

7/1 (2009), 51–68

tmcs@math.klte.hu
<http://tmcs.math.klte.hu>

Teaching
Mathematics and
Computer Science

Teaching Fourier series, partial differential equations and their applications with help of computer algebra system

ILDIKÓ PERJÉSI-HÁMORI and CSABA SÁRVÁRI

Abstract. In this paper, some examples of Fourier series and partial difference equations will be shown to demonstrate opportunities for CAS use in various circumstances. The well-known white-box - black-box teaching-learning techniques and the modularization will be used to allow the use of the same worksheet in different ways.

Key words and phrases: computer algebra in teaching-learning process, didactical questions of CAS, Fourier-series, partial differential equation.

ZDM Subject Classification: B45, C75, U55.

1. Introduction

The Pécs University, Pollack Mihály Faculty of Engineering has launched the two-level (BSc and MSc) educational system, in conjunction with the Bologna process, in 2006. As a result of this process fewer mathematics contact lessons are in the basic level, although the required mathematics themes are not reduced. In order to reach the parallel goals of the undergraduate mathematics it is necessary to make contact

- with the practice (first of all, whom, who will go to the labor market after graduating at Bachelor level),
- with the Master level mathematics and engineering subjects.

The MAPLE computer algebra system is used in the teaching-learning process, in the basic mathematics courses of informatics, mechanical-, electrical- and civil engineering since 1996. The introduction of the usage of computer algebra is gradually only for visualization, on the lectures (in the first semester), for time-consuming numerical calculations after chalk-and talk practicing, at the end of the semester (in the second semester), for symbolical calculations and for complicated applications – all lessons in computer laboratories (in the third semester).

In this paper some example is shown in themes of Fourier series and partial difference equations to demonstrate the possibilities of CAS in several circumstances.

The mathematics curriculum of the engineering and informatics obliges the discussion of the concept of the Fourier-series and partial differential equations at the end of students' third semester. Sometimes (particularly lack of time) they are mentioned only on lecture, there is no trainings for the practice. There are courses, where the practical lessons are in a computer laboratory in the whole semester. In this case the students get deep experiences in using CAS. There are students (especially students of informatics) who get acquainted with the programming techniques of CAS, as well. In this paper the usage of CAS in the above mentioned cases is demonstrated with help of examples.

2. Curriculum based modularization

The starting point is the exploration of connections between CAS and curriculum. Recognizing operative knowledge as a critical component of “doing” mathematics, thoughtful modularization of course material is a crucial first step in the successful implementation of CAS in classroom instruction. We use the concept of module with the following meaning: The module is more or less complex and connected system of generalized knowledge, which can be called as unified whole procedure, without to be executed. The CAS module is an immediately callable cognitive unit. (Dörfler, [3]) The functions of the CAS module are

- (i) Reduction of burden of the thinking
- (ii) Decrease of the complexity.

First of all, we have to take into consideration that mathematics content and students' learning of the content, by its very nature, has a modular structure. If teachers wish to make use of CAS-based functions, programs, or scripts (i.e. “CAS modules”) in classroom instruction, these CAS functions, programs, and scripts

must be incorporated into instruction in a way that is compatible with students' current mathematical understandings. The modularization must be synchronized with the content and learners of the course. This is true whether the module is “built in” with the CAS or constructed by teachers or students.

The effectiveness of mathematical knowledge, in a narrow sense the comprehension itself, can only be approached from an organizational perspective of knowledge. 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. (Hiebert and Carpenter, 1992 [4])

This view of mathematics comprehension is relevant with regard to the use of CAS modules in at least two ways. Firstly, the use of a CAS module allows students the time and cognitive capacity for building more connections among new knowledge elements. Secondly, once the types of CAS modules to be used are determined, the primary issue to be addressed is the role of the new knowledge element in the curriculum. If detailed understanding of the elaborated procedure is necessary afterwards – for instance, to work meaningfully with other elements – then teachers should carry out a white box / black box modularization. If knowing the structure and the type of elaboration of mental picture of the processed algorithm is not necessary to later thickening of the knowing representation net, then we can apply the outsourcing-principle. (Peschek and Schneider[5]). This means that in these cases is allowed and useful the application of CAS module without knowing its inner structure.

In our teaching practice, modularization is a dynamic process. (Sárvári, [6]) Firstly throughout our construction of complex CAS-procedures we apply the following model:

- Firstly, we construct a series of procedural steps from algorithm primitives (i.e. *pseudocode phase*).
- Then we construct the parts of the complete algorithm as stand-alone procedures (i.e. *semi-automatic phase*).
- At the end, we paste the working parts together using the mathematical concept of composition (i.e. *functional programming phase*).

Secondly we use the white box / black box modularization as it mentioned by B. Buchberger (2003) [2]. The white-box black-box principle is recursive and it is used in frame of creativity or invention spiral. Negotiation of all concepts of all algorithms begins in white box phase. In the next step we create a new concept, and can use the previous algorithms-modules- as black box and so on.

3. Teaching Fourier series

Problem: Let's see the group of the periodical functions, which have significant role in engineering practice.

Assume that $y = f(x)$ is given in the range $-\pi$ to π and

$$f(x) = f(x + 2k\pi),$$

where k is an arbitrary integer number, thus f is periodic with period 2π . Assume, that f can be represented by an infinite series, i.e.

$$f(x) = a_0 + a_1 \cos(x) + b_1 \sin(x) + a_2 \cos(2x) + b_2 \sin(2x) + \dots + a_n \cos(nx) + b_n \sin(nx) + \dots = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx)).$$

Our goal is to determine $a_0, a_1, b_1, a_2, b_2, \dots$ coefficients. We would like to know, how and in what kind of conditions can be represented by infinite series of sinus and cosines?

Pseudo code phase

Firstly will be detailed theoretical basis and given series of procedural steps (i.e. pseudo code). In this case the procedural steps are the computing of the coefficients.

White-box: if we have enough time, students have basic knowledge of integration, and series of function.

Let's suppose that the series is convergent, and its sum is $f(x)$. For the determination of $a_0, a_1, b_1, a_2, b_2, \dots$, coefficients we have to prove some simply lemmas.

White-box: (More time) prove the lemmas by hand and after it use Maple only for checking.

Black-box: (Less time required) solving them with *int* command.

Multiply each side of the equation by $\cos(nx)$

$$f(x) \cos(nx) = a_0 \cos(nx) + a_1 \cos(x) \cos(nx) + b_1 \sin(x) \cos(nx) + \dots + a_m \cos(nx) \cos(nx) + \dots$$

Because of the Lemmas, all integrals, but $a_n \cos(nx) \cos(nx)$ are equal to 0, and this one is equal to π .

So

$$a_n = \frac{\int_{-\pi}^{\pi} f(x) \cos(nx) dx}{\pi}$$

and

$$b_n = \frac{\int_{-\pi}^{\pi} f(x) \sin(nx) dx}{\pi}$$

Semi-automatic phase

We construct the parts of the complete algorithm a stand-alone procedures or use the built-in commands (functions, procedures) of Maple

```
> a0:=1/(2*Pi)*int(f(x),x=-Pi..Pi);
> a:=k->1/Pi*int(f(x)*cos(k*x),x=-Pi..Pi);
> b:=k->1/Pi*int(f(x)*sin(k*x),x=-Pi..Pi);
> F_partial_sum:=a0+Sum(a(k)*cos(k*x)+b(k)*sin(k*x),k=1..n);
> F_series:=a0+Sum(a(k)*cos(k*x)+b(k)*sin(k*x),k=1..infinity);
```

In this phase the importance is on the applications. Let's demonstrate on the example of

$$f := x \rightarrow e^x$$

function. The Fourier-coefficients:

$$a_0 = \frac{1}{2} \frac{(-1 + e^{2\pi})e^{-\pi}}{\pi}$$

$$a(k) = \frac{(-\cos(\pi k) + k \sin(\pi k) + e^{2\pi} \cos(\pi k) + e^{2\pi} k \sin(\pi k))e^{-\pi}}{\pi(1 + k^2)}$$

$$b(k) = -\frac{(-k \cos(\pi k) - \sin(\pi k) + e^{2\pi} k \cos(\pi k) - e^{2\pi} \sin(\pi k))e^{-\pi}}{\pi(1 + k^2)}.$$

The fourth partial sum is:

$$\frac{1}{2} \frac{(-1 + e^{2\pi})e^{-\pi}}{\pi} + \sum_{k \sim 1}^4 \left(\frac{(-1)^{k \sim} (-1 + e^{2\pi})e^{-\pi} \cos(k \sim x)}{\pi(1 + k \sim^2)} - \frac{k \sim (-1)^{k \sim} (-1 + e^{2\pi})e^{-\pi} \sin(k \sim x)}{\pi(1 + k \sim^2)} \right).$$

Using command *value* we can evaluate the expression:

$$\frac{1}{2} \frac{(-1 + e^{2\pi})e^{-\pi}}{\pi} - \frac{1}{2} \frac{(-1 + e^{2\pi})e^{-\pi} \cos(x)}{\pi} + \frac{1}{2} \frac{(-1 + e^{2\pi})e^{-\pi} \sin(x)}{\pi}$$

$$\begin{aligned}
 & + \frac{1}{5} \frac{(-1 + e^{2\pi})e^{(-\pi)} \cos(2x)}{\pi} - \frac{2}{5} \frac{(-1 + e^{2\pi})e^{(-\pi)} \sin(2x)}{\pi} \\
 & - \frac{1}{10} \frac{(-1 + e^{2\pi})e^{(-\pi)} \cos(3x)}{\pi} + \frac{3}{10} \frac{(-1 + e^{2\pi})e^{(-\pi)} \sin(3x)}{\pi} \\
 & + \frac{1}{17} \frac{(-1 + e^{2\pi})e^{(-\pi)} \cos(4x)}{\pi} - \frac{4}{17} \frac{(-1 + e^{2\pi})e^{(-\pi)} \sin(4x)}{\pi}.
 \end{aligned}$$

To get to know more about the size of the coefficients, we use the *evalf* command.

$$\begin{aligned}
 & 3.676077910 - 3.676077910 \cos(x) + 3.676077910 \sin(x) + 1.470431164 \cos(2.x) \\
 & - 2.940862328 \sin(2.x) - 0.7352155821 \cos(3.x) + 2.205646746 \sin(3.x) \\
 & + 0.4324797542 \cos(4.x) - 1.729919016 \sin(4.x).
 \end{aligned}$$

Plotting together the function and some of the partial sum of the series are shown on Figure 1.

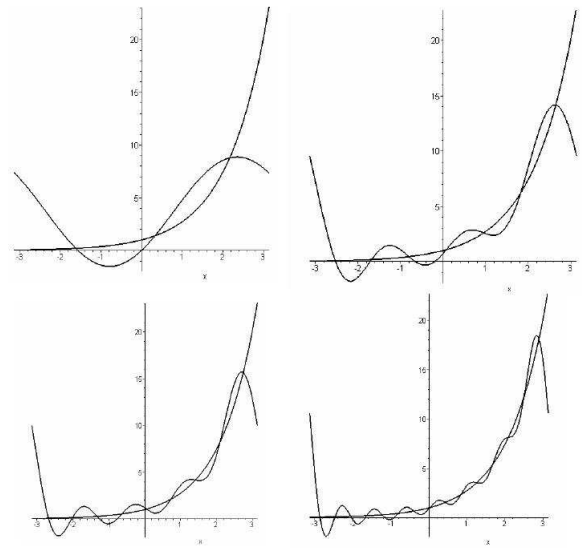


Figure 1. $f(x) = e^x$ and partial sums of it's Fourier series

Automatic phase (functional programming phase)

White-box: with those who want to know more about how to construct Maple programs we discuss the different possibilities of writing procedures for computing of Fourier series. The students guided by teacher construct procedure to animate the function and the partial sums. (informatics students)

Black-box: – more time required – for those who has the basic knowledge of the development of a series, but he/she wants to use it in the engineering practice.

– less time required – for those who does not want to know why but only how to solve the problem.

In this phase we paste the working parts together using the mathematical concept of composition, and construct a procedure, namely Fourier procedure:

```
> Fourier:=proc(f)
  local n;
  global a0,a,b;
  a0:=1/(2*Pi)*int(f(x),x=-Pi..Pi);
  b:=k->1/Pi*int(f(x)*sin(k*x),x=-Pi..Pi);
  unapply(eval(a0+Sum(a(k)*cos(k*x)+b(k)*sin(k*x),k=1..n)),x,n);
end;
```

Now apply it for an example. Find the Fourier series of

$$f(x) = \begin{cases} \frac{(x + \pi)^2}{\pi} & -\pi \leq x \text{ and } x \leq 0 \\ \frac{(x - \pi)^2}{\pi} & 0 < x \text{ and } x \leq \pi \end{cases}$$

$f(x + 2k\pi) = f(x)$, where k is an arbitrary integer number.

```
> f:=x->piecewise(-Pi<=x and x<=0,1/Pi*(x+Pi)^2,0<x and
x<=Pi,1/Pi*(x-Pi)^2);
```

Call the procedure *Fourier* for the function f .

```
F:=Fourier(f):
```

$$F := (x, n) \rightarrow \frac{\pi}{3} + \left(\sum_{k=1}^n \left(-\frac{4(\sin(\pi k) - \pi k) \cos(kx)}{\pi^2 k^3} \right) \right)$$

The 10. partial sum is:

```
> value(F(x,10));
```

$$\frac{\pi}{3} + \frac{4 \cos(x)}{\pi} + \frac{\cos(2x)}{\pi} + \frac{4 \cos(3x)}{9\pi} + \frac{1 \cos(4x)}{4\pi} + \frac{4 \cos(5x)}{25\pi} + \frac{1 \cos(6x)}{9\pi} + \frac{4 \cos(7x)}{49\pi} + \frac{1 \cos(8x)}{16\pi} + \frac{4 \cos(9x)}{81\pi} + \frac{1 \cos(10x)}{25\pi}$$

> 'f(x) '=F(x,infinity);

$$f(x) = \frac{\pi}{3} + \left(\sum_{k=1}^{\infty} \left(\frac{4 \cos(kx)}{\pi k^2} \right) \right).$$

A procedure can be written to animate the function and the partial sums:

```
> picture:=proc(k)
local n,N,kep:
N:=k:
for n from 1 to N do
kep||n:=plot([F(x,n),f(x)],x=-
Pi..Pi,color=[red,blue],thickness=[2,2],discont=true):
od:
plots[display]([kep||(1..N)],insequence=true);
end:
```

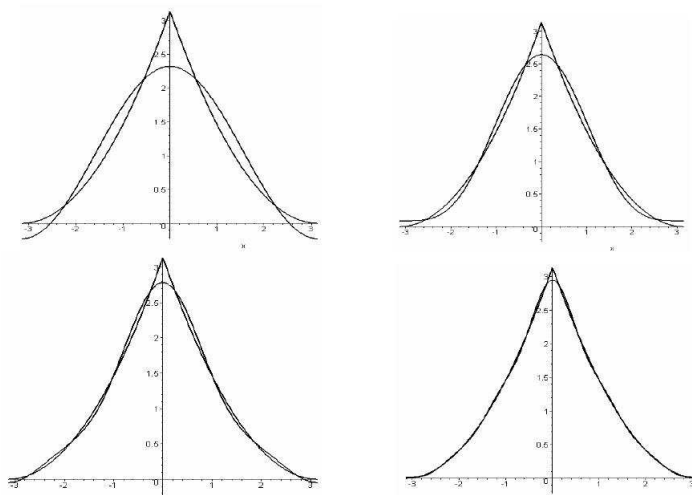


Figure 2. Fourier series of a piecewise function

4. Teaching partial differential equations

4.1. Vibration of a cord (Hyperbolic type partial differential equation)

The problem is: Let's compute the vertical displacement $u(x, t)$ of a cord with length L and fixed endpoints, where t is time and x is distance along the cord (Figure 3). For each instance in this the problem, we must specify the initial displacement of the cord $f(x)$, the initial speed of the cord $g(x)$ and the horizontal wave speed c .

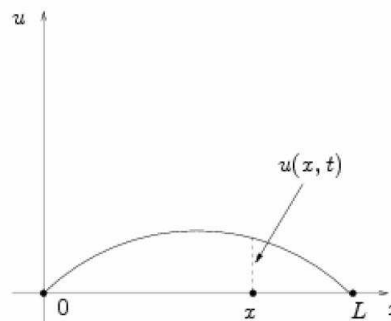


Figure 3. Vibration of a cord

After physical considerations the one-dimensional wave equation in Cartesian coordinates is:

$$\frac{\partial^2}{\partial t^2} u(x, t) = \alpha^2 \left(\frac{\partial^2}{\partial x^2} u(x, t) \right)$$

subject to the boundary conditions:

$$u(0, t) = 0,$$

$$u(L, t) = 0, \quad (\text{fixed endpoints})$$

and initial conditions:

$$u(x, 0) = f(x), \quad (\text{initial displacement of the cord})$$

$$\frac{\partial}{\partial t} u(x, 0) = g(x) \quad (\text{initial speed of the cord}).$$

Here the physical explanation is a *black-box* type consideration in lesson mathematics, and a *white-box* type in lesson mechanics.

4.1.1. The detailed explanation of the solution

The explanation of the solution of the differential equation is *white-box* type if there is enough time for students to have the basic knowledge of the solution of the ordinary differential equations and Fourier series.

The solutions are search of the form $u(x, t) = X(x)T(t)$ by substituting $X(x)T(t)$ into PDE. So the second order PDE is changed to two ODE.

$$\frac{\partial^2}{\partial x^2}u(x, t) = \left(\frac{d^2}{dx^2}X(x)\right)T(t)$$

$$\frac{\partial^2}{\partial t^2}u(x, t) = \left(\frac{d^2}{dt^2}T(t)\right)X(x).$$

After the substitution:

$$\frac{\frac{d^2}{dt^2}T(t)}{T(t)} = \frac{\alpha^2 \left(\frac{d^2}{dx^2}X(x)\right)}{X(x)}$$

the variables. x and t is independent of each other, each side must be fixed constant, β^2 The task is to solve two, second ordered ordinary differential equations.

- *White-box*: (More time required) solve these equations by hand and after it they use Maple for only checking their solution.
- *Black-box*: (Less time required) solving them with Maple's *dsolve* command.

$$de1 := \left(\frac{d^2}{dt^2}T(t)\right) + \beta^2 T(t) = 0$$

$$de2 := \left(\frac{d^2}{dx^2}X(x)\right) + \frac{\beta^2 X(x)}{\alpha^2} = 0$$

$$\text{Sol 1} := T(t) = A \sin(\beta t) + B \cos(\beta t)$$

$$\text{Sol 2} := X(x) = C \sin\left(\frac{\beta x}{\alpha}\right) + E \cos\left(\frac{\beta x}{\alpha}\right).$$

The “step by step” solution of the partial differential equation with help of Maple

$$\text{Sol} := u(x, t) = (A \sin(\beta t) + B \cos(\beta t)) \left(C \sin\left(\frac{\beta x}{\alpha}\right) + E \cos\left(\frac{\beta x}{\alpha}\right)\right).$$

Here A, B, C, E, β are arbitrary constants. Their values depend on the initial and boundary conditions.

The first boundary condition $u(0, t) = 0$, so

$$eq1 := 0 = 1. (A \sin(\beta t) + B \cos(\beta t))E.$$

Its mean, that for every t :

$$E := 0.$$

From the second boundary condition $u(L, t) = 0$, so

$$eq2 := 0 = (A \sin(\beta t) + B \cos(\beta t)) C \sin\left(\frac{\beta L}{\alpha}\right)$$

C is not 0, so $\sin\left(\frac{\beta L}{\alpha}\right)$ has to be 0. It gives a sequence of solutions for β . Assuming k is an integer:

$$eq3 := \frac{\beta_k L}{\alpha} = k\pi$$

$$\beta_k := \frac{k\pi\alpha}{L}$$

$$\text{Sol}_k := u_k(x, t) = \left(A_k \sin\left(\frac{k\pi\alpha t}{L}\right) + B_k \cos\left(\frac{k\pi\alpha t}{L}\right) \right) C_k \sin\left(\frac{k\pi x}{L}\right).$$

Expand the right hand side:

$$\text{Sol}_k := u_k(x, t) = C_k \sin\left(\frac{k\pi x}{L}\right) A_k \sin\left(\frac{k\pi\alpha t}{L}\right) + C_k \sin\left(\frac{k\pi x}{L}\right) B_k \cos\left(\frac{k\pi\alpha t}{L}\right).$$

The linear combination of the solutions is solutions as well. This will give the collection of vibrations of the string and the goal is to sum them.

$$\text{Solution} := u(x, t) = \sum_{k=1}^{\infty} u_k(x, t).$$

Substitute $a_k := C_k A_k$, $b_k := C_k B_k$:

$$\text{Sol} := u(x, t) = \sum_{k=1}^{\infty} \left(a_k \sin\left(\frac{k\pi x}{L}\right) \sin\left(\frac{k\pi\alpha t}{L}\right) + b_k \sin\left(\frac{k\pi x}{L}\right) \cos\left(\frac{k\pi\alpha t}{L}\right) \right).$$

Now the goal is to determine a_k , and b_k .

From the initial conditions $u(x, 0) = f(x)$, and $\frac{\partial}{\partial t} u(x, 0) = g(x)$,

$$f(x) = \sum_{k=1}^{\infty} b_k \sin\left(\frac{k\pi x}{L}\right)$$

$$g(x) = \sum_{k=1}^{\infty} \frac{a_k \sin\left(\frac{k\pi x}{L}\right) k\pi\alpha}{L}.$$

In these equations, b_k are the coefficients of the poor sinus Fourier series of f , if the domain of f is enlarged to be periodical to $2L$, uneven function. This is the so called *Fourier* procedure where the coefficients a_k can be calculated. After performing this procedure all constants are known in the general solution from the boundary and initial conditions, or in other word, the particular solution is determined.

4.1.2. With help of built in procedure

It can be used in two ways:

- *black-box* mode if there is no time or students have only little knowledge of the solution of the differential equation and Fourier series
- *white-box* mode after the detailed explanation for experiments or solving more complicated problems.

With help of Maple *VectorCalculus* package the wave equation is written down using the *Laplacian* function with $\alpha = 1$, $L = 1$.

```
> Cord_Eqn := {diff( u(x,t), t, t) = Laplacian( u(x,t),
'cartesian'[x]) };
```

$$Cord_Eqn := \left\{ \frac{\partial^2}{\partial t^2} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) \right\}$$

The specified boundary conditions are that the cord is fixed at both ends and has 0 initial speeds.

$$BC1 := \{D_2(u)(x, 0) = 0, u(0, t) = 0, u(\pi, t) = 0\}.$$

Two cases are shown below to demonstrate the interactive possibilities of Maple:
Case 1: Cord is plucked from the center

$$BC2 := \left\{ u(x, 0) = \begin{cases} x & -x \leq 0 \text{ and } x - \frac{\pi}{2} \leq 0 \\ -x + \pi & -x + \frac{\pi}{2} < 0 \text{ and } x - \pi \leq 0 \end{cases} \right\}.$$

The PDE boundary-value problem is solved numerically with the *pdsolve* command and *numeric* option specified. The accuracy of the solution is set by specifying the time step and space step of the discretisation over the distance-time rectangle. Maple delivered the solution as a Maple module. The module provides functions for viewing the solution as plots (*plot* and *plot3d*), animations over time (*animate*) and numerical values (*value*).

```
> sol:=pdsolve( Cord_Eqn, BC1 union BC2, numeric,
spacestep=1/200, timestep=1/100);
sol:= module () export plot,plot3d,animate,value,settings; ... end module
> sol:-animate( u(x,t) ,t=0..2*Pi, frames=30,
labels=["x","u(x,t)"], labelfont=[TIMES,ROMAN,14],
scaling=constrained);
```

(Figure 4.)

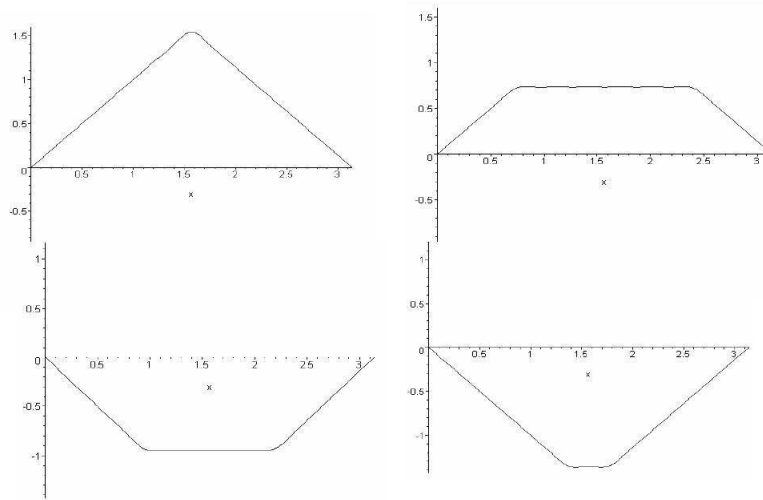


Figure 4. Elements of the animation of a cord with $BC1$ boundary condition

```
> CordTemperature := sol :- value();
CordTemperature := proc () ... end proc.
```

The cord temperature value at a discreet point, in a given time:

```
> CordTemperature( .5, 3 );
[x = 0.5, t = 3., u(x, t) = -0.499767804920668746].
```

Case 2: Cord starts in the shape of different trigonometric function (Figure 5.)

```
> sol:-animate( [[u(x,t) , thickness=1,color=black],
[-u(x,t)/2-sin(4*x),thickness=2 ,color=black ] ,
[u(x,t)/2+sin(6*x) ,thickness=3,color=black]], t=0..2*Pi,
frames=8,labels=["x","u(x,t)"], labelfont=[TIMES,ROMAN,14],
scaling=constrained);
```

This built in procedure is very useful for these experiments which have different boundary conditions and only if we want visualize the solution to get the results only in some discrete points.

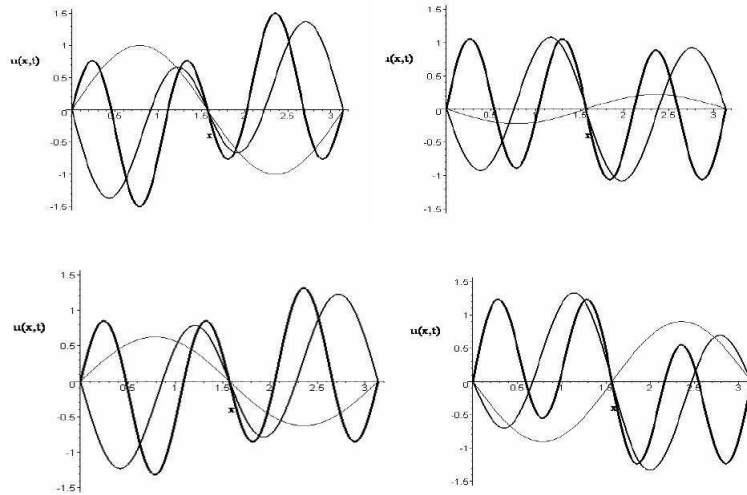


Figure 5. Elements of the animation of a cord with different trigonometric boundary conditions

4.2. Heat flow in a rod (time dependence, one dimension) (Parabolic type partial differential equation)

The following problem is to compute the temperature $u(x, t)$ over a rod, where t is time and x is distance along the rod. (Figure 6.) The assumption is that the ends of the rod maintain a constant temperature. For each part of the problem, we must specify the initial heat distribution $f(x)$ and the thermal diffusivity α of the rod.

The time-dependent, one-dimensional heat equation in Cartesian coordinates:

$$\frac{\partial}{\partial t} u = \alpha^2 \left(\frac{\partial^2}{\partial x^2} u \right)$$

subject to the boundary conditions:

$$u(0, t) = 0,$$

$$u(1, t) = 0,$$

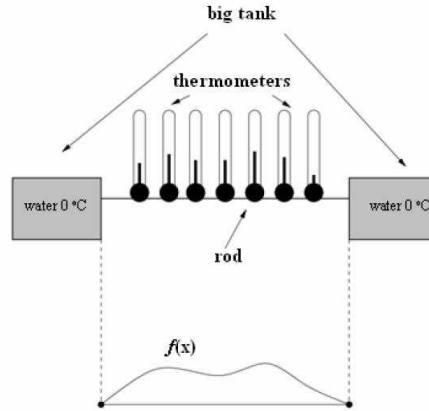


Figure 6. Heat flow in a rod

the initial condition

$$u(x, 0) = f(x).$$

The detailed explanation of the solution is used as a *white-box*:

- if there is sufficient time available,
- if the students have basic knowledge of the solution of the ordinary differential equation, and Fourier series
- if the wave equation was not solved earlier (for example we solve the wave equation with one group and the heat equation with another group).

The built in procedure is used:

- *white-box*: for experimentation
- *black-box*:
 - if no time available,
 - if the students have only few knowledge of the solution of the differential equation, and Fourier series.

We write down the heat equation using the *Laplacian* function. The assumption is, that $\alpha^2 = 1$

```
>Heat_Eqn := {diff( u(x,t), t) = Laplacian( u(x,t),
'cartesian' [x]) };
```

$$Heat_Eqn := \left\{ \frac{\partial}{\partial t} u(x, t) = \frac{\partial^2}{\partial x^2} u(x, t) \right\}$$

The boundary conditions are the ends of the rod are kept at a constant temperature and the initial temperature distribution is $f(x)$.

$$BCs := \left\{ u(0, t) = 0, u(1, t) = 0, u(x, 0) = (-x + 1)(-e^{-10x} + 1) \right\}$$

```
>Heat_Solution := pdsolve( Heat_Eqn, BCs, numeric,
timestep=1/100, spacestep=1/100):
>Heat_Solution :- plot3d(u(x,t), t=0..0.18, shading=zhue,
axes=boxed, labels=["x", "t", "u(x,t)"],
labelfont=[TIMES,ROMAN,16]);
```

(Figure 7.)

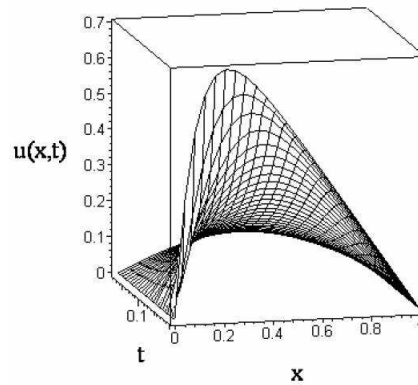


Figure 7. The time-dependent temperature distribution in a rod

To get numerical values the *value* procedure from the solution’s module is used.

```
>RodTemperature := Heat_Solution :- value();
```

The temperature at $x = .5, t = 3$

$$[x = 0.5, t = 3., u(x, t) = 0.798817965072981856 \cdot 10^{-13}].$$

5. Conclusions

Probably the most basic, most complex, and from didactic point of view the most critical question about the usage of CAS concerns the usage of the modules. On the one hand using modules CAS executes the majority of the operative

activities. On the other hand it renders different delineation and representation methods in quick and easy access. Due these two functions experiment-based explorative learning gains an effective tool using white-box – black-box technique in teaching. Using the white box/black box technique we built up the modules, and dependence of the circumstances the modules could be “unpacked”, or not. But the appropriate mode of using modules always must be chosen very carefully. The applied method depends on

- time (practical aspect: how many holidays are included in the semester)
- needs of professions in details:
 - for civil engineering, the concept and the solution,
 - for informatics students the construction of procedures
 - for mathematicians the step by step solution
 - for architects only the built in procedures-interest of students (it might change to students group of the same year).

In our experience using Maple makes it possible to teach Fourier series and partial differential equations deeper and more detailed. Using of white boxes in the whole teaching process results

- the knowledge developed by the students is more flexible and can be connected easier to practical application of other chapters;
- some of the students are able to modify the procedures, to create own procedures for investigating of special cases;
- they gain considerable new skills in flexible using and writing procedures.

Students are working without white boxes can successful use procedures in their work, they spare time for analyzing of occurrences, but of course transportability of their knowing is not to compare with ability of previous group.

Our experiences suggest that moderate use of CAS and the well-considered modularization at university level benefits students in our courses.

References

- [1] B. Buchberger, Should Students Learn Integration Rules?, *ACM SIGSAM Bulletin* **24**, no. 1 (1990), 10–17.
- [2] B. Buchberger, White-Box/Black-Box Principle etc., *Symposium Mathematics and New Technologies: What to Learn, How to Teach?*, Kepler University, Austria, 2003, www.risc.uni-linz.ac.at/publications/download/risc_2191/2003-12-10-A.ppt.

- [3] W. Dörfler, *Der Computer als kognitives Werkzeug und kognitives Medium*, In: Dörfler, W./Schneider, E./Wegenkittl, K. (Hrsg), *Computer-Mensch-Mathematik*. Wien: Hölder-Pichler-Tempsky, 1991, 51–75.
- [4] J. Hiebert and T. P. Carpenter, *Learning and teaching with understanding*, Handbook of research on mathematics teaching and learning, (In: D. A. Grouws, ed.), New York: Macmillan, 1992, 65–97.
- [5] W. Peschek and E. Schneider, CAS in general mathematics education, *Zentralblatt für Didaktik der Mathematik* **34/5** (2002), 189–195.
- [6] Cs. Sárvári, Experiences in the Integration of CAS into Engineering Education at the University of Pécs, *International Journal for Technology in Mathematics Education* **14/3** (2007), 130–136.

ILDIKÓ PERJESI-HÁMORI and CSABA SÁRVÁRI
INSTITUTE OF MATHEMATICS
POLLACK MIHÁLY ENGINEERING FACULTY
UNIVERSITY OF PÉCS
H-7628 PÉCS, BOSZORKÁNY U. 2.
HUNGARY

E-mail: perjesi@witch.pmmf.hu

E-mail: sarvari@witch.pmmf.hu

(Received January, 2008)