

## Assimilation of mathematical knowledge using Maple

ZOLTÁN KOVÁCS and LÁSZLÓ KOZMA

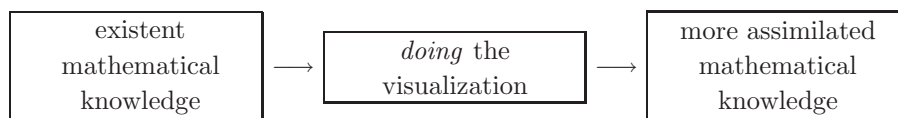
*Abstract.* For more than four years we have been teaching a Maple course at University of Debrecen for prospective mathematics teachers. The aim of the course is that students get some experience on mathematical visualization with Maple. At the last part of the course the student is provided with a problem of geometrical flavor. Within three or four weeks he/she must obtain a solution. In this paper we present and analyze two of student projects: rotation of the hypercube and drawing of complex functions. The concluding remark is that most of the students will profit from using Maple for such type of problems: it helps to assimilate mathematical knowledge.

*Key words and phrases:* using CAS in mathematics teaching, mathematical visualization, hypercube, complex functions.

*ZDM Subject Classification:* R25, R69, G85, I85.

### Introduction

The pedagogical model of our course is briefly the following:



*Figure 1*

We use the phrase “assimilated knowledge” in sense of [5]. The phrase “*doing* the visualization” needs more detailed explanation. Mathematical visualization helps understanding abstract objects by itself. But creating a visualization module (may be technically poor) is more than simply admiring a professional presentation demonstrated by a lecturer.

R. S. Palais named mathematical visualization as marriage of mathematics and computer science [4]. The “mathematical part” is supposed to be existent in our course. The “computer science part” is the Maple itself. Using Maple exempts the student from being a crafted computer programmer. Students seem to have least difficulty with learning the syntax and using thirty or forty Maple commands.

Our 14–15 week (28–30 hours=14–15 double periods) course has been organized as follows:

- General presentation on Computer Algebra Systems and Maple. (1 double period.)
- Samples of the most important mathematical features. (2 d-periods.)
- Tutorial on Maple graphics. (2 d-periods.)
- Introduction to programming with Maple. (1 d-period.)
- Drawing up the projects. (1 d-period.)
- Solution of a visualization problem. (4 d-periods plus homework.)
- Presentation of the projects. (1 d-period.)

The “tutorial part” of the course consists of a general presentation followed by a sequence of practical exercises that the student has to execute in the classroom. This syllabus may be considered as a standard short introduction to Maple.

In the “project part” students work of course independently with our guidance.

Finding adequate projects that can straightly and quickly “Maple-ized” is a cautious job. The first group of projects is connected with the teacher’s practice. For example the student of the project “rotation of the cube around its body-diagonal” gave one of the most successful solutions for four years. (See Figure (a) on the color plate on page 423.) The funny thing about it is that this simple, static figure is more expressive than the animation itself. The original figure was post-processed by the ray-tracing system POV-Ray, see [www.povray.org](http://www.povray.org). The second group of projects is connected with some “higher mathematics”. In this paper we analyze two of the projects of this type. They have one common

feature: they are about visualizing 4-dimensional objects, but the final solutions are completely different.

In the classroom we had got Maple V Release 5, here we updated the worksheets to Maple 7. Moreover we translated the student's comments to English. Apart from this we left the student's work unchanged. We remark that the effectiveness of these procedures could be increased, but the capability of procedures was of minor importance from our point of view. Moreover, we do not claim any originality for these projects. However, we hope that this paper will be stimulating to organize such courses.

## 1. Rotation of the hypercube

Aim of the project: *Visualize the 4-dimensional cube (8-cell) and rotate it.*

Using any type of programming language (such as Maple) requires approaching mathematics structurally. So we divide the project to sub-modules:

- (1) Construction of the 8-cell, describing of the face-structure.  
Preliminary knowledge: Geometry (convexity, regular politopes.)
- (2) Representation of the "4D-world  $\rightarrow$  screen" transformation, which is an  $\mathbb{R}^4 \rightarrow \mathbb{R}^2$  map. Preliminary knowledge: Descriptive geometry (central and parallel projection, axonometry.)
- (3) Rotation. Preliminary knowledge: Linear algebra (orthogonal transformations.)
- (4) Organization of the Maple worksheet.

When we offered the project we reviewed the method which can be found e.g. in the wonderful classic on mathematical visualization *Geometry and the Imagination* [3, 23.§]. First we map the vertices to a hyperplane of  $\mathbb{R}^4$  with central projection, afterwards we apply a parallel projection to  $\mathbb{R}^3$ . Finally for this image in  $\mathbb{R}^3$  we can apply Maple's own algorithm in order to map to the screen ( $\mathbb{R}^2$ ).

### 1.1. Construction of the 8-cell, describing of the face-structure ([1])

The 4-dimensional hypercube is the convex hull of the points

$$\{(i_1, i_2, i_3, i_4) \mid i_j = \pm 1, j = 1, 2, 3, 4\}.$$

Fixing one of them we get the faces of the hypercube, e.g. the convex hull of the points  $(+1, i_2, i_3, i_4)$  is one of the 8 faces. Further on, we get faces of the faces (2-faces) by fixing one more coordinate (e.g.  $(+1, +1, i_3, i_4)$ ). These 2-faces are (2-dimensional) squares. These squares (altogether 24 items) would be represented in our figure.

In the worksheet the vertices are organized in a matrix:

```
#Matrix of vertices
CS:=array(1..16,1..4,
  [[-1,-1,-1,-1], [-1,-1,-1,1], [-1,-1,1,-1],
  [-1,-1,1,1], [-1,1,-1,-1], [-1,1,-1,1],
  [-1,1,1,-1], [-1,1,1,1], [1,-1,-1,-1],
  [1,-1,-1,1], [1,-1,1,-1], [1,-1,1,1],
  [1,1,-1,-1], [1,1,-1,1], [1,1,1,-1], [1,1,1,1]]):
```

## 1.2. The 4D-world to screen transformation

The image hyperplane is a subspace with normal  $A = (A_1, A_2, A_3, A_4)$ , so its equation is

$$\langle x, A \rangle = x_1 A_1 + x_2 A_2 + x_3 A_3 + x_4 A_4 = 0.$$

The center of the projection is  $c = (c_1, c_2, c_3, c_4)$ . Determining the projection of the point  $p \in \mathbb{R}^4$  we find the parameter  $t$  such that:

$$\langle c + t(p - c), A \rangle = 0.$$

Solving for  $t$  we get the central projection of  $p$  to the subspace above:

$$p' = c + \frac{\langle c, A \rangle}{\langle c - p, A \rangle} \cdot (p - c).$$

```
#Normal of the target plane.
A1:=0:A2:=0:A3:=0:A4:=1:
#Center of the projection.
c1:=0:c2:=0:c3:=0:c4:=4:
#Scalar factor.
ff:=(x1,x2,x3,x4)->
  ((A1*c1 + A2*c2 + A3*c3 + A4*c4)/((c1-x1)*A1 +
  (c2-x2)*A2 + (c3-x3)*A3 + (c4-x4)*A4));
#Component functions.
f1:=(x1,x2,x3,x4)->c1 + ff(x1,x2,x3,x4)*(x1-c1);
f2:=(x1,x2,x3,x4)->c2 + ff(x1,x2,x3,x4)*(x2-c2);
f3:=(x1,x2,x3,x4)->c3 + ff(x1,x2,x3,x4)*(x3-c3);
f4:=(x1,x2,x3,x4)->c4 + ff(x1,x2,x3,x4)*(x4-c4);
#The central projection followed by a parallel projection to x4=0.
proj:=(x1,x2,x3,x4)->(f1(x1,x2,x3,x4),f2(x1,x2,x3,x4), f3(x1,x2,x3,x4));
```

### 1.3. The rotation

The rotation (around a two-dimensional subspace) is given by a  $4 \times 4$ -type orthogonal matrix in canonical form.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos \beta & -\sin \beta \\ 0 & 0 & \sin \beta & \cos \beta \end{pmatrix}.$$

In the worksheet the student used a composition of two rotations, doing the worksheet more flexible:

$$\begin{pmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & \cos \beta & -\sin \beta \\ 0 & 0 & \sin \beta & \cos \beta \end{pmatrix}.$$

#The rotation with parameters alpha, beta.

```
rot:=(x1,x2,x3,x4,alpha,beta)->
(cos(alpha)*x1-sin(alpha)*x2, sin(alpha)*x1+cos(alpha)*x2,
cos(beta)*x3-sin(beta)*x4, sin(beta)*x3+cos(beta)*x4);
```

The final map is the composition of the rotation and the projection.

#The composition map.

```
projrot:=proj@rot;
```

#Image of the i-th vertex after the composition map.

```
W:=(i,alpha,beta)->
[projrot(CS[i,1], CS[i,2], CS[i,3], CS[i,4], alpha, beta)];
```

### 1.4. The Maple procedure

The 2-face structure of the hypercube is given and transformed by the composition map. The defined `hypercube` procedure gives the image of the rotated hypercube by the parameters  $\alpha$  and  $\beta$ .

```
hypercube:=proc(alpha,beta)
local V, l1, l2, l3, l4, l5, l6, l7, l8, l9, l10,
l11, l12, l13, l14, l15, l16, l17, l18, l19, l20,
l21, l22, l23, l24;
#Image of the i-th vertex.
V:=i->W(i,alpha,beta);
#2-face structure of the hypercube.
l1 :=POLYGONS([V(1),V(3),V(4),V(2)]):
l2 :=POLYGONS([V(1),V(5),V(6),V(2)]):
l3 :=POLYGONS([V(1),V(45),V(7),V(3)]):
```

```

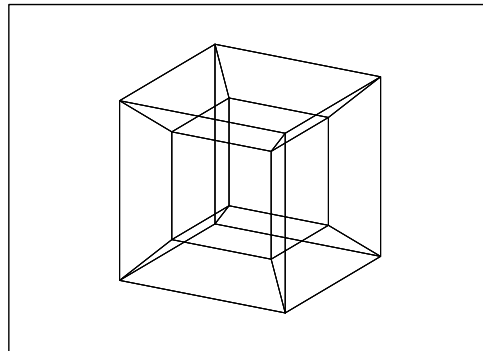
14 :=POLYGONS([V(1),V(9),V(10),V(2)]):
15 :=POLYGONS([V(1),V(9),V(11),V(3)]):
16 :=POLYGONS([V(1),V(9),V(13),V(5)]):
17 :=POLYGONS([V(5),V(7),V(8),V(6)]):
18 :=POLYGONS([V(3),V(7),V(8),V(4)]):
19 :=POLYGONS([V(2),V(6),V(8),V(4)]):
110 :=POLYGONS([V(3),V(11),V(12),V(4)]):
111 :=POLYGONS([V(2),V(10),V(12),V(4)]):
112 :=POLYGONS([V(2),V(10),V(14),V(6)]):
113 :=POLYGONS([V(9),V(11),V(12),V(10)]):
114 :=POLYGONS([V(9),V(13),V(14),V(10)]):
115 :=POLYGONS([V(9),V(13),V(15),V(11)]):
116 :=POLYGONS([V(5),V(13),V(14),V(6)]):
117 :=POLYGONS([V(5),V(13),V(15),V(7)]):
118 :=POLYGONS([V(3),V(11),V(15),V(7)]):
119 :=POLYGONS([V(13),V(15),V(17),V(14)]):
120 :=POLYGONS([V(11),V(15),V(17),V(12)]):
121 :=POLYGONS([V(10),V(14),V(17),V(12)]):
122 :=POLYGONS([V(7),V(15),V(17),V(8)]):
123 :=POLYGONS([V(6),V(14),V(17),V(8)]):
124 :=POLYGONS([V(4),V(12),V(17),V(8)]):
#Displaying the cube in wireframe style.
display([11,12,13,14,15,16,17,18,19,110,
111,112,113,114,115,116,117,118,119,120,121,122,123,124],
style=wireframe, color=black, scaling=constrained);
end:

```

## 1.5. Figures

Figure 2 is a static one.

```
display(hypercube(0,0),orientation=[30,-70]);
```



*Figure 2*

Some phase of the animation is on the Figure 3.

```
g1:=seq(hypercube(0,6.2831853*n/5), n=0..5):
```

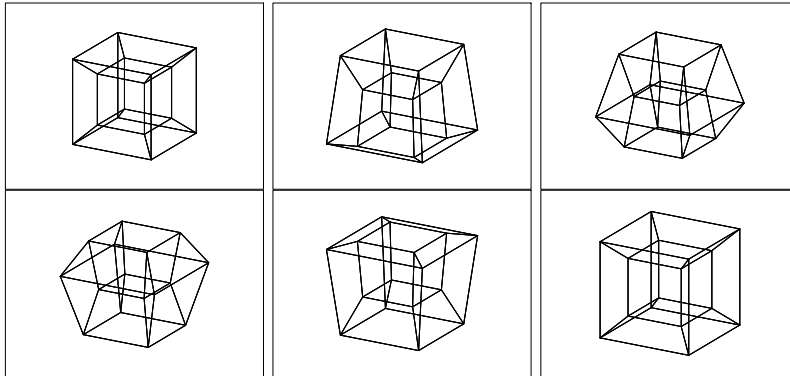


Figure 3

## 2. Visualization of complex functions with color wheel

Aim of the project: *Visualize complex functions with color wheel. Be the colors of the zero and the pole white and black, respectively.*

Let us make some comments on the motivation of the project. The graph of a function  $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  lives in  $\mathbb{R}^4$ , it can not be drawn directly. Maple has an own algorithm to draw complex functions: `plots[complexplot3D]`. This procedure draws the graph of the modulus function (i.e.  $|f(z)|$  – its graph is in  $\mathbb{R}^3$ ) and colors the surface by the polar angle of  $f(z)$ . But we were dissatisfied with the coloring method of this algorithm. For example see the drawing of the rational function  $(z^2 - i)/(z^2 + i)$  on the Figure (b) on the color plate. You can not recognize at a first glance the zeros of the function, to be more precise all complex numbers with imaginary part 0 are colored with red. (All complex numbers with common polar angles are colored with the same color.) In this respect word “coloring” means a map from the set of complex numbers to the manifold of colors. This “coloring” may be visualized by putting a color wheel on the Gauss-plane. Maple’s own color wheel is on Figure (c) on the color plate. (We used only 12 colors to demonstrate the algorithm.) The desired coloring map is on Figure (d) on the color plate. Notice that in this case the color contains the information on modulus and also on polar angle. (We used 12 colors and 10 lightness value.)

The representation of colors in most color-models is by three coordinates. So the map we need is a  $\mathbb{R}^2 \rightarrow \mathbb{R}^3$  map, called later the *coloring map*.

The sub-modules of the project:

- (1) Choosing the color-models. Preliminary knowledge: computer graphics.
- (2) Algorithm of coloring, i.e. construction of the coloring map  $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ . Preliminary knowledge: Geometry, stereographic projection.
- (3) Procedure of coloring according to the coloring map.

### 2.1. Choosing the color-model

In the desired algorithm the polar angle of the complex number gives the *hue* value ( $H$ ) of the color, the module of the complex number is connected with the *lightness* ( $L$ ) of the color. So it is practical to use the HLS color model (hue, lightness, saturation). The colors in the HLS model live in a double cone, see Figure 4. For a given  $(H, L)$  pair we can choose the maximal saturation value. (We choose color from the boundary of the double cone.)

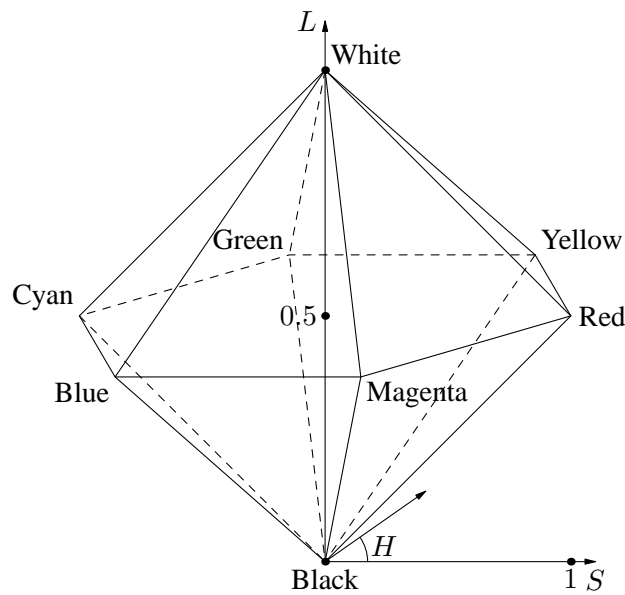


Figure 4. The double cone of the HLS color model, after [2, Fig. 13.35].



Maple uses RGB (Red, Green, Blue) or HSV (Hue, Saturation, Value) color model.

The algorithm for converting from HLS to RGB is given after [2, 13.3.5].

```
#HLS to RGB
val:=proc(x,y,h)
  local hh:
  if h>360.0 then hh:=h-360.0 else
  if h<0.0 then hh:=h+360.0
  else hh:=h
  end if
end if:
if hh<60.0 then x+(y-x)*hh/60.0
else if hh<180.0 then y
else if hh<240.0 then x+(y-x)*(240.0-hh)/60.0
else x
end if
end if:
end if:
end proc:
HLStoRGB:=proc(h,l,s)
  local m1,m2:
  if l<=0.5 then m2:=(l*(1+s))
  else m2:= (l*s-l)
  end if:
  m1:=2.0*m2:
  [val(m1,m2,h+120.0), val(m1,m2,h), val(m1,m2,h-120.0)]
end proc:
```

## 2.2. Algorithm of coloring

Map the Gauss plane by stereographic projection to the sphere of radius 1 centered at the origin. Using polar coordinates  $(\theta, \phi)$  on the sphere,  $\phi(z) = \text{Arg}(z)$ ,  $\theta(z) = \pi - 2 \text{ArcTan}(|z|)$ , see Figure 5.

Moreover,

$$H = \phi \in [0, 360], \quad L = (\pi - 2 \cdot \text{ArcTan}(|z|))/\pi \in [0, 1].$$

```
#Floating-point pi.
mypi:=3.14159;
#Polar angle of z is in [0, 360].
myargument:=proc(z)
  if evalf(argument(z))<0.0
  then evalf((argument(z)+2*mypi)/(mypi)*180.0)
  else evalf(argument(z)/mypi*180.0)
  end if:
end proc:
#Setting S to the maximum.
```

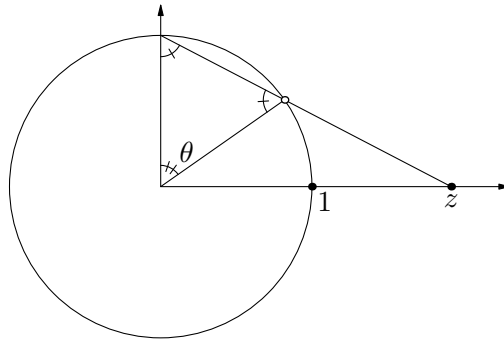


Figure 5

```

smax:=proc(x)
  if x<=0.5 then 2*x else 2-2*x
  end if:
end proc:
#Map from the Gauss plane to the double cone of colors
colormap:=z->(myargument(z),
  (mypi-2*evalf(arctan(abs(z))))/mypi,
  smax((mypi-2*evalf(arctan(abs(z))))/mypi));

```

### 2.3. Coloring of the Gauss plane

We find for this job a ready-to-use module in Maple reference book: [6, 9.9 – Programming with Color].

```

#ComplexColorM colors the rectangle [u1,u2]x[v1,v2] using
#the z->colormap(f(z)) function.
#The original rectangle is divided to nxm sub-rectangles.
#The procedure:
ComplexColorM:=proc(u1,u2,v1,v2,f,n,m)
  local i,j, points, colors, um, vm, flatten;
  um:=u2-u1;vm:=v2-v1;
#points = sequence of corners for rectangles
  points:=seq(seq(evalf(
    [[u1+i*um/n,v1+j*vm/m],
     [u1+(i+1)*um/n,v1+j*vm/m],
     [u1+(i+1)*um/n,v1+(j+1)*vm/m],
     [u1+i*um/n,v1+(j+1)*vm/m]]),
    i=0..n-1),j=0..m-1);
#flatten turns a listlist into a sequence
  flatten:=a->op(map(op,a));
#colors = RGB color values by the colormap
  colors:=COLOUR(RGB,

```

```

    flatten([seq( seq( evalf(
      (HLStoRGB(colormap(f(u1+i*um/n+I*(v1+j*vm/m))))),
      i=0..n-1),j=0..m-1)]));
#The final plot.
  PLOT(POLYGONS(points,colors), SCALING(CONSTRAINED),
    STYLE(PATCHNOGRID));
end proc:

```

On Figure (e) on the color plate is the picture of the rational function  $f(z) = (z^2 - i)/(z^2 + i)$ . See the two zeros!

On Figure (f) on the color plate is the picture of the function  $f(z) = (z^5 - 1)^2$ . It can be seen that the zeros are of second order, because all colors appear twice around a zero.

## References

- [1] Marcel Berger, *Geometrie*, CEDIC, 1977.
- [2] James D. Foley, Andries van Dam, Steven K. Feiner and John F. Hughes, *Computer Graphics, Second edition in C*, Addison-Wesley Publishing Company, 1996.
- [3] David Hilbert and Stefan Cohn-Vossen, *Geometry and the Imagination*, Chelsea, 1952.
- [4] Richard Palais, The visualization of mathematics: towards a mathematical exploratorium, *Notices Amer. Math. Soc.* **46**, no. 6 (1999), 647–658.
- [5] G. Polya, *Mathematical Discovery*, John Wiley and Sons, Inc., 1965.
- [6] *Maple 7 Programming Guide*, Waterloo Maple Inc., 2001.

ZOLTÁN KOVÁCS  
 INSTITUTE OF MATHEMATICS AND COMPUTER SCIENCE  
 COLLEGE OF NYÍREGYHÁZA  
 H-4401 NYÍREGYHÁZA, P.O. BOX 166.  
 HUNGARY

AND  
 INSTITUTE OF MATHEMATICS  
 UNIVERSITY OF DEBRECEN  
 H-4010 DEBRECEN P.O. BOX 12.  
 HUNGARY

*E-mail:* kovacs@nyf.hu

LÁSZLÓ KOZMA  
 INSTITUTE OF MATHEMATICS  
 UNIVERSITY OF DEBRECEN  
 H-4010 DEBRECEN P.O. BOX 12.  
 HUNGARY

*E-mail:* kozma@math.klte.hu

(Received July, 2003)