

A New Application of CAS to \LaTeX Plottings

Masayoshi Sekiguchi, Masataka Kaneko, Yuuki Tadokoro, Satoshi Yamashita,
and Setsuo Takato

Kisarazu National College of Technology,
Kiyomidai-Higashi 2-11-1, Chiba 292-0041, Japan
masa@kisarazu.ac.jp

Abstract. We have found a new application of Computer Algebra System (CAS), *KETpic* which has been developed as a macro package for a CAS. One of aspects in its philosophy is *CAS-aided visualization in \LaTeX documents*. We aim to extend *KETpic* to other CASs, and derive necessary conditions from the basic idea for CASs to accept it, *i.e.*, *I/O functions with external files and manipulating numerical or string data*. Finally, we describe *KETpic for Maple* as a successful example. By using *KETpic* we can draw fine pictures in \LaTeX documents.

Keywords: CAS, \LaTeX .

1 Introduction

In many cases, mathematicians or mathematics teachers, as well as other scientists, need to prepare good illustrations or educational materials. In general, CAS gives us a set of highly accurate numerical data. Therefore, it is quite natural to utilize CAS for the purpose of creating fine pictures. CASs support beautiful and impressive graphics and some of them can output the picture in graphical formats (EPS, JPEG, GIF, BMP, etc). However, the authors have not been satisfied with printed matters obtained as a direct output from CASs, as well as from CADs (Computer Aided Design) or from data/function-plotting programs, like Gnuplot. The reason is that mathematical lettering in their pictures is not clear. We need to optimize their outputs so as to be available for mathematical textbooks or academic papers.

On the other hand, \LaTeX has a quality in lettering high enough to satisfy us sufficiently but no abilities of symbolic or numerical computation (see [5]). It has the picture environment or ability of displaying graphical data files in EPS format. By using *Tpic*, a graphical extension of \LaTeX , we can draw various pictures based on 2D numerical data (see [2,4]). However, it is cumbersome to handle numerical data directly and to generate *tpic* special commands. It is better to write a program generating *tpic* special commands from numerical plotting data. The program will be another individual software or a macro package for a CAS.

The authors have developed *KETpic for Maple*, a Maple macro package. It generates *tpic* special commands, and enables us to draw complicated but 2D

mathematical objects in \LaTeX documents at the highest accuracy. Detailed description of KETpic for Maple is given in [7]. Recently we organized a project in which we aim to extend KETpic to other CASs. We consider necessary conditions for CASs to accept KETpic. Philosophy of designing KETpic includes a basic idea, *CAS-aided visualization in \LaTeX documents* which we call *CAS-aided \LaTeX plottings*. The requirements will be naturally derived from the basic idea.

Section 2 is devoted to construction of necessary conditions for a CAS to generate graphic files which we can include in \LaTeX documents. In section 3, we show that Maple satisfies the requirements, describe how KETpic realizes the idea, and illustrate its outputs.

2 Requirements for CAS-Aided \LaTeX Plottings

In order to realize the idea “CAS-aided \LaTeX plottings”, we decided to develop a macro package for a known CAS. We did not select other ways of development: a new CAS which is designed to realize the idea or another individual software which calls a kernel of CAS as an external computing engine. We believe that it is best to develop a macro package for a known CAS.

Hereafter, we suppose that a standard CAS is equipped with abilities of symbolic or numerical computing, programming, and generating graphical images. In addition, we require the following necessary conditions for a CAS.

- R1.** Loadability of macro packages from external files,
- R2.** Writability of numerical data and strings with formats on text files,
- R3.** Accessibility to raw numerical data in 2D/3D coordinates,
- R4.** Ability of manipulating numerical values or strings to generate graphic codes, *e.g.*, tpic special commands, PostScript, or EPS.

If a CAS satisfies “writability without formats” instead of the condition R2, it can write a sequence of raw data in a text format. In this case, it is necessary to translate the unformatted data into a formatted data. The translation may be done by a post-processor. The condition R4 and the ability of programming enable us to handle a lot of data collectively or iteratively, and optimize their outputs.

For KETpic, we have chosen to generate tpic special commands. Our choice, Tpic, allows us to obtain rich graphical expressions. We believe that Tpic is best because it is wide-spread. Unfortunately a previewer, *Mxdvi* in Mac OS X, does not support Tpic. We offer a particular version of KETpic for the previewer. It generates eepic [4] commands instead of tpic special commands, and is downloadable from our web site [6]. Another way to provide rich expressions is to generate PostScript or EPS (Encapsulated PostScript) files. Many versions of \LaTeX allow EPS files to be inserted in documents. This is still realistic if we are familiar to grammar of EPS format.

A graph with many curves or items is a powerful expression in mathematical documents. Producing the graph becomes easier by collective or iterative operations, *e.g.*, list processing, DO-loop, WHILE-loop, and so on.

Optimization of graphics means fine-tuning of outputs and customizing of graph accessories, *e.g.*, tickmarks, labels of axes, and legends of curves, which make pictures more appealing. For fine-tuning we have added commands to KETpic, by which we can draw various hatchings, dashed lines, and projections of 3D objects. For customizing of graph accessories we have added other commands to KETpic. These operations are realized by programming of CASs.

3 A Successful Example: *KETpic for Maple*

3.1 Maple Satisfies the Requirements

The condition R1 is satisfied by command `read` equipped with in Maple. We can use it as follows.

```
> read 'UsersFolder/ketpicw.m';
```

The condition R2 is satisfied by Maple commands `fopen`, `fclose`, and `fprintf`. They define KETpic commands `openfile` and `closefile`. The usage is as follows.

```
> openfile('UsersFolder/figure1.tex');
...
> closefile():
```

These also satisfy the condition R4. KETpic commands `openpicture`, `closepicture`, and `setwindow` return `\begin{picture}` and `\end{picture}` with option indicating its window size and a unit length. For instance, the following set of commands,

```
> setwindow(0..5,-1.5..1.5):
> openpicture("1cm"):
...
> closepicture():
```

returns a set of commands of the picture environment as follows.

```
{\unitlength=1cm%
\begin{picture}%
(5.00,3.00)(0.00,-1.50)%
...
\end{picture}}%
```

The following Maple command gives plotting data to a variable `g1`.

```
> g1:=plot(sin(x),x=0..5):
```

If we execute a command ending with semi-column instead of column in the command line above, we can see the internal expression of `g1` which takes a list format.

```
> g1:=plot(sin(x),x=0..5);
g1:=PLOT(CURVES([[0.,0.],...
..., [5., -0.958924274663138453]], COLOUR(RGB,1.0,0.,0.)),...))
```

This operation satisfies the condition R3. This internal expression can be constructed through DO-loop operations and string manipulation. Maple command `op` returns n -th operand of its argument.

```
> op(g1);
  CURVES(...), COLOUR(...), AXESLABELS(...), VIEW(...)
> op(3,g1);
  AXESLABELS(...)
> op(1,op(1,g1))
  [[0.,0.],..., [5., -0.958924274663138453]]
> op(1,op(1,op(1,g1)))
  [0.,0.]
```

Maple commands `sscanf` and `convert` can translate characters into numerical data, and vice versa. Other commands `cat`, `substring`, and `length` can be used for string manipulations, *concatenating*, and so on. One of drawing commands of KETpic is `drwline`. The usage is as follows.

```
> drwline(g1):
```

This command returns a set of tpic special codes as follows.

```
> drwline(g1);
\special{pa 0 0}\special{pa 43 -43}...\special{pa 164 -160}%
...
\special{pa 1844 394}...\special{pa 1969 378}%
\special{fp}%
%
```

Commands for hatching area, drawing dashed curves, and customizing graph accessories are as follows.

```
> setax("", "\\xi", "", "\\eta", "", "", ""):
> g2:=plot(cos(x),x=0..5):
> g3:=hatchdata(["ii"], [3,0], [g1,"s"], [g2,"n"]):
> drwline(g3):
> dashline(g1,g2):
> expr([2,1], "ne", "\\displaystyle
\\int_{\\pi/4}^{5\\pi/4}(\\sin\\xi-\\cos\\xi)d\\xi)":
```

Double backslash “`\\`” returns single backslash because single backslash is a control code in Maple. The first command `setax` defines axes, the origin, and their names. In this case, the name of the horizontal axis is ξ , and the vertical one η . The command `hatchdata` returns a set of stripes inside a closed curve obtained

from `g1` and `g2`. Its third argument [`g1`, "`s`"] indicates a region in the south of curve `g1`. Similarly the fourth argument [`g2`, "`n`"] indicates a region. The first argument [`"ii"`] indicates inside areas of them. The second argument [`3,0`] defines a reference point. The command `dashline(g1,g2)` returns plotting data of `g1` and `g2` with dashed lines. The last command line puts a legend $\int_{\pi/4}^{5\pi/4} (\sin \xi - \cos \xi) d\xi$ at a point whose position is slightly different from (2, 1) to the northeast. The resulting figure is given in Fig. 1.

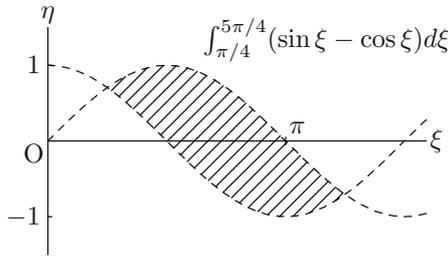


Fig. 1. Output of §3.1

3.2 Special Functions or Functions Defined by Integrals

Using Maple, we can call special functions, calculate values of them, and plot graphs. The Chi-square distribution is defined by the gamma function $\Gamma(x)$ as follows.

$$f_n(x) = \frac{x^{\frac{n}{2}-1} e^{-\frac{x}{2}}}{2^{\frac{n}{2}} \Gamma(\frac{n}{2})}. \tag{1}$$

Curves of the distributions can be obtained by Maple, and can be included in this document by KETpic (see Fig. 2 (left)). The corresponding sequence of KETpic commands are given below.

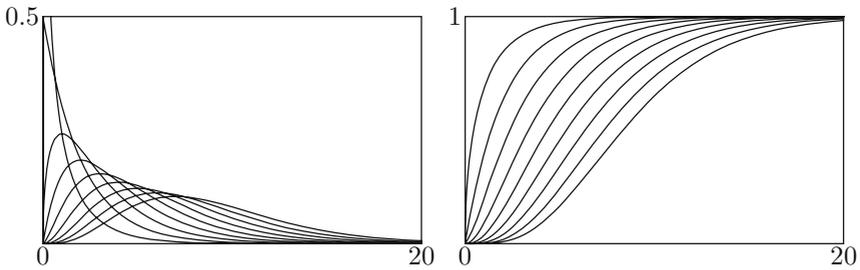


Fig. 2. Chi-square distributions for degrees of freedom $n = 1, 2, \dots, 9$ (left) and their corresponding definite integrals (right)

```
> f:=(n,x)->x^{n/2-1}*exp(-x/2)/2^{n/2}/GAMMA(n/2);
> tmp:=[]:
  for i from 1 to 9 do
    tmp:=[op(tmp),plot(f(i,x),x=0..20)]:
  od:
g4:=display(tmp):
```

The internal expression of g4 is

```
plot(CURVES(...),CURVES(...),...,CURVES(...))
```

One can find a definition of the Chi-square distribution in the first line. This is an advantage of CAS. Another advantage is an iterative operation which one can find in the last three lines. To define a function, one can use an integral form. The following function is a definite integral of the Chi-square distribution (see Fig. 2 (right)).

$$F_n(x) = \int_0^x f_n(t) dt. \tag{2}$$

The corresponding sequence of KETpic commands are given below.

```
> F:=(n,x)->int(f(n,t),t=0..x);
> tmp:=[]:
  for i from 1 to 9 do
    tmp:=[op(tmp),plot(F(i,x),x=0..20)]:
  od:
g5:=display(tmp):
```

3.3 Curves Defined by Implicit Functions or with Parameters

Using Maple, we can draw a curve defined by implicit functions. In general, contours are obtained by the same way. Fig. 3 (left) shows contours of the following Coulomb potential,

$$\phi(x, y) = \frac{1}{\sqrt{(x+1)^2 + y^2}} + \frac{1}{\sqrt{(x-1)^2 + y^2}}, \tag{3}$$

where two electric charges place on $(\pm 1, 0)$. The corresponding sequence of KETpic commands are given below.

```
> g6:=contourplot(((x+1)^2+y^2)^(-1/2)+((x-1)^2+y^2)^(-1/2),
  x=XMIN..XMAX,y=YMIN..YMAX,grid=[100,100],
  contours=[3,2,1.5,1.2,0.95]):
```

There are no technical difficulties in this case. As well, it is not difficult to plot parametric curves. Conformal mappings of complex functions consist of a set of parametric curves. Fig. 3 (right) shows a conformal mapping of the following complex function.

$$g(z) = \frac{1}{z}. \tag{4}$$

In this case, we emphasize its different images of $\text{Re}(z)$ and $\text{Im}(z)$. The images of $\text{Im}(z)$ in Fig. 3 (right) are plotted in bold curves. We explain the technique briefly. The corresponding sequence of KETpic commands are given below.

```

> g7:=conformal(1/z,z=-1-I..1+I):
> g8=[]:
  for i from 1 to 11 do
    g8:=[op(g8),op(i,g7)]:
  od:
> g9=[]:
  for i from 12 to 22 do
    g9:=[op(g9),op(i,g7)]:
  od:

```

The first line simply gives plotting data to a variable $g7$. The second argument in `conformal`, $z=-1-I..1+I$, indicates a range of z , *i.e.*, $|\text{Re}(z)| \leq 1$ and $|\text{Im}(z)| \leq 1$. The value of $g7$ consists of 11 (default) curves of $g(\text{Im}(z))$ and 11 ones of $g(\text{Re}(z))$ in this order. The second (resp. third) line collects the first (resp. remaining) 11 curves and saves them in a variable $g8$ (resp. $g9$). We obtain Fig. 3 (right) by writing $g8$ with the doubled width and $g9$ with the default width in a text file. The corresponding KETpic commands are as follows.

```

> drwline(g8,2):
> drwline(g9):

```

The option, “2” after $g8$ in the first command, implies multiplier of the line width.

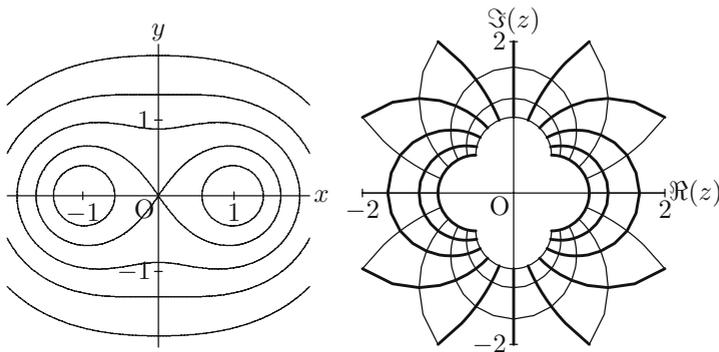


Fig. 3. Contours of a Coulomb potential in (3) (left) and a conformal mapping of a complex function in (4) (right)

4 Conclusions and Future Work

We have clarified the requirements from **R1** to **R4** for a CAS to accept our new application, CAS-aided \LaTeX plottings. We suppose the CAS be standard, which means the CAS is equipped with abilities of symbolic computing and numerical computing, programming, and showing graphical images.

Our first example is a macro package for Maple, which we call KETpic for Maple. The package is able to produce accurate and richly expressive pictures with a minimal input and a reasonable effort. KETpic for Maple is available on major platforms; Windows, Macintosh or Linux. Its minimal configuration is a combination of Maple V release 5 (see [1,3]) and a DVI driver supporting Tpic (see [4]). Anyone interested in KETpic can download the latest version with its command reference and some examples from our web site [6], which are completely free of charge.

KETpic is powerful to create \LaTeX plottings but is relatively weaker at the following aspects. First, it does not support GUI. Therefore, users might have difficulties to handle KETpic. However, it is a necessary consequence of text-based user-interface of CAS which realizes accurate plottings. Second, curve fitting is one of remaining problems of KETpic because GUI environments are the best for fitting a curve. Third, at present, KETpic is not good at 3D drawings, especially surface plottings. Finally, there are no versions for other CASs. We have several plans to extend KETpic to other CASs, *e.g.*, Mathematica or free CASs. We are developing a project to improve KETpic. In addition, we are preparing its user manual.

References

1. Char, B.W. and Geddes, K.O., et al: "Maple V Library Reference Manual", (1991), Springer-Verlag.
2. Goossens, M., Rahtz, S. and Mittelbach, F.: "The \LaTeX Graphics Companion", (1997), Addison-Wesley.
3. Heal, K.M., Hansen, M.L. and Rickard, K.M.: "Maple V Learning Guide", (1996), Springer-Verlag.
4. Kwok, K.: EEPIC: Extensions to epic and \LaTeX Picture Environment Version 1.1, 1988, <http://www.ntg.nl/doc/kwok/eepic.pdf>
5. Mittelbach, F. and Goossens, M., et al: "The \LaTeX Companion", (2004), Addison-Wesley.
6. Sekiguchi, M.: <http://www.kisarazu.ac.jp/~masa/math/>
7. Sekiguchi, M., Yamashita, S. and Takato, S.: "Development of a Maple Macro Package Suitable for Drawing Fine \TeX -Pictures", (2006), Lecture Notes in Computer Science 4151 (eds. A. Iglesias & N. Takayama), pp.24–34, Springer-Verlag.