The Asian Journal of TEX, Volume 4, No. 2, December 2010
Article revision 2010/12/31

The Extension of KETpic Functions

Meta commands and their applications

Masataka Kaneko
Kisarazu National College of Technology,
292-0041, Japan
nkaneko@inc.kisarazu.ac.jp

Setsuo Takato
Faculty of Pharmaceutical Sciences,
Toho University, 274-8510, Japan
takato@phar.toho-u.ac.jp

KEYWORDS Graphics in LATEX documents, KETpic, Computer algebra system

ABSTRACT Though LATEX has become the standard tool for editing high-quality mathematical documents, the use of graphics in LATEX tends to be unsatisfactory. Also it is desirable that capability of generating tables and page layout in the preferred style be added to LATEX. The authors have developed KETpic, a computer algebra system (CAS)-based plug-in for high-quality graphics in LATEX documents. In this paper, we will show how newly developed functions of KETpic can easily generate new environments or graphical commands of LATEX, so that LATEX can be endowed with the above mentioned capabilities.

1 Introduction

Since LATEX offers remarkable publishing features and extensive facilities for automating most aspects of typesetting, quite a number of mathematicians use LATEX to edit their research papers. Moreover LATEX allows such quick and easy editing of mathematical documents that many university teachers also use LATEX as a tool to edit their teaching materials. In fact, of the 378 mathematics teachers at universities and colleges of technology in Japan who responded to our questionnaire survey (executed in 2008) [5], 281 teachers use LATEX to edit their teaching materials. However, only 87 teachers frequently use graphics in their printed class materials edited with LATEX. Though various graphics packages such as PSTricks and Ti*k*Z (applicable to LATEX) have been developed, only a few of the respondents use them. The reason of this seems to be twofold. One is that computable functions or applicable programmings are limited in case of these packages compared to CAS. The other is that the use of these packages requires extra training for ordinary mathematics teachers.

The functions needed for editing printed class materials could be summarized in the following five:

1. high-quality typesetting of mathematical expressions
2. capability to generate various mathematical symbols
3. capability to insert high-quality graphics

4. flexibility in creating tables which are appropriate to the situation
5. capability for fine tuning of page layout

Though editing by hand could provide all these functions to a certain extent, the output tends to lack precision, beauty and reproductivity. Using word processors makes up for these deficits fairly well. They especially enable us to easily insert various graphical images into documents. Moreover page layouts can be modified on demand. The weak point of word processors is that their ability to provide high-quality mathematical expressions and symbols is insufficient. On the other hand, LATEX is endowed with a limited set of capabilities for graphics and page layouts. Therefore, the improvement of such capabilities should make LATEX a more powerful tool for teachers editing their class materials.

Originally, we developed KETpic as a tool to insert fine (precise and expressive) graphics into LATEX documents. KETpic is a plug-in based on computer algebra system (CAS) such as Maple [13, 12], Mathematica [4], R [10], Scilab [7], Maxima, and Matlab [2, 1]. Simultaneous use of CAS and the graphics capability with which LATEX is originally endowed enables us to generate high-quality graphical images in LATEX documents. The authors have actualy been using KETpic as a daily tool. Not only planar graphics but also 3D graphics can be inserted as in Figure 1 [14, 3, 11].

Notice that the elimination of hidden lines endows the figure rich perspective. Moreover, mathematical expressions in the figure have the same quality as in the LATEX document. Since the graphics of KETpic use only monochrome lines (i.e. no colours or shadings), the quality of KETpic graphics is maintained when they are copied. Thus KETpic can create high-quality graphics that can be used for printed materials. It can even be used with projectors. This feature of KETpic could have some educational value [6].

Recently, the ability to produce tables and page layouts in the preferred style has been implemented to KETpic. The aim of this paper is to display these new functions of KETpic. Adding these new functions should enhance the use of LATEX itself.

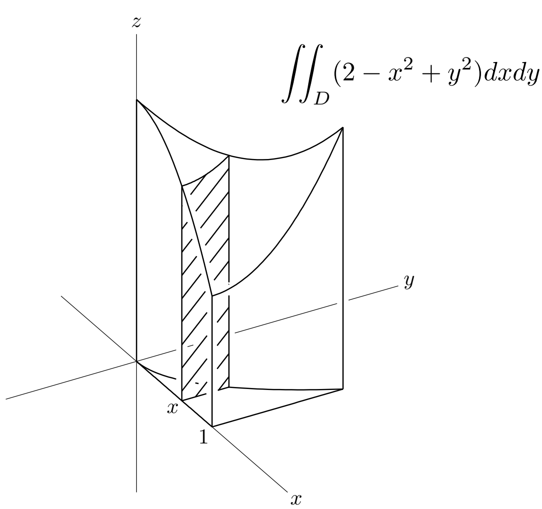


FIGURE 1. 3D graphics of KETpic

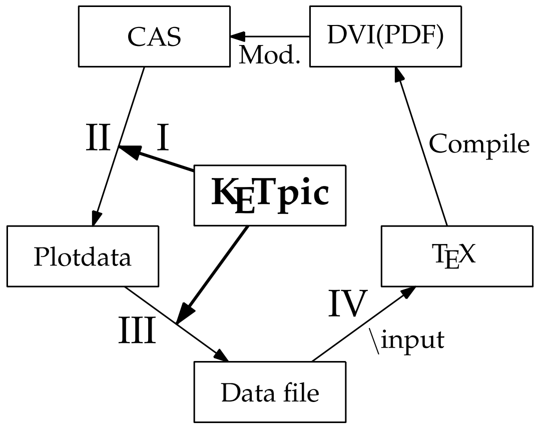


FIGURE 2. KETpic cycle

2 Brief introduction to KETpic

To draw precise figures, CAS is the most popular tool. However, it is not so convenient to use their graphical images in LaTeX documents. Usually we convert the graphical output of CAS into EPS or PDF file format and load the resulting files into LaTeX. In such cases, it is not so easy to add illustrations or to manipulate them. Moreover the quality of the graphics tend to become lost when they are copied. Therefore, we took a different strategy. In fact, KETpic generates LaTeX-readable code with the aid of CAS, so that image files are no longer either loaded or required. The procedure of KETpic drawing is summarized in Figure 2.

To begin, we will show how to make a document containing illustrations with KETpic. We will use as an example the procedure for plotting the Maclaurin expansion of $y = \sin x$ with KETpic for Scilab, and inserting the plot into a LaTeX document as shown below.

In step I, we load KETpic with the following commands:

```
Ketlib=lib('folder:/ketpicsciL5/');
Ketinit();
```

In step II, we generate the plot data of $y = \sin x$ and its Maclaurin expansion as follows:

```
P1=Plotdata('sin(x)','x=[-4.5,4.5]');  //range of x is specified
P2=Plotdata('x-x^3/factorial(3)+x^5/factorial(5)','x=[-4.5,4.5]');
```

In step III, we convert the plot data into Tpic specials code and output to a text file (named "`fig.tex`") with the following commands:

```
Openfile('folder/fig.tex');
Beginpicture('0.5cm');  //defining unit length
Drwline(P1,2);   //2 means the width of the curve P1
Drwline(P2);
Endpicture(1);
Closefile(0);
```

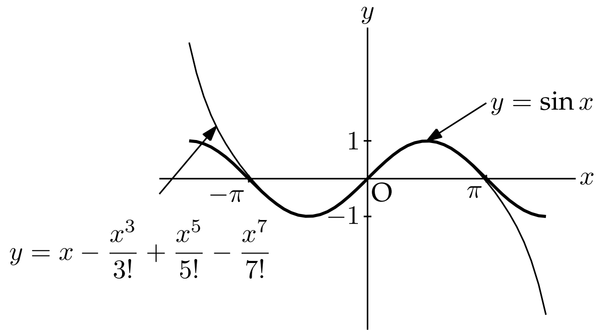


FIGURE 3. Resulting figure

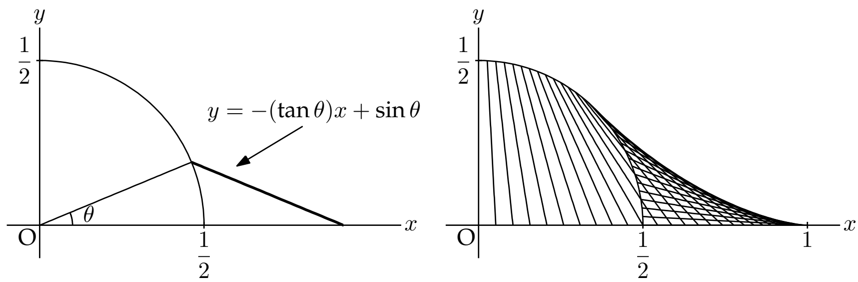


FIGURE 4. Programmability of KETpic drawing

In step IV, we insert the above fig.tex file into a LaTeX document using the following commands:

```
\usepackage{ketpic}
\begin{document}
\input{fig}
\end{document}
```

Thus, after the compilation, we obtain Figure 3. In fact, we also use commands like `arrowline`, `vtickmark`, `htickmark`, and `expr` which are also implemented to KETpic in step II to add some accessories and mathematical expressions. In a KETpic drawing, we can also utilize the programmability of CAS to draw figures like Figure 4 very easily. The corresponding libraries and some interesting examples and documentation are freely downloadable at [15].

As stated in Section 1, Ti*k*Z has a similar feature as KETpic. It is endowed with remarkable capabilities such as colors, animations and various libraries which make its drawings illuminative, as shown in [16]. Since KETpic drawings are monochrome and static, they may be inferior to Ti*k*Z ones in case when they are used with projectors. However, KETpic drawings seem to be superior in case when they are used in mass printed materials, since the quality of them is maintained when they are copied. Moreover, owing to the use of CAS, KETpic can utilize such wide range of computabilities and programmabilities that can not be expected of Ti*k*Z. Though KETpic is not equipped with rich libraries as Ti*k*Z, the flow of KETpic programming seems to be easier to see for ordinary LaTeX users compared to Ti*k*Z case.

TABLE 1. Example of a table

$W^{(\nu-1)} - basis$	$\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{r_\nu}$			
	$\cdots$	$\ddots$		
$W^{(2)} - basis$	$T^{\nu-2}\boldsymbol{v}_1, \cdots, T^{\nu-2}\boldsymbol{v}_{r_\nu}$	$\cdots$	$\boldsymbol{v}_{r_3+1}, \cdots, \boldsymbol{v}_{r_2}$	
$W^{(1)} - basis$	$T^{\nu-1}\boldsymbol{v}_1, \cdots, T^{\nu-1}\boldsymbol{v}_{r_\nu}$	$\cdots$	$T\boldsymbol{v}_{r_3+1}, \cdots, T\boldsymbol{v}_{r_2}$	$\boldsymbol{v}_{r_2+1}, \cdots, \boldsymbol{v}_{r_1}$

3 Flexible generation of tables

Though LaTeX is endowed with the environments for generating tables such as "tabular" or "tabularx", their outputs tend to be unsatisfactory when used in teaching materials. In fact, the height of cells in tables are automatically determined in these environments, so that modification of the table needs much effort. Moreover, using graphical objects in a table is desirable in some cases such as tables of increase and decrease of functions. Recently, the functions for flexible generation of tables have been implemented to KETpic. This development is based on the idea that table is a kind of graphic. For example, the following command lines of the KETpic version for Scilab provides Table 1:

```
Tmp1=list([6,3,5],[10,4,5],16,25,[10,2,5],[20,3,5],[17,4,5]);
Tmp2=list([7.5,4,6],[22.5,2,7],[7.5,3,8],7.5);
Out=Tabledata([-1,-1],Tmp1,Tmp2);
Openfile('e:/latable.tex');
  Beginpicture('1.5mm');
    Drwline(Out(1));
    Putcell(Out,1,1,'l','$W^{(\nu -1)}$-basis');
    Putcell(Out,2,3,'l','$W^{(2)}$-basis');
    Putcell(Out,4,1,'c','$\bm{v}_1, \cdots, \bm{v}_{r_{\nu}}$');
    Putcell(Out,4,2,'c','$\cdots$');
      (The rest is omitted)
  Endpicture(0);
Closefile(0);
```

Here, by using the command `Tabledata`, we can easily specify the width and height of each cell and the range where lines are drawn as we like. Notice that the content of a cell can be located at a preferred position by using the command `Putcell`.

As an application, KETpic enables us to easily offer the tables of increase and decrease as Table 2. Here the images of arrows are generated with the use of KETpic meta commands explained in the next section.

TABLE 2. Table of increase and decrease

x	-2	$\cdots$	-1	$\cdots$	$-\frac{\sqrt{6}}{3}$	$\cdots$	0	$\cdots$	$\frac{\sqrt{6}}{3}$	$\cdots$
y'		$-$	0	$+$		$+$	0	$-$		$-$
y''		$+$		$+$	0	$-$		$-$	0	$+$
y		↘	0	↗		↗	$\frac{5}{2}$	↘		↘
			minimal				maximal			

4 Meta commands of KETpic

Recently we have implemented the commands to generate various LATEX macros to KETpic. Using these commands with KETpic graphical commands simultaneously, we can generate graphical symbols as LATEX macro commands. We call these new KETpic commands "meta commands".

As an example, we will show the KETpic commands to generate the LATEX command named `\cirmark` which is used to offer the graphical symbol .

```
Openphr('\cirmark');
  Setwindow([-1,1],[-1,1]);
  Tmp=Circledata([0,0],1);
  F1=Scaledata(Tmp,2,2.5);
  Beginpicture('1mm');
    Shade(list(F1));
    Drwline(F1);
  Endpicture(0);
Closephr();
```

Also we can generate LATEX commands with parameters. For example, the command `\dashmark` to offer a symbol like 8 can be generated by the following KETpic commands:

```
Openphr('\dashmark#1');
  Beginpicture('1mm');
    Dashline(F1);
    Letter([0,0],'c','#1');
  Endpicture(0);
Closephr();
```

In the above examples, the commands `Openphr` and `Closephr` are sort of meta commands. They are used to generate the package of LATEX command lines corresponding to `\def`.

When generating more complicated LATEX graphical commands, using LATEX programmings to

1. treat variables
2. create conditional branching

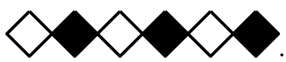
TABLE 3. Other meta commands

Treatment of variables	`Texsetctr`
Conditional branching	`Texif`, `Texelse`, `Texendif`
Loop structure	`Texfor`, `Texendfor`

3. create loop structure

are often needed. However, these programmings are complicated in LaTeX. So that, we have implemented the meta commands in Table 3 into KETpic. They automatically generate the corresponding LaTeX command lines.

As an example, we will show how to generate the LaTeX command `diachain{n}`. Here n specifies the number of daimonds so that the output of `\diachain{6}` is


.

The KETpic commands generating it are very simple as follows:

```
Texnewcmd('\bs diachain',1);
  Tmp=Framedata([0,0],L);
  G=Rotatedata(Tmp,\%pi/4);
  Texfor(1,'1','\#1');
    Beginpicture('1mm');
      Drwline(G,2);
      Texsetctr(2,'Texctr(1)/2*2-Texctr(1)');
      Texif(Texthectr(2)+'=0');
        Shade(G);
      Texendif();
    Endpicture(0);
  Texendfor(1);
Texend();
```

5 Layer environment

In the previous section, we have introduced the meta commands which prevent authors from programming TeX macros. Since the meta commands are so powerful, we have developped a LaTeX environment named "layer". It could be a convenient tool for not only editing research papers or printed materials but also proofreading, editing textbooks, preparing examinations and tutorials, and web-tech based learning.

The "layer" environment enables us to attach some graphical supplements to a document without affecting the other elements already in the document. Though "overpic.sty" has a similar feature, its use is limited to picture environment only. The "layer" environment, on the other hand, graphical supplements can be placed in any position you desire. An example is shown in Figure 5.

The "layer" environment and some typical graphical contents have been integrated into LaTeX macro package named "`ketlayer.sty`". The authors are distributing "`ketlayer.sty`" at [15]. The ketlayer package allows interactive use of graphics and

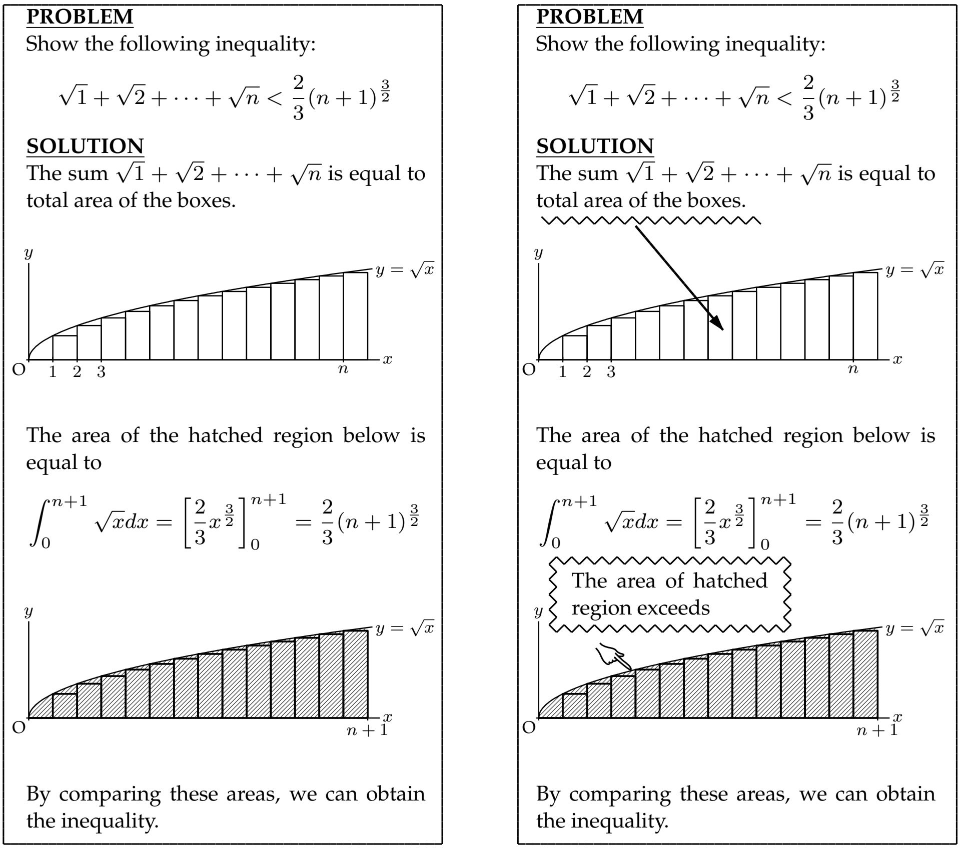


FIGURE 5. Using layer environment

sentences. In fact, it has a switch to turn on and off grid lines which specify where a graphical supplement is located. While polishing supplements, let height parameter of `\begin{layer}{`⟨*width*⟩`}{`⟨*height*⟩`}` declaration be positive. After editing them, modify the height parameter to be `0` as shown in the following code.

```
%\begin{layer}{130}{140}% draft: grid lines are shown
\begin{layer}{130}{0}% final
  % supplements begin
  \hjaggyline{72}{27}{28}
  \arrowlineseg{100}{78}{27}{-130}
  \jaggyboxframe{85}{110}{40}{10}{The area of hatched region exceeds}
  \rightdownhand{96}{126}}
  % supplements end
\end{layer}
```

Note that the layer environment should lie before the target of the supplements.

6 Future works

As explained in Section 2, the graphical output of KETpic is formatted in the form of Tpic Specials codes. Tpic Specials codes are supported by DVI drivers including dvipdfmx and dvips, but Tpic Specials codes are not supported by pdfLaTeX's direct PDF output. On some occasions, the authors were requested to use pdfLaTeX and could not use KETpic. Therefore they are planning to develop another version of KETpic which is applicable to pdfLaTeX. Some adjustments are required for KETpic to be applied to pdfLaTeX. However, KETpic utilizes a limited set of Tpic Specials codes such as `pn`, `pa`, `fp`, (and partially `sh`, `ip`) as showin in the following code. Hence the adjustments seem not to be so difficult.

```
{\unitlength=1cm%
\begin{picture}(4.40000,4.40000)(-2.20000,-2.20000)%
\special{pn 8}\special{pa -787 787}%
\special{pa 787 787}\special{pa 0 -787}%
\special{pa -787 787}\special{fp}%
\settowidth{\Width}{0}\setlength{\Width}{-0.5\Width}%
\settoheight{\Height}{0}\settodepth{\Depth}{0}%
\setlength{\Height}{-0.5\Height}\setlength{\Depth}{0.5\Depth}%
\addtolength{\Height}{\Depth}%
\put(0.0000,0.0000){\hspace*{\Width}\raisebox{\Height}{0}}%
\end{picture}}%
```

Acknowledgments

This work was supported by a Grant-in-Aid for Scientific Research (C) 20500818.

References

1. A. Galvez, A. Iglesias, S. Takato, *New Matlab-Based KETpic Plug-In for High-Quality Drawing of Curves*, Proc. ICCSA 2009, IEEE Press, 123–131. 2009.

2. A. Iglesias, A. Galvez, S. Takato, *Matlab-Based KETpic Add-On for Generating and Rendering IFS Fractals*, Communications in Computer and Information Science **56**. 334–341. 2009.

3. M. Kaneko, T. Abe, H. Izumi, K. Kitahara, M. Sekiguchi, Y. Tadokoro, S. Yamashita, K. Fukazawa, S. Takato, *A simple method of the TEX surface drawing suitable for teaching materials with the aid of CAS*, L.N.C.S., **5102**, 35–45. 2008.

4. M. Kaneko, T. Abe, M. Sekiguchi, Y. Tadokoro, K. Fukazawa, S. Yamashita, S. Takato, *CAS-aided Visualization in LaTeX documents for Mathematical Education*, Teaching Mathematics and Computer Science **8** (2010), no. 1, 1–18.

5. M. Kaneko, S. Yamashita, H. Koshikawa, K. Kitahara, S. Takato, *The Effective Use of LaTeX Drawing in Linear Algebra—Utilization of Graphics Drawn with KETpic—*, Proc. CADGME 2010. `http://home.pf.jcu.cz/~cadgme2010/`

6. K. Kitahara, T. Abe, M. Kaneko, S. Yamashita, S. Takato, *Towards a More Effective Use of 3D-Graphics in Mathematics Education*, to appear in International Journal for Technology in Mathematics Education, 2010.

7. H. Koshikawa, M. Kaneko, S. Yamashita, K. Kitahara, S. Takato, *Handier Use of Scilab to Draw Fine LaTeX Figures—Usage of KETpic Version for Scilab—*, Proc. ICCSA 2010, IEEE Press, 39–48. 2010.

8. Y. Nakamura, S. Takato, *Development of web application for LATEX plotting software KETpic*, Proc. World Conference on E-learning in Cooperate, Government, Healthcare and Higher Education 2009, 2183–2188. 2009.

9. ———, *Development of a graphical user interface for LATEX plotting software KETpic*, Proc. ICCSA 2009, IEEE Press, 109–114. 2009.

10. S. Ouchi, S. Takato, *High-quality statistical plots in LATEX for mathematics education using an R-Based KETpic plug-in*, Proc. of the 15th ATCM Conference, 266–275, 2010.

11. M. Sekiguchi, T. Abe, H. Izumi, M. Kaneko, K. Kitahara, Y. Tadokoro, S. Yamashita, K. Fukazawa, S. Takato, *Monochrome line drawings of 3D objects due to the programmability of KETpic*, Proc. ICCSA 2008, IEEE Press, 277–283. 2008.

12. M. Sekiguchi, M. Kaneko, Y. Tadokoro, S. Yamashita, S. Takato, *A New Application of CAS to LaTeX-Plottings*, L.N.C.S., **4488**, 178–185. 2007.

13. M. Sekiguchi, S. Yamashita, S. Takato, *Development of a Maple Macro Package Suitable for Drawing Fine TeX-Pictures*, L.N.C.S., **4151**, 24–34. 2006.

14. S. Takato, A. Galvez, A. Iglesias, *Use of ImplicitPlot in Drawing Surfaces Embedded into LaTeX Documents*, Proc. ICCSA 2009, IEEE Press, 115–122. 2009.

15. KETpic.com. `http://ketpic.com`

16. TEXample.net, *Examples by The TikZ and PGF manual*. `http://www.texample.net/tikz/examples/author/pgf-manual/`