルを作ります。それを dviout で読み込んで、[Option]→[Setup Parameters]→[Display] とたどって H-scale と V-scale の数値を小さくした上<sup>33</sup>で ‘OK’ を押します。そして、再度メニューバーに戻って、[File]→[Save as image]→[File] とたどって、例えば ‘qfig.bmp’ という画像ファイルに、その図のあるページだけを保存してください。あとはこのファイルの中から図の部分、例えば IrfanView 等でトリムして、最後に Gimp 等で ps にすれば終わりです。解像度をどうしたらいいかは、試行錯誤が必要かもしれません。ちょうど latex2html のように、Qfig の中からコンパイル・トリム・読み込みができれば、このようなことは不要にはなるのですが。図 6 が、「長い破線」の場合の図をこの方法で ps 画像にした例です。

<sup>33</sup> 小さくすると解像度は上がりますが図そのもののサイズが大きくなります。

## 15.7 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。最近では論文集等で、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が、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。

## 16 PostScript ファイルそのものの編集と利用

### 16.1 必要なパッケージ

例えば ‘graphics.sty’ あるいは ‘graphicx.sty’ を使います。ps ファイルそのものは

```
\includegraphics*{qfig3ple.ps}
\includegraphics*[width=3cm]{qfig3ple.ps}
```

のようにして読み込みます。サンプルを参照してください。これ以外にもパッケージがあると思います。私も学生さんの真似をして何か別のものを使っていたことがありますが、例えば ‘dvipdfmx’ のサポート等によって是不具合が生じます。これについては回りの人に聞いてください。

### 16.2 横向きの文字列にしたい

横向きにしたい場合には、例えば

```
/Times-Roman findfont 10 scalefont setfont
100 60 moveto
(Vertical) show
```



となっているとすると、それに回転を加えて

```
/Times-Roman findfont 10 scalefont setfont
100 60 moveto
90 rotate
(Vertical) show
```

とするだけです。クロスが Qfig の定義点（この例の (100,60) という座標）です。ただし、回転させた文字列の部分がすべて BoundingBox 内に表示されません。

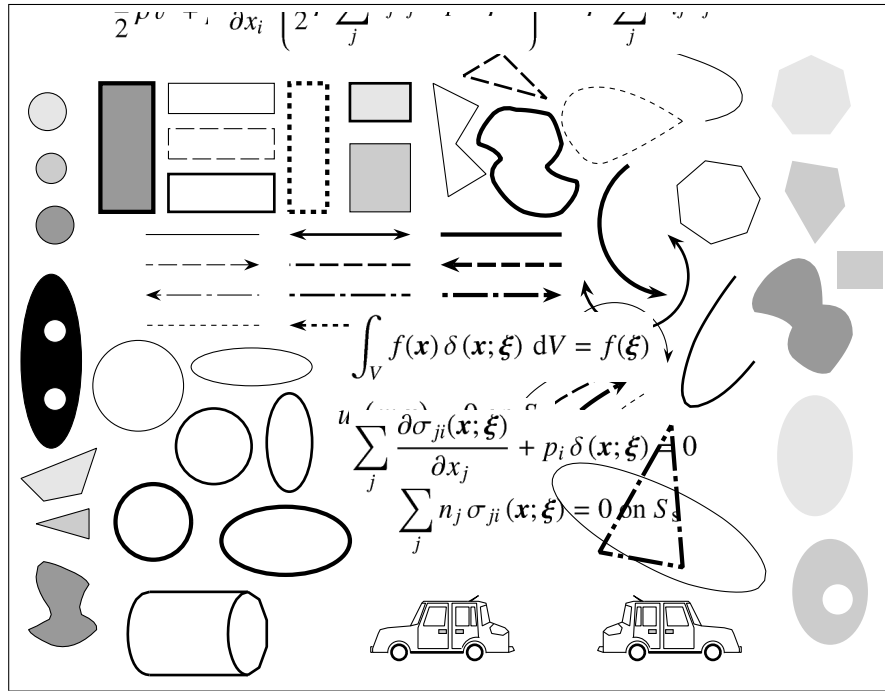


図 7 PostScript 出力に数式をいくつか入れた例；枠線がほぼ BoundingBox

### 16.3 文字列に数学記号を入りたい

一文字ならイタリックフォントで誤魔化すという手はあります。うーむ。さて式等はどうしましょう。ちょうど latex2html のように、Qfig 内部から L<sup>A</sup>T<sub>E</sub>X コンパイル・トリム等ができれば（できるのでしょうか）いいのですが、泥臭い方法を思いつきましたので、以下に書いておきます。数学記号部分を画像 (image) としてファイルに挿入する方法です。

1. まず挿入したい数式表現を L<sup>A</sup>T<sub>E</sub>X で作成し、その該当する式の部分だけを第 15.6 節で述べた方法で画像にした式の ps ファイルにしておきます。数式の横の長さは L<sup>A</sup>T<sub>E</sub>X の dvi を見てできるだけ正しい寸法 (Gimp で保存するときの横幅設定) で ps ファイルに変換し、例えば 'math1.ps' に保存します。
2. 次にその ps ファイルをエディタで読み込んでみると、例えば Gimp の出力の場合は

```
%!PS-Adobe-3.0
.... 10 数行の省略
% Translate for offset
0 0 translate
% Translate to begin of first scanline
0 25.734760683004513 translate
113.38582677165356 -25.734760683004513 scale
% Image geometry
445 101 8
% Transformation matrix
[ 445 0 0 101 0 0 ]
currentfile /ASCII85Decode filter /RunLengthDecode filter
%%BeginData:          6861 ASCII Bytes
```

```

image
JcC<$JcE[gJcC<$JcE[gJcC<$Jc.....
JcG9?i;X$2!!"IFs8QX.!!&mIs+13$s763LrrA.....
ここから画像データが並ぶ....
.....
%%EndData
showpage
%%Trailer
end
%%EOF

```

のようになっています。そして、% Translate to begin ... の行より前と %%EndData よりも後ろの部分削除します。したがって、% Translate to begin ... で始まって %%EndData で終わるファイルになります。ここでは簡単のために、ファイルを編集するような説明をしていますが、この切り出した部分をコピーするだけですから、慣れてしまえば、実際には編集する必要はありません。

3. Qfig で作成する図では、その数式を配置する予定位置に例えば嘘の文字列 'Eq(1)' を定義して、Qfig に PostScript ファイルを出力させます。すると、その文字列の部分は（図形番号や文字列で検索してください）例えば

```

gsave
/Times-Roman findfont 12 scalefont setfont
100 200 moveto
(Eq\ (1\)) show
grestore

```

のように書き込まれています。これを

```

gsave
% /Times-Roman findfont 12 scalefont setfont
100 200 translate
% (Eq\ (1\)) show
grestore

```

のように変更してください。'moveto' の座標に 'translate' させ、他の情報をコメントアウトしました。'gsave' と 'grestore' には触れないでください。

4. 以上の準備の上で、先ほど作った 'math1.ps' の「中身」を、この 'grestore' のすぐ上に挿入してください。つまり

```

gsave
% /Times-Roman findfont 12 scalefont setfont
100 200 translate
% (Eq\ (1\)) show

```



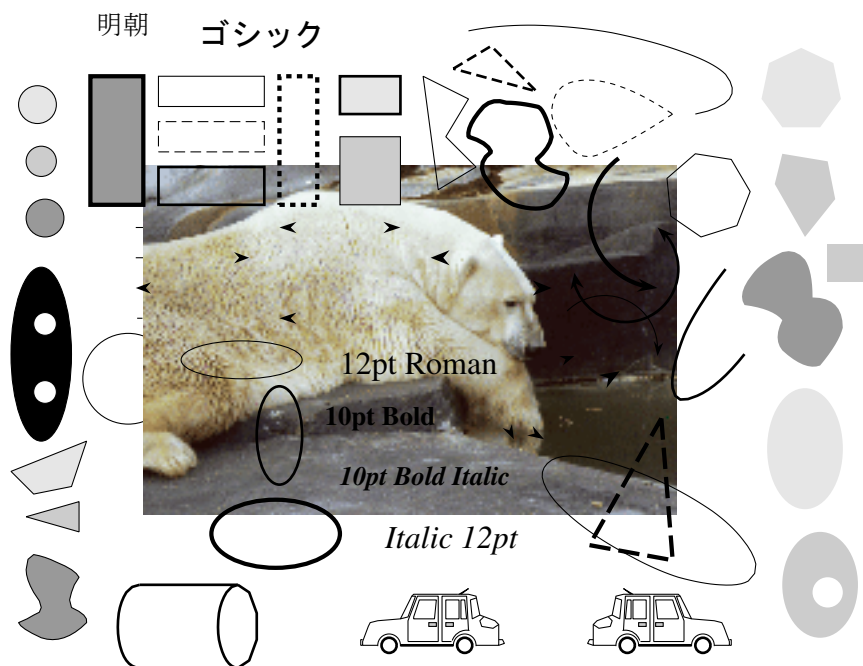


図 8 PostScript 出力に写真を入れた例

```
% Translate to begin of first scanline
0 25.734760683004513 translate
113.38582677165356 -25.734760683004513 scale
% Image geometry
..... 省略
..... そのあとに画像データが並ぶ.....
.....
%%EndData
grestore
```

のようになります。これで、画像にした数式が PostScript ファイルに挿入できたことになります。

図 7 がその例で、6 箇所の文字列のところに画像にした式を配置したものです。上方の式は **BoundingBox**（枠線の若干内側）からはみ出たため表示できていません。また中ほど下の数式も、文字列を定義した順番に定義していきますから、式同士が重なっているところがあります。面倒ですが可能だということです。ということは写真も・・・という例が図 8 です。左下にあった太い円の中心に、シカゴの **Brookfield** 動物園にいたころの私、否、私と同じくなまけものの白熊の写真<sup>34</sup>を配置した例です。矢印だけが線と分離して並んでいるところを見れば、図形の順番のこと等がわかる面白い例です。呵呵。

<sup>34</sup> この 'Qfig.exe' のアイコンは、この熊の顔の目から上の部分です。決して、いやらしい写真とかおじいさんの顔ではないのです。

## 17 多分どうでもいいこと

### 17.1 たくさんの図形が必要

デフォルトでは 2048 個の図形までしか描けず、また一つの図形の節点は 64 個が最大になっています。それ以上の節点が必要な場合には二つの図形で誤魔化すことが一つのアイデアです。この限界に近づくとメッセージが出るようになってるので、すぐに保存して作業をやめましょう。がしかし、両方共増やすことは可能です。旧版でもそうでした。プログラムを置いたディレクトリに 'Qfig.ini' というファイルができています。なので、そこの設定の数字（配列を 0 から始めることができるため、どちらも 1 だけ引いた数値になっています）を増やせば対応できます。ただ、この ini ファイルを消してしまったり、最大値を減らしたりすると、たくさんの図形を定義したデータファイルの読み込みに失敗しますから、一旦増やしたら、そのまま使い続ける方が安全です。

### 17.2 描画の途中誤って Qfig を終了してしまった

終了した時点での描画の Qfig 生データが、プログラムが置いてあるディレクトリに 'qfig-bak.qfg' という名前のファイルに保存されています。ただ、終了する度に重ね書きされているので、最後の結果のみしか手に入らないので、やはり注意が必要です。

### 17.3 ワードプロセッサ等に挿入したい

ま、そんな要望は無いとは思いますが、PostScript ファイルの頭で引用しているネット（懐かしい小波先生のところ）の情報から、例えば GhostScript が c:\gs\gs9.04 にインストールされているとして

```
c:\gs\gs9.04\bin\gswin32 -q -dEPCrop -dSAFER -dBATCH -dNOPAUSE
-sDEVICE=png16m -r240 -dText -dGraphicsAlphaBits=2 -sOutputFile=%1.png %1.ps
```

と（本当は一行）記述した 'ps2png.bat' というバッチファイルを作っておけば

```
ps2png qfig3ple
```

で、ps ファイルが png ファイルに変換されます。これは 1 ページ分の画像で、図そのものは左下に配置されています。あとはこのファイルから IrfanView 等で図の部分だけを切り出して他のソフトウェアに挿入することができます。png 形式をサポートしていない場合にも IrfanView 等で他の形式に変換できます。あるいはもっと簡単に GSview で読み込んで、png16m 形式の 240 ドットで変換しても同じ結果です、多分。

### 17.4 旧版にデータを渡したい

旧版では 'curves.sty'<sup>35</sup> や 'eepicemu.sty', 'eclepic.sty' 等のスタイルファイルや、P<sub>CT</sub>E<sub>X</sub><sup>36</sup> にも対応していました。もしそれが必要な場合には、この版の生データファイル 1 行目の

```
% Output of qfig.bas [Visual BASIC 2010]
```

の鍵括弧あたりを削除して、旧版のヘッダである

```
% Output of qfig.bas
```

に変更してください。これで読み込めるはずですが。一部、網掛けや塗りつぶしは無視されますが。

<sup>35</sup> 今は配布されていないような。 \besier が重複定義されるため、エラーが出ます。

<sup>36</sup> P<sub>CT</sub>E<sub>X</sub> は古いのかも知れません。パッケージ読み込み前に \def\fiverm{\tiny} 等が必要になるようです。

表 1 現バージョンにおける図形の基本情報

	情報 1	2	3	4	5	6	7
直線	1	node	node	node	thick	ltype	arrow
多角形	2	node	node-1	node-1	thick	ltype	fill
曲線	3	node	node	node	thick	ltype	arrow
閉曲線	4	node	node-1	node-1	thick	ltype	fill
円	5	2	1	-1	thick	0	fill
円弧	6	4	2	-1	thick	0	arrow
楕円	7	2	1	-1	thick	0	fill
箱	8	1	1	-1	thick	ltype	fill
文字列	10	klen	0	-1	point	文字種	全半角
矢印	11	2	-1	-1	thick	object#	node#

### 17.5 Qfig 生データ仕様

ファイルの 1 行目は Qfig データであるかどうか、旧版のそれであるかどうかの判定に使うコメント行です。次の行が、描画された領域の  $x$  座標の最小最大値と  $y$  座標の最小最大値になっています。それ以降が各図形毎のデータになっています。ファイル中では、ある行（下の例では A）に整数 7 個が並んでいるはずです。この行が各図形の最初の基本データになっています。

```

1 2 2 2 1 2 0 <--- A
56 216 <--- B
72 224
64 240

```

そのうしろ（上の例では B 以下）に 2 組のデータが複数行並んでいます。これが図形の座標値等の情報です。

まず A は各図形の基本的な情報で、表 1 にその中身を示しました。表 2 には旧版の情報を書いておきましたが、箱以外の閉じた図形に網掛けをしなかったことが主な変更点です。また、旧版の網掛けには塗りつぶしはありませんでした。上の例の各図形の 1 行目の整数の情報の説明は左から順に以下の通りです。

**情報 1:** 図の識別番号（旧版は網掛け箱が 9 でした）

**情報 2:** 座標等の情報最大数、つまり B に並ぶ行数 -1。1 だけ引かれているのは、配列が 0 から始まるから。文字列の場合はその文字数 klen（バイト数ではない）。たいていは節点数 node。

**情報 3:** 編集時にマークすべき節点数（コピーや移動等）

**情報 4:** 編集時にマークすべき節点数（回転や対称コピー等）

**情報 5:** thick は線の太さで、0=\thinlines, 1=\thicklines, 2=\Thicklines に対応。PostScript の場合は 'x setlinewidth' で x は 0.4, 1.0, 1.6 です。文字列の場合はフォントサイズ point で 10 か 12。

**情報 6:** ltype は線の属性で、0= 実線, 1= 破線, 2= 長い破線, 3= 点線。それぞれの設定はサンプルファイルを参照のこと。文字列の場合は文字種で、矢印の場合はそれを付けた図形番号。文字種は、0= 明朝と roman, 1= ゴシックと boldface, 2=italic, 3=bold-italic。

表 2 旧バージョンの基本情報

	情報 1	2	3	4	5	6	7
直線	1	node	node	node	thick	ltype	0
多角形	2	node	node-1	node-1	thick	ltype	0
曲線	3	node	node	node	thick	ltype	0
閉曲線	4	node	node-1	node-1	thick	ltype	0
円	5	2	1	-1	thick	0	0
円弧	6	4	2	-1	thick	0	0
楕円	7	2	1	-1	thick	0	0
箱	8	1	1	-1	thick	ltype	0
網掛け箱	9	1	1	-1	thick	0	fill
文字列	10	klen	0	-1	point	文字種	0
矢印	11	2	-1	-1	thick	object#	node#

情報 7: fill は網掛けパターンで,  $1=\backslash\text{shade}[0.1]$ ,  $2=\backslash\text{shade}[0.2]$ ,  $3=\backslash\text{shade}[0.4]$ ,  $4=\backslash\text{blacken}$ ,  $5=\text{whiten}$ 。PostScript の場合は 'y setgray' で y は 0.9, 0.8, 0.6 と 0 と 1 です。輪郭無しの場合には負の値になります。一方 arrow は矢印の個数。文字列の場合は全角 (=1) 半角 (=0) の区別。矢印の場合はそれが付いている節点番号 (始点か終点)。

次に, B 以下に並ぶ情報は表 3 の通りで, 基本情報 2 の数値 +1 行並びます。座標は, 描画窓の左上が原点で, 右に x 軸で下に y 軸となっています。直線と曲線以外はメモリの無駄ですが, ま, 最近ではコンピュータ側のハードウェアが良くなったので問題は生じないでしょう。

## 18 お問合せと最新版

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

<http://mechanics.civil.tohoku.ac.jp/%7Ebear/soft/qfig/>

に置いておきます。また, コマンド窓右上のバージョン情報 (当然ではありますが宇宙連邦の宇宙暦で, まだ紀元前です) のところをクリックすると, Qfig 公開のウェブページのブラウザ表示が現れますので, 最新版の確認ができます。[HELP] ボタンを押して, さらにその [HELP] ボタンを押すことによっても連絡先は表示されます。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。

ところで, Qfig を Google 検索すると「Qfig.exe とは? その修復方法は?」というページが見つかります。中身は全く理解できませんが, そこからリンクされているページにある 'Qfig.exe' は, 私とは全く関係の無いファイルだと想像されます。怖いのでダウンロードもしていません。私のこの Qfig の発端は 1992 年 10 月 4 日が Ver.0.9b のマニュアル日付になっている MS-DOS 上のプログラムで, Quick BASIC で作成したものです。そして例えば, DOS-V 版 Qfig (Ver.1.0) のクレジットを見るとわかるように

表 3 図の節点情報等

	情報 1	情報 2	備考
直線	始点の $x$ 座標 節点 $j$ の $x$ 座標	$y$ 座標 $y$ 座標	全部で <b>node+1</b> 行
多角形	始点の $x$ 座標 節点 $j$ の $x$ 座標	$y$ 座標 $y$ 座標	全部で <b>node+1</b> 行 最後のデータは最初のデータと同じ
曲線	始点の $x$ 座標 節点 $j$ の $x$ 座標	$y$ 座標 $y$ 座標	全部で <b>node+1</b> 行
閉曲線	始点の $x$ 座標 節点 $j$ の $x$ 座標	$y$ 座標 $y$ 座標	全部で <b>node+1</b> 行 最後のデータは最初のデータと同じ
円	中心の $x$ 座標 始点の $x$ 座標 半径	$y$ 座標 $y$ 座標 0	
円弧	始点の $x$ 座標 中心の $x$ 座標 終点の $x$ 座標 半径 0	$y$ 座標 $y$ 座標 $y$ 座標 始まりの角度 (rad) 終わりの角度 (rad)	旧版は反時計回り 水平から時計回りに $0 \leq \text{角度} < 2\pi$ 始点角度 < 終点角度
楕円	中心の $x$ 座標 始点の $x$ 座標 長い方の半径	$y$ 座標 $y$ 座標 半径比	$x$ 方向半径に対する $y$ 方向半径の比
箱	左上の角点の $x$ 座標 右下角点の $x$ 座標	$y$ 座標 $y$ 座標	
文字列	左上点の $x$ 座標 0	$y$ 座標 <b>AscW(文字)</b>	以下 <b>klen</b> 行
	<b>x=Int(AscW(文字)/256)</b>	<b>AscW(文字)-256*x</b>	旧版の場合の一文字情報
矢印	矢始めの $x$ 座標 要点の $x$ 座標 矢終わり $x$ 座標	$y$ 座標 $y$ 座標 $y$ 座標	

qfig Ver.1.0 (5/25/1993)modified for J-3100/IBM compatible PCs

by: William Ofosu-Amaah(w---@---.toshiba.co.jp)

from: qfig NEC Ver.0.9c+1 (3/20/1993)by: Mr T.Iwakuma (bear@---.tohoku.ac.jp)

と（メールアドレスは一部伏字）なっていて、開発者は私と、当時東芝におられた Ofosu-Amaah さん<sup>37</sup>だとい  
うことがわかります。私が作成した exe ファイルは、当時日本ではメジャーだった某社のコンピュータ上でし  
か動かず、それを世界共通である IBM 系のコンピュータ上でも動くように Ofosu-Amaah さんが加筆修正変換  
してくださいました。それは v.1.1 までであり、私のウェブページからダウンロードすれば XP マシン上で今で  
も<sup>38</sup>使えます。そしてさらに CTAN サイトの /tex-archive/support/qfig/ でもオープンソースで配布され  
ています。そこいらが上述の変なファイルにつながるのかもしれませんが。

また拙作の Windows 版 Qfig は VBasic4 で開発して公開してあるのは Ver.1.21.6 (宇宙暦 [-30]6775: 2001  
年 6 月 12 日) が最終版です。その後非公開で若干の変更をした VBasic のソースファイル等の最終の更新日付

<sup>37</sup> その後、このアドレスでは連絡できなくなっております。

<sup>38</sup> 実は拙作 Windows 版よりも DOS-V 版 v.1.0 の方を長年多用してきました。

は 2001 年 11 月で、最終的に整備されたマニュアルの日付は 2003 年ですが、アイコン等の作成整備日付から 1998 年 6 月には開発と公開がほぼ完了していることがわかります。またそれは直接レジストリ等に触れることは無いと思われますし、ソースは公開しておりませんので、この「Qfig.exe とは? その修復方法は?」というページに書いてある ‘Qfig.exe’ は、本 Qfig とは全く関係の無いファイル<sup>39</sup>であろうと想像できます。

私自身がこのパッケージをアップロードした先は東北大学と Vector のみです。必ずこの二箇所からダウンロードしてお使いください。なお、第 2 節でも説明したように、ダウンロードしただけの Qfig 起動はブロックされていますが、これは Windows の通常の security 設定だと思われます。使い続ける場合には、ウィルスチェック後にブロックの解除をしてください。

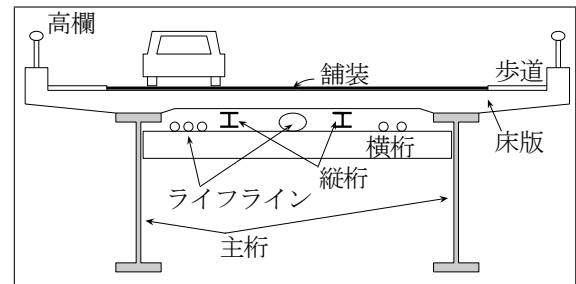
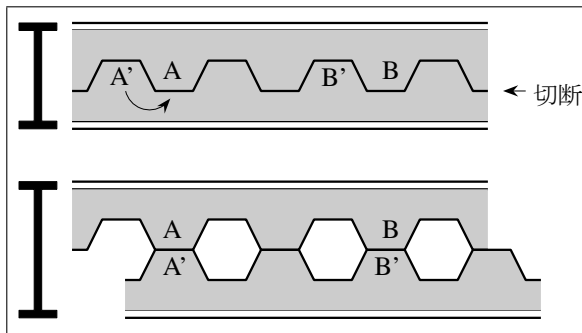
## ヒントになるような項目一覧

アルトキーの不思議, 4	楕円 (外接する), 14
ウィルス, 28	楕円 (回転できる), 11, 14
円弧 (正確な), 19	楕円 (傾いた), 13, 14
‘curves.sty’ の利用, 26	楕円の半分, 13
キーボード割り当て, 4	P <sub>CT</sub> E <sub>X</sub> の利用, 26
鏡像, 16	ピッチ (キーボード) 変更, 4
矩形 (回転できる), 11	ピッチ (マウス) 変更, 3
グリッドを消す, 12	ファンクションキー 10 番の不思議, 4
左右反転, 16	ブロックの解除, 2
縮小した図, 20	文字サイズが変, 8
正多角形, 13	文字列中の数式, 8
正多角形 (回転できる), 13	矢印の向きが変, 7
	ロールアップダウン, 5

---

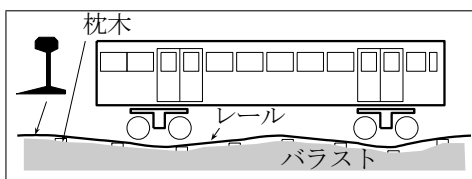
<sup>39</sup> 私どもで開発したプログラムには Ver.1.0.0.0 というものは存在しませんし、悪意のある人がファイル名をそうしただけの何らかの実行ファイルがどこかにあって、それを駆除するのがこのページにあるプログラムなのかもしれません。なお、地震関係のソフトウェアに同名のものがあることがインターネット検索でわかりましたが、それとも関係ありません。その版の日付らしいもので一番古いものが 5-9-1997 のようですから、命名は私どもの方が早いと想像できます。すぐにわかるとは思いますが、UNIX 上の xfig が命名のヒントでした。

# 講義資料等で使った $\text{\LaTeX}$ ファイルの例

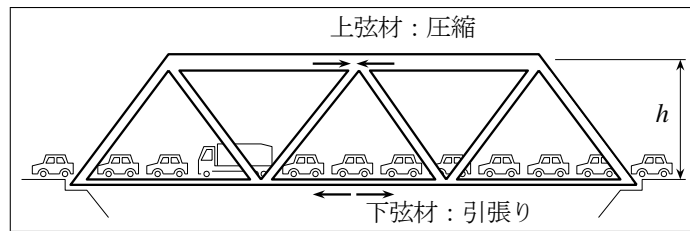


↑  
 $\text{\LaTeX}$  ファイルで `\unitlength` を 0.15 mm に変更

↑  
輪郭線除去  
↓

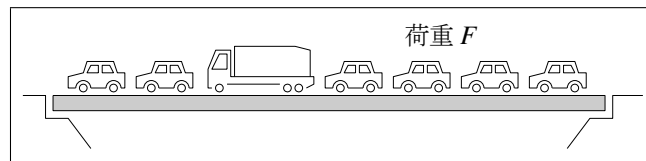


↑  
 $\text{\LaTeX}$  ファイルで  
`\unitlength` を 0.125 mm に変更

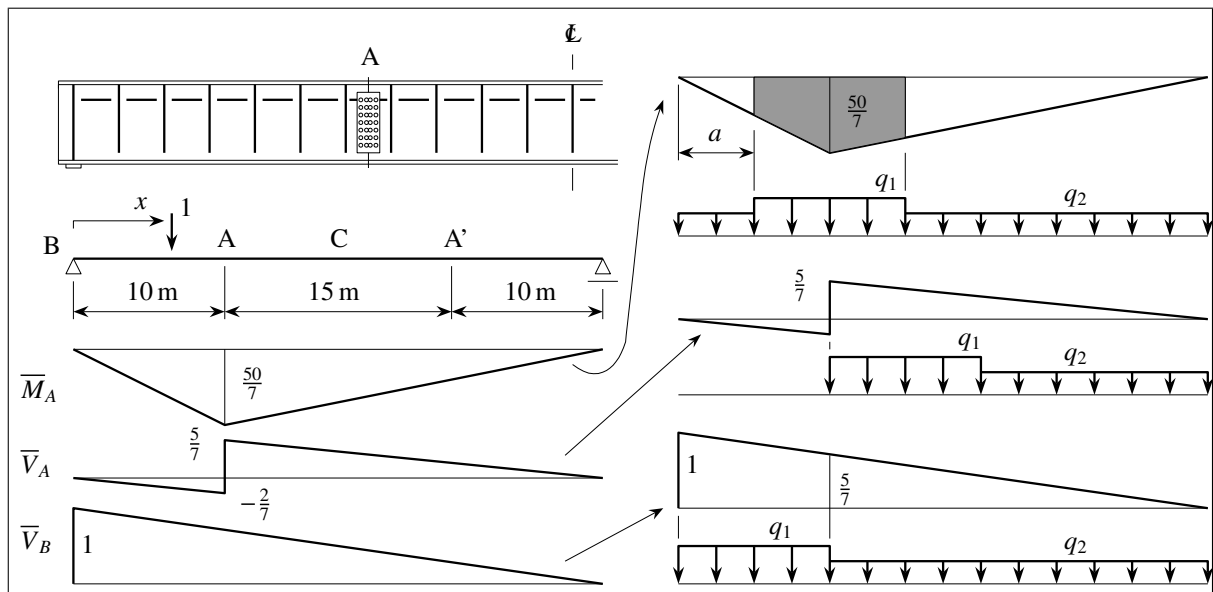


↑  
 $\text{\LaTeX}$  ファイルで `\unitlength` を 0.18 mm に変更  
斜材：輪郭線除去+白塗りつぶしで自動車を隠す  
太い矢印：旧版より尖らせた改善例

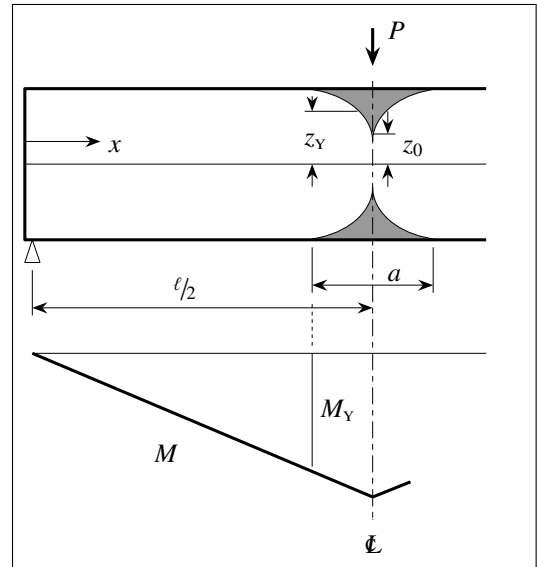
↑  
数学記号  $h, F$  は  $\text{\LaTeX}$  ファイルで  $\$$  マーク挿入  
↓



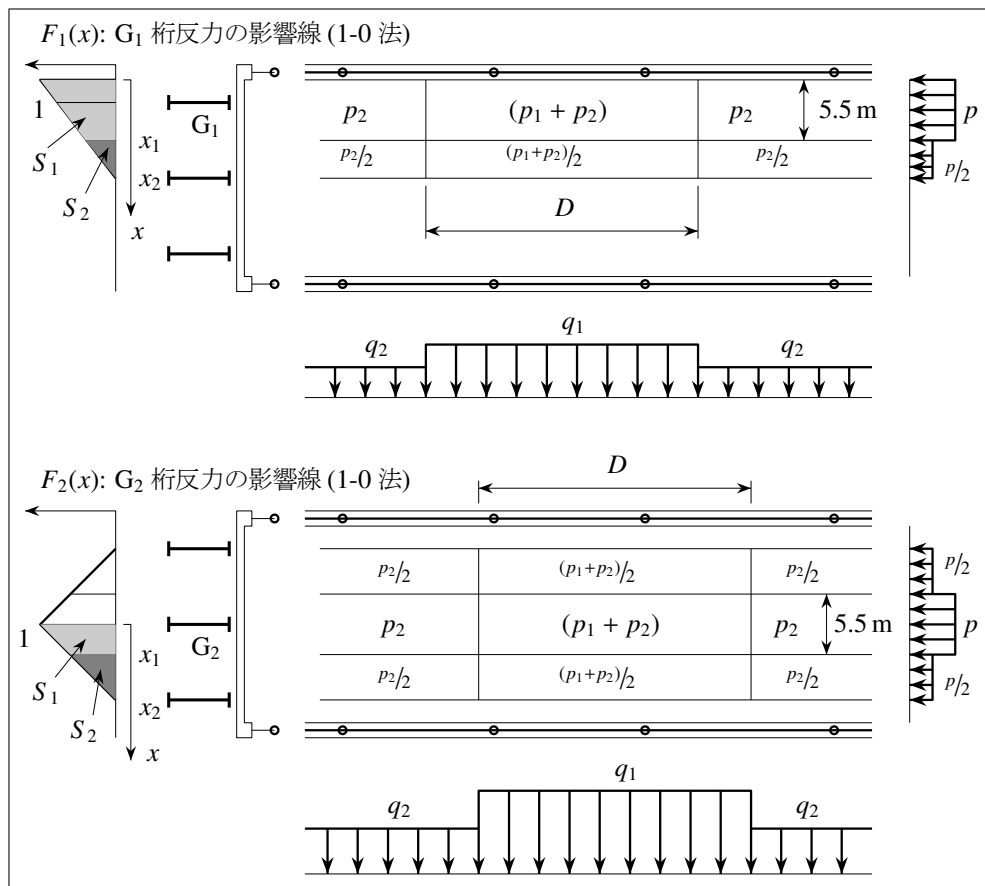
網掛けと数字の定義順に注意  
このくらいで図形が 226 個  
↓



網掛けした任意の図形を白塗りで隠す ⇒  
 どうやってあるか想像してみてください。  
 ついでに一点鎖線もご覧ください



旧 Qfig のデフォルトの図形数が 256 個  
 だったこともあって結構苦労した例：270 個の図形





## 更新ログ

以下はほぼすべてがファイル変換時の虫取りと改訂ですから、生データは旧版も含めてそのまま扱えます。つまりその生データを新しい版で変換すると、結果が改善されることになります。なお本開発は私の本来の業務の一つであると考えています<sup>40</sup>が、たいていは、寝入りばなやプールで泳いでいる最中に思いついたことや週末・夏休みにぼおーっとしていて気づいたことを用いて改訂していますので、特に最初のころは数日で版が上がってしまっています。pdfのportabilityはv.3.3から実現されています。

**v.3.3.2:** 2014/8/12

### 昆虫採集:

1. 円弧の矢印がずれた原因は何と！座標値の整数化。何故整数化したのか忘れた。これは生データを整数化してしまっているの、気になる方は、円弧以外は矢印の削除・再定義、円弧の場合は円弧の対称コピーを二度使って定義し直した上で、矢印の再定義をどうぞ。
2. HELPの加筆修正。
3. Ngraph用ファイルの円の白塗りだけ忘れていた。何故?

### 変更・追加・削除:

1. Ngraph v.6の箱をちゃんと‘B’命令にした。また塗りつぶしの無い曲線もちゃんと‘T’命令にした。最終結果にはほとんど差異は生じないが、導入当初の命令内容の理解に誤解があったため変更。
2. 図のより正しい領域定義のために`\end{picture}`の後ろに%加えた。図の枠外に半角スペース程度の余白が加わるのを嫌ったため。L<sup>A</sup>T<sub>E</sub>Xのマニュアルにも記述は無かったかと・・・
3. 再描画時にグリッドを消せるようにした。不完全だが・・・対処不能。
4. Ngraph ボタンを追加し、LaTeX ボタンとは独立させた。
5. Ngraph v.6の箱だけでは輪郭線を付けられるようにした。
6. 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。Gra2Eepicとの連動。

**v.3.3.1:** 2014/7/30

### 変更・追加・削除:

1. 破線等に`\dashline`等を使わないでいいように改善した。これでpdfファイルにも一点鎖線が出るようになった。ようやくportabilityが得られたため主バージョン番号を上げた。Ngraphの一点鎖線等の出力方式に感謝。
2. そのため、一点鎖線をデフォルトにし、長い破線は出力時オプションとした。

**v.3.2.4:** 2014/7/26

### 変更・追加・削除:

1. Ngraph v.6に合成できるgraファイル出力で、長破線と一点鎖線の選択をオプションにした。

**v.3.2.3:** 2014/7/21

### 変更・追加・削除:

1. 円弧定義の右クリック中、つまり中心位置探索中の座標表示は円弧の中心座標にした。

---

<sup>40</sup> 旧版の表示を24世紀仕様に準じたことから、業務怠慢による成果だと匿名のメールでお叱りを受けたことがあります。論文執筆のためのプログラム開発なので業務だと考えております。ま、こっちに方が楽しいのも事実ですが。また今回は、一部のボタンの表現を除き、21世紀仕様で作成してみましたので音声も無く面白くないかもしれません。おっとそういえば、音声そのものは著作権上かなり問題があったので削除したのでしたっけね。

2. 対称コピーができる図形に円弧も加えた。
3. Ngraph に合成できる gra ファイル出力を加えた。旧版のサブルーティン流用で済んだのと、ユーザーはほとんどいないと想像して主バージョン番号は上げず。

v.3.2.2: 2014/7/8 早くも・・・

#### 昆虫採集:

1.  $\text{\LaTeX}$ , PostScript 出力で一点鎖線が選べなくなっていたミス。

v.3.2.1: 2014/7/5 ミスがかかなりあったことから主バージョン番号を上げた

#### 昆虫採集:

1. Copy/ Move/ KILL の領域指定で、領域内に何も無いときのリセットし忘れ。
2. 2 節点線分を描いたあと円弧が描けなくなるミス。デバッグ時制御の削除し忘れ。
3. 円弧の  $\text{\LaTeX}$ , PostScript 出力で  $2\pi$  以上の円弧になるミス。arctan の処理は難しい。
4. 文字列が一番下にあるとき、それを全部はカバーできない描画範囲になるミス。
5. コマンドボタンとファンクションキー上下切替のシンクロナイズし忘れ。
6. 全図形が削除された場合でもそれを認識せず余計なことを尋ねようとするミス。
7. 文字列入力のときの描画・再描画の順序のミス。
8. EDIT のバックアップで矢印の図形番号がずれるミス。
9. 直線描画等での最初の右クリックで変な図形が出る、あるいはエラーになるミス。

#### 変更・追加・削除:

1. **Obj #** で右クリックしたときに、周辺の節点にマーキングして選べるようにした。
2. グループに参加した図形を一時的に赤くした。
3. 各種設定小窓が出ている間は、他の窓にはアクセスできないようにした。
4. **EDIT** 右クリックでは、親図形の属性や塗りつぶしを引き継ぐことにし、その変更は左クリックだけにした。
5. 円・楕円・箱の **EDIT** 左クリックでは、属性や塗りつぶしのみの変更とし、半径等の寸法変更を伴う場合には **Shift** キーを押すことにした。
6. 円弧の **EDIT** 右クリック中の座標表示は、円弧の中心座標にした。
7. 円と箱の **EDIT** 右クリックによる内外接多角形・外接楕円への変換で、**Shift** キーを押した場合には親図形を温存するようにした。そのため、箱の外接円の機能は無くした。
8. 描画窓タイトルバーに、今扱っているデータファイル名を表示した。
9. キーボード操作時の **KILL!** と **Redraw** のキー割り当ての追加。
10. 箱描画時の右クリックでの塗りつぶし機能の解消。
11. ini ファイルにコマンド窓の最後の位置を保存して次回利用。
12. DrawString の位置が若干ずれているような気がして、また旧版との整合を得るためにも画面表示の位置を調整した。 $\text{\LaTeX}$ , PostScript 出力とほぼマッチするようにした積もり。
13. キーボード操作のときにはデフォルトでファンクションキー機能を on にした。
14. 文字列入力確定に **Enter** キーが使えるようにした。入力窓も小さくした。

v.3.1.1: 2014/6/15 に公開。その後、暫定版 v.3.X.1, v.3.Y.1, v.3.Z.1。

v.2.21.6: 2014/1 頃、v.1.21.6 [VB4] の [VB2010] への移植を始めようかなと思案開始。窓一つだけにしようとしたが、キーボード操作機能の組み込みが安定せず断念。複数窓版に変更。