

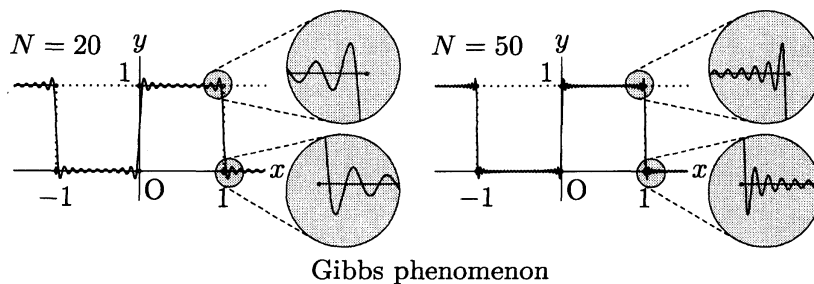
K_ETpic を用いた教材作成

長野工業高等専門学校・一般科 前田 善文 (Yoshifumi Maeda)
Faculty of General Education,
Nagano National College of Technology
長野工業高等専門学校・一般科 濱口 直樹 (Naoki Hamaguchi)
Faculty of General Education,
Nagano National College of Technology
東邦大学・薬学部 高遠 節夫 (Setsuo Takato)
Faculty of Pharmaceutical Science,
Toho University

1 はじめに

学生の理解向上を図るための効果的な方法の一つとして、内容に即した分かり易い図入り教材を配付することである。図やグラフ、表を挿入するために K_ETpic は大変便利なツールである。これまで、K_ETpic を用いて様々な教材を作成してきた。

一例として、下図はフーリエ級数の収束に関するギブス現象を説明するためのものである。



2 研究の内容

Scilab 版 K_ETpic を用いて、より簡単に教材作成ができるように K_ETpic のライブラリとして Pdfdisp, Pagedisp, Gscreen を開発した。

2.1 Pdftdispについて

2.1.1 Pdftdispの概要

これまで Scilab 版 $\text{K}_{\text{E}}\text{T}_{\text{P}}\text{I}_{\text{C}}$ では、作成された図の詳細をみるとき、Scilab のプログラムを実行した後に、PDF ファイル作成用の $\text{T}_{\text{E}}\text{X}$ ファイルをコンパイルする必要があった。それほど時間がかかる訳ではないが、多少煩雑であった。

開発した Pdftdisp は、その煩わしさを解消するための Scilab のライブラリである。基本的な Pdftdisp の使い方を紹介しよう。

※プログラムの初期化

```
Ketlib=lib('c:/work/ketpicsciL5/'); Ketinit();
// $\text{K}_{\text{E}}\text{T}_{\text{P}}\text{I}_{\text{C}}$  の初期化 必ず実行する。
Pdftdisp=lib("c:/work/pdftdisp/lib"); Pdftdispinit();
//Pdftdisp の初期化 必ず実行する必要がある。
//Scilab プログラムが存在しているフォルダ名とプログラムのファイル名を取得
Fname=Getfilename();
//ファイル名をプログラムで使用可能にする (拡張子は ".tex" に変更)
```

※図データの作成プログラム

```
... (プログラムを記述)
Openfile(Fname,"lcm"); //取得したファイル名でファイルを開く
```

※図データの描画プログラム

```
... (プログラムを記述)
Closefile("1"); //ファイルを閉じる
Pdftdisp(); // $\text{T}_{\text{E}}\text{X}$  ファイルをコンパイルして、PDF ファイルを作成する
```

このプログラムによって、Scilab のプログラムを実行すると、図の PDF ファイルを自動的に作成して表示するため、文字位置等の細部を確認しながら、プログラムを修正して再度実行することができるようになった。

Pdftdisp を使用するためには、フォルダ `c:/work/pdftdisp` にライブラリなどを置き、既存の Scilab プログラムの先頭に

```
Pdftdisp=lib("c:/work/pdftdisp/lib"); Pdftdispinit();
Fname=Getfilename();
```

ファイルを開くとき

```
Openfile(Fname,"lcm");
```

ファイルを閉じた後

```
Pdftdisp();
```

と Scilab プログラムに追加すればよい。

既に作られたプログラムが必要となり、これを探するとき、Scilab プログラムと出力された $\text{T}_{\text{E}}\text{X}$ ファイルだけでは、どのような図を作成したものか分かり難く、プログラムを見て判断するしかなかった。Pdftdisp ではプログラムの中でフォルダ名を指定しなくても自動的に Scilab プログラムのある場所にプログラムと同じ名前でも拡張子だけを変えた $\text{T}_{\text{E}}\text{X}$ ファイルと作成された図の PDF ファイルを保存することができる。Explorer で

プログラムを探すとき、PDF ファイルを見ることによってプログラムがどのようなものであるか判断することができる。特に、Explorer のプレビュー表示を利用すると簡単に図を探し出すことができるようになる。

自動的に Scilab プログラムのある場所をプログラムの中で取得できるため、プログラムをどのフォルダにコピーしてもプログラムを変更することなく Scilab を実行することができる。また、Scilab プログラムの名前を変えると作成される TeX, PDF のファイル名も自動的に変更されるため、前に作成されたファイルを上書きして消去してしまうこともなくなった。

2.1.2 Pdfdisp のプログラム

(1) の pdfdisp.tex は c:/work/pdfdisp/files の中に置かれている TeX ファイルである。

最初に Pdfdisp は (2) の TeX ファイル page.tex をサブフォルダ/pages の中に作成する。フォルダ名、ファイル名は Scilab プログラムで取得したものである。

次に (3) のプログラムによって、pdfdisp.tex をコンパイルし、DVI から PDF を作成する。作成された PDF の名前を指定 (プログラムから取得) した名前に変更して、Scilab プログラムと同じフォルダにコピーする。

(1) pdfdisp.tex

```

\begin{document}
  \input{pages/page.tex}
\end{document}

```

(2) page.tex の主要部

```

\newslide[0]{\bf\color{NavyBlue} ファイル名.tex} //スライドのタイトル
\begin{layer}{130}{0}
  \putnotese{5}{5}{\input{フォルダ名/ファイル名.tex}}
\end{layer}

```

(3) pdfdisp のプログラム主要部

page.tex の作成部分に続いて

```

[_Tmp, bOK]=powershell("platex pdfdisp.tex"); //pdfdisp.tex をコンパイル
if bOK==%f then //コンパイルに失敗したとき
  disp("TeX Compile Error");
else //コンパイルに成功したとき
  powershell("taskkill /IM "+Pdf_viewer);
  //表示されている PDF ファイルを画面から消去
  powershell("dvi2pdf pdfdisp.dvi"); //DVI を PDF に変換
  powershell("start pdfdisp.pdf"); //PDF の表示
  powershell("copy pdfdisp.pdf "+Pdf_Dirname1+Pdf_Name);
  //pdfdisp.pdf を指定のフォルダに名前を変更してコピーする
end

```

《注》 powershell : windows のコマンドラインの命令を実行する。

2.2 Pagedisp のにつて

2.2.1 Pagedisp の概要

K_ETpic は、図入り配付教材作成を主目的に考えているが、図入り配付教材の理解をさらに深めるために、プロジェクタによる提示教材や導入教材も必要になる。ここで、大切なことは配付教材の図と提示教材の図が同じであることである。配付教材の図から動画作成等を行うための作成ツールとして、Pagedisp を開発した。

Pagedisp は基本的には [4] で紹介した動画作成の方法を自動的に作成できるようにプログラム化したものである。当初は、動画作成のためのプログラムとして開発したが、プレゼンテーション用のスライド作成にも利用することができる。

次に Pagedisp のコマンドを紹介する。

Pdfdispinit : 数値等の初期化
 Setpdfviewer : 通常使う PDFviewer を指定する (デフォルトは acrobat)
 //Setpdfviewer("acrobat"); or "AcroRd32" or "PDFXCview" など
 Getdirname : 実行している Scilab プログラムの存在するフォルダ名を取得する
 Setdirname : 図ファイルを書き込む、または、図ファイルが保存されているフォルダ名を指定する
 Getfilename : 実行している Scilab プログラムのファイル名を取得する
 Setfilename : 図ファイルの名前を指定する
 SetTitle : スライド画面のタイトルの名前を指定する
 SetZuichi : スライドに書き込む図の位置を指定する
 SetBun : スライドに書き込む文章を指定する

2.2.2 Pagedisp プログラムの例

※プログラムの初期化

```
Ketlib=lib("/Work/ketpicsciL5/"); Ketinit();
Pdfdisplib=lib("c:/work/pdfdisp/lib"); Pdfdispinit();
Dirname0=Getdirname(); //このプログラムの存在するフォルダ名を取得
Fname0=Getfilename(0); //このプログラムの名前を取得 (拡張子なし)
Setdirname(Dirname0+"zu"); //どこのフォルダを指定してもよい
//Dirname0+"zu" がなければフォルダを自動作成する
```

※1 ページ目のスライド作成

```
SetBun(0,30,... //この後に"TeX の文" を書く
" ¥hspace{18mm} ¥scalebox{2}{ ¥bf¥color{NavyBlue}Pagedisp のサンプル }",...
" ¥vspace{5mm} ¥¥",...
: //TeX の文を 1 行ごとに" ",... の中に書く
" ..... "); //文書の終わり
Setfilename(); //図のないページを作成する
```

※ 2 ページ目以降のスライド作成

```

SetTitle("平行移動"); //スライドのタイトル名をセット
SetBun(); //文章のないスライドの作成
SetZuichi(50,20); //表示する図の位置
Setwindow([-3,5],[-1,8]); G0=Plotdata("x^2","x"); //基本データの作成
for k=0:10, //スライドを 11 枚作成する
    Fname=Setfilename(Fname0+string(k)); //作成する図のファイル名
    G=Translatedata(G0,0.2*k,0.1*k); //G0 を平行移動
    Openfile(Fname,"0.5cm");
        Texcom("¥color[cmymk]{1,0,0,0}");
        if k =0 then Arrowline([0,0],[0.2*k,0.1*k]); end
        Texcom("¥color[cmymk]{0,0,0,1}");
        Drwline(G);
    Closefile("1");
end
SetTitle("平行移動 (頂点の座標表示) ");
for k=10:-1:0,
    SetBun(10,20, sprintf("頂点の座標 $(%1.1f, %1.1f)$", 0.2*k, 0.1*k);
        //[10, 20] の位置に頂点の座標を表示
    Setfilename(Fname0+string(k)); //すでに作成されている図も使用できる
        //Setfilename より前に Setdirname でフォルダの場所を指定しておけば,
        //他のプログラムで作成された図がどこにあっても使用できる
end
Pagedisp();
//TeX ファイルを作成しコンパイルして, Ketslide の PDF ファイルを作成する

```

《注》 基本的なコマンドの順番

- (1) SetTitle("スライドのタイトル");
1 ページ目で省略すると, 1 ページ目はタイトルのないスライドページとなる.
一度指定した後, 省略すると同じタイトルのスライドページとなる.
 - (2) SetBun(10,20,"TeX 文","TeX 文",...);
一度指定した後, 省略すると同じ文章のスライドページとなる.
SetBun(); とすると文章のないスライドページとなる.
 - (3) Setfilename("ファイル名");
省略できない. スライドページの区切りとしても用いられている.
Setfilename(); とすると, 図のないスライドページとなる.
- (1)~(3) を繰り返して Ketslide のページを作成する.

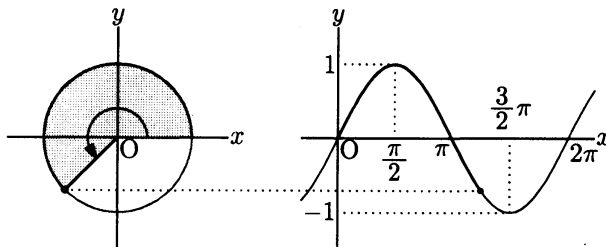
《注》 他の Scilab プログラムで図を作成して利用してもよい.

Scilab プログラムのあるフォルダの中に, サブフォルダとして pages が作成されて, その中に Ketslide のページデータ page.tex が作成される.

2.3 Gscreen について

2.3.1 Gscreen の概要

右図のように、一つの図の中に複数の図やグラフを作成するとき、別々に図を作成して並べて配置することもできるが、図のように相互の関係をもって2つの図を作成する場合は不可能である。これを1つの図として作成するときの問題点としては、一方の図を基本とすれば他方は平行移動等の座標変換をしなければならない。また、すでに作成されている図を1つの図としてまとめ、関係を示す場合は大幅なプログラムの修正をしなければならない。この煩わしさを解消するために Gscreen を開発した。



Gscreen のプログラム例

- (1) `Setwindow([-9,7],[-3,3]);`
//全体の画面設定, OpenGscreen の前に設定する. これを絶対座標とする.
`OpenGscreen(Fname,"5mm");`
//Fname は TeX ファイルの名前, unitlength は cm と mm の単位で設定
- (2) `Setwindow([-1,7],[-1.5,1.5]);` //この Setwindow は局所的な画面の設定
`Setscaling(2);` //y 座標を x 座標の 2 倍の length とする
`Lsnum1=BeginLscreen([0,0],1);` //Lsnum1 はこの画面に割り振られた番号
//局所的な画面の原点を絶対座標 [0,0] に設定
//引数の 2 番目の数値は, この画面の unitlength (全体座標に対して)
..... //図データの作成プログラム部分
..... //図データの描画プログラム部分
`EndLscreen("1");` //座標軸を描いて, この画面を終了する.
- (3) `Setwindow([-1.5,1.5],[-1.5,1.5]);`
`Lsnum2=BeginLscreen([-6,0],2);`
//局所的な画面の原点を絶対座標 [-6,0] に設定
//2 番目の数値が 2 の場合は, この画面の 1 を絶対座標の 2 とする.
..... //図データの作成, 描画プログラム部分
`EndLscreen("1");` //これを繰り返して複数画面を作成
- (4) `BeginLscreen();` //何も指定しないと絶対座標となる.
`P=TrnsPt(P1,Lsnum1); Q=TrnsPt(Q1,Lsnum2);`
//TrnsPt はそれぞれの画面の点をこの画面の座標に変換
//点だけではなく, 図形も変換できる.
..... //図データの作成, 描画プログラム部分
`EndLscreen("0");` //座標軸を描かず, この画面を終了する.
- (5) `CloseGscreen();` //プログラムの終了

2.3.2 Gscreen のプログラム

Scilab 版 $\text{K}\epsilon\text{T}\mu\text{c}$ で、`Setwindow([-9,7],[-5,4]); Openfile(Fname,"5mm");` としてプログラムを実行すると、作られる図の $\text{T}\epsilon\text{X}$ ファイルは次の通りである。

```

\unitlength=5mm%
\beginpicture%
(16.00000,9.00000)(-9.00000,-5.00000)%
..... %描画コマンド
\endpicture%

```

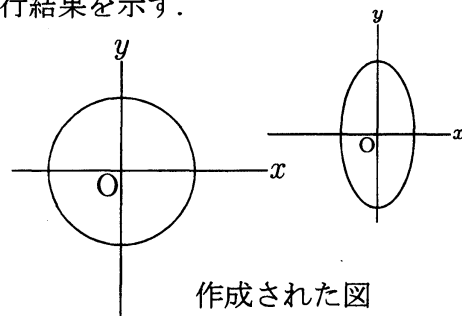
`\beginpicture%` の後の数値の意味は、横 (x 軸) 方向の長さが 16、縦 (y 軸) 方向の長さが 9 の長方形領域が描画範囲であり、左下隅の座標が $(-9, -5)$ と考えればよい。したがって、 $(-9, -5)$ を変えれば原点の位置を変更できる。

次に、例として Gscreen を用いたプログラムの実行結果を示す。

```

C=Circledata([0,0],1);
Setwindow([-9,7],[-5,4]);
OpenGscreen(Fname,"5mm");
Setwindow([-3,2],[-1.2,1.5]);Setscaling(2);
BeginLscreen([1,0],1);
  Fontsize("ss");
  Drwline(C);
EndLscreen("1");
Setwindow([-1.5,2],[-2,1.5]);
BeginLscreen([-6,-1],2);
  Fontsize("n");
  Drwline(C);
EndLscreen("1");
CloseGscreen();

```



作成された図

出力された図の $\text{T}\epsilon\text{X}$ ファイル

```

\unitlength=0.5cm%
\beginpicture%
(16.00000,9.00000)(-10.00000,-5.00000)%
..... %描画コマンド
\endpicture%
\hspace{-8cm}%
\unitlength=0.5cm%
\beginpicture%
(16.00000,9.00000)(-3.00000,-4.00000)%
..... %描画コマンド
\endpicture%

```

Gscreen のプログラムの概要

はじめの `Setwindow([-9,7],[-5,4]);` と `OpenGscreen(Fname,"5mm");` によって、描画領域と `unitlength` を決め、次の `BeginLscreen` によって、原点の位置を決めている。 `BeginLscreen` の

前の `Setwindow` は無視して描画領域の大きさは変えていない。 `EndLscreen` は基本的には $\text{K}\epsilon\text{T}\mu\text{c}$ コマンドの `Endpicture(数値)` を実行しているだけである。さらに、 `BeginLscreen` があつた場合は、図の位置を初期状態にもどすため `\hspace{-unitlength\times 横幅 cm}%` を出力して繰り返すという単純なプログラムである。 `BeginLscreen` の前の `Setwindow` は出力された $\text{T}\epsilon\text{X}$ ファイルでは無視されているが、 $\text{K}\epsilon\text{T}\mu\text{c}$ の Scilab プログラムの中では変更されていないため、仕様通りに正常に動作する。

`TrnsPt` コマンドは、画面ごとに保存された原点の位置やスケール `SCALEX`, `SCALEY` のデータを用いて、座標変換を行っている。

3 まとめ

Pdfdisp, Pagedisp は共通する部分が多いため、1つのライブラリにまとめられている。現在のところ Pdfdisp, Pagedisp が動作可能なのは Windows だけである。これは Windows のコマンドラインに関する Scilab のコマンド powershell を利用しているからである。Mac においても、sh シェルで実行するコマンドを `unix.s("コマンドの文字列")`; として使用可能である。プログラムでは Windows か Mac かを判定して実行経路を分岐させ、Mac の場合は次の命令を実行するようにすればよい。

```
powershell("platex pdfdisp.tex");    →  unix.s("platex pdfdisp.tex");
powershell("dvipdfmx pdfdisp.dvi");  →  unix.s("dvipdfmx pdfdisp.dvi");
powershell("start pdfdisp.pdf");     →  unix.s("open -a preview pdfdisp.pdf");
```

他の命令でも Windows のコマンドラインを使用している箇所があるから、今後 Mac でも使用可能となるようにプログラムを修正していきたい。

Gscreen は単純なプログラムではあるが、便利なツールである。先に示したギブス現象の説明図では多数の画面を1つの画面にまとめている。ギブス現象の図は左右に N の値だけ違う同じ図が配置されている。これは for で2回繰り返し、 N の値と原点の位置を変更した図を描いただけである。同じような図が多数ある場合、平行移動等をすることなく、基本的な図データを作成しておけば、原点の位置を変更することによって簡単に実現できる。また、画面ごとに原点の位置とスケールを変更できるため、拡大と縮小に関する図を作成するときにも非常に便利である。

図入り教材の作成を支援するソフトとして、Pdfdisp, Pagedisp, Gscreen を開発してきた。これからも、図入り教材の作成をする中で、Scilab 版 K_εTpic を用いた便利な教材作成支援ソフトを開発していきたい。

参考文献

- [1] CAST_εX 応用研究会：『K_εTpic で楽々 T_εX グラフ』，イーテキスト研究所，2011.
- [2] 山下哲，高遠節夫，「K_εTpic による教材作成と Symbolic Thinking」，数理解析研究所考究録，Vol.1780，pp.72–82，2012 年.
- [3] 前田善文，高遠節夫，「陰影を付けた立体図の K_εTpic による描画」，数理解析研究所考究録，Vol.1780，pp.154–159，2012 年.
- [4] 前田善文，高遠節夫，「K_εTpic の有用性と可能性について：授業における教材提示と増減表の自動作成」，数理解析研究所考究録，Vol.1865，pp.72–78，2013 年.
- [5] 前田善文，高遠節夫，「ワードによる教材作成と T_εX+ K_εTpic による教材作成」，数理解析研究所考究録，Vol.1909，pp.8–16，2014 年.