

K_ET_Cindy の開発について

東邦大学・理学部 高遠 節夫 (Setsuo Takato)
Faculty of Sciences,
Toho University
K_ET_Cindy 開発チーム (K_ET_Cindy developpers)¹

1 数学教材作成と T_EX

大学・高専（以下，カレッジ級）や高校における数学の授業では，教科書，黒（白）板，印刷教材，プレゼン提示，インタラクティブな授業などいろいろな教材や教育方法が利用されるが，これらのうち，授業担当者（以下，教員）が作成する印刷配付教材（以下，プリント）は重要な位置を占める．これらのプリントには，当然ながら数式入力が必要であるため，プリント作成のツールとしては，MS Word などのワープロ，StudyAid および T_EX がよく用いられている．数学プリントの要件とこれらのツールの使い勝手を掲げると以下ようになるであろう．ただし，T_EX については，ここでは標準的なパッケージを使用した L^AT_EX の場合に限っておく．

	Word	StudyAid	T _E X
数式入力	△	△	◎
数学記号	△	△	○
図	○	△	△
表	○	△	△
画像	○	×	○
レイアウト	○	○	△
手軽さ	○	○	△

言うまでもなく，T_EX は慣れれば慣れるほど数式入力が容易になって，プリント作成で大きなアドバンテージを持つことになる．実際，大学・高専教員の 8 割以上が T_EX でプリントを作成している．これは，彼らが論文作成等に T_EX を用いる機会が多いことも大きな理由であろう．一方，高校教員の T_EX 利用率はずっと低くて，多分 1 割程度である．彼らの多くは Word や Studyaid を使っている．Studyaid には問題データベースが付随していて，そこから問題を持ってくれば簡単に問題プリントを作成できる点が大きいが，T_EX は（１）インストールが大変，（２）コマンドを覚える必要がある，（３）文書入力の前（プレアンブル）の設定も煩雑である，（４）入力結果がすぐに見えず，エラー

¹阿原 一志（明治大学）入谷 昭（磐田南高校）大内 俊二（下関市立大学）金子 真隆（東邦大学）北原 清志（工学院大学）久保 康幸（弓削商船高専）小林 茂樹（長野高専）中村 泰之（名古屋大学）西浦 孝治（福島高専）野町 俊文（都城高専）濱口 直樹（長野高専）前田 善文（長野高専）牧下 英世（芝浦工業大学）松宮 篤（明石高専）山下 哲（木更津高専）

処理が難しい、などの敷居が高く、独力では無理と敬遠されていると考えられる。また、高校教員が多くの仕事を抱えていて、かつ同僚や先輩の教員が $\text{T}_\text{E}\text{X}$ を使わないなど高校での教育環境の問題もある。さらに、上記のうち、通常の $\text{T}_\text{E}\text{X}$ が数式入力以外の要件に不十分であることも大きな理由であろう。しかし、数式入力が容易かつ数式記号が多いことは $\text{T}_\text{E}\text{X}$ の最大の長所であり、自動番号付けなどの便利な機能も多く、慣れれば慣れるほど教材作成が容易になる利点もある。 $\text{T}_\text{E}\text{X}$ のソースファイルはテキストファイルであるため、ワードなどの文書ファイルに比べて長期にわたって再利用できる。すなわち「時空を超えた自他との共有」を可能とするのである。

前述のように、通常の $\text{T}_\text{E}\text{X}$ には弱点も多い。これらを克服するための方策として

- (1) 軽くてオフラインでインストールができる $\text{T}_\text{E}\text{X}$ を配布する。
- (2) 煩わしいプリアンプルを自動生成するツールを提供する。
- (3) 多くのコマンドを覚えるのでなく、必要なものはネットで拾うことを推奨する。
- (4) 図表の作成環境を充実させる。
- (5) レイアウトが自由にできるようなパッケージを作成・提供する。

などが考えられる。

図表は、数式の次に重要な要素であるが、 $\text{T}_\text{E}\text{X}$ を利用しているカレッジ級の教員でも適切な図表を入れたプリントを作成している人は恐らく3割にも満たない。 $\text{T}_\text{E}\text{X}$ での挿図の方法は、大きく

- (1) パワーポイント、Maple, Mathematica, Geogebra など他のソフトで作った図表を `\includegraphics` で入れる。
- (2) Emath, Tikz, Asymptote など、直接インラインで図を作成する。

の2つがあるが、前者には、(1) フォントなどが違い違和感があり、(2) 不要な要素がそのまま残ってしまっていて加除修正に苦労することが多い、(3) カラー画像はモノクロ印刷したとき美しくない、などの問題がある。特に(2)は数学プリント作成には決定的な欠点である。また、後者には、数学プリント作成に必要な機能が不足しているか、逆にコマンドが多すぎるといった問題があり、必ずしも使いやすくない。

2 $\text{K}_\text{E}\text{Tpic}$ の開発

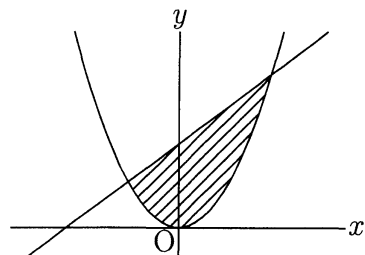
そこで、私たちは2006年から、MapleやMathematicaなどの数式処理ソフトやScilab, Rなどにより図のTpicコードを生成するパッケージ $\text{K}_\text{E}\text{Tpic}^2$ を開発してきた³。さらに、作表コマンド、メタコマンドをサポートして、レイアウトのためのマクロパッケージなども開発してきた。すなわち、現在では $\text{T}_\text{E}\text{X}$ の総合支援ツールに発展している。

$\text{K}_\text{E}\text{Tpic}$ は数学プリント作成にジャストフィットであり、正確で美しい図を作成できる。例えば、次の左側のコード(Scilabの場合)により、右側の図が得られる。ただし、冒頭に初期設定などが必要である。

²ketpic.com

³使用するコマンドはpn,pa,fp,ip,shのみで、他のコードへの移植も容易である。

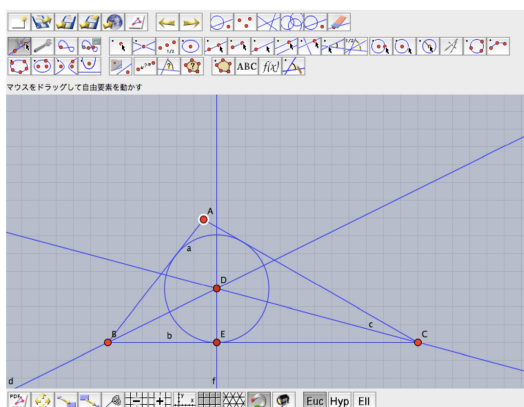
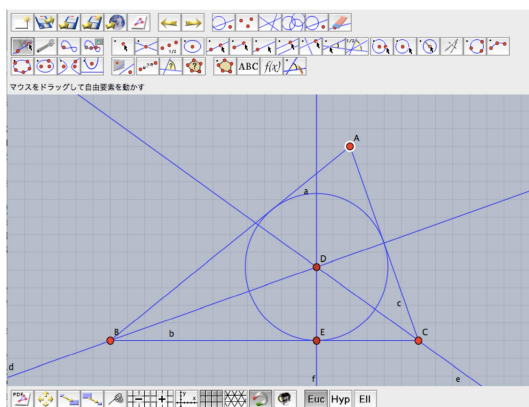
```
G1=Plotdata("x^2","x");
G2=Lineplot([-2,0],[2,3]);
Gh=Hatchdata("ii",list(G1,"n"),list(G2,"s"));
Beginpicture("picure.tex","1cm");
Drawline(G1,G2,Gh);
Endpicture("1");
```



スクリプトの可読性は高いので、慣れてくれば書きやすく再利用もしやすい。ただし、コマンドラインで全てを記述することは、やはり特に初心者にとって大きなハードルであった。そのため、 $\text{K}_\text{E}\text{Tpic}$ に GUI を付加してほしいとの要望が多く寄せられていた。

3 動的幾何 Cinderella と $\text{K}_\text{E}\text{Tpic}$ の連携

動的幾何ソフト (DGS, インタラクティブ幾何ソフト IGS ともいう) は、作図ツールの 1 つであるが、作図手順を記憶しているため、幾何要素を動かしても幾何的な性質を保つ点が特徴である。数学教育、特に中学校において、図形の普遍的な性質を調べる数学的探究活動のツールとして利用されている。1980 年代から Cabri などいくつかの DGS が開発されているが、Cinderella⁴ もその 1 つであり、1998 年にバージョン 1 が、さらに 2006 年にバージョン 2 が公開されている。



点 A を動かしても内接円は保たれる

上述のように、DGS は幾何教育では非常に有用なツールであるが、カレッジ級や高校の数学では、幾何そのものを扱う機会はそれほど多くはない。それでも日常のプリントには図が必要である。インタラクティブに作図できる Cinderella と、 $\text{T}_\text{E}\text{X}$ で正確な図を作成できる $\text{K}_\text{E}\text{Tpic}$ の両者に連携の可能性があるのではないかと。私たちは、幾度かの国際会議で Cinderella 開発者の主要なメンバーである Ulrich Kortenkamp 氏 (現 Potsdam 大) と話し合う中で、そのことを確信するようになった。そこで、2014 年に Kortenkamp 氏を日本に招聘して、昨年の RIMS 研究集会「数学ソフトウェアとその効果的教育利用に関する研究」で講演してもらった後、東邦大学に戻って 2 日間、連携のあり方について

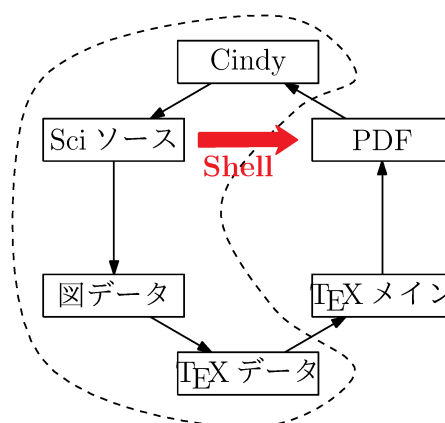
⁴cinderella.de, 入谷昭氏が日本語のサポートページ ketcindyjapan を立ち上げている。

ての詳細打ち合わせを行った。その結果、両者のコラボとしての K_ET Cindy が生まれたのである。すなわち、等式で表すと次のようになる。

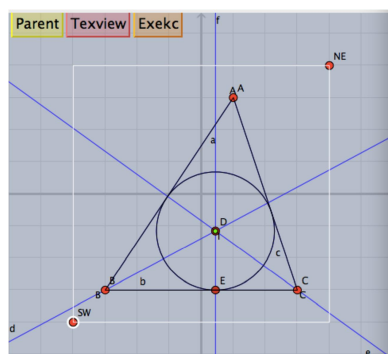
$$K_{E}TCindy = K_{E}Tpic + Cinderella$$

4 K_ET Cindy の仕組み

Cinderella (以下, Cindy) は, CindyScript というユーザーフレンドリーなプログラミング言語を持っている。K_ET Cindy はこのマクロパッケージである。ただ, 他の K_ETpic パッケージ (Scilab 版, R 版など) とは異なり, 直接 Tpic コードを生成するのではなく, Scilab 版 K_ETpic のスクリプトを作り, 後の処理を Scilab に任せる仕組みになっている。すなわち, 右図のように, (1) Cindy から Scilab ソースを作り, (2) Scilab, T_EX コンパイラ, PDF ビューアをバッチ処理する, という流れに沿って図を作成する。なお, T_EX への読み込みは `\includegraphics` ではなく `\input` である。



内接円を例にして作図の流れを説明する。Cindy (のサンプルファイル) を立ち上げると, 下左図 (Cindy 画面) が現れる。画面内の左上にあるのは, K_ET Cindy のボタンである。Cindy 画面には, 三角形や円, 角の二等分線などの幾何要素を Cinderella の作図ツールを用いて描くことができる。通常は, 点は赤色, 線は青色で表示される。次に CindyScript のメニューを選ぶと, 下中図の画面 (Script 画面) が現れる。ここに, Listplot や Circledata などの K_ET Cindy のコマンドを書いて実行すると, Script 画面の下側 (コンソール) に結果が表示され, Cindy 画面の該当する線 (三角形 ABC と内接円) が黒くなる。Cindy 画面上の白線は描画領域で, 点 NE, SW を動かして変更できる。他の点も動かして, 適当な位置が決まったら, Texview ボタンを押すと, (1) Scilab ソースファイル, (2) 表示用の T_EX メインファイル, (3) バッチファイルが更新され, Exekc ボタンを押すことで, (3) のバッチファイルが実行され, 下右図が PDF で表示される。出来上がりの図を見て, さらに修正や要素の加除を施せばよい。このように, インタラクティブに作図できる点が K_ET Cindy の最大の特長である。



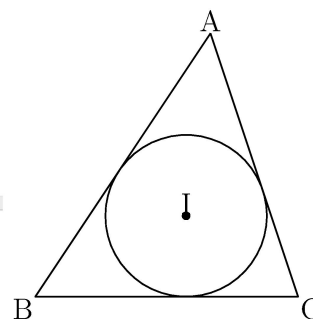
```

Fhead="inscribed";
Ketinit();
Addax(0);

Listplot([A,B,C,A]);
Circledata([D,E]);
Pointdata("I",D,["size=5"]);
Letter([A,"n","A",B,"sw","B",
C,"se","C",D,"n","I"]);

Windispq();
  
```

KETCindy V.1.6.5 (2015.09.20)
 Setwindow([-4,4],[-4,4])
 generate Listplot sgABCA
 generate Circledata crDE
 generate pointdata pt1

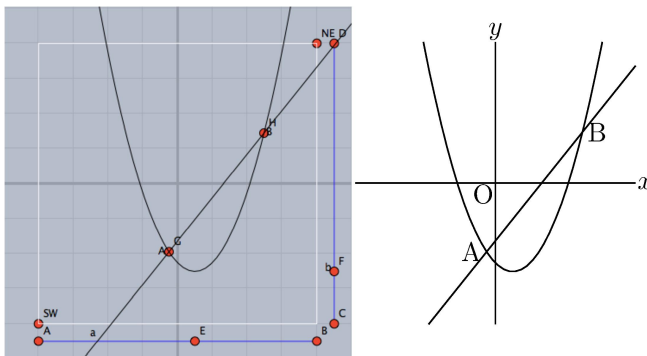


5 K_ET_Cindy の作図例

K_ET_Cindy は, Cindy と連携して, プリント作成に必要な要素を描き入れるための豊富なコマンドを持っている. ここでは, 前節で取り上げた幾何図形以外の作図例のいくつかを Script (の主要部分) とともに紹介することにする⁵.

関数のグラフ

```
Fhead="exgraph";
Ketinit();
Setunitlen("0.5cm");
Plotdata("1","(x-E.x)^2+F.y",
"x");
PutonCurve("G","gr1");
PutonCurve("H","gr1");
Lineplot([G,H]);
Letter([G,"w2","A",
,H,"e2","B"]);
Windispg();
```

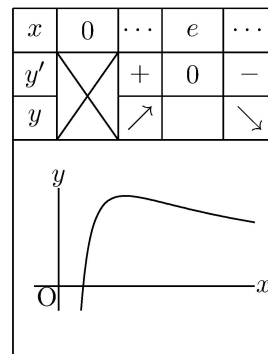
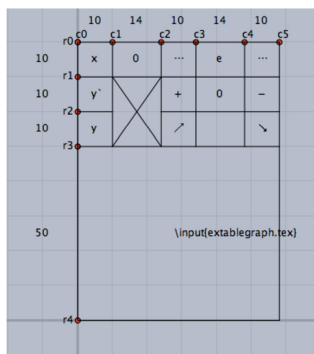


Cindy 画面の右と下にあるのはスライダーである. 例えば線分 AB にとった点 E は AB 上だけを動かすことになる. E (または F) の座標を使うことで, 曲線を自由に動かすことができる. また, G, H も曲線上を自由に動かすようにしている. すなわち, Cindy 画面でそれらの点を動かせば, それに伴って直線も変わることになる.

作表

周知のように, T_EX における作表はかなり面倒である. また, Excel などにしても, 例に挙げた表を作成することは容易ではないが, K_ET_Cindy では以下の Script で簡単に作成できる. Cindy 画面の左および上部にある点を自由に移動して, 表の高さと幅を変更することができ, セルをまとめたり, 図表をおいたりすることも簡単にできる.

```
Xl1st=[10,14,10,14,10];
Yl1st=[10,10,10,50];
RmvL=["r2c1c2","c1r3r4","c2r3r4",
"c3r3r4","c4r3r4"];
Tabledata("",Xl1st,Yl1st,RmvL);
Tlistplot("1",["c1r1","c2r3"]);
Tlistplot("2",["c2r1","c1r3"]);
Putrowexpr(1,"c",["x","0",
"\cdots","e","\cdots"]);
Putrowexpr(2,"c",["y'","", "+",
"0","-"]);
Putrowexpr(3,"c",["y","",
"\nearrow","", "\searrow"]);
Putcell("c0r3","c5r4","c",
"\input{exstablegraph.tex}")
```



⁵sites.google.com/site/ketcindy も参照

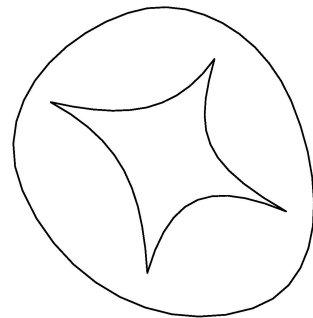
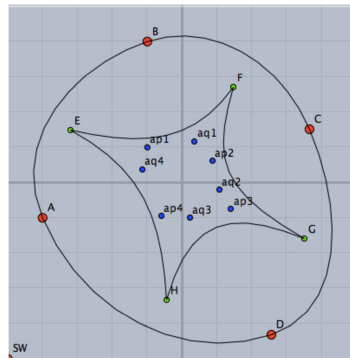
自由曲線

K_ETCindy には、2 次または 3 次のベジェ曲線を用いるコマンド群を組み込んである。主なものを列挙すると次のようになる。ただし、曲線が通る点を節点、それ以外の点を制御点と呼ぶことにする。

Bezier(番号, 節点, 制御点)	3 次 (2 次) のベジェ曲線を作る
Mkbezierptcrv(節点)	節点だけを与える
Bspline(番号, 制御点)	2 次 B-スプライン
Ospline(番号, 節点)	大島利雄氏考案の方法で制御点を自動的にとる

節点や制御点を Cindy 画面で動かして、曲線の形状を変えることができる。以下に、Ospline と Mkbezierptcrv の例を示す。

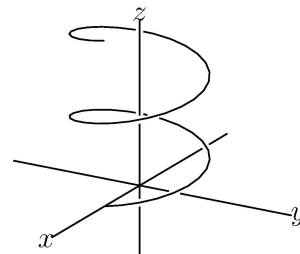
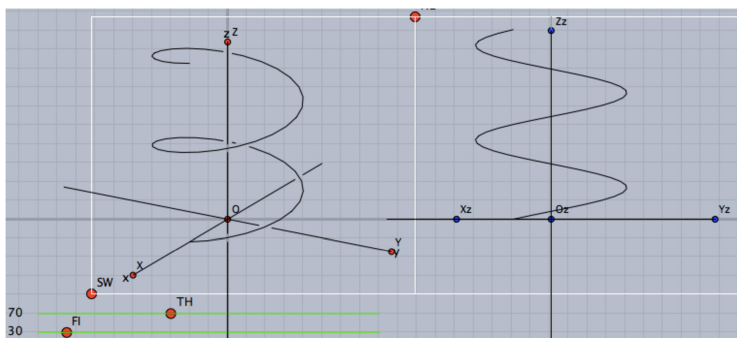
```
Fhead="exbezier";
Ketinit();
Addax(0);
Ospline("1", [A,B,C,D,A]);
Mkbezierptcrv([E,F,G,H,E]);
Windisp();
```



空間曲線と多面体

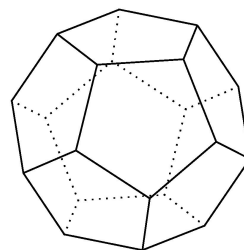
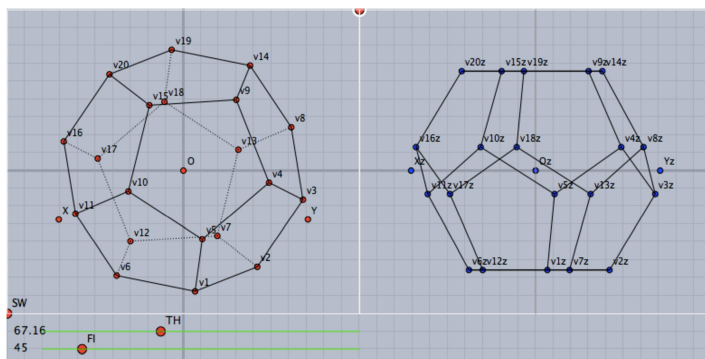
空間図形については、Cindy 画面に主画面と副画面の両方を表示するようにしてあり、両者の組み合わせで空間点の位置が決定される。主画面の下部には θ, φ のスライダーを設け、投影方向を変えられるようにしている。図の螺旋は以下のコマンド列により描かれるが、奥にある線の一部を切る立体表現（スケルトン処理）をとっている。

```
Start3d();
Ptseg3data();
Putaxes3d(5);
Xyzax3data("", "x=[-5,5]", "y=[-5,5]", "z=[-5,5]");
Spacecurve("1", "2*[cos(t),sin(t),0.2*t]", "t=[0,4*pi]");
Skeletonparadata("1", ["r", 1.5]);
Windisp();
```



また、次図の 12 面体は、小林光夫氏、鈴木卓治氏が作成した整面凸多面体の座標データを三谷純氏が OBJ 形式に変換したものを利用して作図したもので、不透明モデルとして陰線処理を施しているが、透明モデルとしてスケルトン処理することも可能である。

```
Setdirectory("/ketlib/polyhedron");
tmp=Readobj("r04.obj",["size=-3.5"]);
Setdirectory(Dirwork);
tmp1=VertexEdgeFace("1",tmp,["Pt=fix","Edg=nogeo"]);
Nohiddenbyfaces("1","phf3d1",["Hid=do"]);
Windispg();
```



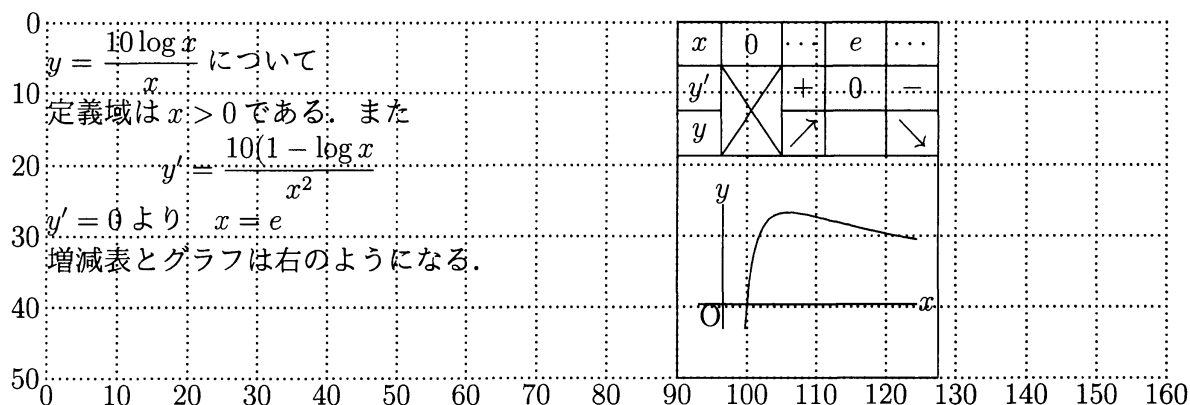
6 レイアウト

1 節で触れたように、レイアウトは教材作成に不可欠な要素であるが、通常の $\text{T}_\text{E}\text{X}$ では思った通りにレイアウトすることは難しい。そこで、 $\text{K}_\text{E}\text{T}_{\text{pic}}$ に組み込んだ $\text{T}_\text{E}\text{X}$ のコマンド作成（メタコマンド）の機能を用いて、レイアウトのためのスタイルファイル（ketlayer.sty）を作成した。このパッケージの layer 環境を用いれば、図表などを自由な位置に配置することができる。例えば

```
\begin{layer}{160}{50}
\putnotese{90}{0}{\input{fig/extable.tex}}
\end{layer}

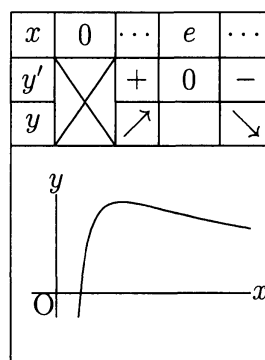
\noindent
 $y = \frac{10 \log x}{x}$  について \\\
定義域は  $x > 0$  である。 また \\\
 $\hspace*{4zw} y' = \frac{10(1 - \log x)}{x^2}$  \\\
 $y' = 0$  より  $\hspace{1zw} x = e$  \\\
増減表とグラフは右のようになる。
```

とすると、本文の位置は変えずに横 160、縦 50（単位は mm）の格子が現れ、その中の (90, 0) を基準点として右下 (se) の位置に図が配置される。方向については、東西南北など 8 方向と中心 (c) をサポートしている。



図の位置が決まったら、layer の高さを 50 から 0 に変えれば、格子が消えて図だけが残ることになる。

$y = \frac{10 \log x}{x}$ について
 定義域は $x > 0$ である。また
 $y' = \frac{10(1 - \log x)}{x^2}$
 $y' = 0$ より $x = e$
 増減表とグラフは右のようになる。



wrapfig や emath などのパッケージと併用すれば、本文を図のまわりに回り込ませることもできる。この場合は、図は layer で入れるので、図のスペースだけをとっておけばよい。以下は、この部分のソースである。

```

\begin{layer}{160}{0}
\putnotese{110}{0}{\input{fig/extable.tex}}
\end{layer}

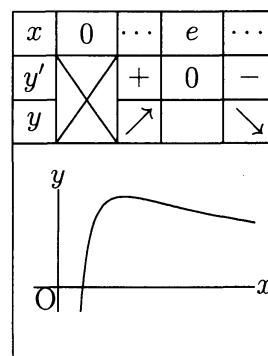
```

```

\begin{wrapfigure}[7]{r}{50mm}
\end{wrapfigure}

```

wrapfig や emath などのパッケージと併用すれば、本文を図のまわりに回り込ませることもできる。この場合は、図は layer で入れるので、図のスペースだけをとっておけばよい。



7 他ソフトウェアとの連携

Cindy には一通りの初等関数が組み込まれているが、特殊関数ではなく、統計についても、一様分布と正規分布の密度関数とそれぞれの分布からの乱数発生程度である。一方、R にはよく用いられる確率分布の関数がすべて統一的にサポートされている。例えば、正規分布は `norm` で表され、その密度関数、分布関数、逆分布関数、乱数は

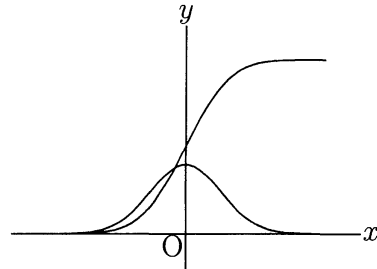
`dnorm, pnorm, qnorm, rnorm`

であり、他の分布についても同様である。そこで、`KETCindy` からバッチ処理で R の関数を呼び出して、結果をテキストファイルとして `KETCindy` に戻すコマンド群を実装した。当然ながら、R をインストールしておく必要がある。以下にいくつかの例を示す。

密度関数と分布関数のグラフ

`PlotdataR` は R の関数のグラフを描くコマンドである。このコマンドにより、(1) R のソースファイルの生成、(2) バッチファイル (`kc.sh/bat`) の更新、を行い、`kc` をターミナルで実行して結果のファイルを作ることになる。なお、既にファイルがあるときは、デフォルトでは `kc` は実行されないが、`"make"` のオプションをつけると、強制的にバッチ処理が行われて、結果ファイルが更新される。

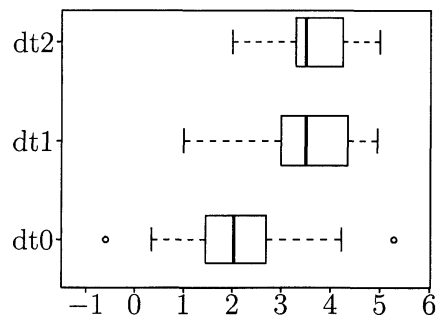
```
Ketinit();
Setscaling(5);
PlotdataR("1","dnorm(x)","x=[-4,4]","");
PlotdataR("2","pnorm(x)","x=[-4,4]","");
Windispg();
```



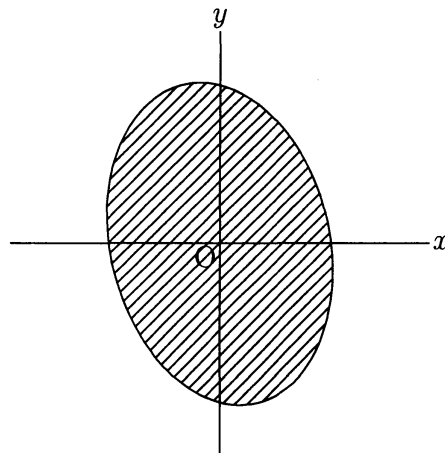
乱数の発生と箱ヒゲ図

`CalcbyR` は一定の書式で R のコマンドを記述したリストから R のソースファイルを作り、`kc` を実行する関数であり、R の呼び出しに関するすべての関数の内部で用いられている。以下の例では、 $N(2, 1^2)$ から 100 個の乱数を発生して `dt0` に代入している。また、`dt1`, `dt2` は、`Readcsv` により `csv` からデータを読み込んで割り当てられたものである。

```
cmdL=[
    "=rnorm",[100,2,1]
];
CalcbyR("dt0",cmdL,["r"]);
Readcsv("1","t4sample2.csv","");
dt1=apply(rc1,#_1);
dt2=apply(rc1,#_2);
Boxplot("1",dt0,1,1/2,[""]);
Boxplot("2",dt1/20,3,1/2,["r"]);
Boxplot("3",dt2/20,5,1/2,["r"]);
Framedata2("1",[A,B]);
tmp1=["r",-1,6,1];
tmp2=["f",1,"\mbox{dt0}",3,"\mbox{dt1}",5,"\mbox{dt2}"];
Rulerscale(A,tmp1,tmp2);
```



Scilab 版 K_ETpic に組み込まれている Hatchdata (閉曲線の内部に斜線を引く) や Implicitplot (陰関数プロット) など一部の関数については、アルゴリズムが複雑で CindyScript には移植していない。そのような関数を使うときには、R の呼び出しと同様に、いったん Scilab を呼んで、テキストとして戻されたデータによって描画する。次の例は、この方法を用いて描いた陰関数と内部斜線のグラフである。ここで、陰関数のプロットデータ imp1 は外部データなので、次の Hatchdata で利用できるように、"Out=yes" のオプションをつけている。

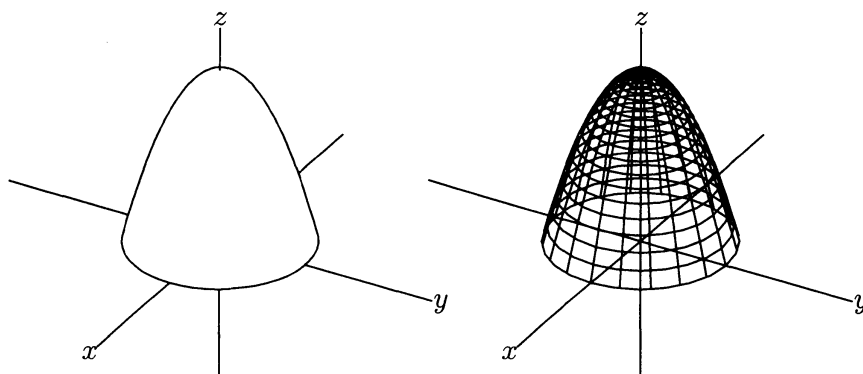


```
Implicitplot("1", "4*x^2+x*y+2*y^2=10",
  "x=[-3,3]", "y=[-3,3]", ["Out=yes"]);
Hatchdata("1", ["i"], ["imp1"]);
```

空間曲面も同様な方法によって描画される。

```
Xyzax3data("", "x=[-5,5]", "y=[-5,5]", "z=[-5,5]");
fd=["z=4-(x^2+y^2)", "x=R*cos(T)", "y=R*sin(T)", "R=[0,2]", "T=[0,2*pi]", "e"];
//Sf3data("1", fd);
Sfbdparadata("1", fd, ["dr,1.5"], []);
Crvsfparadata("1", "ax3d", "sfbd3d1", fd, [], []);
```

Sfbdparadata, Crvsfparadata などのオプションの 2 番目 (最後の引数) で陰線の描き方を設定できる。例えば、["do"] とすると点線で描かれる。また、3 行目を生かすと、陰線処理はしないが、スライダでリアルタイムに動くワイアフレームが表示される。



8 まとめ

Cindy を GUI として利用することで、インタラクティブな T_EX 作図を可能にした。さらに、本稿ではふれないが、K_ETCindy によって PDF のアニメーションやプレゼンスライド作成も簡単にできる。これらのことは、Cindy の利用可能性を大きく広げることにもなった。今後は、Maxima, Risa/Asir など数式処理との連携を図っていきたい。

謝辞 本研究は JSPS 科研費 25350370, 15K01037, 15K00944 の助成を受けている。