

# **Experiences in the Integration of CAS into Engineering Education at the University of Pécs**

**Csaba Sárvári, Pécs (Hungary)**

## **Abstract**

CAS has a significant influence on all elements of the mathematics teaching and learning process. The involvement of CAS in teaching alters the structure of the material, the development of students' mathematical abilities, and attitudes of both students and teachers. There is an overall change in the pragmatic and epistemic values present in the education process. For instance, problem solving activities may become richer by means of new heuristics and novel approaches to experimentation and modelling.

The integration of CAS in mathematics education provides significant and permanent didactical challenges and requires substantial reconsideration of teaching approaches. In the Department of Engineering at the University of Pécs, we have been employing a variety of CAS applications for the teaching of informatics students since 1998. Applying our accumulated experience, we have been continuously upgrading CAS-assisted teaching approaches in our courses. We consider modularization as a fundamental didactical issue in improving these courses.

## **Characteristics of courses and the learning environment**

The adequate form for the use of CAS depends strongly on course characteristics. Our CAS rich courses have the following characteristics:

- courses designed for informatics students in our university
- courses employ characters, therefore, pragmatic elaboration is necessary
- course schedule is tight.

The previous features make it desirable to use ICT, particularly CAS, and carry important challenges in using CAS.

The main technological elements of our courses are the following:

- We use Maple CAS in all elements (lectures, practical sessions, exams) of teaching
- Students may access all course material on our intranet and our E-Learning website
- The computer laboratory is equipped with a Tight-VNC control system allowing several connections to the same desktop. With the help of this system students' screens can be projected as well. Thus, nice solutions and common errors can be shared easily with all the students. In this way we can realise effective instrumental orchestration.

## **Curriculum based modularisation**

The use of CAS causes fundamental changes for the whole teaching and learning process in mathematics. Our starting point is to make a connection between CAS and curriculum. If we study the theme of operative knowledge and skills as a critical and important component of teaching, then the crucial point for us is the modularization of the course material.

First, we have to take into consideration that mathematics and curriculum have a modular structure. If we use CAS-modules, built in or constructed by a learning community, we have to synchronize CAS-modularization with both of them.

For the mathematical way of thinking and communicating it is necessary to represent the elements of mathematical structures, as well as the structures themselves. The representations can be sorted into two types.

For effective communication we need external representations, presented as physical objects, pictures, and spoken language (Lesh, Post and Behr, 1987). In the development of the thinking process internal representation is used. These representations concern mental images corresponding to internal formulations we construct from reality. According to results in cognitive science, there is a correlation between internal and external representations. This indicates that the effectiveness of teaching for comprehension of mathematical concepts greatly depends on using the appropriate representation. On the basis of examining the mathematical learning process, we can say that the internal representations are greatly determined by the mathematical representations.

The effectiveness of mathematical knowledge, in a narrow sense the comprehension itself, can only be approached from an organisational perspective of knowledge. Hiebert and Carpenter (1992) define comprehension in mathematics as follows:

A mathematical idea, procedure or fact can only be understood if its mental representation is part of the network of representations. The degree of the comprehension is determined by the number and strength of connections. A mathematical idea, procedure or fact is understood thoroughly if it is linked to existing networks with strong and numerous connections.

There are two important statements we draw from this definition. First, the use of a CAS module can allow time and the cognitive capacity for building more connections between new knowledge elements. Second, if we decide the kind of CAS-modules, then the main objective must be the role of the new knowledge-element in the curriculum. If detailed understanding of the elaborated procedure is necessary to treat other elements appropriately, then we have to carry out a white box / black box (or black box/white box) modularization.

The main benefit of using CAS in teaching and learning is that CAS offers the opportunity to extend creative learning; the possibility of structured knowledge-building. A CAS enables the ambitious student to build, with the help of teachers, their own mathematical microworlds. Therefore, it appears that the role of CAS can not be identified with a “model for the communication between mathematical experts and laypersons”, explained by W. Pescheks’ in his plenary paper (Peschek, 2005). On the contrary, application of CAS in learning mathematics is successful only if students are continuously ready to use it in a constructive manner.

In our teaching practice modularisation is a dynamic process of constructing complex CAS-procedures. We apply the following model:

- First, from the algorithm we determine a primitive series of steps or procedures.
- Second, we construct the parts of the complete algorithm as stand alone procedures. This is the semi-automatic phase.
- At the end, we paste the working parts together using the mathematical concept of composition. (Functional programming)

Creating and applying the CAS module in the described manner has two benefits. On the one hand, making such a procedure is real creative work. On the other hand, the work with self-made procedures reduces complexity and the burden of thinking. The completed procedures can be the algorithm-primitives of procedures of the next chapters. Creative use and design of

a CAS-procedure is a complex and constructive activity. Students are able to write procedures only after thorough conscientious arrangement. Nevertheless, without this activity there is not complete learning. We give examples to illustrate how in our department we have attempted to tackle the didactical challenges posed by CAS.

Using instrumentation via computer technology changes the operative activities, but does not decrease their importance. This change modifies the pragmatic, epistemic and heuristic values of the operative activities. New technology allows us to enrich the curriculum through accurate negotiation of elements, which were dealt with only superficially before. Our first example shows how problem solving with the help of CAS supports the efficient execution of these activities.

### Example 1

After studying the method of least squares, we give a procedure for computing the regression line. Then we take one after the other the experimental formulas. Using the logarithmic fitting formula as an example, we obtain a widely applicable CAS step-series, leading to a pattern for treating similar cases. Thus for a measured values data set we find a best fit function of the form

$$f(x) = a \ln(x) + b$$

We substitute  $z = \ln(x)$ , thus instead of pairs  $(x_i, y_i)$  we are working with pairs  $(\ln(x_i), y_i)$  where  $(i = 1, \dots, n)$ . With the latter pairs we compute the regression line and its coefficients are coefficients of the logarithmic function too. We execute the following steps.

1. We store the values in lists and then we zip together the two lists. The obtained structure is list of lists and immediately available to graph.

```
> X:= [seq(i, i=1..6)];
           X := [1, 2, 3, 4, 5, 6]

> Y:= [1.557258033, 3.228601822, 4.070962393, 4.785270808,
5.362137759, 5.747027974];
           Y := [1.557258033, 3.228601822, 4.070962393, 4.785270808,
5.362137759, 5.747027974]

> points:=zip((x,y)->[x,y],X,Y);
           points := [[1, 1.557258033], [2, 3.228601822], [3, 4.070962393],
[4, 4.785270808], [5, 5.362137759], [6, 5.747027974]]
```

2. We graph the points, and try to decide the type of function appropriate to fit to the points. (See Figure 1)

```
plot(points, style=point, symbol=circle, color=red, view=[0..6, 0..6]);
```

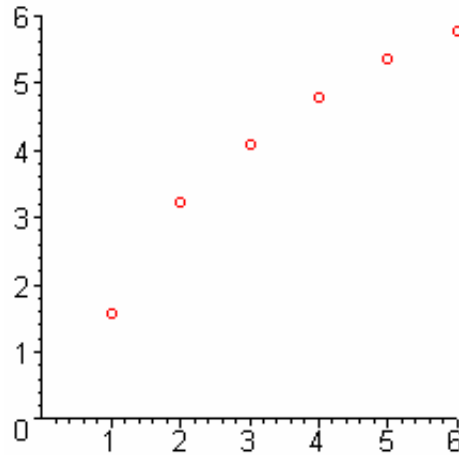


Figure 1. The graph of data points

3. We execute the transformation. It seems to us that the appropriate function is of the form  $f(x)=a \ln(x)+b$ . Therefore we have to make the transformation  $u=\ln(x)$ . This can be made with command `map` as follows:

```
> X1:=evalf(map(ln,X));
      XI := [0., 0.6931471806, 1.098612289, 1.386294361, 1.609437912,
            1.791759469]
```

4. We create a list of the transformed pairs and then compute with the help of our procedure the regression line. (Fig. 2.)

```
> pointsln:=zip((x,y)->[x,y],X1,Y);
      pointsln := [[0., 1.557258033], [0.6931471806, 3.228601822],
                  [1.098612289, 4.070962393], [1.386294361, 4.785270808],
                  [1.609437912, 5.362137759], [1.791759469, 5.747027974]]
```

```
> regressionline(pointsln);
      Sum of the squares of the deviations, 0.007413776190
      Regression line, 2.336902565 x + 1.562698290
```

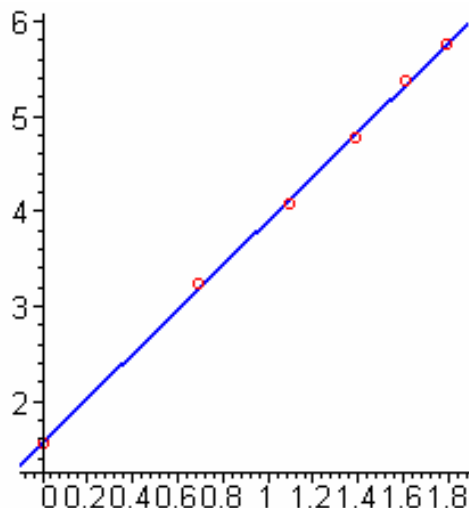


Figure 2. The output of the procedure

5. We create the function, as a suitable CAS-object.

```
> f:=unapply(a*ln(x)+b,x);
      f:=x → 2.336902565 ln(x) + 1.562698290
```

6. We graph the points and the fitted function. By so doing we examine our work at first graphically (Figure 3).

```
> abra:=plot(f(x),x=op(1,X)-
1..op(nops(X),X),color=blue,thickness=2):
> kep:=plot(points,style=point,symbol=circle):
> plots[display]({abra,kep},title=`The searched function of
form a*lnx+b`);
```

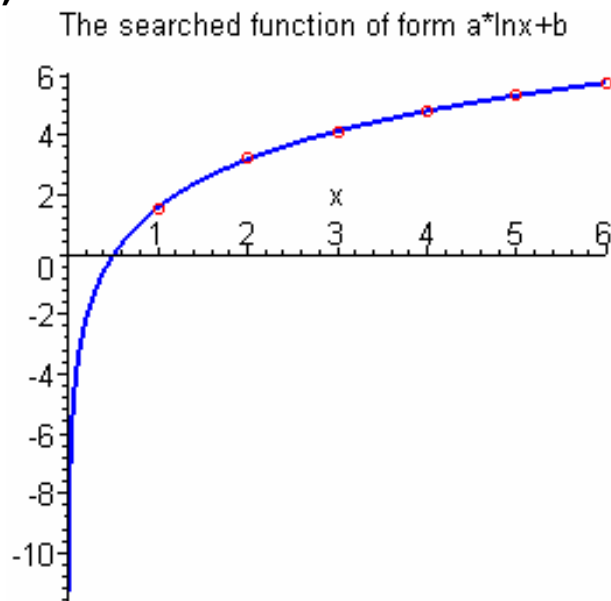


Figure 3. Graph of fitted function with the data points

7. We compute the square of deviances to obtain a numeric control.

```
> seq(evalf((f(X[i])-Y[i])^2),i=1..nops(X));
      0.00002959639623, 0.002123929351, 0.003491128571, 0.0002911231392,
      0.001469946705, 0.8052058888 10-5
```

```
> Sum('((f(X[i])-Y[i])^2)',i=1..nops(X))=sum('evalf((f(X[i])-
Y[i])^2)',i=1..nops(X));
```

$$\sum_{i=1}^6 (f(X_i) - Y_i)^2 = 0.007413776221$$

The next example shows, how can we apply the built in and self-made procedures together to obtain an effective instrument. This approach allows us to connect distant components of the curriculum. Therefore we can create new knowledge-elements that without CAS could not be part of the curriculum.

### Example 2 (Minmax approximation)

Let  $x_1, x_2, \dots, x_N$  be base points and  $y_1, y_2, \dots, y_N$  be their corresponding function values.

**Problem:**  $p^*(x)$  is a maximum  $n$ th degree polynomial ( $n + 2 \leq N$ ).  $p^*(x)$  is the polynomial that satisfies the condition that  $p^*(x)$  has the minimum of the maximum distances of  $y_1, y_2, \dots, y_N$  from the initial values at  $x = x_1, x_2, \dots, x_N$ . Then, the maximum distances are defined as

$$\max(|y_k - p(x_k)|, \quad k = 1, 2, \dots, N)$$

which is true for all maximum  $n$ th degree polynomial. Following, we choose the polynomial that has the minimum distance from all, then rephrase it as a simple linear programming problem. Denote the maximum deviation by  $E$ . Then  $p(x)$  satisfies the following inequalities:

$$\begin{aligned} -E &\leq y_1 - p(x_1) \leq E \\ -E &\leq y_2 - p(x_2) \leq E \\ -E &\leq y_3 - p(x_3) \leq E \\ &\dots\dots\dots \\ -E &\leq y_N - p(x_N) \leq E \end{aligned}$$

The polynomial  $p(x)$  can be written in the form

$$p(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n$$

Then the inequality system becomes

$$\begin{aligned} y_k - c_0 - c_1 x_k - c_2 x_k^2 - \dots - c_n x_k^n &\leq E \\ y_k - c_0 - c_1 x_k - c_2 x_k^2 - \dots - c_n x_k^n &\geq -E \end{aligned}$$

We add  $E$ , the minimum objective function, to the inequalities. Thus, it becomes a complete linear programming problem for  $c_0, c_1, \dots, c_n$  coefficients.

It can be observed that this topic is fairly pragmatic, indeed, important for general teaching practice. Numerical approximation is one of the most central topics of mathematics. However, to be able to properly cover approximation in courses we must combine seemingly distinct areas of mathematics. Furthermore, while covering the topic we must clearly highlight the specific topic and try to leave the other elements in the shadow of the central problem with the intention of reducing complexity. Thus, while teaching approximation I must focus on the pre-developed elements to assist students' initial understanding. To illustrate my method lets review an example based on the simplex-method.

#### 1. Solution in details

1.1. Lets assign a vector to  $x_i$  and  $y_i$  points. Then let's find the best-fit third degree polynomial.

> restart;

X:=[1,2,3,4,5];

Y:=[2,5,6,1,3];

1.2. We are looking for the third degree polynomial in the form:

p:=x->c0+c1\*x+c2\*x^2+c3\*x^3;

1.3. Constraints are:

```
n:=nops(X);constraints:=NULL:  
for i from 1 to n do  
  constraints:=constraints,Y[i]-p(X[i])<=e,Y[i]-p(X[i])>=-e:  
od:
```

1.4. Using the 'minimize' procedure from the 'simplex' package, we can define the 'e' objective-function respecting the given constraints. I must emphasize again that I am using a special case of the simplex-method, which is not a part of the regular curriculum due to time constraints. However, using the given procedure as a black-box element of the course, we can successfully assist students' general understanding of important concepts which are essential to the overall course. Additionally, the integration of such specific elements is not a common teaching practice.

```
> simplex[minimize](e,{constraints});  
> assign(%);
```

1.5 Now, lets sketch the best-fit polynomial.

```
points:=plot([seq([X[i],Y[i]],i=1..n)],style=point,  
             symbol=diamond,color=black):  
graf:=plot(p(x),x=X[1]..X[n],color=red):  
> plots[display]([points,graf],view=[0..6,0..7],scaling=constrained);  
> evalf(p(x));
```

## 2. The development of the procedure

To be able to experiment with the approximation of polynomials we must define parameters, rather than constants, that students can easily modify. Without parameters students must cut and copy a long command line for their modifications. However, by using a pre-developed procedure we could ease this burden, because students can conveniently change the values of parameters.

Bellow, you can find the code of such a pre-developed procedure.

```
> minmax:=proc()  
local i,X,Y,n,constraints,points,graf,k,j;  
global p;  
X:=evalf(args[1],12):  
Y:=evalf(args[2],12):  
if args[3]>nops(X)-2 or nops(X)<>nops(Y) then RETURN( Error, uncorrect data` ) fi:  
unassign('e');  
unassign('c'): p:=unapply(sum(c[k]*x^k,k=0..args[3]),x);  
n:=nops(X);constraints:=NULL:  
for i from 1 to n do  
  constraints:=constraints,Y[i]-p(X[i])<=e,Y[i]-p(X[i])>=-e:  
od:  
simplex[minimize](e,{constraints});  
assign(%);p(x);  
points:=plot([seq([X[i],Y[i]],i=1..n)],style=point,  
             symbol=diamond,color=black):  
graf:=plot(p(x),x=X[1]..X[n],color=[blue,red]):  
print(p(x):  
plots[display]([points,graf],scaling=constrained,  
               axes=framed);
```

**end:**

Similarly as in phase 1 we use the same data and execute the previous procedure.

$X := [1, 2, 3, 4, 5]$ ;  $Y := [2, 5, 6, 1, 3]$ ;

$X := [1, 2, 3, 4, 5]$

$Y := [2, 5, 6, 1, 3]$

> **minmax(X,Y,3);**

$-14.93750000 + 22.75000000 x - 7.625000000 x^2 + 0.7500000000 x^3$

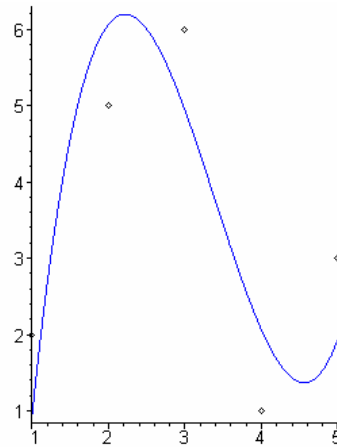


Figure 4. The fitted polynomial

The multifaceted illustration of concepts and their multiple representations clearly add to the process' epistemological value. It is essential to apply as much as possible descriptive, graphical, and numerical representations. Using such an approach can result in dynamic objects.

The extensive progress in studying highly complicated chaotic dynamical systems is mostly due to the advent of complex computer programs such as Computer Algebra Systems, which integrate numerical, graphical and symbolic capabilities. This evolution was affected by both the theoretical and the practical approaches of chaos theory and related topics, for example, fractal geometry. New results obtained by computer simulation are verified theoretically, because computations are not reliable. However, the scientists often obtain inspiration from the visual and numerical experiences generated by means of computer programs.

### Example 3

In our courses we mention and compare four classical one dimensional discrete dynamical systems: the *logistic map* with parameter 4, the *tent map* and the *sawtooth map* on the interval  $[0,1]$ , and the *Bernoulli-shift on the symbol space* demonstrating these processes (Klincsik & Sárvári, 2003). Each of them realizes features of chaotic behaviors, however these properties can be represented with more or less difficulty by using the CAS. For example, in the case of the tent map and the sawtooth map the orbits are always periodic for all rational initial values. So we can't see the chaos from these cobweb diagrams, because we have to truncate the irrational numbers when we draw these points (See Fig. 5).

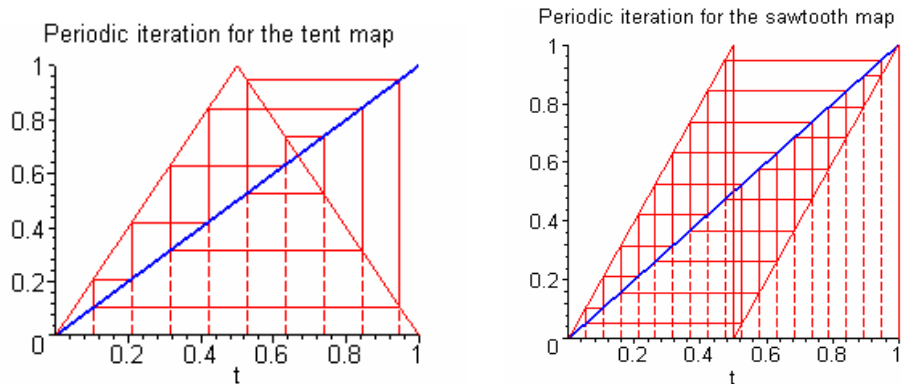
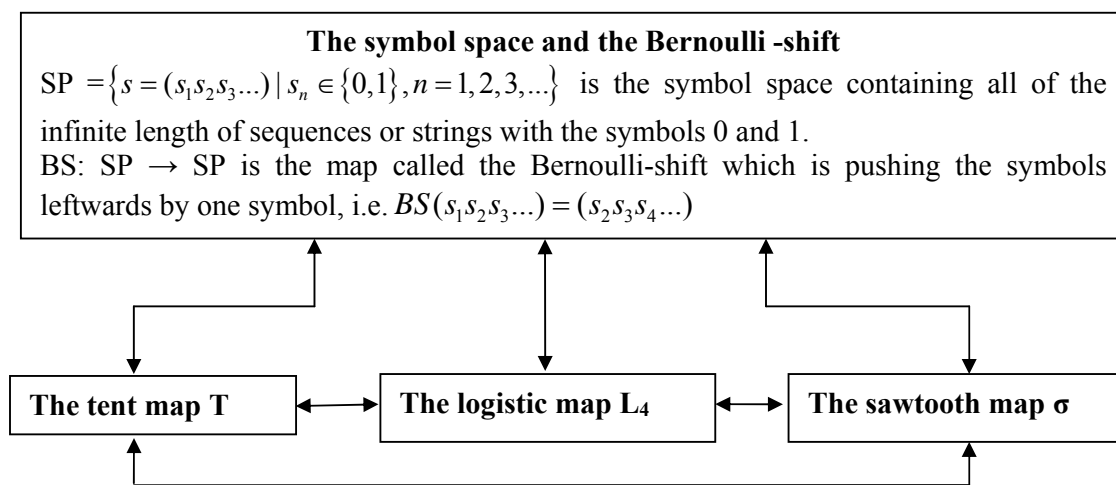


Figure.5. The orbits are periodical if we choose rational initial points for the tent and the sawtooth map

In the figure bellow we draw the four well-known chaotic discrete dynamical systems or models, where the equivalencies are indicated by the double arrows between the boxes. We give several equivalence relations from these - which are called conjugacies in the theory of dynamical systems - and we hope that the students will discover the others (See Fig 6).



Each model is a variation of the same theme, which is the chaos. The symbol space together with the Bernoulli shift plays a central role among the four models. The well-known chaotic behaviors (i.e. the existence of an **everywhere dense orbit** in the space, **the sensitive dependence from the initial values**, and the **topological mixing** property of the Bernoulli-shift) can be demonstrated easily by using the descriptive representation. The CAS does not give significantly more information for verifying these properties than the method of mathematical description, because in each individual case we can demonstrate these properties, but there always remains the infinite case without check at all. So we have to use the descriptive methods for solving the problem in the general case.

For example, when we construct the infinite string  $s^*$  containing 0's and 1's

$$s^* = (0|1|00|01|10|11|000|001|010|011|100|101|110|111|0000|\dots),$$

where we put 0's and 1's into a list and separate by vertical lines all of those substrings whose length are orderly equal to 1, 2, 3, ....., obviously we can give only a finite version of  $s^*$  by using the CAS. So we can verify that the orbit of this  $s^*$  string is everywhere dense in the symbol space by using the CAS only for cases when  $0 < \varepsilon < 1$  is a given accuracy and the strings

have finite length. Thus, in this case, we can't take a significant step towards more demonstrative tools by using the CAS.

We emphasize that the computer simulations in relation to logistical maps were one of the first experiments to emerge from the idea of chaos. Moreover, Lorenz (1963) established similar chaotic behaviors of the solutions of a particular 3-dimensional system of differential equations and he demonstrated them by computer simulations. However, he could convince the scientific world about this chaotic phenomenon only when he reduced the questions to the investigation of a one-dimensional map (Alligood, K.T., Sauer, T.D. & Yorke, J.A., 1997). The idea of the reduction was based on studying and drawing the subsequent maximums of the z-coordinates of the trajectories. When Lorenz drew the points  $[z_n, z_{n+1}]$  of the maximums  $z_n$  for  $n=1, 2, 3, \dots$ , then he got a similar figure to the tent map or logistic map (See Fig. 7).

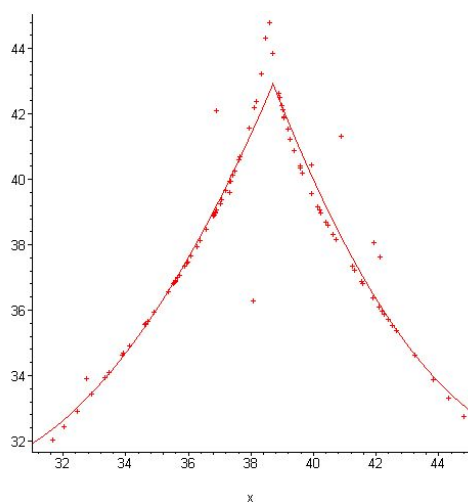


Figure 7. Pairs of  $[z_n, z_{n+1}]$  of maximum z-coordinates of the trajectory for the Lorenz differential equations and the fitted curve

It is reasonable to believe that the behavior of the whole trajectory is much more erratic than their maximum values. However, the unpredictability of the maximums is proved exactly by using the theory of discrete dynamical systems with this calculated map. The central idea in these questions was that computer programs that are able to exhibit such representation were very useful in developing these ideas.

From these examples we can conclude that computer algebra systems are often suitable for demonstrating, realizing, and initiating discoveries not only for researchers, but also for the students.

### References

- Lesh, R., Post, T., & Behr, M. (1987). Representation and translations among representation in mathematics teaching and problem solving. In C. Janvier (Ed.), *Problems of representation in the teaching and learning of mathematics* (pp. 33-40). NJ: Lawrence Erlbaum Associates
- Hiebert, J., & Carpenter, T.P. (1992). Learning and teaching with understanding. In D.A. Grouws (Ed.), *Handbook of research on mathematics teaching and learning* (pp. 65-97). New York: Macmillan.
- W. Peschek, The impact of CAS on our understanding of mathematics education, Theme 2, CAME Meeting, October 2005,
- Cs. Sárvári and M. Klincsik, From iteration to one-dimensional discrete dynamic systems using CAS, *Teaching Mathematics and Computer Science* 1 (2003) 2, 271-296.

Alligood, K.T., Sauer, T.D. & Yorke, J.A. (1997). *Chaos, An introduction to the dynamical systems*, Springer.