

Interfacing KetCindy and CASs, and its Applications to Scientific Problems Modeling

Setsuo Takato¹, José A. Vallejo², Masataka Kaneko³

¹ Toho University, Japan, takato@phar.toho-u.ac.jp

² Universidad Autónoma de San Luis Potosí, México, jvallejo@fc.uaslp.mx

³ Toho University, Japan, masataka.kaneko@phar.toho-u.ac.jp

1 Introduction

Nowadays the majority of college math instructors use \LaTeX when preparing materials for their classes. \LaTeX is a powerful editor specifically designed for writing scientific and mathematical formulas. In this regard, the scope of its capabilities ranges from the simplest to the most complex. However, it falls somewhat short when it comes to dealing with graphics. Making graphics with \LaTeX can be a challenging and cumbersome task, requiring additional coding skills that can divert the attention from the main goal, which is teaching the subject matter.

What most \LaTeX users do is to create their graphics with the aid of a separate program and then import them in the working \LaTeX document. Of course, more advanced users can resort to TiKZ , which is a $\text{T}_{\text{E}}\text{X}$ package with its own commands and syntax, but this is difficult for the beginner or the occasional user, due to the steep learning curve. To overcome this issue the authors developed a package called KETpic [1], using a CAS to generate the \LaTeX code for graphics. Here is a quick example of how it works: A CAS (say, Scilab [2]) user would download the folder `ketpicsciL5` and the file `ketpic.sty` both from [1]. For definiteness, suppose that, in a Linux box, they are downloaded to the directory `/home/username/ketpic`. Then, in a Scilab session the user would write the following sequence of commands:

```
cd("/home/username/ketpic");
Ketlib=lib("ketpicsciL5");
Ketint();
gr1=Plotdata("x^2","x");
gr2=Lineplot([-1,1],[2,4]);
h1=Hatchdata("ii",list(gr1,"n"),list(gr2,"s"));
Openfile("fig1");
Drwline(gr1,gr2,h1);
Expr([2,4],"se","y=x+2",[2,5],"nw","y=x^2");
Closefile("1");
```

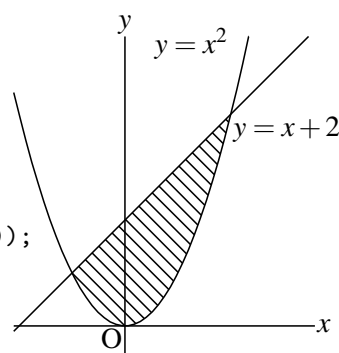


Figure 1

This creates a `fig1.tex` file in the directory `/home/username/ketpic`, containing the \LaTeX code that generates Figure 1. To use it, the user must add `\usepackage{ketpic}` to the preamble of her document; then, the command line `\input{fig1.tex}` will do the job. The resulting `.tex` document should be compiled with \XeLaTeX or \LaTeX for better results.

2 Migration from \KETpic to \KETCindy

\KETCindy [1] is an upgrade of \KETpic which combines its features with those of the popular dynamic geometry software Cinderella. Cinderella works as a GUI for \KETCindy : Graphics are shown in Cinderella via a \KETCindy script while Scilab runs in the background. The interactive features of the latter allow editing the figures via the CindyScript editor. A brief procedural step of the process is shown in Figure 2.

1. Draw geometric objects on the Cinderella worksheet.
2. Write a script as in Section 1 to generate the \LaTeX code for the figure.
3. Execute a batch process of Scilab, \LaTeX compiler and a PDF viewer.

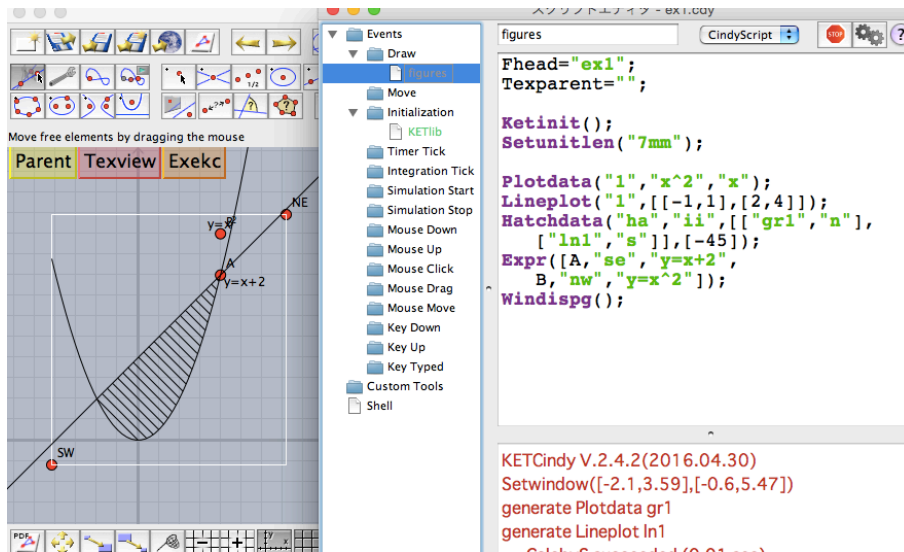


Figure 2

\KETCindy 's codes are short and compact, yet they generate high resolution images as good as \KETpic 's ones, suitable for inclusion in high quality documentation projects or research papers.

3 Interfacing with CASs

The most recent version of $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ includes capabilities of interfacing with Computer Algebra Systems others than Scilab. In this paper we illustrate the case of Maxima [4], but there exist versions for Fricas [5] and Risa/Asir [6]. The flow chart of these interfaces is shown in Figure 3.

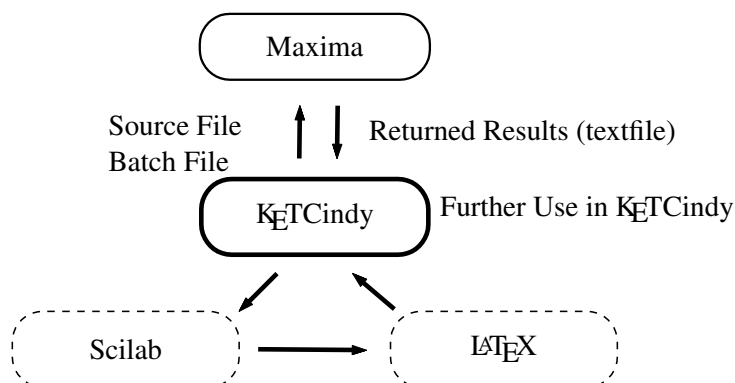


Figure 3

When interfacing with Maxima, the command `Mxfun` is all we need to complete the task. Other commands such as `calcbym` and `Mxtex` may be used for multistep and code conversion to $\text{L}\text{A}\text{T}\text{E}\text{X}$, respectively. The output of the CAS is returned to $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ as a character string for further processing of the graphics. Used this way, $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ proves to be a powerful companion to CASs.

4 Examples

4.1 Fourier Series

The second author has developed a package for computing the Fourier series of piecewise defined functions in Maxima, called `fourier_sec`. Below we show the code needed to compute the Fourier expansion of degree 10 of a square wave.

```
cmdL=Concat(Mxbatch("fourier_sec"), [
  "hs(x):=if (x<0 and x>=-1) then 0 "
  +"elseif (0<=x and x<=1) then 1", [],
  "c:fourier_sec_coeff", ["hs(x)", "x"],
  "c[1]::c[2]::c[3]", []
]);
Calcbym("c", cmdL, []);
```

```

c0=text(c_1); cn=text(c_2);
Mxtex("2",c_3);
sn=replace(c_3,"%pi","pi");
fs=c0;
forall(1..10,
  tmp1=assign(("+cn+)*cos(n*pi*x)","n,#);
  tmp2=assign(("+sn+)*sin(n*pi*x)","n,#);
  fs=fs+" "+tmp1+" "+tmp2;
);
Plotdata("2",fs,"x",["Num=400"]);
Expr([D,"e","s_n="+tx2,E,"e","n=10"]);

```

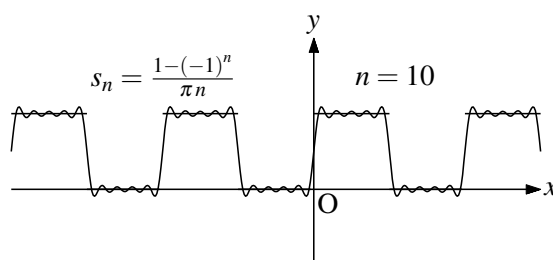


Figure 4

4.2 Generating 3D Models

One of the appealing features of K_ET_Cindy is its capability of displaying 3D graphics (through .obj files). These are to be viewed with programs such as Meshlab [7]. Once the basic 3D graphics has been coded, and prior sending the file to a 3D printer, using Maxima the surface can be fine-tuned by giving it the desired thickness (this is required by the 3D printing process), in addition to other features. The final code, along with its 3D printed version, is shown in Figure 5.

```

fd=["p","x=U*cos(V)","y=U*sin(V)","z=cos(V)^2-sin(V)^2",
  "U=[0,2]","V=[0,2*pi]","e"];
tmp=Mkobjnrm("1",fd);
cmdL=["assume",["U>0"],"a:trigsimp",[tmp],"a",[]];
CalcbyM("ans",cmdL);
Mxfun("1","solve",[den,"C"]);
cmdL=["assume",["U>0"],
  "a1:limit",[norm1,"V","0"],...];
CalcbyM("lim1",cmdL,[""]);
norm="if or(V=%p/2*[0,1,2,3,4]) then Out=[0,0,1];"
  +"else Out="+norm+"end";
cmd=Mkobjthickcmd("1",fd,norm,[0.05,"+n+s-e-w+"]);
Mkviewobj("ds",cmd),["m","v"];

```

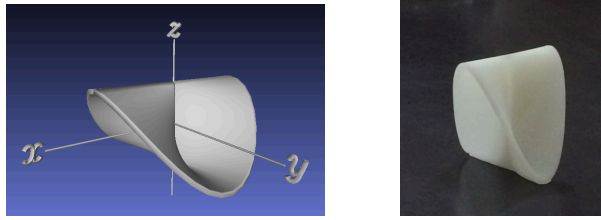


Figure 5

5 Conclusions

The goal of the paper is to show by way of examples how our package $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ can be used to generate high quality 2D and 3D graphics to be used within technical $\text{L}\text{A}\text{T}\text{E}\text{X}$ documents. Also, a brief description of the interface between several CASs and $\text{L}\text{A}\text{T}\text{E}\text{X}$ through $\text{K}\epsilon\text{T}\text{C}\text{i}\text{n}\text{d}\text{y}$ is presented.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 25350370, 15K01037, 15K00944. The second author was partially supported by the Mexican Consejo Nacional de Ciencia y Tecnología, Project CB-2012 179115.

References

- [1] KeTpic/KeTCindy, <http://ketpic.com/?lang=english>
- [2] Scilab, <http://www.scilab.org>
- [3] Cinderella, <http://www.cinderella.de>
- [4] Maxima, <http://maxima.sourceforge.net>
- [5] Fricas, <http://fricas.sourceforge.net>
- [6] Risa/Asir, <http://www.math.kobe-u.ac.jp/Asir/asir.html>
- [7] Meshlab, <http://meshlab.sourceforge.net>
- [8] Kaneko M., Yamashita S., Kitahara K., Maeda Y., Nakamura Y., Kortenkamp U, Takato S., KETCindy— Collaboration of Cinderella and $\text{K}\epsilon\text{T}\text{p}\text{i}\text{c}$, Reports on CADGME 2014 Conference Working Group, The International Journal for Technology in Mathematics Education, 22(4), 179–185, 2015.
- [9] Takato S., Hamaguchi N., Sarafian H., Generating Data of Mathematical Figures for 3D Printers with $\text{K}\epsilon\text{T}\text{p}\text{i}\text{c}$ and Educational Impact of the Printed Models, ICMS 2014, LNCS, vol. 8592, pp. 629–634, Springer, Heidelberg, 2014.
- [10] Kaneko M, Takato S., The effective use of LaTeX drawing in linear algebra, The Electronic Journal of Mathematics and Technology 5(2), pp. 1–20, 2011.