Programming in KeTCindy with Combined Use of Cinderella and Maxima

S. Takato[1], S,Yamashita[2], J.A,Vallejo[3]

[1] *Toho University, Japan, {takato}@phar.toho-u.ac.jp*
[2] *National Institute of Technology, Kisarazu College, Japan, yamasita@kisarazu.ac.jp*
[3] *Universidad Autónoma de San Luis Potosí, México, jvallejo@fciencias.uaslp.mx*

Mathematics teachers at the college level often distribute printed materials to their alumni. For such materials, figures presented as line drawings are better suited, because students can write their own remarks over them on the paper. KeTCindy, a macro package of CindyScript (which is a programming language implemented in Cinderella), can produce fine figures for LaTeX. Furthermore, KeTCindy supports line drawing of 3D figures as explained below.

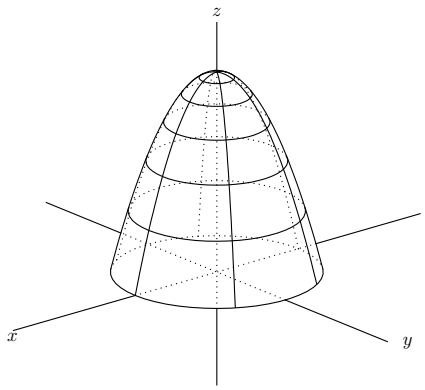

Fig.1

To produce these 3D figures, KeTCindy follows the following steps:

1. To find silhouette lines of the surface, in the figure above those are given by $x = u\cos v,\ y = u\sin v,\ z = 4-u^2$. Data are obtained from an implicit function of the form

$$J(u,v) = \frac{dX}{du}\frac{dY}{dv} - \frac{dX}{dv}\frac{dY}{du} = 0\,,$$

   where $(X,Y) = \mathrm{Proj}(x,y,z)$ is the map to the plane of projection.

2. To find the intersections of silhouette lines and a projection curve.

3. To divide the curve by these intersects, and to decide whether each separation is hidden by the surface or not.

Of the above, the second item is of fundamental importance, but it represents a difficult task in the case of contacting curves because curves are numerically polygonal lines. The following figures demonstrate this setting: The right panel shows an enlarged figure at a contact point presented on the left.

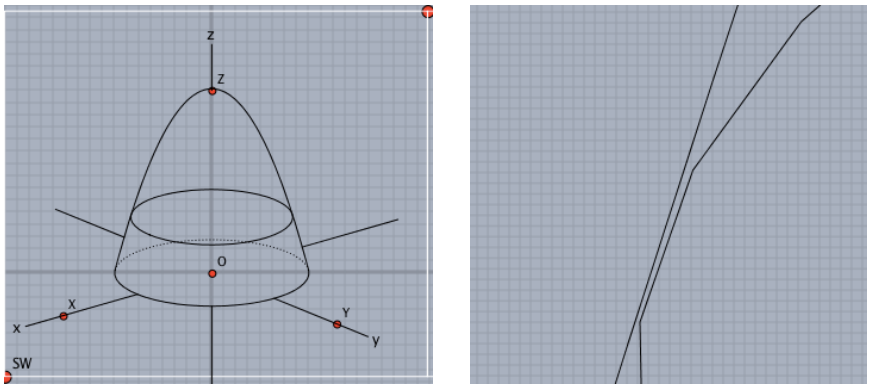

Fig.2

To refine the calculation of item 2, we have adopted an interpolatory scheme using Bézier curves near the contact point. Then we use a formula developed by Oshima[**?**] to decide the control points.

The left in the following is a further enlarged figure. The right shows Bézier curves in red color.

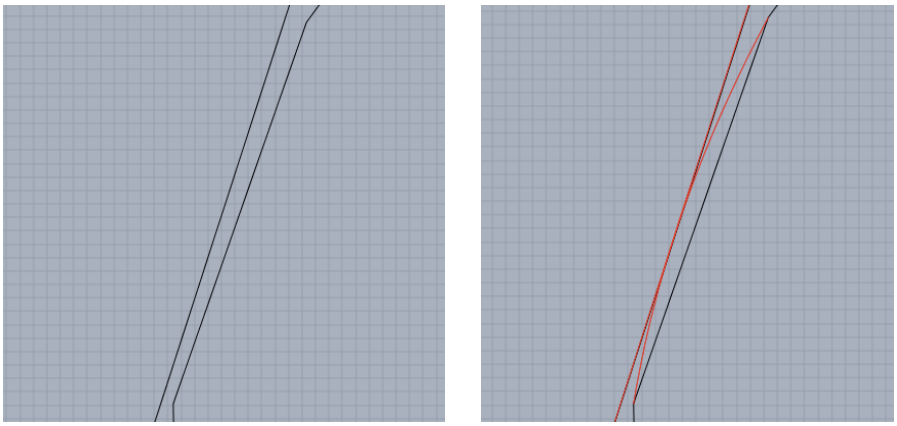

Fig.3

In this setting, the intersect is represented by a cluster of points. One of them is

$$\mathrm{P} = [-1.65827, 1.20578]. \tag{1}$$

KETCindy can also call Maxima from Cinderella and return a result back to Cinderella. For example, the intersect for Figure 2 is calculable using the following script in CindyScript. The result is:

$$\mathrm{P} = [-1.656701299244927, 1.210755779027779], \tag{2}$$

confirming that (1) is a very good approximation to the contact point.

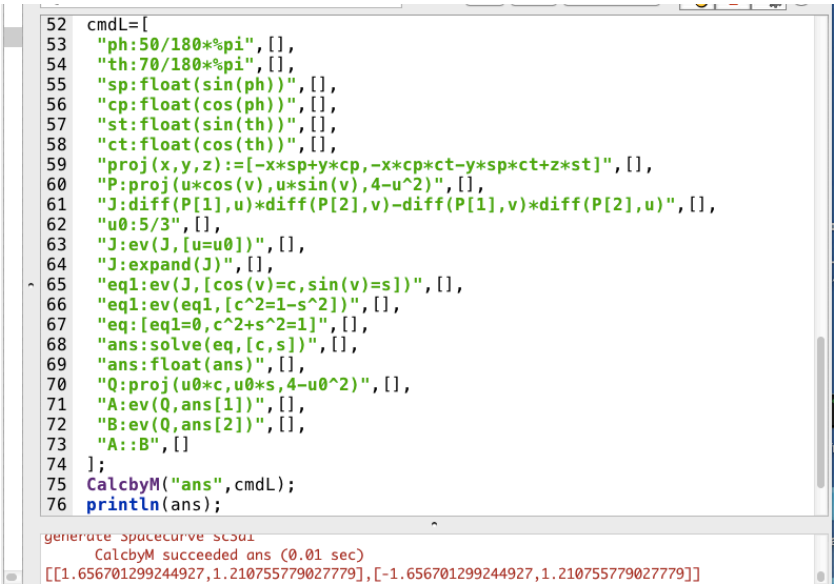

Fig.4

As a conclusion, we could say that the combined use of KETCindy, Cinderella, and Maxima is an effective tool to develop programs for surface drawing.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 16K01152.

References

[1] Oshima, T., Drawing curves, Symposium MEIS2015: Mathematical Progress in Expressive Image Synthesis, MI Lecture Notes 2015 **64**, 117–120, Kyushu University, 2015

[2] Takato S., What is and how to Use KeTCindy – Linkage Between Dynamic Geometry Software and Collaborative Use of KetCindy and Free Computer Algebra Systems and LaTeX Graphics Capabilities –, Mathematical Software –ICMS 2016, LNCS **9725**, 371–379, Springer, 2016.