

Ipe7 マニュアル

鎌田正雄

No:H2011-9

2015年4月10日

目次

| | | |
|----------|---------------------------|-----------|
| 1 | 概要 | 1 |
| 2 | 参考文献 | 1 |
| 3 | Ipe の概要 | 1 |
| 4 | Ipe のファイル | 1 |
| 5 | 一般事項 | 2 |
| 5.1 | オブジェクトの順位 | 3 |
| 5.2 | 対象の選択 | 4 |
| 5.3 | オブジェクトの移動とサイズ変化 | 6 |
| 5.4 | ストロークと色の塗りつぶし | 6 |
| 5.5 | ペン、点線、矢、斜パタン | 8 |
| 5.6 | 透明度 | 9 |
| 5.7 | シンボリック属性と絶対属性 | 9 |
| 5.8 | ズームとパン | 10 |
| 5.9 | グループ | 10 |
| 5.10 | レイヤー | 10 |
| 5.11 | マウスのショートカット | 11 |
| 6 | オブジェクトの型 | 11 |
| 6.1 | パス・オブジェクト | 12 |
| 6.2 | テキスト・オブジェクト | 17 |
| 6.3 | イメージ・オブジェクト | 19 |
| 6.4 | グループ・オブジェクト | 19 |
| 6.5 | 参照オブジェクトとシンボル | 20 |

| | | |
|-----------|------------------------------|-----------|
| 7 | スナッピング | 20 |
| 7.1 | グリッド・スナッピング | 21 |
| 7.2 | コンテキスト・スナッピング | 21 |
| 7.3 | 角度スナッピング | 21 |
| 7.4 | スナッピング・モードの相互作用 | 22 |
| 7.5 | 例題 | 23 |
| 8 | スタイルシート | 27 |
| 9 | プレゼンテーション | 28 |
| 9.1 | プレゼンテーションのスタイルシート | 29 |
| 9.2 | ビュー | 29 |
| 9.3 | ブックマーク | 30 |
| 9.4 | 陰影のパターン | 30 |
| 9.5 | テキスト・オブジェクトから用いられる Ipe のシンボル | 30 |
| 10 | 進んだ話題 | 31 |
| 10.1 | Latex 文書と Latex 定義を共有する | 31 |
| 10.2 | <i>ipelet</i> を書く | 31 |
| 10.3 | LaTeX 変換のトラブルシュート | 31 |
| 10.4 | Truetype フォントの使用 | 31 |
| 10.5 | 日本語処理 | 31 |
| 11 | 謝辞 | 32 |
| 12 | 後記 | 32 |

1 概要

Ipe は \LaTeX の入るドローソフトである。 \LaTeX を含む図を製作できるユニークな特徴を有する。英文マニュアルはあるが、難解な表現もかなりあり、習得は容易ではない。ここでは参考文献の抄訳を行い、適当な例を付け加えて、理解に便利のように編集する。なお、術語は Ipe の操作画面と一致させるために、あえて翻訳せず英語のままとする。ただし、その意味は解説する。

Ipe7 自体はまだベータ版である。できるだけ最新版を用いるように努めているが、マニュアルには訳者の理解が及ばず、現実の Ipe と矛盾する点がある。気づいた限りで注をしておいた。

2 参考文献

1. The Ipe Manual, Otfried Cheong, February 23, 2015

3 Ipe の概要

数学記号あるいは数式を含む図を作成のために、Ipe は大変有用である。資料の準備の際に \LaTeX コマンドを用いることができるからである。Ipe によれば各種の基礎的な幾何学的図、例えば線、薄板、ポリゴン、円などを含む図を編集し、あるいは準備することができる。

Ipe は、他のグラフィックソフトと異なる特徴として、図にテキストを付け加えることができる。Ipe はこれらのテキストを \LaTeX のオブジェクトとして取り扱う。これにより図にあらゆる \LaTeX コマンドを入れることができる。すなわち、数学的表現ができるのである。もちろん簡単な“ q_i ”のようなラベルも許容される。Ipe は \LaTeX のソースを処理し、ポストスクリプトもしくは PDF の図として入れ込む。

さらに、Ipe は通常は高級なグラフィックソフトあるいは CAD にしか見られない編集機能を有している。例えば、スナッピング機能である。

Ipe の独特な機能に拡張能力がある。使用者は使用者独自の機能を *ipelet* として定義でき、いったん Ipe に登録できれば、Ipe の本来の機能であるかのように使用できる。*ipelet* は Lua で書くことができる。そのほかにはボロノイ図を描ける *ipelet* がある。

4 Ipe のファイル

Ipe7 はポストスクリプトあるいは PDF ファイルを生成する。これらのファイルは GhostView、Acrobat Reader、Xpdf でみることができるファイルである。ただし、Ipe のファイルには Ipe のオブジェクトを示す特殊なコードが含まれている。そこで Adobe Acrobat によりいったん Ipe のファイルを編集するともはや Ipe では読めないファイルとなる。

Ipe で製作した図をストアするフォーマットとして、eps、pdf、ipe の 3 とおりの方法がある。eps の場合には単一ページから構成される文書のみであり、複数ページには対応していない。

Ipe 文書は以下のとおりである。

1. L^AT_EX 文書の図 :
2. プレゼンテーション用の図 : Ipe は Powerpoint や Keynote のようなプレゼンテーション・ツールではない。Ipe のプレゼンテーションは単に Ipe により作成された PDF ファイルである。従って、Acrobat Reader あるいは LCD により、プレゼンテーションを行う場合に使用される。
3. SVG ファイル : SVG ファイルは web ページで図の伸張が可能である。Ipe は直接 SVG フォーマットでストアできるわけではないが、*iperender* を用いれば、Ipe ファイルを SVG ファイルに変換できる。この変換は一方向である。ただし、補助ツール *svgtoipe* を用いれば、SVG の図を Ipe フォーマットに変換できる。
4. Bitmap : *ioerender* を用い、PNG フォーマットにおけるビットマップに変換できる。

5 一般事項

Ipe をスタートすると下図のような画面が現れる。

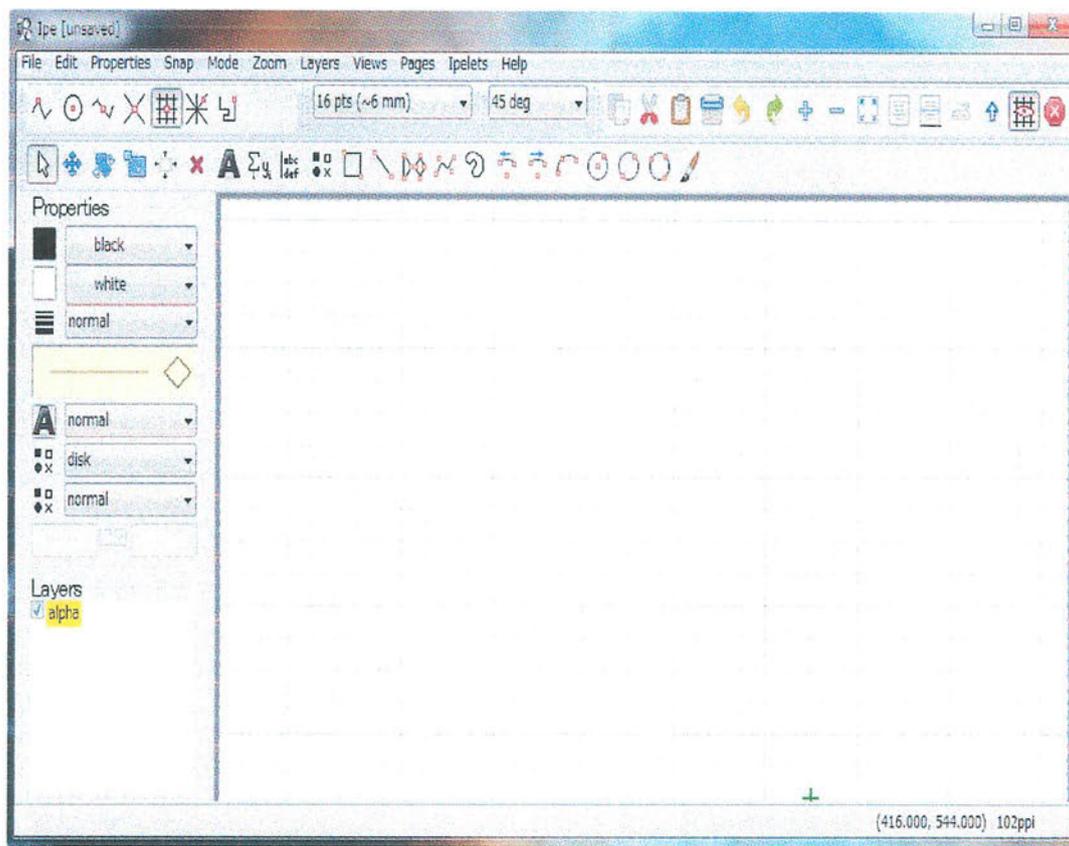


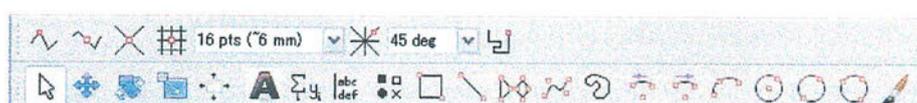
図 5.1: Ipe7 のスタート画面

中央に見える格子の入った白い画面はキャンバス (*canbas*) という。白い画面を”シート”という。これは文書の第一ページである。

ウィンドウのトップには、メニューバーがある。



メニューバーの下、キャンバスの上部にツールバーが2つ見える。



上部に見えるのはスナッピング・モード選択のツールバーである。格子サイズを選択、角度分解能の選択ができる。下部のツールバーはモード選択である。

キャンバスの左部はオブジェクトプロパティを示す。上からストロークの色彩、塗りの色彩、線幅、パスプロパティ、テキストサイズ、マークの形状とサイズが示される。最下部には表示ページのレイヤーのリストが示される。すべての要素には説明が付けられている。マウスを当てて、少し待てば説明が現れる。

モード・ツールバーにより Ipe のモード設定ができる。大雑把に言えば、モードとはマウスの左ボタンを図に置きクリックしたときの状態である。ツールバーの左から5つはオブジェクトのモードを選択し、変換する機能である。残りのボタンは新しいオブジェクトを生成するための選択ボタンである。

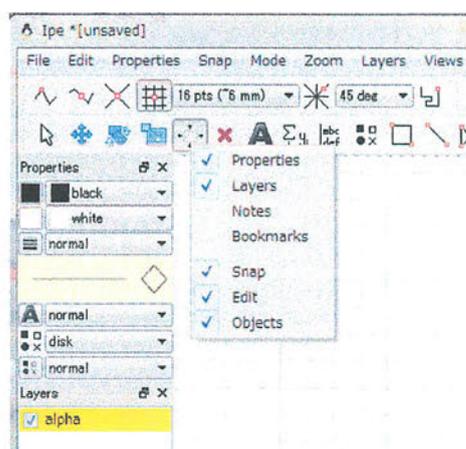


図 5.2: コンテキスト・メニュー

オブジェクトの順位は変更することもできる。これには編集メニューの”Front”もしくは”Back”機能を用いる。

マウスの右ボタンを選択したモードに置くとオブジェクトのコンテキスト・メニューが示される。コンテキスト・メニューとは *Properties*、*Layers*、*Bookmarks*、*Snap*、*Object* をいう (図 5.2)。

5.1 オブジェクトの順位

Ipe の図面は幾何学的オブジェクトの並んだものである。オブジェクトの順位は重要である。オブジェクトが2つ重なるとき、第一順位のオブジェクトは他を隠すことになる。新しいオブジェクトを作ると他のすべてのオブジェクトの前面に出ることになる。ただし、

図 5.3 と図 5.4 には、矩形オブジェクトを生成し、それぞれ色塗りをした上で、編集メニューの "Front" もしくは "Back" 機能を用いて、順位を変更した例を示す。

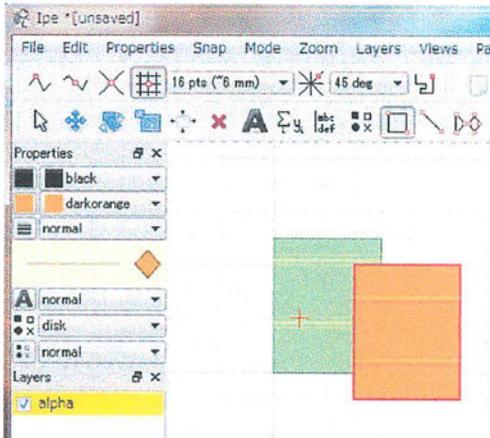


図 5.3: オブジェクトの順位 : 1

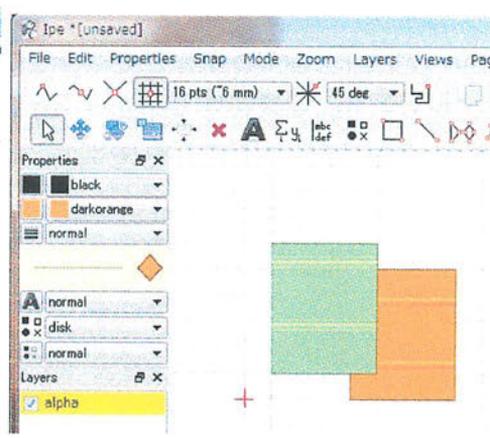


図 5.4: オブジェクトの順位 : 2

5.2 対象の選択

Ipe 機能を用いるときにはその機能を適用しようとするオブジェクトを決めておかななくてはならない。これをオブジェクトの選択という。あるオブジェクトの近傍でマウスの左ボタンを押すと、そのオブジェクトが選択される。近傍とは(正確な距離が *prefs.lua* において、*select distance* として設定される。) 図 5.5 において示される円である。オブジェクトが選択されたことはそのオブジェクトの枠の色彩が変化する(例えば、円あるいは矩形オブジェクトならその形状を示す線)。

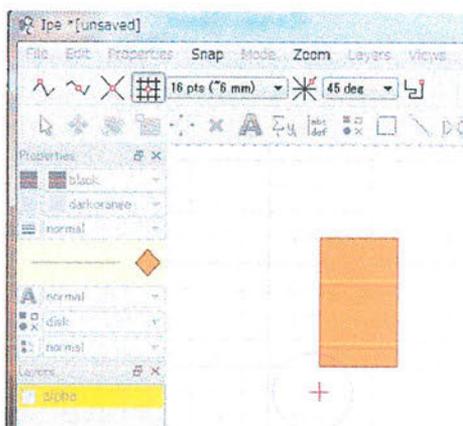


図 5.5: オブジェクトの選択

キャンバスに複数のオブジェクトがあるとき、選択されたオブジェクトは第一オブジェクト (*primary object*) (このオブジェクトはグループであってもよい。) といい、ほかのオブジェクトは第二オブジェクト (*secondary object*) という。ある機能 (例えば、コンテキストメニュー) は第一オブジェクトにのみ作用する。他の機能は第一オブジェクトと第二オブジェクトを別々に取り扱う。例えば、整列機能 (*align function*:メニューバーの *Iplets*) では、第一オブジェクトを基準として第二オブジェクトに整列機能を作用させる。

例: 図 5.6 に 2 つのオブジェクトを示す。左側のオブジェクトを第一オブジェクトとし、右側のオブジェクトを左側のオブジェクトに整列させる。

選択にはマウスを近傍にして (図 5.5)、モード選択のツールバーから矢印マークを選択し、編集モードの "すべてを選択" を選ぶ。(図 5.6) メニューバーの *Iplets* → *Align&Distribute*

→ **Align top** により、図 5.7 のように整列が行われる。Ipelets の機能選択によりさまざまな整列ができる。

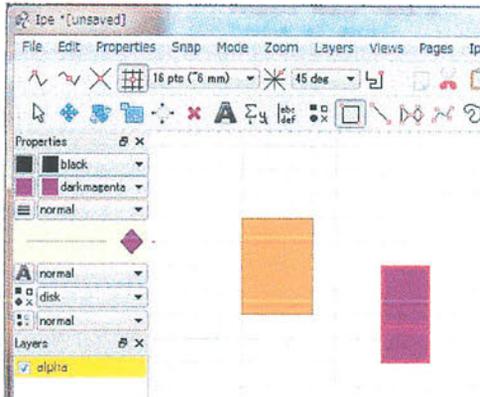


図 5.6: 2つのオブジェクトの選択

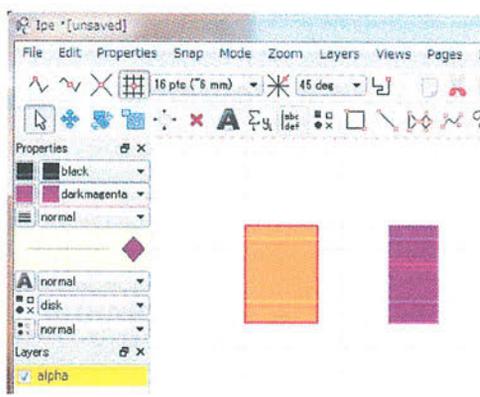


図 5.7: オブジェクトの整列

選択は選択されたオブジェクトを赤枠の色付けで示す。第一オブジェクトの枠はやや太く、第二オブジェクトの枠は少し細い。図 5.6 からは分かりにくいですが、Ipe の実行画面では明瞭である。

第一及び第二の区分けは選択モードで設定される。あるオブジェクトに近いところでマウスの左ボタンを押すことによりそのオブジェクトは第一選択部分となる。他のすべてのオブジェクトは選択外とされる。左マウスをクリックしている間に **Shift** キーを押し続けていると、マウスに最も近いオブジェクトが選択オブジェクトに加わり、あるいは選択オブジェクトから除去される。マウスとともに矩形をドラッグし、マウスを放すと矩形のすべてのオブジェクトが選択される (図 5.6)。**Shift** キーを用いれば、矩形内のすべてのオブジェクトの状態をスイッチすることができる。

他のオブジェクトの下位にあるとか近いオブジェクトの選択を容易にするには、オブジェクトの選択がどのように行われるかを正確に理解する必要がある。マウスボタンを押すと、マウスの位置 (正確な距離が `prefs.lua` において、`select distance` として設定される。) に十分近いすべてのオブジェクトのリストが計算される。次にこのリストからマウスからの距離により順位付けされ、図における深度を増加させる。**Shift** キーが押されていないときには、現在の選択オブジェクトはクリアされない。提示されるオブジェクトはリストにある最初のオブジェクトである。

そこで、マウスを押している間に **Space** キーを用いて、深度と距離の順序でオブジェクトのリストを変更できる。マウスの右ボタンを離すとオブジェクトが選定され、もしくは選定から除外される。

マウスの位置からオブジェクトにいたる距離を測定するときには、Ipe はオブジェクトの周辺のみ考慮する。オブジェクトが塗りつぶしであるときには、その内部をクリックせず、周辺に近い位置をクリックしなくてはならない。

オブジェクトを選定するほかの方法は編集メニューの **Select all** 機能を用いることである。ページにあるすべてのオブジェクトが選定される。同様に **Layer** メニューの **Select all in layer** 関数を用いれば、アクティブレイヤーのすべてのオブジェクトを選定できる。

5.3 オブジェクトの移動とサイズ変化

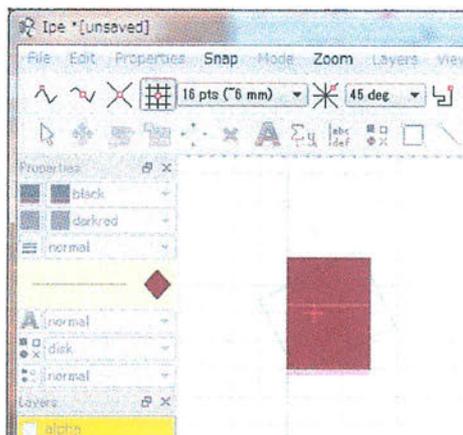


図 5.8: オブジェクトの回転

重ね書きすることができる。軸が設定されると原点は中心として用いられる。

例：回転 (rotate)

マウスでオブジェクトを選択し、ツールバーのモード選択で **rotate** を選択し、マウスでオブジェクトを掴むと回転できる (図 5.8)。

通常、オブジェクトの変換は現在のオブジェクト選択に作用する。しかし、異なる多くのオブジェクトの周辺で移動させるようにするには、例外がある。カーソルが現在の選定の近くでないときでカーソル近傍に他のオブジェクトが存在するときに、マウスボタンが押されると、そのオブジェクトは移動、回転、サイズ変化をさせることができる。

サイズ変化と伸縮は境界ボックスの対角点を変換の固定点として用いる。さらに軸システムの設定を行うと、軸システムの原点が代わりに用いられる。軸の設定には F_1 を押しながらマウスで原点を設定する。軸システムのリセットには $\text{Ctrl}+F_1$ を押す。

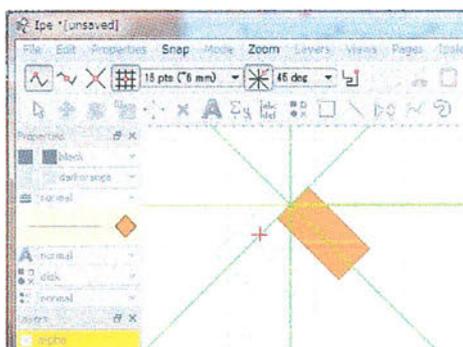


図 5.9: 軸設定時のオブジェクトの伸長

オブジェクトの頂点の周囲で回転し、あるいはサイズ変化をすると便利である。軸システムの原点をオブジェクトの頂点に設定すると容易にできる。軸システムを **Snap** により頂点に設定することにより可能になる (図 5.9)。

メニューバーの **Mode** もしくは、ツールバーのモード選択 (左から 2~4 番目) により、オブジェクトの変換が可能になる。オブジェクトの変換には 3 つのモードがある。 **translate**(移動)、 **stretch**(伸縮)、 **rotate**(回転) である。 **translate** によりいずれの方向にもオブジェクトの移動が可能になる。 **stretch** 機能により伸張、縮小が可能であるが、左マウスボタンを押している間、 **Shift** キーをホールドしていると、伸縮機能はオブジェクトの伸張比を維持する (scaling という)。 **rotate** 関数は選定オブジェクトの **Bounding Box** の中心に対して回転する。 **Bounding Box** とは **TeX** で読み取り可能な書式で画像のサイズを保存したファイルである。この動作は角度軸を規定すると

オブジェクトの頂点の周囲で回転し、あるいはサイズ変化をすると便利である。軸システムの原点をオブジェクトの頂点に設定すると容易にできる。軸システムを **Snap** により頂点に設定することにより可能になる (図 5.9)。

5.4 ストロークと色の塗りつぶし

Path オブジェクトでは輪郭をストローク (stroke)、内部をフィル (fill) と呼ぶ。この呼び方はポストスクリプトの用語である。 **Properties** ウィンドウにおいてストロークとフィルの色は独立に選定できる。手書きの図面をペンと黒色のインクで書くとしよう。 **Ipe** が **stroke color** であることは黒色のインクを用いて、書くことに相当する。お

そらくペンでオブジェクトを塗りつぶすことはないだろう。このためにはブラシを使うだろうし、水彩のような違う色も必要だろう。fill color とはブラシに相当する。

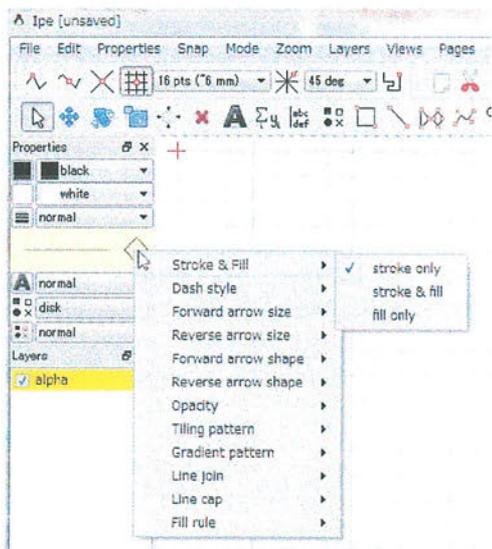


図 5.10: path properties の選択

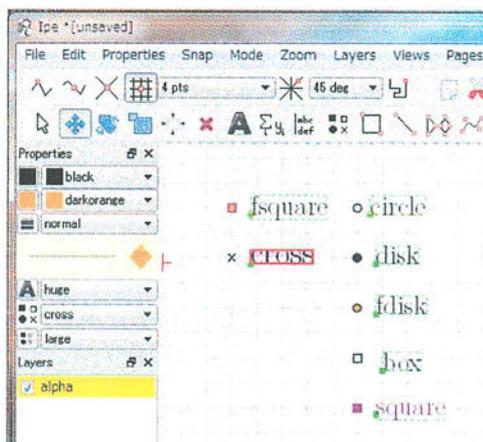


図 5.11: マークのオブジェクト

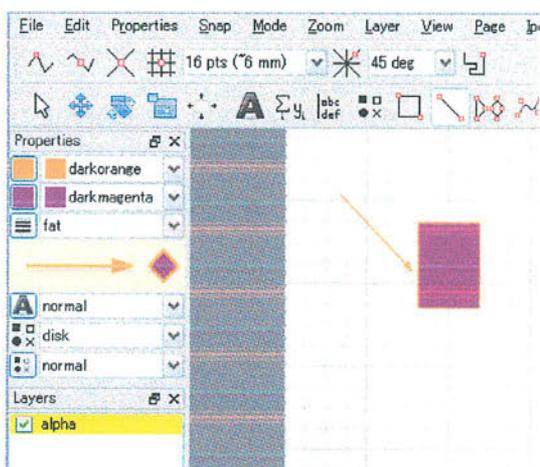


図 5.12: stroke&fill のオブジェクト

パス・オブジェクトを作るとき、ipe にそれがストローク、フィルのいずれか、あるいは両方を用いるかどうかを指示しなければならない。これは **Path Properties** フィールドで設定できる。**Path Properties** フィールドの右端 (菱形図形) をクリックすると **stroke**、**stroke&fill**、**filled** の 3 つのモードを交互に循環する。**Path properties** フィールドでもマウスを対象に当てると説明が現れる (**context menu**)(図 5.10)。

テキストと矢印のオブジェクトの色はストローク色により決まる。

これは矢印の塗りでもおなじである。これらにはブラシ (**fill color**) でなく、ペン

(**stroke color**) を用いることもできる。mark 形の **disk** と **square** は **stroke color** のみを用いる。マーク形の **fdisk** と **fsquare** を用いて、2 色のマークを作ることできる。図 5.11 にマークを作った例を示す。**disk** と **square** は **stroke** 色で塗りつくされているが、**fdisk** と **fsquare** は 2 色に塗り分けられている。**circle** は輪郭のみで色が決まるが内部は透明となる。

5.5 ペン、点線、矢、斜パターン

path properties フィールドはパス・オブジェクトのすべてのプロパティを設定するために用いられる。ただし、ペン幅はパスプロパティの上にあるセクタを用いて設定される。セクタの選択モードは **hevier**、**fat**、**ultrafat**、**normal** である。点線、ダッシュ、のパターン (固定、点線、ダッシュ) はパス・オブジェクトの境界に影響する。例えば、ポリゴン、ポリゴナル線、スプライン、円、楕円、矩形、円弧などである。これはテキストやマークには影響しない。図 5.13 にパス・オブジェクトの例を示す。

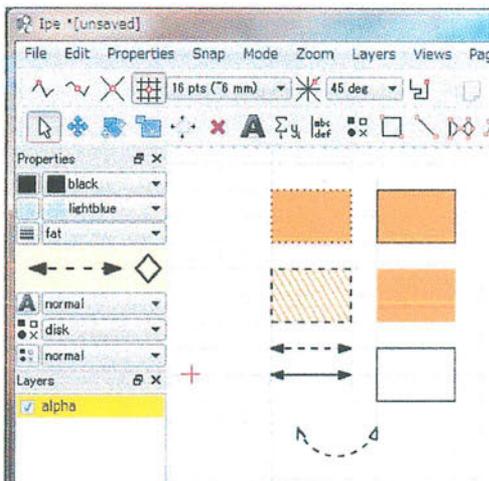


図 5.13: パスオブジェクトの選択例

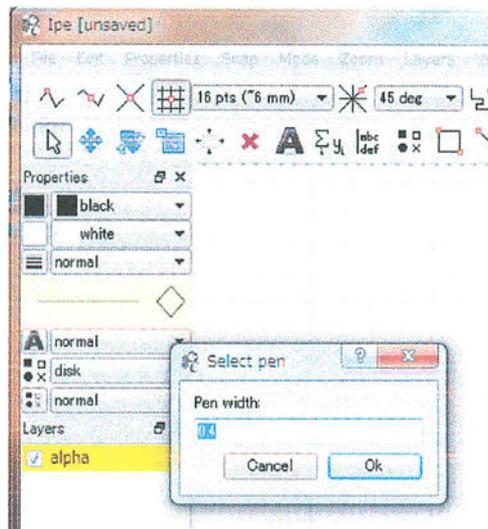


図 5.14: 線幅の設定

線幅はポストスクリプト・ポイント (1/72 インチ) で与えられる。0.4 もしくは 0.6 程度がよい。(図 5.14)

パスプロパティ・フィールドにあるセグメントの終端をクリックすると先端及び終端の矢印をオンオフできる。多角形の線、スプライン、円弧だけが矢印をつけられる。矢印の形とサイズはパスプロパティのコンテキストメニューにより利用できる。スタイルシートを用いて他の形やサイズを付け加えることもできる矢の形の **arc** と **farc** は特別である。パス・オブジェクトの最終セグメントが円弧のときには、これらは弧の半径に依存するカーブした形をとる。これらはもっと小さな半径をもつ弧に対しても正しく見えるように設計されている。傾きのパターンは固定色で塗る代わりにパス・オブジェクトに斜線で塗り分けるようにする。パス・オブジェクトのみが斜線パターンで塗ることができる。パターンはハッチの傾き、厚み、密度を決め、塗りの色はオブジェクトの塗り色からとられる。パスプロパティフィールドにあるコンテキストメニューを用いて、傾きパターンを決めることができる。スタイルシートにより独自の傾きパターンを決められる。

5.6 透明度

Ipe は透明度 (Opeque) については簡単なモデルをサポートしている。パス・オブジェクトとテキスト・オブジェクトの不透明度を設定することができる。1.0 の不透明度は完全に不透明である。0.5 は半透明を意味する。文書で用いようとする透明度の値はスタイルシートで規定しなければならない。(図 5.15) 透明度を含む文書は eps フォーマットではセーブできない。(ポストスクリプトは透明度をサポートしないからである。) このため標準の Ipe スタイルシートは (これは新規の文書には自動的に付け加えられる。) 透明度を規定しない。そこで自分のスタイルシートを用いなければならない。スタイルシートで規定すると、図 5.15 のようにコンテキストメニューに透明度が表示される。

5.7 シンボリック属性と絶対属性

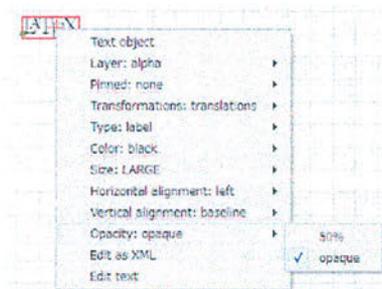


図 5.15: 透明度の指定

色、線幅、ペン、マークサイズ、テキストサイズなどの属性は絶対値 (色を規定する値など) もしくは名称のようなシンボリックで規定できる。シンボリック属性はスタイルシートにより絶対値に解釈されなければならない。

スタイルシートの一つの目的はプレゼンテーション用に作成した論文の図を再使用できることにある。プレゼンテーションの図は大きなフォント、マーク、矢、太いラインを用いなければならないことは明らかである。元の図がシンボリック属性をもっていれば、スタイルシートの交換により達成できる。

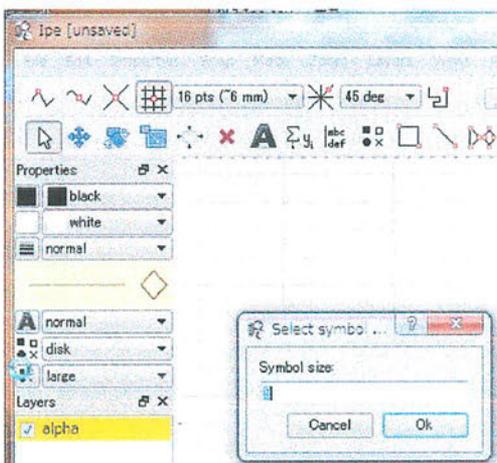


図 5.16: マークサイズの指定

Ipe のユーザーインターフェースはシンボリック属性を用いて調整できる。シンボリック名のセレクタの左のボタンをクリックすると絶対色値、ペン幅、テキストサイズ、マークサイズを使用できる (図 5.16)。

オブジェクトを作ると現在のユーザーインターフェース設定から属性を取得する。もし、絶対値属性を選択するなら絶対属性を得ることになる。絶対値の属性には任意の値を選定できる利点がある。例えば色彩調整による任意の色彩などである。シンボリックモードでは現在のスタイルシートにより与えられた選択を使用できる。

Ipe のユーザーインターフェースにより与えられたシンボリック属性はスタイルシートにより選択

される。

5.8 ズームとパン

マウスのホイールもしくはズーム機能を用いて、ズームインもしくはズームアウトできる。分解能の最小、最大を調整できる。Ipe のディスプレイの右下の分解能の表示で知ることができる。

関連するものとして、ノーマルサイズ機能がある。(インチあたり 72 ピクセルに設定する。) Fit Page 機能 (現在のページがキャンバスいっぱいであるような分解能である。) Fit object 機能 (スクリーンをカバーするように分解能を設定する。) Fit selection 機能 (選定されたオブジェクトにのみ適用する。) これらのすべてがズームメニューにある。

パンモードではマウスにより図面を左右に動かすパンができる。あるいはキャンバスのどこでもマウスを置いた位置でキーボードの "x" キーを押せばカーソルの位置がキャンバスの中央に来るように動く。このショートカットにより、図の中央において仕事が行えるようにできる利点がある。おなじことがズームイン、ズームアウトでも適用され、ほかのオブジェクトを編集したり、加えるときに図の注目点に戻ることができる。

5.9 グループ

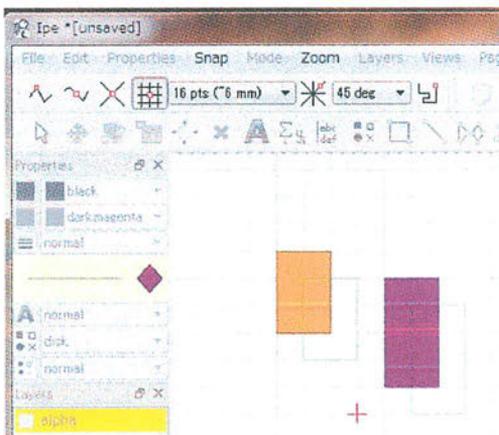


図 5.17: グループオブジェクトの移動

オブジェクトの集合を単一にまとめられれば便利である。グループにするには、矢印マークでオブジェクトを選択し、**Edit → Group** もしくは **Ctrl+G** とする。これは grouping object (オブジェクトのグループ化) という。この結果が幾何学的オブジェクトとなる。これにより移動、伸縮、回転などを全体としてできる (図 5.17)。ある部分の編集を他の部分に関連して、行うにはグループ化の状態ではできず、オブジェクトの分解をしなくてはならない。グループオブジェクトの分解はオブジェクトメニューにより Ungroup 機能を用いる。

グループオブジェクトは他のオブジェクトの要素となりうる。これによりオブジェクトの階層ができる。グループにクリッピング・パス¹を加えることができる。これによりグループの図をクリッピング・パスの内部に制限することができる。

5.10 レイヤー

Ipe 文書の 1 ページは 1 もしくはそれ以上のレイヤーから構成されている。そのページの各オブジェクトはレイヤーに属する。レイヤーの間と前後にはなんの関係もない。それゆえ、レイヤーは実際にはオブジェクトの属性である。

¹Ipe:Ver7.1.4 ではクリッピング・パスは機能していない。

レイヤーはいくつもの属性を有する。これらは編集可能であり、ロックもできる。レイヤーが編集可能である場合にはオブジェクトが修正できる。レイヤーは可視にも、不可視にもできる。レイヤーはスナッピングできるし、はずすこともできる。レイヤーがスナッピングされているならオブジェクトは磁化されているように動く。Ipe のページを編集するときには少なくとも1のアクティブ・レイヤーが存在する。新しいオブジェクトは常にアクティブレイヤーに生成される。レイヤーはまた、Acrobat Reader に分割して表示されるようにページを生成する。いろいろなレイヤーにオブジェクトを分布するならばビューを生成できる。これはページのレイヤーが示される順序を規定するのである。

5.11 マウスのショートカット

初心者は選定 (selection) あるいは変換 (transformation) モードを左マウスボタンと共に行うのがよい。Ipe になれたユーザーはショートカットを気にせず、作業中のモードはそのまま選定 (selection) あるいは変換 (transformation) できる。

表 5.1: マウスのショートカット

| | 左マウス | 右マウス |
|---------------------|--------------------------------------|---------------------|
| <i>Plain</i> | (*) | <i>context menu</i> |
| <i>Shift</i> | (*) | <i>pan</i> |
| <i>Ctrl</i> | <i>select</i> | <i>stretch</i> |
| <i>Ctrl + Shift</i> | <i>select non - destructively</i> | <i>scale</i> |
| <i>Alt</i> | <i>translate</i> | <i>rotate</i> |
| <i>Alt + Shift</i> | <i>translate horizontal/vertical</i> | <i>rotate</i> |

(*) とマークしたフィールドは作業モードによる。マウスボタンの中央は常にキャンパスのパンを行う。右マウスボタンはオブジェクトのコンテキストメニューを引きだす。通常はセンターボタンを使うのに2ボタンのマウスでIpeを使用しなければならないときには、Shift キーをホールドし、右マウスを使えばよい。

6 オブジェクトの型

あるページにおけるオブジェクトには5種類ある。

- パス・オブジェクト (このオブジェクトにはストロークによるもの、フィルによるものすべてを含む。(これには (poly)lines polygons、splines、splinegons、circles ellipses、circular 及び elliptic arcs、rectangles がある)。
- テキスト・オブジェクト

- イメージ・オブジェクト
- グループ・オブジェクト
- 参照オブジェクト（これはそのページのある点においてシンボルが使われている場合）である。

パス・オブジェクトとテキスト・オブジェクトは Mode → 該当オブジェクトにおいてキャンバスの該当位置で左マウスをクリックすると生成される (図 6.1)。グループオブジェクトは編集モードでグループ機能により生成される (Edit → Group)。イメージ・オブジェクトは Insert image ipelet を用いて文書に付加される。reference オブジェクトは mark mode を用いるか symbols ipelet を用いて生成される。

6.1 パス・オブジェクト

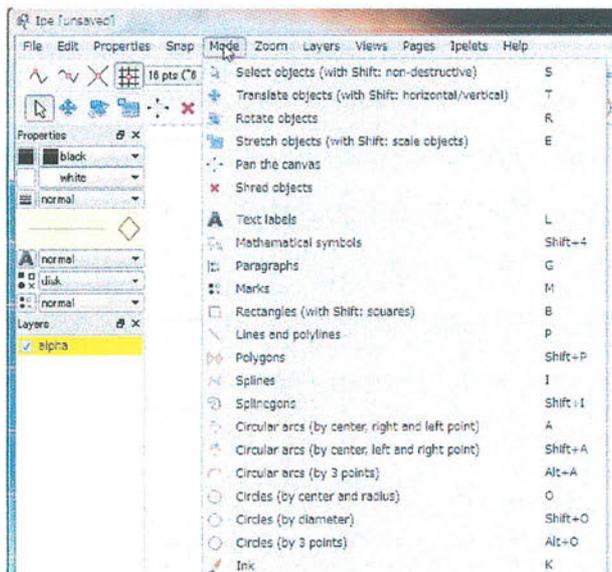


図 6.1: パス・オブジェクトとテキスト・オブジェクト

パス・オブジェクトはサブパス（要素）の集合である。すなわち、キャンバスに書かれた曲線の集合である。それぞれのサブパスは、直線のセグメント、パラボラセグメント（等価的には quadratic Bezier spline）、cubic Bezier spline、cubic B-spline セグメントから構成され、開曲線、もしくは閉曲線である。曲線はストローク色、点線スタイル、ライン幅、規定されたオブジェクトの内部のフィル色により作られる。

オープンとクローズのサブパスの差はストロッキングに限って意味がある。オープン・サブパスのフィル（塗り）については暗黙裡にクローズとなる。

サブパスの組のストロッキングは個別にストロッキングすることと同じである。ただし、フィル（塗り）については同じではない。各種のサブパスを用いると、穴のあるオブジェクト、さらにはもっと複雑なパターンを形成できる。フィル（塗り）のアルゴリズムは通常は Postscript/PDF の偶数奇数ルールによる。このルールはある点が塗りの中にあるかどうかを決めるために、その点から放射線を引き、放射線と交差するパス・セグメントの数を数え、この数が奇数なら点は内部にあり、偶数なら外部にあるとするルールである。

Ipe はパス・オブジェクトの最初と最後のセグメントにおいて矢をつけることができる。ただし、そのセグメントがオープン・サブパスの一部である場合に限る。

パス・オブジェクトを生成する Ipe のモードにはいろいろあり、異なる方法でできる。すべてのモードは、単一のサブパスのみから構成されるオブジェクトを生成する。穴があるよう

な複雑なパス・オブジェクトを作るには、境界になる部品を別々に生成する。次にそれらすべてを選択し、編集メニューの構成パス (Compose paths) 機能を用いる。逆動作は分解パス (Decompose Path) である。これはいろいろなサブパスを有するパス・オブジェクトのコンテキストメニューにある (図 6.2 最下部のマウス矢印)。

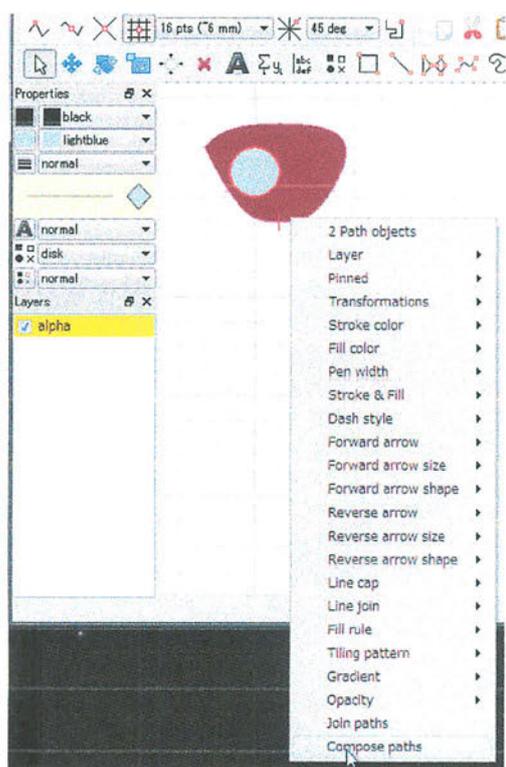


図 6.2: 穴のあるオブジェクト

曲線を付け加えることにより、複雑なパスを生成することもできる。これをするためには、一つのパスの最終端は次のパスの始端にほぼ同一 (スナッピングを用いてできる程度) でなくてはならない。曲線の付け加えは、それをしようとするパス・オブジェクトを選び、編集メニューの Join Path による。この方法で一つの閉じたパスに、いくつものオープンパス・オブジェクトを加えることができる。

円はツールバーにある 3 とおりの方法で、入れることができる。楕円の生成には、円を作り、伸張し、回転させる方法がある。円弧もツールバーの 3 とおりの方法で入れることができる。円弧の 3 点をクリックする方法、円の始端と中心点をクリックし、円弧の頂点を最後にクリックする。これはポストスクリプトのやり方で塗ることができ、矢もつけられる。また、円弧の伸張もできる。

円弧の共通の応用として図において角度をマークすることがある。スナップキーがそのような円弧を生成するのに有用である。

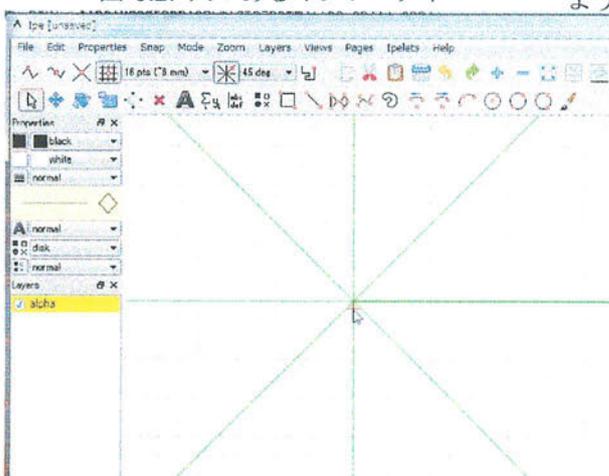


図 6.3: スナップライン設定

まず、スナップのためにポイントを選び、F1 キーを押す。図 6.3 のようにスナップラインが示される。スナップの条件はメニューバーで予め設定しておく。中心と 2 点の円弧 (center & 2pts) の生成のため、メニューバーで Shift+A とする。スナップライン上でクリックし、スナップする。境界にスナップする。最初に角度の中心点をクリックし (これがスナップ作用である。)、次に 2 つの境界線でクリックする。図 6.4 に例を示す。スナップラインをリセットするには、Ctrl+F1 とする。さらに複雑なパス・オブジェ

クトを生成するには、2つのモード（ラインモード、ポリゴンモード）がある。これらモードの違いはラインモードはオープンパスを生成し、ポリゴンモードは閉パスを生成することである。結果として、ラインモードは現在の矢の設定を用い、ポリゴンは用いない。

[直線]

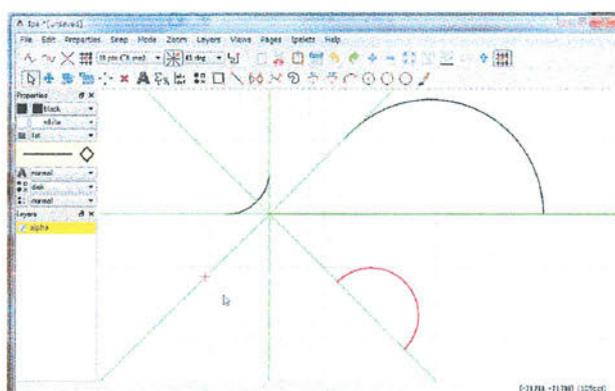


図 6.4: スナップラインを用いた図

最初の2つの管理ポイントのポリラインモードにおいて2回クリック（1と2の管理点）する。次にキーqを押し、第3の管理（最後の）ポイント3を右クリックする。図6.6に例を示す。

3次元のBezier スプラインセグメントを加えるには、最初の3つの管理ポイントに関するポリラインモードにおいて3回クリックする。キーcを押し、最後の管理ポイントをクリックする。

[円弧]

キー”p”を押し、ポリラインモードにする。最初は円弧の始点A、次に想定する円弧の切線の延長ポイントBでクリックする。キー”a”を押し、円弧の最終管理ポイントCで右クリックする。円弧は始点の切線から始まることが図6.7に示される。

キー”p”によりスタートする。左マウスが始点となり、右マウスで終点となる。

[B-spline]

B-スプライン曲線とは、与えられた複数の制御点から定義される滑らかな曲線である。キー”i”により B-spline にスイッチできる。左マウスを押して、制御点を決める。図6.5に例を示す。点、A~Dは制御点である。

[放物線]: 2次のBezier スプラインセグメント

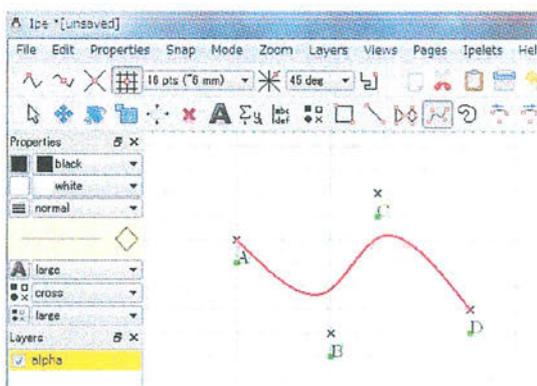


図 6.5: B-spline による図

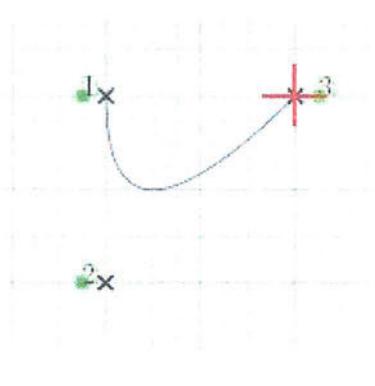


図 6.6: 放物線

[同一の切線を有するセグメントの接続]

[1] 異なったタイプのセグメントで同一の接線を有する曲線を作るには、新しいセグメントを作る点でキー”y”を押す。これによりセグメントの最初の点を中心とする座標が設定される。

そして前のセグメントへの接線に接続する。

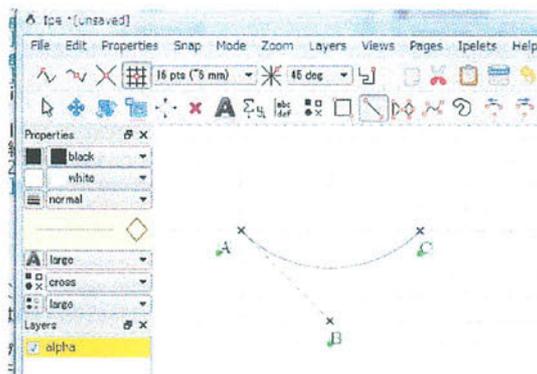


図 6.7: 円弧

[2] 2 次の Bezier スプラインセグメントのテスト

[A]. F5 を押し、[Snap to boundary] のモードにし、キー”p”を押し、ポリラインモードにする。

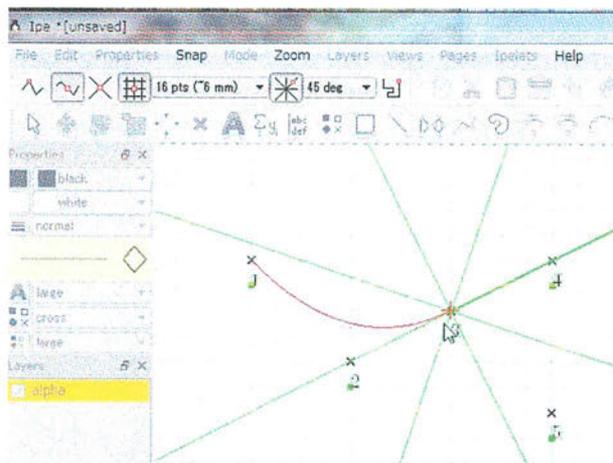


図 6.8: Bezier スプライン曲線の切線

[B]. 左クリックを 2 回行い (”1”と”2”の管理ポイントを指示)、キー”q”を押す。次に管理ポイント”3”をクリックし、ここで、キー”y”を押すと、スナップラインが現れ、切線方向のスナップラインが表示される。図 6.8

[C]. 切線の延長上の管理ポイント”4”を左クリックで指示する。これは B における管理ポイント”2”に相当する。

F1+Ctrl でスナップラインをリセットする。次の管理ポイント”5”が指示できる。ここで、右クリックすると終了する。右クリックしないで、キー”y”を押すと、[B]に戻る。管理ポイント”4”が切線上にあることが確認できる。図 6.9

このようにして、切線方向を継続してなめらかな曲線を継続できる。

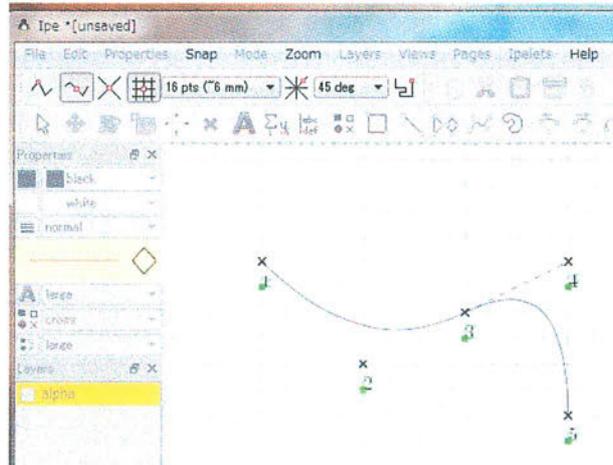


図 6.9: Bezier スプライン曲線の終了

数学的表現を含むのであれば、サブパスについてより詳細な記述が可能になる。詳細は Foley 他²によること。

サブパスはセグメントが連なったものである。各セグメントは直線セグメント、楕円、4次元 Bezier スプライン、3次元 Bezier スプライン、均一な2次元 B-スプラインである。

p_0 、 p_1 及び p_2 により定義される Bezier スプラインは次の曲線である。

$$P(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2$$

ここで、 $0 < t < 1$ 。

これはライン $p_0 p_1$ に接する点 p_0 からスタートし、ライン $p_1 p_2$ に接する点 p_2 を終点とする。3点で囲まれる凸な殻に含まれる。いかなる放物線のいかなるセグメントも4次元の Bezier スプラインで表現できる。

例えば、 $x = a$ と $x = b$ で囲まれる放物線 $y = x^2$ は次の管理ポイントで生成される。

$$\begin{aligned} p_0 &= (a, a^2) \\ p_1 &= \frac{a+b}{2}, ab \\ p_2 &= (b, b^2) \end{aligned}$$

どの放物線のどの部品であっても、これらの点に対して affine transformation(アフィン変換)により生成することができる。

管理点 p_0, p_1, p_2, p_3 を有する3次元の Bezier スプラインは次の曲線である。

$$R(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3$$

² J.D.Foley, A. Van Dam, S.K. Feiner, and J.F. Hughes, Computer Graphics: Principles and Practice, Addison-Wesley, 1990

均一な 3 次元 B スプラインは $m + 1$ の管理ポイント $p_0, p_1, \dots, p_m, m \geq 3$ の順列であつて、 $m - 2$ の多項式曲線セグメント $s_0, s_1, s_2, \dots, s_{m-3}$ から構成される。それぞれの該当曲線セグメントは 4 つの管理点により定義される。事実、曲線セグメント s_i は点 p_i, p_{i+1}, p_{i+2} 及び p_{i+3} により定義される。もし曲線が閉曲線なら (splinegon)3 つの付加的な曲線セグメント s_{m-2}, s_{m-1}, s_m を含み、順列の終端に p_0, p_1, p_2 を付加することにより定義づけられる。Ipe のパス・オブジェクトのオープンサブパスにおける均一な B スプラインセグメントは最初と最後の管理点を 3 回繰り返し、セグメントの最初と最後を 3 点でつくることになる。

セグメント s_i は以下のパラメータ化を含む 3 次元の曲線セグメントである。

$$Q(t) = \frac{(1-t)^3}{6} + \frac{3t^3 - 6t^2 + 4}{6} + \frac{-3t^3 + 3t^2 + 3t + 1}{6} p_{i+2} + \frac{t^3}{6} p_{i+3}$$

ここで、 $0 \leq t \leq 1$ 。

点 $Q(t)$ は 4 つの管理点による凸形状を形成するので、曲線セグメント s_i の組合せは p_i から p_{i+3} に至る凸点による殻形にある。さらに、管理点にアフィン変換を適用することにより曲線に応用することができる。管理点 p_i は 4 つの曲線セグメント $s_{i-3}, s_{i-2}, s_{i-1}$ 及び s_i においてのみ影響度を有する。かくして、スプライン・オブジェクトを編集し、さらに管理点を移動させるときには管理点の近傍のスプラインの短いピースのみが移動する。

6.2 テキスト・オブジェクト

テキスト・オブジェクト³ は、単純なラベルとミニページの二つがある。実際には、さらに変化して、ページタイトルとしてのラベルとページの全幅にわたるミニページに分かれる

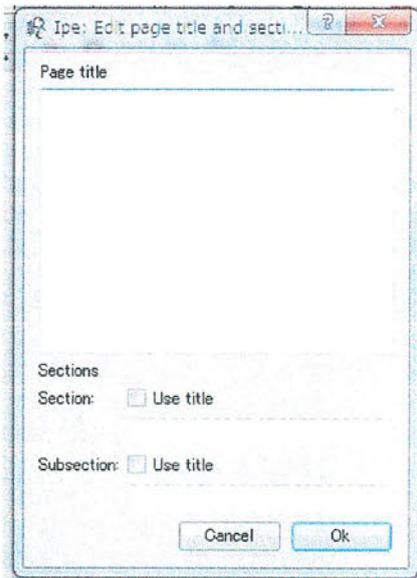


図 6.10: タイトル入力ウインドウ

ラベルオブジェクトを生成するために **Ctrl+P** をタイプする。ポップアップウインドウが現れ、 \LaTeX のソースコードを入力できる。図 6.10。テキストコマンドを入力してもよい。その場合に、クリックする点は、通常最も左端のベースラインポイントである。(ただし、これはオブジェクトの水平、垂直機軸を変更すれば移動できる。)ミニページオブジェクトは単純なテキスト・オブジェクトではない。その幅が定義事項の一部となっている。ミニページ・オブジェクトを生成するときには、ミニページのための水平セグメントをドラッグアウトしなければならない。これはミニページのトップページとして用いられるが、テキストを書くにつれ下方に伸張する。ミニページは \LaTeX のミニページ環境を用いてフォーマットされている。 \LaTeX は可能な限り与えられた bounding box を満たすようにする。center 環境、lemma その他の多くをミニページで行うことができる。テキストボッ

³ テキストを選び、 \LaTeX コードを *Run LaTeX* によりコンパイルすると、緑色の点が付着する。カラーページでなければ問題はないようである。MikTeX と GSVIEW の相性の問題かもしれない。

クスを生成するには単に F10 キーを押す。Ipe は自動的にオブジェクトがページの全幅になるように置く。図 6.11 にコンテキストメニューを示す。左の窓には `normal`, `center`, `itemize`, `item` の 4 つがある。右の窓はテキストのサイズを規定する。左の窓の `normal` は通常のテキストが書ける。左上部から出現する。`center` では中央上部に出現する。`itemize` とすると、レイアウトの設定はスタイルシートにより決まり、サイドのスペースも決まる。そしてすでにあるテキストボックスの下部に置く。これは多くのテキストがあるプレゼンテーションを生成するときや、交互に現れるアイテムについて便利である。

タイトルオブジェクトは Ipe が自動的に管理する。これらは Page メニューの編集タイトル項を用いて生成される。色彩、サイズ、整列、位置などはスタイルシートにより決定される。`makebox`(ラベル用) 内部もしくはミニページ内部において正しいいかなる $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ コマンドも用いることができる。。

ハイパーリンクとか外部イメージを含むコマンドのような PDF の非線形変換は使用できない。

テキスト・オブジェクトにおいて `textcolor` コマンドを用いてテキスト・オブジェクトにおいて色彩を使用できる。

例えば、

```
This is in black. \textcolor{red}
```

```
{This is in red.} This is in black.
```

現在のスタイルシートのすべてのシンボリックカラーは `textcolor` のコマンドのアーギュメントとして使用できる。また、絶対色彩を用いることもできる。

例。

```
This is in black. \textcolor[rgb]{1,1,0}{This  
is in yellow.} this is in black.
```

付加的な $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ パッケージにある $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ コマンドを必要とするなら、編集メニューにある Document Properties に設定できる $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ プレアンブルにある `usepackage` を用いることができる。

テキスト・オブジェクトを生成あるいは編集後に、Ipe のスクリーンは $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ のソースの始まりを示す。オブジェクトの PDF/Postscript 表現を生成するために File メニューから Run Latex を選ぶ。これによりすべてのテキスト・オブジェクトをただちに変換する。Ipe はその後にはテキストの正しい表出をする。

もし、Latex 変換においてエラーとなった場合には、Ipe は自動的にログファイルを生成する。問題を解決できなければ、8.3 章のトラブルシュートを見てほしい。

ギリシャ文字はテキストオブジェクトにおいて数式モードで使用できる。漢字、日本語、などのユニコードを使用したければ、必要なスタイルシートとフォントを設定することができる。(8.5 章)

Ipe があるテキストの bounding box の計算をするときには LaTeX で与えられた諸元に従う。時には絵文字の場合には与えられた諸元を越えることがある。この場合には下図のように 'A' と 'G' は色つきの枠の上からはみ出し、'y' は下にはみ出す。実際にはみだすか否かは LaTeX

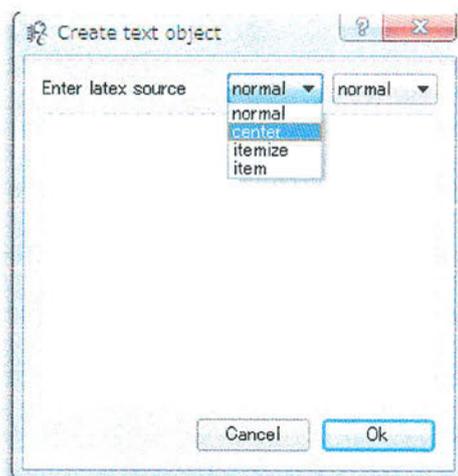


図 6.11: テキストボックスのコンテキストメニュー

によるので一概には言えない。

A Glyph

GSView により図を見ながら bounding box⁴ のサイズを調節することにより、下図のようにはみ出さないようにもできる。

A Glyph

6.3 イメージ・オブジェクト

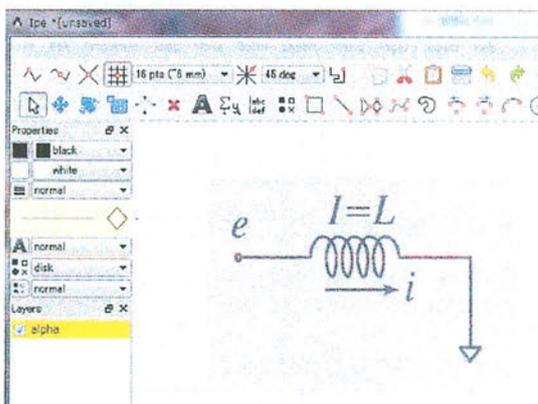


図 6.12: イメージの挿入

である。クリップボードのイメージをペーストすることもできるが、JPEG についてはできない。

Insert bitmap は JPG ファイルを許容しない。

イメージは、PDF フォーマットにおいて効果的にストアされる。Ipe において JPEG のほとんどの写真について PDF 化できる。Ipe は 7 ビットのポストスクリプト・ファイルを作り出すのでポストスクリプトでセーブするのは効率的ではない。

イメージは **File** → **Insert Images** により挿入できる。イメージは伸長、縦横維持・変更、回転できる。図面のスキャンを行い、Ipe に挿入することもできる。(図 6.12)

イメージの挿入には 3 通りの機能：一般的なイメージ、JPEG、クリップボードからの挿入がある。

用いようとするイメージが PNG、GIF、BMP のような画質の劣化がないビットマップ・イメージなら最初の機能が適している。

JPEG(JPG) フォーマットなら 2 番目が適当

6.4 グループ・オブジェクト

グループ・オブジェクトは任意のオブジェクトを選び、編集メニューからグループ機能を用いてグループにできる。グループ・オブジェクトは単一のオブジェクトのように取り扱える。

⁴ bounding box はエディターにより図を読みだせば、知ることができる。例えば、上の画像ファイルをエディターで読みだす。

グループ・オブジェクトを修正するには、Ungroup 機能を用いて、いったん部分に分解しなければならない。

グループ・オブジェクトにクリッピング・パスを付け加えることができる。このグループはそのパスにクリップされる。クリッピング・パスの外部からは何も持ってこられない。クリッピング・パスを加えるためには第一選択 (*primary selection* としてグループを選び、第二選択としてグループのコンテキスト・メニューから *Add clipping path* を選択する。⁵

6.5 参照オブジェクトとシンボル

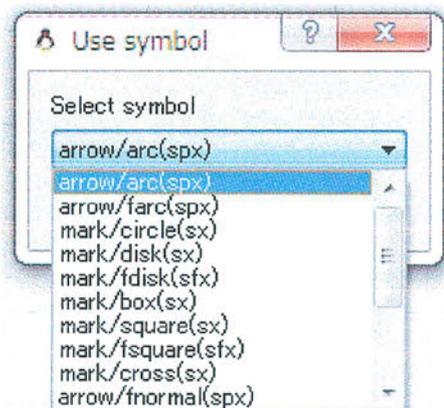


図 6.13: シンボルの利用

シンボルはドキュメントのスタイル・シートにおいて定義される単一の Ipe オブジェクトである。もちろん、これはグループ・オブジェクトであっても良い。参照オブジェクトはシンボルの参照である。シンボルはストロークと塗潰し色、ペン幅、シンボルサイズによりパラメータとして取り扱える。シンボルがパラメータを受け取るには名称の決まりを守らなければならない。名称は"s"、"f"、"p"、"x"のいずれかを含み、かっこでくくっていなければならない。パラメータとされるシンボルの参照によりシンボルが受け取るすべての属性が含まれる。(図 6.13)。シンボルを選択するには、*Ipelets* → *Symbols* とする。

7 スナッピング

Ipe の素晴らしい機能の一つに他のオブジェクトに対してマウスによるスナッピングができることがある。画面にあるものが磁性を帯びたかのような特性を持ち、オブジェクトを相互に整列させることや、+新しいオブジェクトを今画面にあるオブジェクトに対し、適切な位置に置くことができる。

スナッピングには、3つある。グリッド・スナッピング、コンテキスト・スナッピング、角度スナッピングである。

一般的に言って、スナップツールバーからボタンを押すか、Snap メニューから必要な機能を選択すれば、スナッピング・モードになる。ボタンは独立であって、別々にオン・オフできる。ただし、スナッピング・モードが独立だということではない。キーボードによるショートカット・メニューを用いれば、必要なオブジェクトを描いている途中でスナッピング・モードを切り替えることができる。

スナッピングモードの一つをイネーブルにしているときには、カーソル位置近傍で小さなクロスが現れる。これは第二のカーソル *Fifi*⁶ である。Fifi はマウスがスナップされている点を

⁵ Ipe:Ver7.1.4 では機能しない。

⁶ *Fifi* は 1980 年代に UNIX システムに組み込まれたゲームにある追かける犬のことである。いつも足元を走り回るからである。

示す。

7.1 グリッド・スナッピング

グリッド・スナッピングとはマウスの位置が最も近いグリッド（格子点）に切り上げられることを意味する。格子点とはその座標がスナップ・フィールドで設定される *grid size* の整数倍である。可能な格子サイズから適当な値を選択できる。単位はポストスクリプト点である (E_T_PX では *bp* とする。) 1インチの 1/72 である。グリッドサイズはツールバーに示されている。

7.2 コンテキスト・スナッピング

コンテキストスナッピングがイネーブルになると現に画面上にある図が他のオブジェクトを引寄せることができるようになる。引寄せ点は屈曲点 (vertex) (F4)、境界 (F5)、交差部位 (F6) である。

マウスが対象となる図から遠すぎるとスナッピングはできない。スナッピング距離はツールバーから変更できる。

- 屈曲点 ポリゴン曲線、スプライン曲線、円もしくは楕円の中心、円弧の終端、マークポイント
- 境界 ポリゴン、スプライン、円、楕円、円弧の境界
- 交差点 パスオブジェクトの交差点

7.3 角度スナッピング

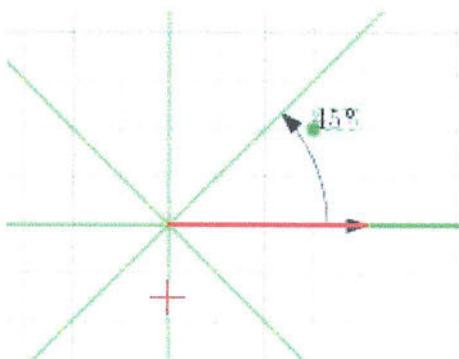


図 7.1: スナップライン

角度スナッピングをイネーブルにすると、マウスは設定された座標システムに制約される。マウスを動かせるのは基底となる方向の角度に限られる。かつ、基底となる方向はスナップ角の整数倍である。スナップ角はツールバーで設定でき、値の単位は (度) である。45° の場合には、図 7.1 のようになる。ここでは、基底線は水平 (→) としている。

角度スナッピングの原点はマウスで指示し、F1 キーを押せばよい。スナップ角度が 180° であれば、単なる直線になる。原点を動かすと、角度スナッピングは無効になる。

角度スナッピングの基底角度を変更するには、変更したいとする通過点にマウスを置き、F2 を押す。図 7.2 は図 7.1 の基底軸を図のクロス点を通るように変更した例である。一旦、原点と基底、角度が設定されると、伸張、回転等の作業は新規座標システムに基づいて実行される。

Shift-F2 を押すと、現行の座標システムはリセットされる。角度スナップングもリセットされる。

マウスがポリゴンオブジェクトの端点の近傍にあるときに、**F3** を押すと、原点と基底方向が同時に設定できる。原点は端点の終点に基底角は端点の方向に一致する。この方法は与えられた端点に並行するオブジェクトに有効である。

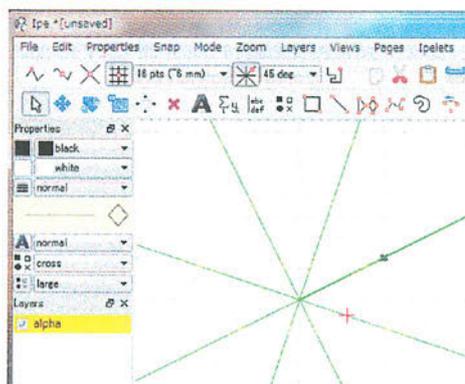


図 7.2: スナップライン基底角度の変更

オブジェクトを描く自動角度スナップングを用いることができる。

スナップングモードのもう一つの利点は、ポリゴンオブジェクトを作り出すまでは何もしないことである。たとえ自動角度スナップングをスタートさせていても、自由にポリゴンの最初の点を自由に決められる。次に残る頂点は c-oriented ポリゴンを構成するように適切に整列される。

自動角度スナップングモードはポリゴン以外のオブジェクトには何らの働きもしない。特に、既定の方向にオブジェクトを移動させようとする際には、通常の方法スナップングを用いなければならない。

最後に、角度スナップングでできる多くのことはコンテキストスナップングを用い、補助線を使うことでも可能である。どちらが良いかはどちらのモードがより適当しているかの判断にかかっている。

7.4 スナップング・モードの相互作用

すべてのボタンを押しているからと言って、すべてのスナップングモードが同時にアクティブになることはない。ここでありうる相互作用とスナップングの優先度について詳しく見ることにする。

角度スナップングがある場合には、座標軸により設定された軸にマウスの位置が制限される。それゆえ、角度スナップングでは屈曲点スナップング (F4)、交差スナップング (F6)、格子スナップング (F7) は無効となる。

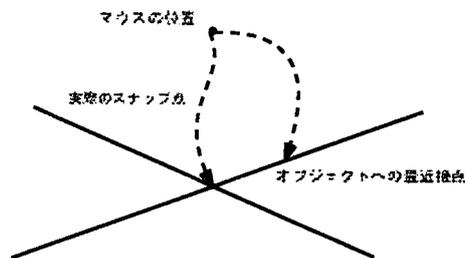
一方で、角度スナップングモードでは、境界スナップング (F5) を用いることは可能であり、正確に実行される。この2つのモードが同時に On とされる場合、IPE はオブジェクトの境界

とスナップラインとの交差点を計算し、マウスがスナップライン上にあり、交点に近い所で、マウスは交点にスナップされる。

2つの角度スナップモードは同時に同様の仕方で存在しうる。角度スナップ、自動角度スナップのいずれもイネーブルの時、IPEは2つの原点とそこでのスナップにより規定されるスナップラインとの交差点を計算する。もし、スナップラインが並行であったり、一致したりすれば自動角度スナップが用いられる。

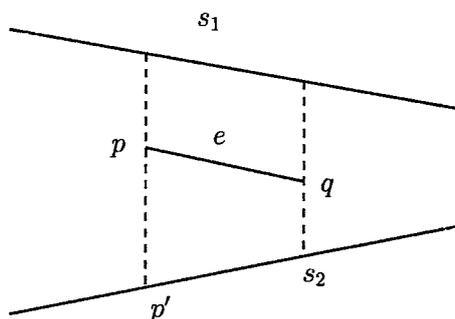
角度スナップモードがアクティブな時には、IPEには選択枝が3つある。まず、IPEは最近接の屈曲点もしくは交点が十分に近接しているかどうか点検する。この条件が満たされない場合には、最近接の境界屈曲点が決定される。もし、この点も遠すぎればIPEは格子スナップに切り替わる。

このことはスナップがオブジェクトに最も近い点を決定するのではないことを注意してほしい。特に2つの直線が交わる時には、最近接点は交点ではありえない。以下の図を参照されたい。



7.5 例題

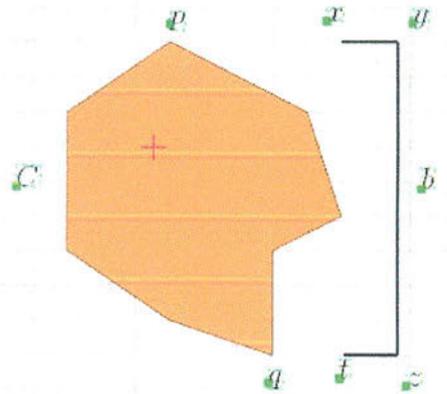
例1 図のようにセグメント s_1 、 s_2 、 e がある。 p 点および q 点を通る鉛直線を描け。



- **F4** と **F5** をセットし、ラインモードを設定する。**Shift-F2** とし、一旦座標軸をリセットする。
- p の近傍で **F1** と **F8** をセットし、原点を定め、角度スナップモードを設定する。

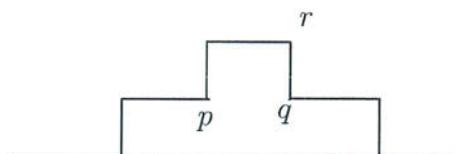
- p' の近傍で s_1 までラインを引く。
- q の近傍で、**F1** を押し、原点設定をする。同様にして、 s_2 からの線を引く。
- **Ctrl+F1** により、スナッピングモードをリセットし、 p 、 q を通るセグメントで、右クリックし、コンテキストメニューから点線化をする。

例2 図のポリゴン C がある。垂直軸の頂点と低点を示すブラケット b を描きたい。



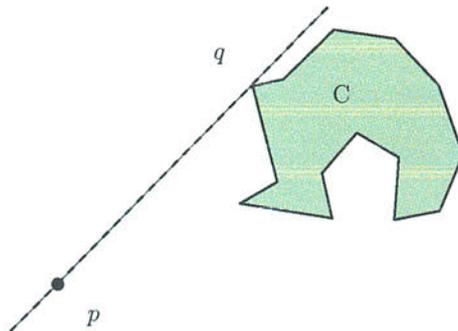
- **F4** と **F8** をセットし、ラインモードを設定する。**Shift-F2** とし、一旦座標軸をリセットする。スナップ角を 90° にセットする。
- p の近傍で **F1** と **F8** を押し、原点設定と角度スナッピングの設定をする。
- x に行き、左マウスをクリックし、セグメントを y まで引く。左クリックする。
- マウスを q の近傍におく。 y からのコントロールラインをそのままにして、**F1** と **F8** を押し、原点をリセットし、角度スナッピングにする。 q を通り、 z に至る水平線が得られる。コントロールラインの終端を y からの下垂線となるようにして、右クリックし $y \rightarrow z$ の線を引く。これで、2つの角度スナッピングがオンになる。スナップラインは z で交差する。
- z で左クリックする。 x に行き、**F1** を押し、 t まで引き、ブラケットを完成させる。

例3 図のようにスカイラインを描きたい。問題点は点 p と点 q の水平線が合致していることである。



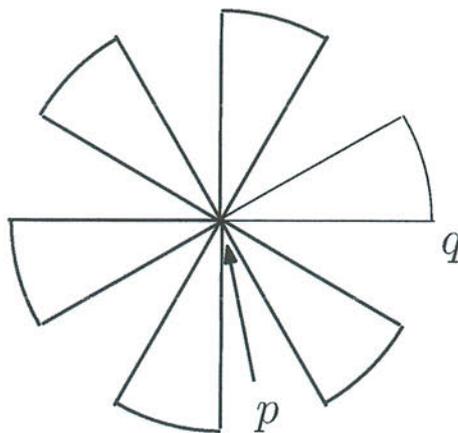
- **F9** をセットし、水平線を設定する。
- p 点で境界スナップリング (**F5**) とし、 r まで自動角度スナップリングにより直角線を引く。スナップリング角は 90° にする。
- p 点で、**F1**、**F8** を押し、 q でスナップラインが交錯するようにする。**Shift-F2** により、角度スナップリングを終了させる。最終ポイントは境界スナップと一致する。

例4 ポリゴンCに対し、 q で接し p を通る線を引け。



- 屈曲点 (**F4**) スナップリングをオンとし、 p を原点とするように **F1** を押す。
- q に行き、**F2** を押す。これにより基底線が p から q にひかれる。
- **F8** で角度スナップリングとし、線を引く。

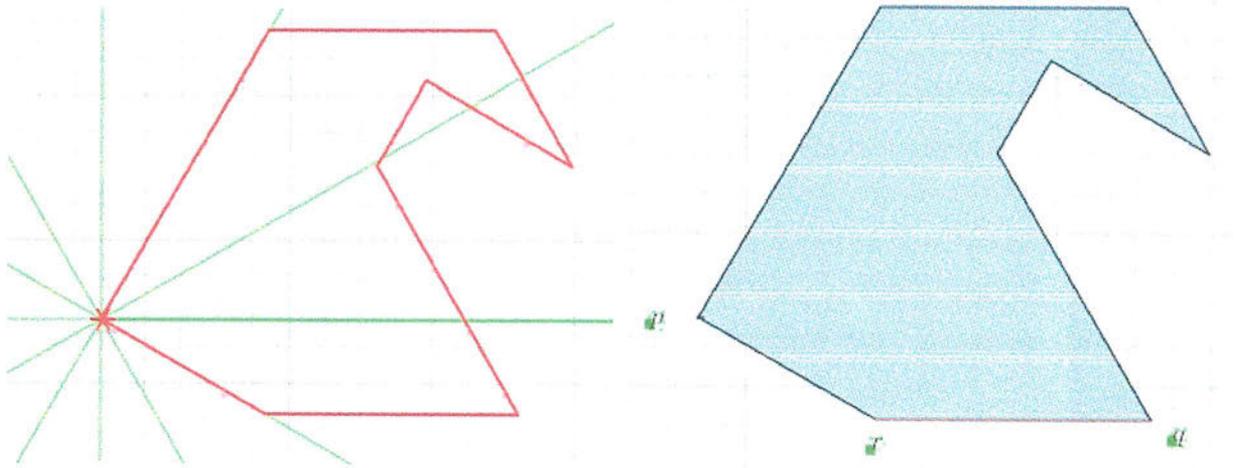
例5 次のような風車を描け。羽の相互の角度は 30° とする。



- 屈曲点 (**F4**) スナップリングをオンとし、スナップ角度を 30° にする。**Shift-F1** により座標軸をリセットする。

- p に行き、**F9** により角度スナップングを設定する。ラインモードによりセグメント pq を引く。
- p に行き、**F1** により原点設定をし、**F8** により、角度スナップングモードを設定する。
- セグメント pq を引く。
- q で edit → Duplicate とし、 30° 回転させる。
- p 点で円弧モードの”center & two points”を選択する。
- p 点を中心とし、 q 点から次の分枝点まで円弧を引く。
- edit → Select All → Group とする。
- 角度スナップングの状態では edit → Duplicate とし、 60° 回転させる。全周させるまで繰り返す。

例6 c-oriented ポリゴンを描きたい。連続する角度は 30° の倍数とする。Automatic Snap[F9] を選択した上で、ポリゴン描画 [Shift+P] をするのであれば、作図は容易である。ここでは、ポリゴン描画の機能を用いず、c-oriented ポリゴンを描きたい。この場合の問題点は、 $q \rightarrow r \rightarrow p$ のラインを描くことである。 r を決めると同時に p とどのように整合させるかの方法が問題となる。



- スナップ角を 30° に設定し、Automatic Snap[F9] をオンにする。
- ラインモード [P] にし、 p 点から q 点まで描く。最終点の2つ前である。
- r 点まで線を延ばす。 q 点及び p に対してあるべき位置であること。Automatic Snap[F9] を維持していれば、点 q に対してはスナップ角は維持できている。 r で右クリックしないようにすること。
- マウスを p 点まで持っていき、[F4] を押す。 p 点を原点とするため、F1 を押し、角度スナップングとなるように、F8 を押す。 r が定まる。

8 スタイルシート

Ipe 文書におけるシンボリック属性は、その文書に付属するスタイルシートにより解釈のために絶対値に変換される。文書は複数のカスケード状のスタイルシートを有することができる。シートはスタックを構成し、シンボルは先頭から底部まで閲覧される。いかなるスタイルシートのカスケードであっても、その底部では Ipe に組み込まれる最少要件の標準のスタイルシートが存在する。

新規文書の作成をスタートすると、自動的に標準スタイルシートがコピーされる。(さまざまな属性についての正規の属性を定義するだけである。) さらに、Ipe は事前に定義されたスタイルシートを挿入する。これらのスタイルシートのリストは Ipe のコマンドラインオプションと環境変数により、*ipelet* を用いれば、希望に従い変更できる。デフォルトでは新しい Ipe 文書には Ipe の基本スタイルシートが入る。

スタイルシート・ダイアログ (スタイルシートの編集メニューにある。) により、作者の文書に関連するスタイルシートのカスケード (階層) を点検できる。スタイルシートは加除訂正が可能であり、独自のスタイルシートのセーブもできる。

文書のスタイルシートでは Ipe のユーザー Ipe のユーザーインターフェースにあるシンボリックの選択もできる。たとえば、Ipe が色彩、ペン幅等々について適切でないと感じた場合には、独自のスタイルシートを作成することもできる。Ipe のスタイルのディレクトリには例もあって、みることもできる。*color.isy* スタイルシートは X11 色彩データベースのすべての色彩を規定し、自らの用途に合わせて選定できる。

Ipe の文書にスタイルシートが加えられると、スタイルシートファイルの内容が Ipe 文書にコピーされる。スタイルシートファイルの修正は Ipe 文書には何らの効果も及ぼさない。自身のスタイルシートを修正する正しい方法はそれを再び加えるかスタイルシートのカスケードから古いコピーを削除するか、あるいは、編集メニューでスタイルシートの機能を更新することである。この機能はスタイルシートファイルが文書と同一のディレクトリにあることを仮定している。また、ファイル名がスタイルシート名 (プラス拡張スタイルファイル) と一致していると仮定している。

スタイルシートの移動もしくは交換は文書のシンボリック属性を未定義の状態にする。これは事故ではなく、Ipe は単純に未定義のシンボリック属性になんらかのデフォルト値を与えるだけである。この問題を警告させるためには、Ipe はすべての未定義属性の警告リストを示す。

スタイルシートは用紙サイズ、フレームサイズの決定にもかかわっている。Ipe のデフォルト・用紙サイズは A4 である。いわゆるレターサイズを用いたい場合には、次のスタイルシートを用いること。

```
<ipestyle name="letterpaper">  
  <layout paper="612 792" origin="0 0" frame="612 792"/>  
</ipestyle>
```

[シンボル]

スタイルシートはマーク、矢印、後景パターン、ロゴ等々のシンボルを含むことができる。これらは文書から参照できる名称のある Ipe オブジェクトである。もし、読者の文書のスタイル

シートがシンボル名つきの後景を規定しているなら、すべてのページにそれが自動的に配置される。*Symbols ipelet* を用いて、シンボルを作成し、使用できる。例として、すこしおかしな後景を有するスタイルシートを示す。

```
<ipestyle name="background">
<symbol name="Background" xform="yes">
<text pos="10 10" stroke=black" size="LARGE">
Background text
</text>
</symbol>
</ipestyle>
```

`xform` 属性の使用は、後景が1回だけPDF文書に実現されることに注意すること。文書がコンパイルされる場合には、相違が大きくなる。

シンボルをストロークカラー、塗カラー、ペンサイズ、シンボルサイズでパラメータを持たせるようにもできる。これはこれらの属性の実際の値がシンボルが文書で用いられる際に設定されるだけであり、シンボルの定義で決まるのではないことを示す。

パラメータつきシンボルの名称は次のいずれかの文字を含む両閉じかつこで終わらなければならない。`"s"`(stroke)、`"f"`(fill)、`"p"`(pen)、`"x"`(symbol size) かつ順番はこの順である。

また、付加的なマーク、矢型、傾きを定義するスタイルシートを用いることができる。

Latex preamble スタイルシートに文書の `LATEX` preamble を定義することができる。テキストオブジェクトが `LATEX` により処理されるなら、用いられる `preamble` はスタイルシート階層の一部を成す。これには文書それ自体の `preamble` が続く。⁷

Transparency もし、文書内で透明度の使用を望むならスタイルシートで透明度の値の定義をしなければならない。デフォルトのスタイルシート `basic.is` は透明度の定義はしない。透明度を用いる文書はポストスクリプト・フォーマットではセーブできないからである。透明度を定義する最小限のスタイルシートは以下のとおりである。

```
<ipestyle name="transparency">
<opacity name="50%" value="0.5"/>
</ipestyle>
```

PDFのプレゼンテーションのためのスタイルシートは次項で述べる。

9 プレゼンテーション

⁸Ipeのプレゼンテーションとは具体的には、Acrobat Readerとビデオプロジェクターを用いて行う、IpeによるPDF文書である。Ipeはこのようなプレゼンテーションを容易に行えるようにするいくつかの特徴を持っている。

⁷いまの所、テストは不成功

⁸テスト未了

Dimitriv Morozov による IpePresenter が参考になる。IpePresenter はあるスライドを外部スクリーンに投影し、自身のディスプレイにはタイマーを始め、次のスライドや現在投影しているページのノートや次のスライドも示す。

Ipe の Windows 用バイナリーパッケージには IpePresenter が含まれている。

9.1 プレゼンテーションのスタイルシート

プレゼンテーションにはそれ用のスタイルシートがなければならない。プレゼンテーションでは通常の図に用いるよりも、大きなフォントを使う必要がある。単にテキストサイズを”LARGE”とか”huge”にするのではなく、それ用のスタイルシートを用い、”normal”で大きなテキストサイズとなるように定義しておくこと。

Ipe はプレゼンテーションに用いられるスタイルシート *presentation.isy* を装備している。プレゼンテーションをするには、単に、次のように書けばよい。

```
ipe -sheet presentation
```

presentation.isy を使用する際は *basic.isy* は不要となることに注意すること。

このプレゼンテーション・スタイルシートはすべての標準サイズを 2.8 倍に拡大する。*textstretch* 要素はテキストの拡大に用いることに注意のこと。

```
<textstretch name="normal" value="2.8"/>
```

```
<textstretch name="large" value="2.8"/>
```

Ipe のユーザーインターフェース (例、”*large*”) からテキストサイズを選択することは事実上は 2 つのシンボリック属性、すなわち *textsize* (例えば、*large* は `\large` に変換される。) と *textstretch* (標準のスタイルシートでは、なんら伸張しない。) テキストストレッチを選択すると、フォントの拡大ができる。

さらに、このスタイルシートの *layout* 要素はペーパーサイズをプロジェクターに適合するように定め、ペーパーのより小部分をフレームとして定義する。フレームはコンテンツのために用いられる。*inset text box* 機能はフレームの幅を正確に満たすオブジェクトである。

<titlestyle> 要素はページの *<title>* のスタイルをフレームの外側で定義する。Page メニューの”Edit title &”を用いて各ページにタイトルを設定することができる。

さらに、リスト環境をより余白が少なくなるようにし、テキストスタイルによりパラグラフが左右されないようにする。(*<textstyle>* 要素)。 *<preamble>* 要素において定義される IAT_EX の *preamble* は標準フォント型を *<cms>* (Computer Modern Sans Serif) に再定義する。スクリーン上では多くの人はサンセリフ書体が読みやすいと感じているようである。

Acrobat Reader の *page transition* 効果を利用したいのであれば、 *<effect>* 要素を用いて、スタイルシートにより、ビューメニューにおいて *<Edit effect>* により設定できる。

9.2 ビュー

Acrobat Reader による PDF のプレゼンテーションをするとき、しばしば、ページごとに提示したいことがある。たとえば、ポリゴンを示したいとする。次に、三角分割をし、最後に頂点に色づけしたいとする。ビューはこれをうまくやってのける。

Ipe 文書はいくつかのページから構成されている。それぞれは任意の数のビューから成る。PDF をセーブするとき、それぞれのビューは分離された PDF のページを発生させる。(もし、結果だけみるなら、2つのページが異なった Ipe のページなのか、あるいは、同一の Ipe のページの2つのビューなのかはわからないであろう。)

9.3 ブックマーク

Ipe の文書の表題と項目を各ページにつけることができる。これらは `<bookmarks list>(pages->Bookmarks)` でみられる。表題をダブルクリックするとそのページに飛ぶ。長い文書の場合には、便利である。

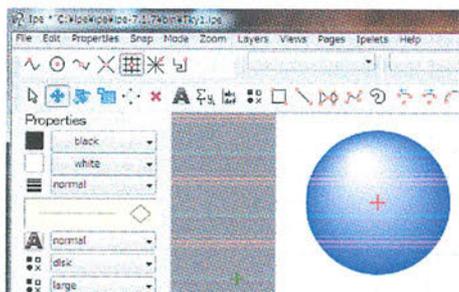
9.4 陰影のパターン

ガラスボールのように見えるシンボルの定義ができる。次のようなスタイルシートになる。

```
<ipestyle name="ball">
  <gradient name="ball" type="radial" coords="-4 10 2 0 0 18">
    <stop offset="0" color="1 1 1"/>
    <stop offset="1" color="0 0 1"/>
  </gradient>
  <symbol name="ball(x)" transformations="translations">
    <path fill="1" gradient="ball"> 18 0 0 18 0 0 e </path>
  </symbol>
</ipestyle>
```

次に `Use symbol` の `ipelet` によりシンボル `ball` を使用するように設定し、スタイルシートに `"ball"` が加わるように設定する。

`transformations="translations"`はこの図の伸張が固定されていることを示す。(x) という記述はマークのサイズ変更指示を利用してサイズ変更できることを示す。



9.5 テキスト・オブジェクトから用いられる Ipe のシンボル

省略

10 進んだ話題

10.1 Latex 文書と Latex 定義を共有する

省略

10.2 *ipelet* を書く

ipelet は Ipe の拡張である。Ipe7 はスクリプト言語 Lua を用いていて、スタート時に Lua で書かれた *ipelet* をロードする。詳細は IpeLib 文書を参照のこと。

10.3 LaTeX 変換のトラブルシューティング

省略

10.4 TrueType フォントの使用

省略

10.5 日本語処理

この項は英文マニュアルではユニコード・テキストと題されている。この日本語マニュアルでは日本語以外の話題は不要と思われるので、すべて省略する。ギリシャ文字については、`preamble` で必要なパッケージを導入しておけば、数学モードですべて使用できる。あえて UTF-8 を用いる必要もなく、Shift-JIS が使用できる。

日本語処理については、`preamble` に次のコードを加える。

```
\usepackage[whole]{bxcjkjatypex}
```

図 10.5 に示すように `Edit` → `Document properties` で `preamble` を例示のようにする。なお、最初の行以外は使用者の任意である。`preamble` 部分は utf-8 で書くこと。⁹

このまま、`Ipe\bin` に例えば、`Tky.ipe` のようにセーブし、日本語処理をしたい場合は呼び出し、`Edit` → `insert text box` により図 10.2 のように記入すればよい。あるいは、`Mode` → `text labels` によってもよい。コンパイラーは MikTeX なのでフォントは自由には選定できない。出てきたもので我慢するしかない。今後の進展に期待したい。フォントの大きさは `LATEX` の命令でかなり調節できる。

XELATEX ではかなりの範囲で日本語フォントの選択ができることされている。しかし、IPE7 は XELATEX は範囲外とされ、`pdflatex` で処理される。

⁹原マニュアルにはスタイルシートを加えることで可能なように書かれているが、現在のところは、うまくいかない。

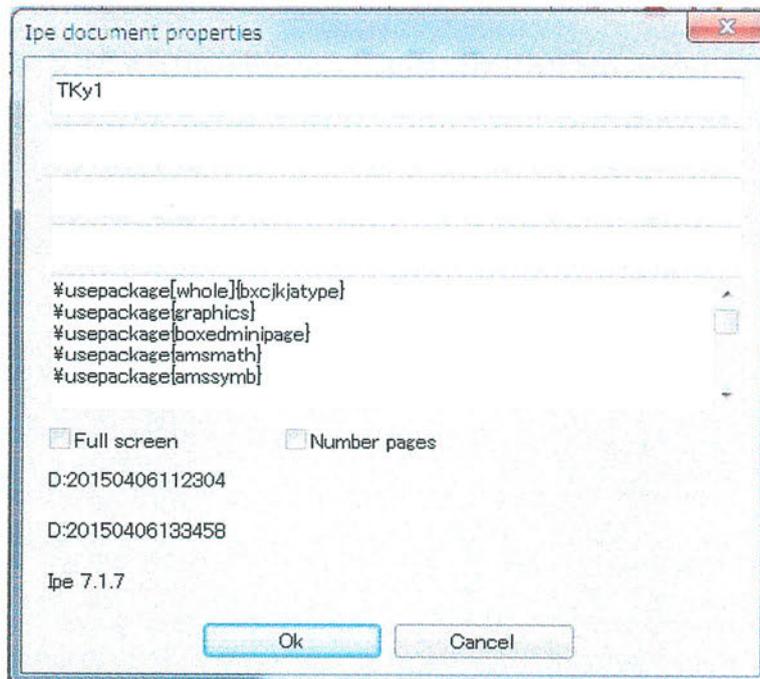


図 10.1: Preamble の規定

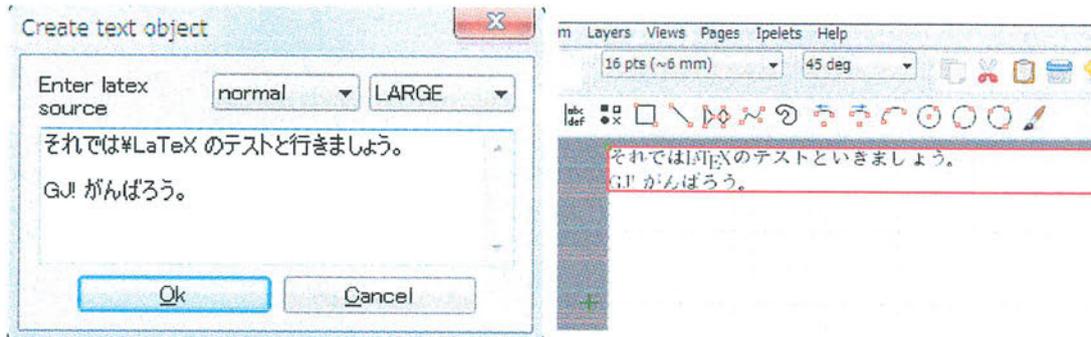


図 10.2: 日本語処理の例

図 10.3: 日本語処理の結果

11 謝辞

Ipe7 という素晴らしいソフトを公開くださった Odfried Cheong 氏に感謝いたします。

12 後記

Ipe7 は GNU Generic Public License に基づく Free Software とされています。Copyright は Odfried Cheong 氏にあることは、もちろんですが、GNU ライセンスのいう”Free”が、マニユ

アルの翻訳ないし抄訳も許可しているかどうかは不明です。したがって、この日本語マニュアルは許可の範囲にあるとの仮定に基づいています。問題を生じた場合、著者の理解を得るよう
に努力したいと思っています。

また、Ipe7 はまだベータ版であり、マニュアルにも説明がない沢山の機能が盛り込まれています。たとえば、複数のサブパスを設定後に `vertex` を操作し、変形させる機能などです。スタート画面でプロパティの下部に空白の部分がありますが、クリッピング パス操作を予定しているのではないかと考えられます。

この日本語マニュアルもベータ版であり、今後進化するものをご理解ください。おそらく多くの誤りの可能性もあると思っています。忌憚のないご指摘をお待ちします。以上