1/2 (2003), 383–404
tmcs@math.klte.hu
http://tmcs.math.klte.hu

Teaching
Mathematics and
Computer Science

Illustrated analysis of Rule of Four using Maple

György Maróti

Abstract. Rule of Four, as a basic didactic principle, was formulated among the NCTM 2000 standards (see [14]) and since then it is quoted by numerous books and publications (see [4], [9], [12]). Practically we can say it is accepted by the community of didactic experts. The usage of the Rule of Four, however, has been realized mainly in the field of calculus, in fact certain authors restrict the wording of the principle to the calculus itself (e.g. [3]).

Calculus is a pleasant field, indeed. A sequence of values of a function provides us with example for numeric representation, while the formula and the graph of the function illustrate symbolic and graphical representations, respectively. In the end by wording the basic features of the function on natural language we gain textual representation.

This idyllic scene, however, becomes more complex when we leave the frame of calculus. In this paper we investigate the consequences of the usage of Rule of Four outside calculus. We discuss the different types of representations and show several examples which make the multiple features of representation evident. The examples are from different fields of mathematics and are created by the computer algebra system Maple, which turns out to be an excellent tool for illustration and visualization of the maim features of mathematical objects.

Next we introduce the concept of basic representation and rational representation, which is considered as the mathematical notion of “didactic usable” or “didactic rational” representation. In the end we generalize the notion of numeric representation, which leads us a more widely usable didactic principle which can be considered as a generalization of Rule of Four.

Key words and phrases: representation theory, Rule of Four, Maple.

ZDM Subject Classification: D20.

Remark 1: Multiple representation

The wording of rule of four sounds as follows:

> *“Every topic should be presented numerically, graphically,*
> *symbolically and verbally.”*

This sentence evokes the impression that mathematical problems always have to be represented in four different manners. On the same time a given mathematical notion, problem or phenomenon can be represented several different ways, which yields that the different kinds of representations are far from being unique. As we will see later, in many cases it is practical to represent the investigated problem by more than one graphical or symbolic representation.

Example

Consider the series of real numbers for which

$$T_n = 2 * (4^n - 1)/3.$$

The series is given by formula, so the initial representation is symbolic. Displaying the numeric value of first ten members we obtain numeric representation.

```
> T := n− > 2*(4ˆn−1)/3;
```

$$T : n \to \frac{2}{3}4^n - \frac{2}{3}$$

```
> seq(T(i),i = 1..10);
```

$$2,\ 10,\ 42,\ 170,\ 682,\ 2730,\ 10922,\ 43690,\ 174762,\ 699050.$$

It can be shown that our series complies with the following recursive definition:

$$T_1 = 2,$$

$$T_{n+1} = 4 * T_n + 2.$$

The first equality can by checked easily, while the second can be proved by the usual Maple technique. We form the difference of the two sides and simplify the result.

```
> 4*T(n) + 2−T(n + 1);
```

$$\frac{8\ 4^n}{3} - \frac{2\ 4^{(n+1)}}{3}$$

```
> simplify(%);
```

$$0.$$

The recursive formula provides us with a newer symbolic representation. Next change the form of numbers in the equalities. Namely let us use the binary forms, i.e. 10 besides 2 and 100 besides 4. We reach the equalities

$$T_1 = 10,$$

$$T_{n+1} = 100 * T_n + 10,$$

which gives the third symbolic representation of the original problem.

Now consider the second equality and try to interpret it. If u is binary number then $100 * u$ coincides with $u00$, while $100 * u + 10$ is of the form $u10$. What we have done is nothing else than we have written two binary digits 1 and 0 after the number u. As the first number is 10, the second must be 1010, the third is 101010, etc. We can state that the members of series T_n as binary numbers are described by the set

$$L = \{10,\ 1010,\ 101010,\ 10101010, \dots\}.$$

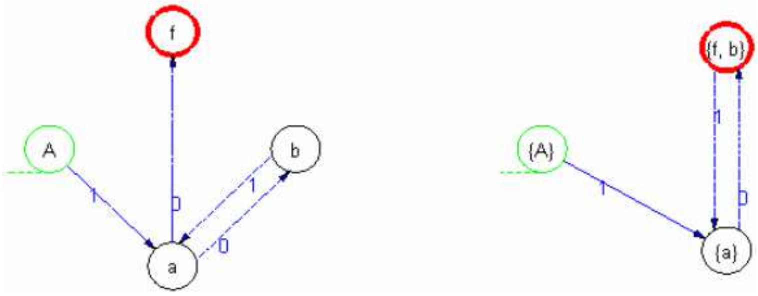
This set, however, can be considered as a language over the alphabet $X = \{0, 1\}$. As L is regular it can be recognized by finite automaton.

The underlying maple commands display the transition graph of a deterministic and a nondeterministic automaton accepting the language L. This steps offers two different graphical representations of the original problem.

```
> L = {seq(convert(2*(4^i−1)/3,binary),i = 1..5)};
```

$$L = \{10,\ 1010,\ 101010,\ 10101010,\ 1010101010\dots\}$$

```
> with(aut):
  Xi := genaut([A,1,a,a,0,{b,f},b,1,a],A):
  p1 := plotaut(Xi,[−2,0]):
  p2 := plotaut(ConDet(Xi),[2,0]):
  plots[display](p1,p2,scaling = unconstrained);
```



Let us summarize what we have done up to now. We took a series of real numbers, which we represented symbolically in its original form. Next we created a numeric and two further symbolic representations, by giving the decimal and binary form of recursive definition. The consideration of a set of binary numbers as a language over the alphabet $\{0, 1\}$ is an implicit switch from binary representation to a symbolic one. In the end we used the result of automata theory and obtained two graphical representations.

Types of representations

The example above shows that we can not speak of the best representation. Neither can we speak of unique representation. In the contrary, all four symbolic representations have had its role so that we could reach the two graphical representations by the end of our examination.

We can see that the attributes included in rule of four (numeric, graphic, symbolic and verbal) specify *types of representations* instead of concrete representations. The concrete representations are special appearances of these types.

The freedom of choosing representation

If we have more than one thing from something, then both the possibility and the constraint of choice arise. What type of representations and what concrete representation of a specific type should be chosen in the course of solving a mathematical problem?

According to my experience this could cause serious difficulties for the students. The faulty attitude, what students acquire in elementary school says that the solution of mathematical problems is nothing else than a sequence of predefined deterministic steps. In other words the solution of problems has only one way, and students have nothing else to do to find this way.

It is not easy to face the fact that the exact nature of mathematics does not mean unanimity of solutions of problems. The solvable problem does not determine uniquely the way which leads us to the solution of the problem.

How should we choose the appropriate representation? The answer is we decide it. We can choose freely, but we should keep in mind that different representations gives different level of support in the course of problem solving.

Remark 2: Base representation

Different problems have different appearance when coming into existence. One of the representations describing the phenomenon in question plays highlighted role. Namely, the one which gives the initial form of mathematical object. This representation is called the *base representation* of the problem.

Using this new notion we can state that the base representation of example of the previous section is symbolic, as we defined the problem by formula.

Remark 3: Properties of representation

Representations are not for themselves. A new representation is introduced in order to visualize, to highlight or to make easier a certain aspect or property of the subject of examination. To do so one has to provide the transfer between the different representations. The bidirectional transfer is necessary for us to be able to formulate the original problem in the new representation, and as well as to formulate the solution of the problem in the second representation in the original representation.

Example

Consider the curve given by the following system of equations

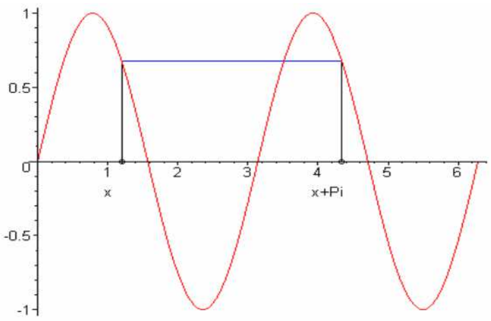
$$x = t$$

$$y = \sin(2 * t).$$

As this system of equations is nothing else that the parametric form of the function $y = \sin(2 * x)$, we state that the curve is periodic and its period is π.

How can we visualize this property of the curve in the Cartesian coordinate system? The answer is well known. For all possible value of the independent variable x the values of function $y(x)$ and $y(x+\pi)$ coincide. Indeed, the graphic below shows that the line which connects the two values $y(x)$ and $y(x+\pi)$ is parallel to the x axis.

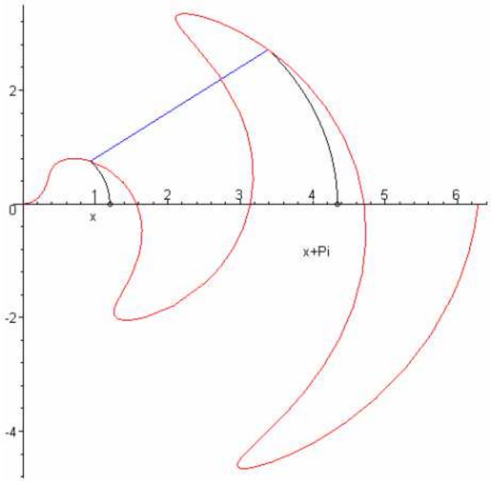
```
> period(cartesian);
```



Remark. As for the usage of period procedure the reader is referred to Appendix of this paper.

Now switch to another graphical representation and draw the function in polar coordinate system.

```
> period(polar);
```



The curve has been significantly changed. How can we formulate the periodicity of the function of this representation? First of all convince ourselves of that vertical lines in Cartesian coordinate system are transformed to circles whose center coincides with the center of polar coordinate system, while the horizontal lines are transformed to lines which contains the center of polar coordinate system. In this way the periodicity of our function assumes the following awful form. “For every nonnegative number x if we draw a circle with radius x around the center of coordinate system, and another circle with radius $x + \pi$ then connecting the points of intersections of two circles and the function curve we obtain a line which contains the center of coordinate system.”

It is obvious that there are serious problems with polar coordinate representation from didactic point of view. For first it is not ease to handle the transfer between the two graphical representation, for second the property in question is formulated clearer and in this way more easily understandable in the original representation.

Summarizing we can state that if the problem is the periodicity of sinus function then the choice of polar coordinate system for graphic representation is fare from being usable. This raises the question under which condition can the representations be considered didactically usable, or didactically rational?

Rephrasebility

Indispensable condition of the usability of a representation is that the essential properties of the examined phenomenon formulated in base representation should have the form − should be rephrased into the form − whose solution is known or easier to solve than that in the original form. In this case we say that the notion, or property is rephrasable for the given representation.

Example

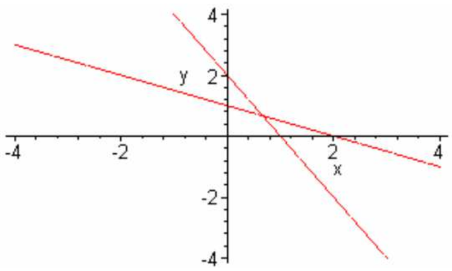
As an example let us consider the solution of linear system of equations with two unknowns. We can add two further graphical representations to the problem whose base representation is symbolic (see [9]). In the first representation we visualize the problem with two lines while in the second the coefficients of variable x and y are considered as two dimensional vectors of a plane.

```
> M := linalg[matrix](2,3,[1,2,2,2,1,2]):
  M[1,1]*x + M[1,2]*y = M[1,3];
  M[2,1]*x + M[2,2]*y = M[2,3];
```

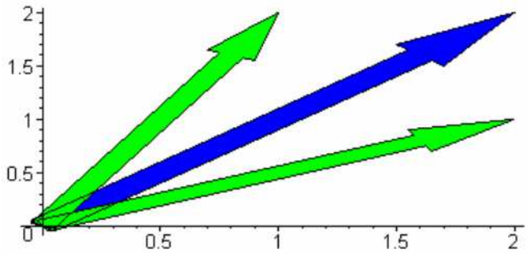
$$x + 2y = 2$$

$$2x + y = 2$$

```
> plots[implicitplot]({%,%%},x = −4..4,y = −4..4);
```



```
> u1 := Vector([M[1,1],M[2,1]]):
  u2 := Vector([M[1,2],M[2,2]]):
  u3 := Vector([M[1,3],M[2,3]]):
  plots[display](plots[arrow]({u1,u2},color = green), plots[arrow](u3,color = blue));
```



In the base representation the exercise is to find the algebraic solution of linear system of equations. The exercise is rephrasable for the second graphical representation as in this one the problem is to find the point of intersection of two lines. The solution of linear system of equations is also rephrasable for the second graphic representation, since in this representation the original problem is reduced to the determination of a linear combination of two vectors. On the other hand we recall that the periodicity of sinus function is not rephrasable for the polar coordinate graphic representation.

In the end we notice that the notion rephrasability is representation dependent. We mean that a problem can be rephrasable for one representation while can not be for another one. As in most cases the problem is specified first and we usually look for representation to this problem and not conversely. In other words, given the problem and our task is to find a representation for which this problem is rephrasable.

Transferability

The easy rephrasability is the second necessary condition of the usability of representations. We can not consider as acceptable the representation, for which the transfer is too complex, hard to execute and in this way it does not support the understanding, but it causes unnecessary difficulties.

In the previous example the transfer is easy between the representations. The equations in the symbolic representation describe themselves equations of the geometric lines. Similarly, we can easily perform the transfer between the symbolic and the second graphical representation as well.

The importance of transferability is also highlighted by the fact, that in elementary mathematics we teach nothing else than the transfer form one representation of a problem to another one. When the pupil solves math exercises by two variable linear system of equations he/she transfers the problem from verbal representation to symbolic representation. If after that he/she draws the line, as we have done before, he/she transfers from symbolic to graphic representation.

In this sense the ability to perform transfer from one representation to another one is the measure of mathematical knowledge level of students.

Remark 4: Rational representation

It is practical to introduce new name for representations which are didactic usable from the point of the problem. We call a representation *rational for the examined problem* if

1. the representation is rephrasable for the base representation of the problem and
2. the problem or the essential features of the problem is transferable for this representation.

The rational representation are those which have "didactic rationality" from the point of examined notions, phenomenon or problem.

It can be seen that the rational nature of a representation is problem dependent. The same representation can be rational for one problem while not rational for another one.

Remark 5: Cognitive efficiency

The notions of cognitive efficiency was introduced by Ziegenbalg (see [17]) for algorithms and programming languages. The cognitive efficiency, i.e. the easy understandability, the visualizations power and the seamless integration to the cognitive structure of students can be easily extend to representation as well.

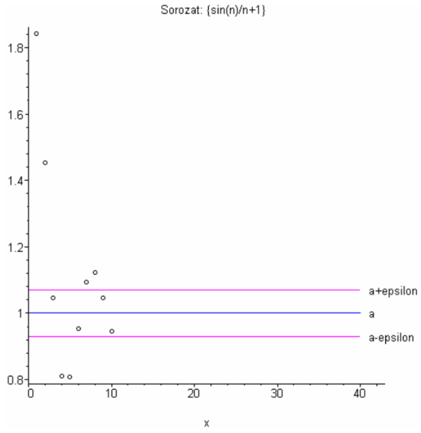
The cognitive efficiency of representations of different type, more specifically the cognitive efficiency of different models belonging to the same type of representation may differ significantly. This is because the cognitive efficiency of representation is problem dependent. It would be very bad if it was not so. Just this property of representations makes them suitable to help us in problem solving, to contribute to the evolution of human thinking.

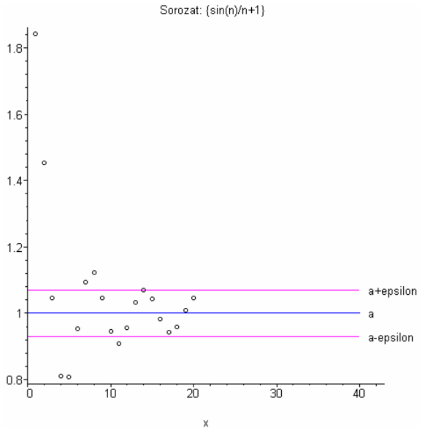
Example

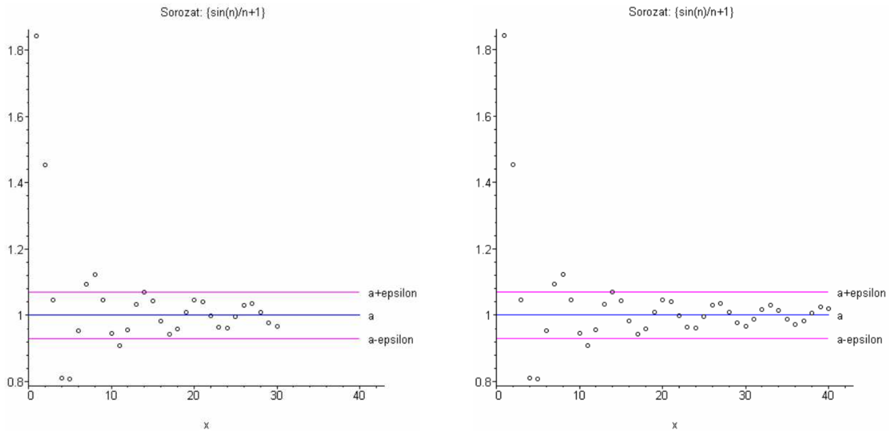
The limit of sequences of number belongs to problems, which constitute big didactic challenge to all teachers. Several authors discuss this problem (see [4]). The symbolic representation of limit is very hard to understand, indeed. We have to state that the cognitive efficiency of this representation very low. Although the verbal representation is of better cognitive efficiency the right solution is given by the animation technique of Maple.

The output of procedure convergence is an animation (see Appendix). We only show four phases of this animation here, which is hopefully appropriate to illustrate the process.

```
> convergence(sin(n)/n + 1, n, 0.07);
```







Remark. As for the usage of convergence procedure the reader is referred to Appendix of this paper.

In general, in most cases the best cognitive efficiency can be reached by graphic representation. This is not by chance, as the graphic representation acts at iconic level promoting the inductive thinking.

Remark 6: Concretized representation

As we already mentioned at the beginning of this article the application of Rule of Four is restricted first of all to different fields of calculus. According to this the numerical representation is realized by a sequence of calculated values, or by building a table of values of function. As an example we mention that Fuchs gives numeric representation for the notion of convergence by showing Heron iteration (see [5]). Sďż˝rvďż˝ri does the same in [16] using Newton iteration.

What happens if we leave the circle of real functions? For example in case of polinomial ring over a field the values of polinomials are members of the field, which may and may not be numbers. Can we speak of numerical representation here? Or let us consider the theory of automata. Although we can call all the states by numbers…

```
> printaut(randaut(3,2));
```

$$\begin{bmatrix} STA \setminus INP & 1 & 2 \\ 1 & \{3\} & \{3\} \\ 2 & - & \{3\} \\ 3 & \{1,3\} & \{2,3\} \end{bmatrix}$$

Set of initial states: , $\{3\}$

Set of final states: , $\{1,2\}$

… but the question arises if we can consider the transition table above as numeric representation of the automaton. Or could we give rational numeric representation for languages over the alphabet $\{x, y\}$?

We have to face the fact that numeric representation can not be interpreted in certain fields or even more over a certain level of abstraction. In this way if we want neither to forgo the usage of numeric representation nor to reduce its usage to calculus, we have to generalize it. In other words we have to try to supplement it by a more general principle, which gives the habitual numeric representation for the know cases, but it is generally usable.

Let’s take a look at what we do when we count the value of a function. It is nothing else than to show the examined general phenomenon by concrete examples. In this way we produce a concrete appearance of the mathematical object, we concretize the problem, assuming that this *concretized representation* is known by the students.

Notice that concretized representation produces a form stands for a lower level of abstraction, it creates special cases of general phenomenon. The concretized representation is a generalization of numeric representation, as applying it to objects specified on real numbers it results in real numbers providing us with numeric representation of examined phenomenon.

Summary

As all of the great and deep principles the Rule of Four is realized in an extremely simple form. The need of its application compels the user to rethink his/her notions concerning representations.

As we have seen in many cases the different representation types appear multiply in the course of the examination of a problem. According to this instead of speaking of four different representations, we can speak only of the usage of a certain number of representations belonging to four different types.

It seems to be more suitable to use the notation of *Rule of Multiple Representations* instead of Rule of Four. This principle can be formulated by using new notions introduced in this paper as follows. *Mathematical topics should be represented by multiple representations being rational to the subject of examinations.* These representations can belong to the *concretized, graphic, symbolic* or *verbal* representation types.

By the introduction of concretized representation we obtain a generalization of Rule of Four, while the stipulation of using rational representations only provides us with the didactic rationality of representation arising during the discussion of mathematical notion, object or phenomenon. In this way the usage of representation like these contributes to better understanding and to a higher cognitive efficiency of discussion.

Appendix

Although the goal of this paper is not to teach how to write procedures, or more specifically animation in Maple, we give a short clarification of two procedures *convergence* and *period* used in the main text. Both of these procedures are used to visualize mathematical notions, namely the first for the notion of convergence of number sequences, while the second for the demonstration of periodicity of sinus function.

We presume the reader is familiar with Maple language, with basic syntactic rules, with Maple's data structures and the usage of procedures located in different packages. We concentrate to our aim, i.e. to write a procedure which is of help to us to visualize the notion of convergence.

Procedure: convergence

Let us commence with a specifying number sequence.

```
> numseq := sin(n)/n + 1;
```

$$\mathrm{numseq} := \frac{\sin(n)}{n} + 1.$$

We intend to illustrate the elements of this sequence as points of the plane in Cartesian coordinate system. Next commands produce a sequence of two element lists. Each list can be considered as the Cartesian coordinate of elements of numseq.

```
> i := 12;
```

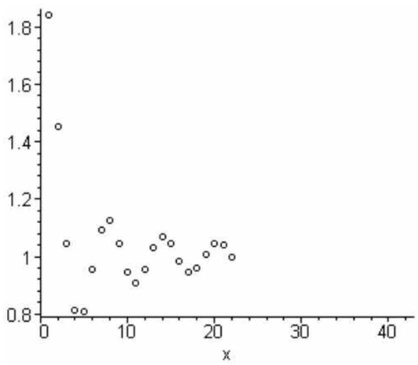
$$i := 12$$

```
> pseq := seq([k,subs(n = k,sorozat)],k = 1..i);
```

$$\begin{aligned}\text{pseq} := {} & [1, \sin(1)+1], \left[2, \frac{1}{2}\sin(2)+1\right], \left[3, \frac{1}{2}\sin(3)+1\right],\\ & \left[4, \frac{1}{4}\sin(4)+1\right], \left[5, \frac{1}{5}\sin(5)+1\right], \left[6, \frac{1}{6}\sin(6)+1\right],\\ & \left[7, \frac{1}{7}\sin(7)+1\right], \left[8, \frac{1}{8}\sin(8)+1\right], \left[9, \frac{1}{9}\sin(9)+1\right],\\ & \left[10, \frac{1}{10}\sin(10)+1\right], \left[11, \frac{1}{11}\sin(11)+1\right], \left[12, \frac{1}{12}\sin(12)+1\right],\end{aligned}$$

The variable pseq can be drawn by Maple's plot procedure.

```
> plot([pseq],x = 0..N + 3,style = point,symbol = circle,color = black);
```



In order to produce an animation which shows the points of number sequence one after another, we encapsulate our commands into a cycle and create $N(= 40)$ different plot objects $pd_1, \ldots, pd_N$. Each of these plot objects contains the illustration of certain elements of number sequence, namely dp_i illustrates the firts i points of numseq.

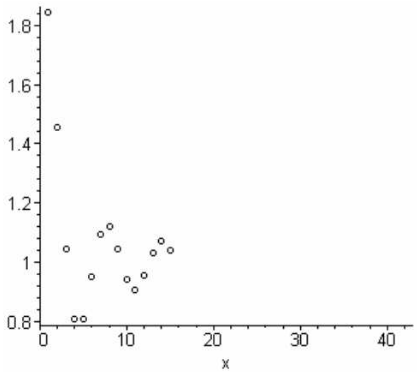
```
> N := 40;
```

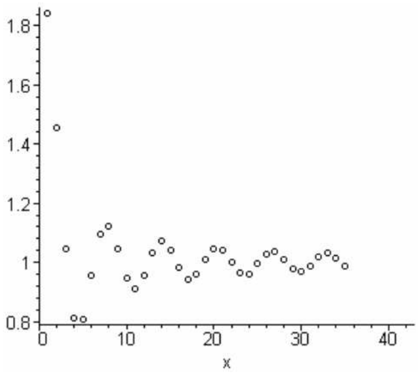
$$N : 40$$

```
> for i to N do
     pseq := seq([k,subs(n = k,sorozat)],k = 1..i);
     ppoints := plot([pseq],x = 0..N + 3,style = point,symbol = circle,color = black):
     pd[i] := plots[display](ppoints):
  od:
```

The result of commands above is 40 plot objects which are located in an array called *pd*. The array element pd_1 visualizes one point, pd_2 visualizes the first two points, and so on. We can check this by plotting for example pd_{15} and dp_{35}.

```
> pd[15]; pd[35];
```





To become more illustrative it would be nice to see three horizontal lines as well. One for the limit A of numseq, another two for values $A+\varepsilon$ and $A-\varepsilon$, respectively. For that reason we first determine the limit of the number sequence. Next we specify a value for and produce the plot objects plimit, pupper and plower, one for each required horizontal line. In the end we encapsulate the three plot objects into a fourth one, called plines.

```
> A := limit(sorozat,n = infinity);
```

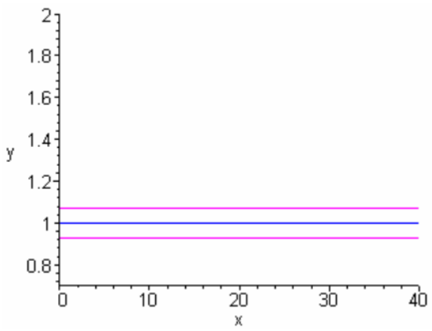
$$A := 1$$

```
> epsilon := 0.07;
```

$$\varepsilon := 0.07$$

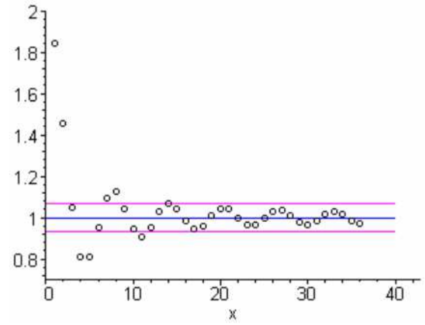
```
> plimit := plot(A,x = 0..N,y = 0.7..2,color = blue):
  pupper := plot(A + epsilon,x = 0..N,color = magenta):
  plower := plot(A−epsilon,x = 0..N,color = magenta):
  plines := plots[display]({plimit,pupper,plower}):
```

```
> plines;
```



What we have to do now is to alter the cycle above. In order to see the three horizontal lines in each plot, we place the variable plines as a parameter of display procedure in the body of the cycle. The second command below displays one of the resulted plot objects, namely pd_{36}.

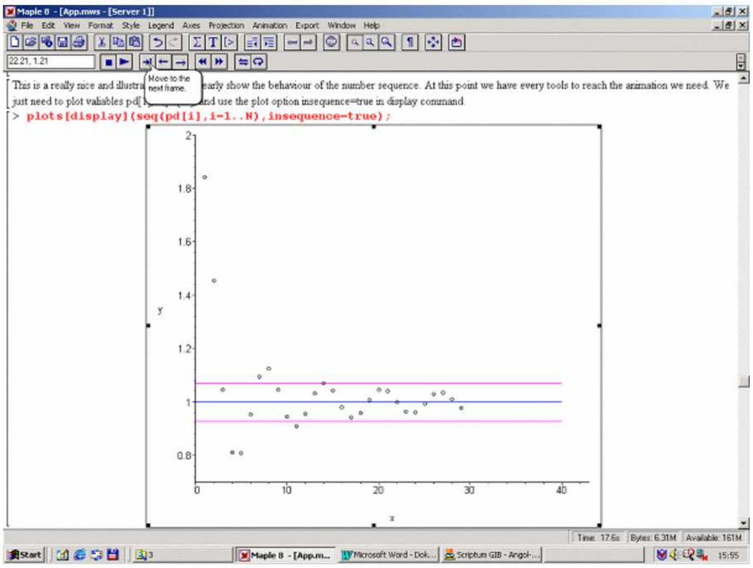
```
> for  i to N do
      pseq := seq([k,subs(n = k,sorozat)],k = 1..i);
      ppoints := plot([pseq],x = 0..N + 3,style = point,symbol = circle,color = black):
      pd[i] := plots[display]({ppoints,plines}):
  od:
> pd[36];
```



This is a really nice and illustrative plot, which clearly show the behavior of the number sequence. At this point we have every tools to reach the animation we need. We just have to display the variables $pd_1 \ldots, pd_N$ on the same coordinate system and use the plot option insequence=true in display command.

```
> plots[display]([seq(pd[i],i = 1..N)],insequence = true);
```

The result of this command is a specific animation screen, which allow us to show the plot objects pd_i ($i = 1, 2, \ldots, 40$) one after another or to play the whole animation as a movie.



Maple 8 - [App.mws - [Server 1]]
File Edit View Format Style Legend Axes Projection Animation Export Window Help
22.21, 1.21
Move to the next frame.
2
1.8
1.6
1.4
1.2
1
0.8
y
0
10
20
30
40
x
Time: 17.6s Bytes: 6.31M Available: 161M
Start
Maple 8 - [App.m...
Microsoft Word - Dok...
Scriptum GIB - Angol-...
15:55

This is a really nice and illustra early show the behaviour of the number sequence. At this point we have every tools to reach the animation we need. We just need to plot valiables pd[nd use the plot option insequence=true in display command.

```
> plots[display](seq(pd[i],i=1..N),insequence=true);
```

We end our work by encapsulate all commands in a procedure called convergence. The first parameter of convergence is the number sequence we investigate, the second is the name of variable with which we describe the sequence, while the third parameter is the value of ε. Note that procedure convergence does not check whether or not its first parameter is convergent. As a consequence of this the user must make sure to give convergent number sequence to this procedure Have fun to play with it.

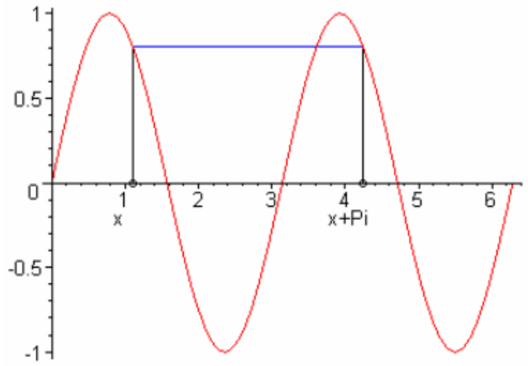
```
> convergence := proc(numseq,variable,epsilon)
  local A, N, k,i,ppoints,plines, pupper, plower, ptext, plimit, pseq, pd:
  N := 40;
  A := expand(limit(numseq,variable = infinity)):
  plimit := plot(A,x = 0..N,color = blue):
  pupper := plot(A + epsilon,x = 0..N,color = magenta):
  plower := plot(A-epsilon,x = 0..N,color = magenta):
  ptext := plots[textplot]([[N + 1,A + epsilon,'` A+epsilon` '],[N + 1,A,'`  A` '],
                           [N + 1,A-epsilon,'` A-epsilon` ']],align = RIGHT):
  plines := plots[display]({plimit,pupper,plower,ptext}):
  for i to N do
     pseq := seq([k,subs(n = k,numseq)],k = 1..i);
     ppoints := plot([pseq],x = 0..N + 3,style = point,symbol = circle,color = black):
     pd[i] := plots[display]({ppoints,plines}):
  od:
  plots[display]([seq(pd[i],i = 1..N)],insequence = true)
  end:
```

Procedure: period

The second procedure period is used to illustrate the periodic feature of sinus function. More specifically we work with the parametric form of function $\sin(2^*x)$, whose periodic feature is very well illustrated by the following commands:

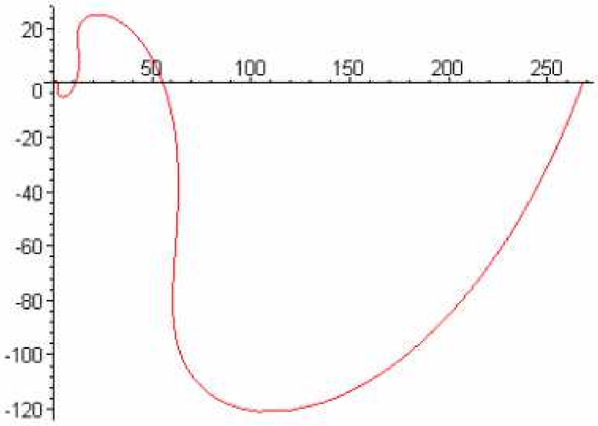
```
> s := [t,sin(2*t),t = 0..2*Pi];
```

```
> p := plot(s) : x := 1.1:
  p := p,plot([[x,0],[x + Pi,0]],style = point,symbol = circle,color = black):
  p := p,plots[textplot]([[x−0.2,−0.2,"x"],[x + Pi−0.2,−0.2,"x + Pi"]]):
  p := p,plot([[x,t,t = 0..sin(2*x)],[x + Pi,t,t = 0..sin(2*(x + Pi))]],color = black):
  p := p,plot([t,sin(2*x),t = x..x + Pi],color = blue):
  plots[display]([p]);
```

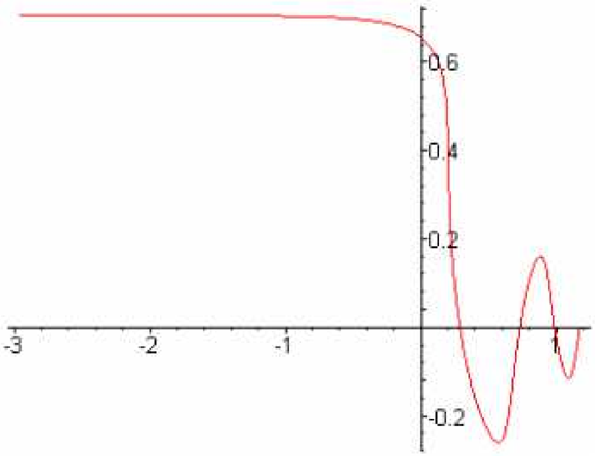


There is a rarely used but very interesting features of Maple’s plot procedure, the option coords. If we omit this option the default value is cartesian. On the other hand we can set several different values, among others elliptic, logarithmic, parabolic or polar. For further details the reader is referred to the help system of Maple. We can produce interesting curves if we change the coordinate system in plot procedure.

```
> plot(p,coords = elliptic);
```



```
> plot(p,coords = logarithmic);
```

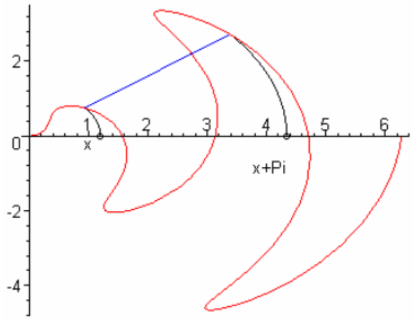


We want to investigate how our illustration of periodicity of sinus function changes when we alter the coordinate system in plot procedure. To make out task easier we encapsulate our commands in a procedure which has one parameter, the name of coordinate system. Notice, that we inserted the option coords $= k$ in the end of all plot procedure. This ensures that the procedure process its parameter correctly.

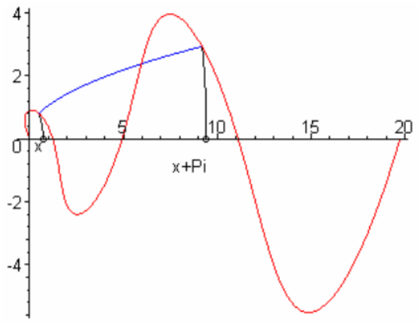
```
> period := proc(k)
  local x,p,s:
  s := [t,sin(2*t),t = 0..2*Pi];x := 1.2:
  p := plot(s,coords = k):
  p := p,plot([[x,0],[x + Pi,0]],style = point,symbol = circle,color = black,coords = k):
  p := p,plots[textplot]([[x−0.2,−0.2,"x"],[x + Pi−0.2,−0.2,"x + Pi"]],coords = k):
  p := p,plot([[x,t,t = 0..sin(2*x)],[x + Pi,t,t = 0..sin(2*(x + Pi))]],color = black,coords = k):
  p := p,plot([t,sin(2*x),t = x..x + Pi],color = blue,coords = k):
  plots[display]([p]);
  end:
```

The procedure period is very convenient tool to investigate the affects of different changes of coordinate system. We just call the procedure and give the name of desired coordinate system as its parameter as the following commands show.

```
> period(polar);
```



```
> period(parabolic);
```



References

[1] A. Ambrus, *Bevezetés a Matematikadidaktikába, (Introduction to Didactic of Mathematics)*, Eötvös Kiadó, 1995 (in Hungarian).

[2] A. Ambrus, *Nemzetközi Tendenciák a Matematika Oktatásában, (International Trends in Teaching of Mathematics)* (in Hungarian), `http://xml.inf.elte.hu/~mathdid/tendenc.pdf`.

[3] E. Connaly, D. Hughes-Hallett and A. Gleason, *Functions Modeling Change: A Preparation for Calculus*.

[4] J. Cottrill, Ed. Dubinsky, K. Schwingendorf, K. Thomas and D. Vidakovic, *Understanding the Limit Consept: Beginning with a coordinated Process Schema*.

[5] K. J. Fuchs, *Computer Algebra Systems in Mathematic Education*, International Symposium, Anniversary of Pollack Mihály College of Engineering, 2002.

[6] J. E. Hopcroft and J. D. Ullman, Introduction to Automata Theory, *Languages, and Computation*, Addison-Wesley, Reading, MA, 1979.

[7] M. Klincsik, Gy. Maróti, *“Maple 8 tételben”*, Novada, 1996.

[8] Kozen, *Automata and Computability*, Springer-Verlag, New York, 1997.

[9] W. Lindner, CAS-Supported Multiple Representations in the Elementary Linear Algebra, *The case of Gaussian Algorithm*, International Symposium, Anniversary of Pollack Mihály College of Engineering, 2002.

[10] W. Lindner, Konstructivischer CAS-intensive Mathematikunterricht laengs der APOS-Theorie – die Falltudie Faerben von Graphen, (Constructive CAS-intensive Teaching of Mathematics based on APOS Theory – Case Study of Coloring of Graphs), *Beitraege zum Mathematikunterricht*, 2003 (in German).

[11] D. Mackie, *Using Computer Algebra to Encourage a Deep Learning Approach to Calculus*, International Conference on Teaching of Mathematics (ICTM2), July, 2002.

[12] Gy. Maróti, Didactic Approach for Teaching Nondeterminism in Automata Theory, *ZDM* **35**(April), Number 2 (2003).

[13] Gy. Maróti, CAS based approach for discussing subset construction, *ZDM* **35**(April), Number 2 (2003).

[14] M. B. Monagan, K. O. Geddes, K. M. Heal, G. Labahm, S. M. Vorkoetter, J. McCaron and P. DeMarco, *Maple 8 Advanced Programming Guide*, Waterloo Maple, 2002.

[15] National Council of Teachers of Mathemetics (NCTM), Principles and Standards for School Mathematics, 2000.

[16] Cs. Sárvári, Network Based Math Teaching Using CAS, *ZDM* **35** (April), Number 2 (2003).

[17] J. Ziegenbalg, *Algorithmen Von Hammapuri bis Gödel, (Algorithms from Hammapuri till Gödel)*, Spektrum Akademischer Verlag, 1996 (in German).

GYÖRGY MARÓTI
UNIVERSITY OF PÉCS
PÉCS
HUNGARY

(Received July, 2003)