

# Computer Technologies and Live Lectures of Mathematics

*Miroslaw Majewski*

e-mail: majewski@mupad.com

School of Arts and Sciences

New York Institute of Technology

UAE

## Abstract

*This paper contains an overview of personal experiences of the author in teaching mathematics with computer technology. Different aspects of teaching mathematics with computer technology are analyzed. The usability of selected computer tools for mathematics classes is discussed.<sup>1</sup>*

## 1 Introduction

We do have a number of powerful computer programs that can perform mathematical operations, both numerical and symbolic with lightning speed, and help us with tedious computing tasks. Such software has proved its usefulness in many areas including education—teaching mathematics, statistics, engineering and a few other, high school as well as university, courses. There is no doubt that we should not ignore existence of such software and find the best possible way of using it. In this article, I will summarize my experiences in teaching mathematics and quantitative mathematics courses with a few mathematical packages. Finally, I will show what we may expect from a computer program when we use it in our classes and what problems we may face.

A number of authors have presented a very enthusiastic, sometimes very naive, opinions about the role of computer technology in mathematics classroom. Most of these opinions concentrate on advantages of technology without going deeper into details or showing how their authors use technology in the classroom. In this paper I will describe what problems a mathematics instructor may face while implementing technology in his lectures.

Throughout the whole paper we will concentrate mainly on live lectures of mathematics with computer software. In most of cases the expression "teaching mathematics" will mean live lectures of mathematics. Although we discuss here some aspects of lecturing with a computer software, the same software will be frequently used by students to do homework,

---

<sup>1</sup>This paper contains a summary of my lectures and discussions at NYIT in Nanjing and East China Normal University in Shanghai

projects and sometimes even exams. Therefore, some of the discussed here issues should be considered in a wider context.

Finally, in this paper I do not mention issues related to various types of calculators. This paper is mostly focusing on technology for live lectures and I cannot imagine how one can type a lecture on a calculator, even the best one, and display it on a large screen.

## 2 A brief history of my involvement in technology for teaching mathematics

My first contact with technology for teaching mathematics was about 20 years ago, with one of the very first versions of Maple. I was very much impressed with its ability to produce great mathematical graphs. However, at this time, I considered Maple's syntax as a serious burden for poor students who, struggling with mathematical concepts, will have to learn a number of sophisticated computer commands with their specific parameters and rigid syntax. Even at that time, though, I realized that Maple would change completely my teaching if I could find a good way of incorporating it into my courses.

Slightly later I obtained, from Waterloo Maple, my first copy of LiveMath. This small Computer Algebra System (CAS) for primary and high school students was at this time a fresh property of Waterloo Maple. This was, I think, the very first visual CAS where students were able to experiment with mathematics by typing mathematical formulae rather than complex commands, and then transform formulae and graphs using mouse pointer. From a didactic point of view, LiveMath was, and still is, the most student-friendly CAS. I could use it with my Papua New Guinea students without telling them how to use LiveMath. They learned in a few minutes how to deal with the program and every day they were discovering some new features. LiveMath, being a great mathematical laboratory, was a bit limited for teaching my engineering mathematics courses.

A few years later, I obtained from Roger Hunter one of the very first versions of Scientific Notebook (SNB). This was a turning point in my view on mathematical software. I realized that I have in my hand a technology that can be used for lecturing most of mathematical courses in a way that was never possible before. I became one of a few enthusiasts of SNB and associates of MacKichan Software Inc. Today, I am still involved in SNB matters like before. We will return to SNB later on.

Initially SNB was using the Maple computing engine. At some stage, due to complicated relations with Waterloo Maple, MacKichan Software decided to change the engine and replace it by the computational kernel from MuPAD. At this time I was contacted by Enrique Wintergerst who asked me to familiarize myself with MuPAD and evaluate it. This was the next turning point in my career, and MuPAD filled my life almost completely. I wrote a number of MuPAD related papers, four MuPAD books and I have done a countless number of MuPAD workshops and presentations (see for example [4]).

## 3 Live lectures with technology

Although I had been using SNB and MuPAD in my classes for years, I always used the whiteboard as the main teaching aid and these two programs as supporting devices. So,



while talking about mathematics, I frequently wrote mathematical facts and formulae on the whiteboard and occasionally I would turn to a computer program to get a graph or quickly calculate something that had a minor significance for the current teaching topic, and only the final result was necessary for proceeding further. The situation changed completely when I moved to NYIT. I found that a small white board 80x80cm is not enough for my lectures and the beautiful smart board in my classroom would lose synchronization after the first ten minutes of the lecture. As a result, the only thing I could use was a computer and a data projector. Therefore, I needed software that I could use as an electronic white board and that had all the features necessary to demonstrate a mathematical topic in the right way. I had available a number of mathematical packages including Scientific Notebook, MuPAD, Maple, Mathematica and Maxima. After a few days of testing various features, I eliminated most of these programs. Here are some the reasons: complex syntax of commands, missing formula editing tools, strange or very overloaded user interfaces. My main objective was the simplicity — simplicity of the interface as well as simplicity of using the tool. At the end I was left with the two most obvious choices—Scientific Notebook and MuPAD. None of these programs had all the features that I needed for my lectures. However, having good experience in using both of them, I could combine both programs together in a reasonably easy way. Below I describe some of my observations.

Scientific Notebook is a combination of a very good word processor, with perfect formula editing tools and a good computing engine taken from MuPAD. Thus, during my lecture, I could type a well-formatted document, transform or recalculate formulae with a single key shortcut, obtain the graph of a formula and change it to a computer animation. Most importantly, I could perform most of these operations in the background without distracting my students. There is nothing worse in a lecture with a computer than hunting for symbols in various menus and panels, doing some tricks to get the right graph, or scrolling through multiple menus in order to perform an operation. Here are a few simple examples.

In SNB in order to type in the formula,

$$\int_{-b}^b \int_{-b}^b \frac{\sqrt{2}xy}{x^2 + y^2} dx dy$$

we need to press [Ctrl+M] — this will switch to the formula editing mode. Then, by pressing [Ctrl+g] and [i] we will insert the symbol of integral, [Ctrl+arrow up/down] will allow us to type the top and bottom limits of the integral, finally [Ctrl+f] and [Ctrl+r] will produce the fraction and the root. After a very short practice all these key shortcuts become a natural habit, like using a knife and fork while eating. In fact, for any specific course of mathematics we may need to learn about 10 to 20 key shortcuts, and this will be enough for us to type comfortably a lecture in front of our students (see [2]). Scientific Notebook has a huge library of mathematical symbols and most of them can be also entered with a single key shortcut. Practically any symbol available in TeX can be typed in SNB. In fact, SNB documents are saved in TeX format.

From typing in a formula to recalculating it is usually a single step. For instance, leave the mouse pointer in the formula or just at the end of it, and press [Ctrl+e]. This way we may obtain the result, if such exists. For example, like below,

$$\int_{-b}^b \int_{-b}^b \frac{\sqrt{2}xy}{x^2 + y^2} dx dy = 0$$

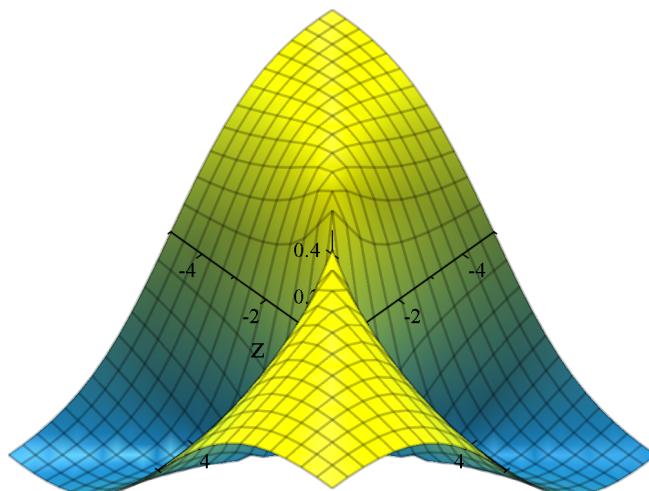


Figure 1: *Standard graph of the function  $\frac{\sqrt{2xy}}{x^2+y^2}$  in SNB*

If we wish to demonstrate our students why this integral is zero, we can plot the function under the integral and show them what is happening here. We need to leave the pointer in the formula or just after it and choose from menu *Compute>Plot 3D>Rectangular*. Unfortunately, this last operation does not have a keyboard shortcut. In our example we could produce the graph shown in figure 1.

If this graph is not enough, we can still modify it using the graph properties panel, add more elements to it or animate it. We have to perform all these operations through the graph properties panel, and this is a very useful activity for our students. They will see how the mathematical graphs are plotted and how to use different options to produce graphs that will explain to them the nature of mathematical objects. For example, in the graph that is enclosed in figure 2, we can easily see that between our surface and the  $xy$ -plane there are four identical 3D sectors, two above the plane and two below. This makes the integral equal to zero.

Although SNB has a number of interesting features that make it a great choice for live lectures of mathematics, it was important for me how my lectures will be presented using various data projectors that I had to use, especially those with low resolution. Here there are two things that helped me in my lectures. One was this very unusual feature of the program is that in SNB on the computer screen a line of the text will never flow beyond of the right border of the editing window. The line of text will simply wrap down before reaching the right edge of the window. This way we can magnify the view of the text to such level that characters will be well visible on the screen even using a very low resolution projector. Finally, we can develop our own text styles—one for displaying my lectures and another one for printing it. Thus, in the first case, we can use fonts like Trebuchet or Verdana that are particularly well displayed on a computer screen or with a data projector, and in the second one we could use fonts like Times or Georgia that work well in printed documents. This way, immediately after the lecture, my students were able to grab all my lecture notes, converted

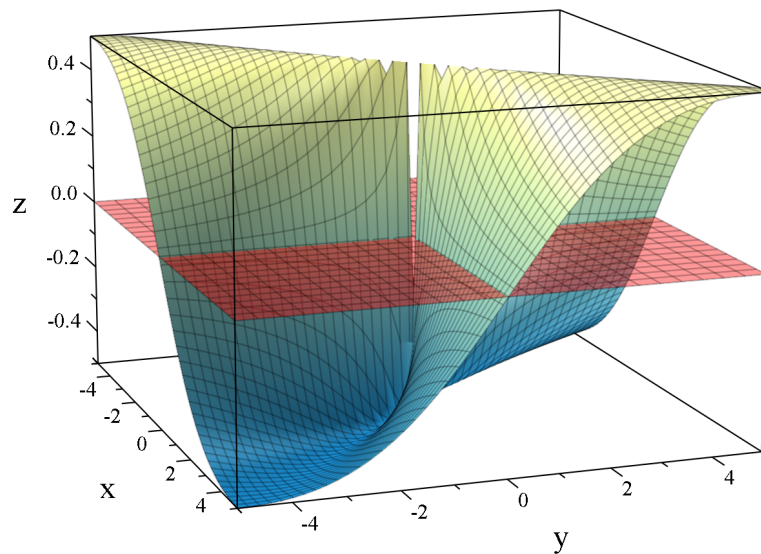


Figure 2: Graph of  $\frac{\sqrt{2}xy}{x^2+y^2}$  after modifications in SNB

quickly to PDF format. In fact, at the end of each of my classes I have been uploading all my lecture notes on the Moodle web site for my courses.

SNB has great editing and computing functionality, but still lacks many things for lecturing statistics, discrete mathematics and a few other disciplines. Here enters MuPAD with its programming language and very impressive graphics. This way we will gain a lot but also we will lose a lot. We are still able to produce a very good display of our text on the screen, use the line wrap features of MuPAD to see on the screen our text magnified to any sensible size. We can even mix our text with formulae. However, this requires a completely different approach. We type in our formulae as MuPAD commands and then copy and paste them into our text. For example, a command like the one below

```
(sqrt(2)*x*y)/(x^2+y^2)
```

will produce the formula  $\frac{\sqrt{2}xy}{x^2+y^2}$  which can be then copied and pasted into text of our lecture. Typing in formulae in such way can be quite tedious but it is still acceptable as long as other features of the program will allow us to produce results that are not available in other programs. In my teaching career, MuPAD has proven itself as a perfect tool for working with algorithms especially those in numerical methods or in discrete mathematics. Here is an example that I used in one of my lectures to demonstrate how the fern fractal algorithm works. The complete MuPAD file of the fern fractal is enclosed with this paper as a supplementary file (see supplement [1]).

```
fern := proc(x0, y0, nrofpoints)
  local r, points, x, y;
begin
  s[1] := (x,y) -> [ 0.77*x + 0.04*y, -0.04*x + 0.77*y + 1.6];
  s[2] := (x,y) -> [ 0.21*x - 0.24*y, 0.24*x + 0.21*y + 1.6];
```

```
s[3] := (x,y) -> [-0.19*x + 0.27*y, 0.27*x + 0.19*y + 0.44];
s[4] := (x,y) -> [0, 0.25*y];

de := proc(x, y, r)
begin
  if r<71 then
    s[1](x,y)
  elif r<88 then
    s[2](x,y)
  elif r<97 then
    s[3](x,y)
  else s[4](x,y) end_if;
end_proc;

[x,y] := [x0,y0];
points := [NIL $ nrofpoints];
for i from 1 to nrofpoints do
  r := random(100)();
  [x,y] := de(x,y,r);
  points[i] := [x,y];
end_for;
plot::PointList2d(points,
  PointSize=0.01,
  PointColor=RGB::Olive
);
end_proc:
w := fern(0.325,0.789,10000):
plot(w, Scaling=Constrained, Axes=None)
```



Although, syntax of the programming language in MuPAD is the simplest one one can imagine, there is another thing that my students appreciate very much. With the recent developments of so called easy plot package developing graphs in MuPAD was simplified to an absolute minimum—wrap any mathematical object in the `plot` command, just like this `plot(sin(x))`, and MuPAD will display the graph using some default parameters. In a very similar way we can create animations of mathematical objects. In fact, in MuPAD we can animate almost everything what can be seen on our plots (see the figure 4 and supplementary file [2]).

A number of other features make MuPAD a good choice for many courses. However, for some other courses MuPAD can be completely useless. For example, teaching synthetic geometry with MuPAD or SNB may not be possible at all. Here we face a very frequently asked question—which tool is best for teaching mathematics or which tool will be best for my course. This first question does not make any sense. Such a dream and universal tool simply does not exist. The second question makes more sense and we will try to find a response to it in the next section of this paper.

## 4 Get the right tool

Whatever we decide to teach we have to choose the tool that will give us most of the features that we need for our class. Otherwise our classes may turn into a complete disaster. We can easily imagine advanced statistics classes with MuPAD or SNB. However it will make more sense if we choose for them Minitab or Excel. We can also imagine a calculus or linear algebra classes with Excel but definitely SNB and MuPAD will make them more enjoyable and more successful. We can imagine a geometry class with MuPAD. This will be possible if we start using coordinates of points and equations of lines. However, Cabri, GSP or Geometry Expressions will serve much better this purpose. Finally, we can use Minitab or Statistica for quantitative methods for business students but for MBA students Excel will be the optimal choice. They have it installed on any Windows computer or Mac and they gain much more from using Excel than from any other software.

While considering computer technology for our classes, we should take into consideration many aspects. Some of them are listed here:

1. How difficult is the tool to learn for us and for an average student?
2. Does this tool contain things that we we wish to use in our class?
3. Are there any printed or online teaching materials for our software?
4. How much time will we spend adjusting the tool for our classes?
5. What is the real cost of the software for us, for the school, and for students?

## 5 Visual or menu-driven

While choosing a tool for our mathematics classes, we often feel that our students are overloaded with the amount of material they have to learn and learning how to work with a computer tool will be for them an additional burden. Therefore, we tend to choose tools that look easy and most of operations can be found through menus. We often forget that digging out some operations from menus can be also difficult, and many students will never learn all possibilities that are hidden in menu options.

Here we are coming to a very important issue of choosing between a visual, i.e. menu-driven computer program, and a command-line program. By a menu-driven program we mean here a program like SNB, Derive or LiveMath, where all operations are available through menus or toolbars, and where they are usually grouped by operations functionality. A combination of menus and toolboxes can be very complicated and difficult to learn.

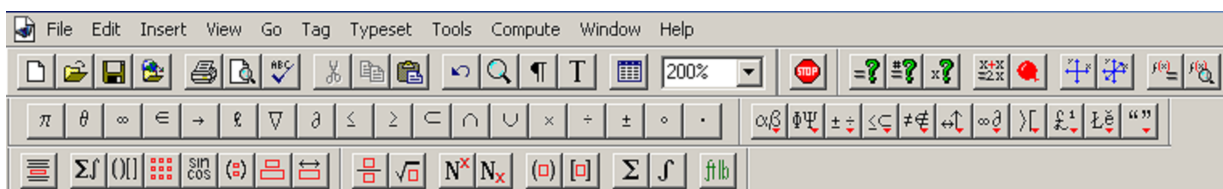


Figure 3: Menu and selected toolbars in SNB

However, many visual programs offer a way to modify the interface, hide some unnecessary toolbars and show only the minimum required for a student. This way is constructed SNB. For the same reason, the ascetic-looking Cabri has always had more users and real fans than some other geometry programs with a very sophisticated user interface. For exactly the same reason MuPAD is a favorite CAS for many undergraduate students. Its interface is so simple (see fig. 4) and so clear that after a very short time students can find there everything they need. In 2001, at the Inter-University Institute of Macao, the students of the mathematics education program evaluated a number of computer programs as tools for their high school mathematics classes. After a few weeks of experimenting they pointed out Cabri, SNB and MuPAD as the most suitable tools for high school mathematics classes.

There is one important thing that we should be aware of—the number of operations available in visual tools is usually limited to those that were programmed in menus and toolboxes. Fortunately the users of SNB can expand its functionality by adding operations that were programmed using the MuPAD programming language. MuPAD is the computing engine in SNB, so in this way some missing operations can be brought to SNB. However, this method has its limitations. For example, we cannot add many types of plots even if they are available in MuPAD. In programs like Derive or LiveMath there is no way to add user defined operations or procedures.

A command-line program, in opposition to a visual program, uses commands to produce mathematical calculations and graphs. Typical command-line programs are MuPAD, Maple or Mathematica. The concept looks simple—type in a command, press the [Enter] key and get a result. Then type in the next command, press the [Enter] key, etc.

Command-line tools have usually limited editing, text formatting and page layout functionality. However, the number of available commands and options is much larger than in

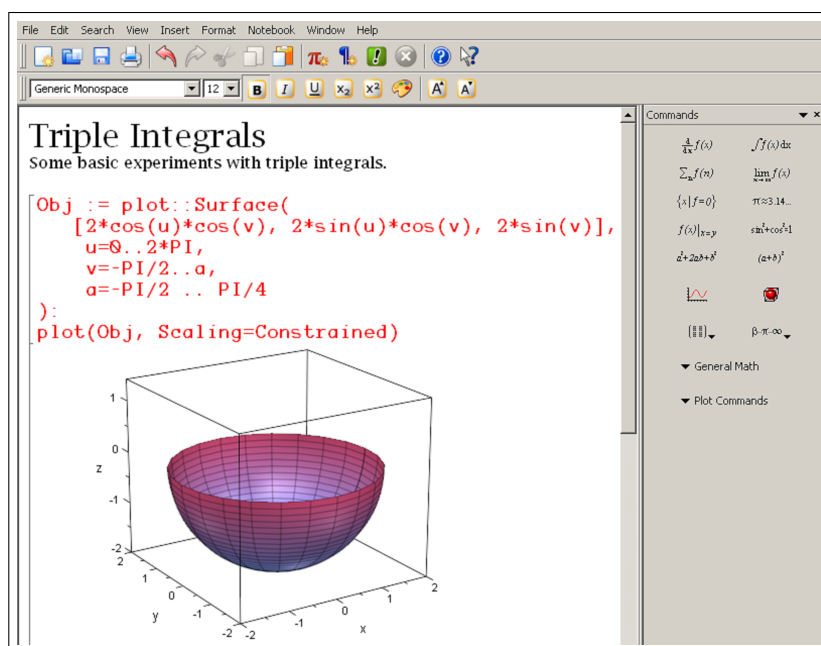


Figure 4: The interface, commands and output in MuPAD

visual programs. We can also write our own procedures using the programming language available. A typical argument against command-line programs is that the quantity and syntax of commands can be overwhelming for an average student. In fact, it is even overwhelming for an experienced user of such program. However, we do not need to master the program. We can introduce, as well as learn ourselves, only the commands that we need for the given topic. A typical single undergraduate class may need 0 to 5 new commands to experiment with a given topic. In the next class we can introduce some more commands and this way, step-by-step get a lot of experience in using a command-line tool in our course.

Another thing that seems to be a problem for many mathematics instructors is the programming language available in such programs. The most frequently used argument is "I do not know how to program, so I cannot use such tool". This is a very popular misunderstanding. The availability of something doesn't mean that we have to use it. In fact, about 90% users of a command-line mathematical program have never had to write a single program.

In summary—if we decide to use a visual tool, we usually gain good text editing and page formatting functionality as well as menu and toolboxes with a lot of goodies. However, usually we have limited opportunities to expand computational functionality of the program. If we decide to use a command-line tool we will have less developed text editing and page formatting functionality, but we will get a rich environment with good computational and graphing functionality.

## 6 No tool is perfect

A popular way of thinking is that we will get a tool, start it in the classroom, and everything will be ready to teach any mathematics course. The choice of a tool for a mathematics course should be consulted with the people who have used it for years. We should consider all the available options, as well as our time, our skills and our adaptability to a new environment. Buying a tool and adopting it in any mathematics course may have serious consequences later on. One of the most important things is to check if the software offers all or most of the features that we need. An acceptable situation is when we find about 80% operations that we will need for our course. We can borrow the remaining 20% from another tool or simply explain topics using the old typical approach—paper and pencil or whiteboard and marker.

If we miss more than 20% of the needed features in our software, we will have to spend a significant amount of time to expand its functionality. This may not be easy or even possible. For example, the great Geometry Expressions program has many things that we need for analytic geometry. However, this program still is lacking many features and is not especially convenient for Euclidean geometry. MuPAD or Maple are great for developing graphs using coordinates but trying to use them in analytical geometry may not be the best idea. We will be able to develop all geometry objects. However, the amount of coding will be much larger than the amount of work that we have to put into developing a geometry scene in Geometry Expressions. Moreover, we will never get the dynamic approach that is so important in dynamic geometry software.

## 7 Nothing is free

Finally, we come to the cost of adopting software in our classes. This is not necessary the cost in terms of money. There are more factors in this bill.

While visiting various universities in Asia, I found in a few places a box with Mathematica or Maple on the bookshelf. Sometimes the box have been covered by dust, showing that it hasn't been touched for a long time. A typical situation was that a given institution received the software as a gift, or they spent significant amount of money from a fund and then they found that there is nobody skilled enough to use the software or they do not have mathematical problems that can be solved with this program. In each case the money could be used in a better way.

One of the popular phrases that I have heard in some third world countries was "if I had money to buy Mathematica, I could become a world-known scientist". We do not become world class scientists or teachers by owning an expensive computer package. We need a few more things — necessary ideas that we want to investigate, necessary skills to use the program, patience to work many hours, days or years, and many things more.

There are many wonderful computer programs for mathematics that are completely free, and they are still rarely used in the classroom. One of the important issues here are the teaching materials—books, tutorials, examples that we can load to the program and analyze, as well as people using it already. If we cannot find printed or online materials showing how we can use the software in the classroom, then we will have to develop them on our own. Here comes the question—do we have time to do this and necessary skills? The existing



examples show that programs having a lot of printed teaching materials are the winners on the educational software market even if they are not necessarily the best choice from a computational point of view. At the same time, programs without such materials become hardly noticeable. The Mathforum web site lists a number of geometry programs that never caught the attention of mathematics teachers (see [5]).

Another important source of teaching materials is the software users' community and online resources. If such a community exists and is active, we can usually get help from somebody for a given topic, obtain tutorials for various teaching topics and ready examples that we can use in class. Many commercial programs have quite well developed community networks. For example, Maple web site contains numerous links to Maple resources, conferences, events and many other things (see [6]). A huge collection of Mathematica resources is available on Wolfram web site (see [7]). A very active MuPAD Education Group provides a lot of MuPAD materials through their web site (see [8]). There is also an online MuPAD magazine, *mathPAD Online*, with MuPAD related articles (see [9]). Finally, we can find on the Internet forums and discussion groups for various mathematical packages. For example, SNB forum (see [10]). Some other discussion groups can be found on Google Groups. For example open the link <http://groups.google.com/> and search for Maple.

Finally, we should also consider the money that we have to spend to buy a class or school license and see how often we will have to upgrade the software and how much we will have to pay for software upgrades. Some software producers give minor upgrades for free, while others ask for payments with every new chunk of code. It is worth to check if the software producer provides a free, time-limited, copy of the program so we can use it for a while and see how useful it will be for our classes. For example, a copy of SNB or MuPAD, working for one month, can be downloaded from MacKichan Software Inc. or SciFace Software web sites (see [11],[12]).

In summary—the total cost of incorporating mathematical software into our course can be much higher than the cost of the license.

## 8 The black-and-white-box issue

While teaching mathematics with a computer program, we often experience situation where a computer tool works for us like a black-box. For example in SNB we can solve a differential equation by selecting menu *Compute>Solve ODE>Exact*. Here is such an example:

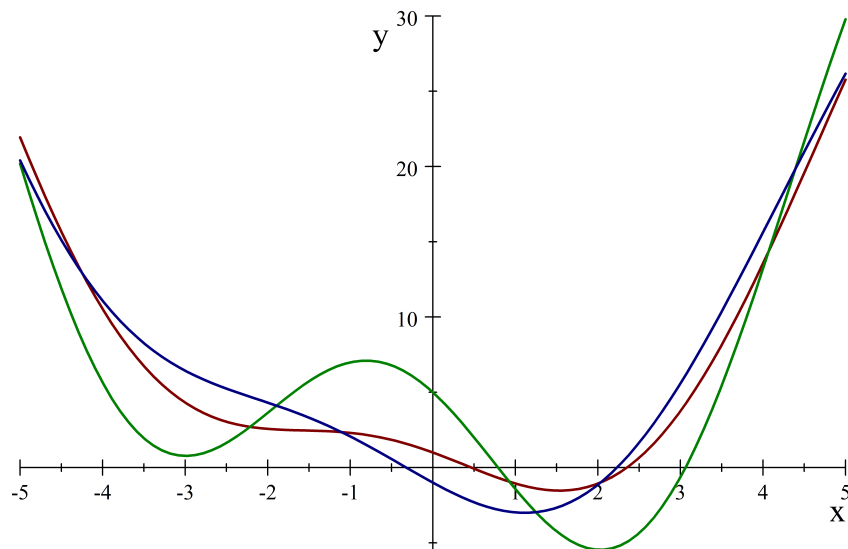
$$y'' + y = x^2, \text{ Exact solution is: } \{C_2 \cos x - C_3 \sin x + x^2 - 2\}$$

Now, another black-box can be used to plot some of the solutions obtained by using different values for constants  $C_2$  and  $C_3$ . For example,

$$3 \cos x - 2 \sin x + x^2 - 2, 7 \cos x - 5 \sin x + x^2 - 2, \cos x - 3 \sin x + x^2 - 2$$

This is the point where many mathematics instructors begin to realize that the strategy of teaching mathematics with a computer program should be crafted very carefully. When teaching mathematics with a computer program, we have two stages—the white-box stage and the black-box stage. In the white-box stage we explain and build the concept. After the concept is well worked out and we are sure that our students understood it correctly, we can show them tools that will automatize solutions with a single command or a menu option.

In one of my classes, I needed to explain what the standard deviation of a dataset is, and how to calculate it using the step-by-step method. Then, a student asked me to skip this



explanation and show them how to produce standard deviation using an Excel command. I did it. I used the Descriptive Statistics tool that is a typical black-box and I asked the student to explain to me how she understands the obtained output. She agreed that without a proper knowledge of the terms used in the Descriptive Statistics output, she is not able to understand anything. Figure 5 shows the two phases of calculating the standard deviation for a given set of data. On the left side, we show the formulae that the student has to generate and all intermediate steps to get the final result. On the right side, we show typical black-box use of the Excel tool known as Descriptive Statistics.

Data	Deviations	Squares	Mean	=AVERAGE(data)	Descriptive statistics output	
0.735022854773298	=A2-\$F\$1	=B2^2	Data count	=COUNT(data)	Mean	0.598889656581674
0.226201584609154	=A3-\$F\$1	=B3^2	Total squares	=SUM(C2:C21)	Standard Error	0.0568890840728971
0.104333034468509	=A4-\$F\$1	=B4^2	Variance	=F3/(F2-1)	Median	0.653268130047495
0.871369678373034	=A5-\$F\$1	=B5^2	Std. deviation	=SQRT(F4)	Mode	#N/A
0.824016088249054	=A6-\$F\$1	=B6^2			Standard Deviation	0.254415718329397
0.134861911222763	=A7-\$F\$1	=B7^2			Sample Variance	0.0647273577330631
0.813160631375565	=A8-\$F\$1	=B8^2			Kurtosis	-0.556342834878159
0.559929232127076	=A9-\$F\$1	=B9^2			Skewness	-0.773742652458803
0.409567178649554	=A10-\$F\$1	=B10^2			Range	0.827183594060877
0.632690259758077	=A11-\$F\$1	=B11^2			Minimum	0.104333034468509
0.931516628529387	=A12-\$F\$1	=B12^2			Maximum	0.931516628529387
0.825714107237854	=A13-\$F\$1	=B13^2			Sum	11.9777931316335
0.817008556466353	=A14-\$F\$1	=B14^2			Count	20
0.58578458250101	=A15-\$F\$1	=B15^2				
0.733080683483045	=A16-\$F\$1	=B16^2				
0.600301648645582	=A17-\$F\$1	=B17^2				
0.49750209292649	=A18-\$F\$1	=B18^2				
0.765715412536379	=A19-\$F\$1	=B19^2				
0.673846000336914	=A20-\$F\$1	=B20^2				
0.236170965364381	=A21-\$F\$1	=B21^2				

Figure 5: Calculations of standard deviation in Excel using the white-box method (left) and the same results produced by Descriptive Statistics tool (right)

In reality, the white-box phase should never be avoided in any mathematics course with a computer program. In fact, we should emphasize this part of the teaching even more than

in mathematics classes without computer. The major reason is that in traditional classes, students after learning the concept in the white-box phase, will still be practicing the concept in further examples and situations. With a computer software, after learning the concept in the white-box phase, students will start using the black-box tools and the learned concepts will begin to deteriorate in their memory. A very detailed discussion of the black-box and white-box concept was discussed by B. Buchberger (see [1]).

## 9 Summary

There is a large selection of powerful mathematical programs. However, in order to use them in our lectures we have to consider a number of factors:

1. What type of software is the most suitable for our courses?
2. Do we prefer easy to use visual software or harder to use command-line programs?
3. Do we have in our software all the tools needed to teach our course?
4. Are we able to expand the functionality of our program?
5. Are there any teaching materials for the program that we are going to use in our course?

And of course it could be worthwhile to:

1. Develop methodologies for teaching mathematics with computer software, and strategies for their introduction in our classrooms.
2. Produce teaching materials (textbooks, notebooks, online materials), that will incorporate CAS into mathematics content according to good educational practices and principles.
3. Organize workshops, seminars and training sessions for current mathematics teachers as well as introduce CAS into teaching programs for mathematics teachers at the university and teachers' colleges.
4. Initiate changes in the official mathematics curriculum as well as changes in examination policies allowing CAS to be used in exams.
5. Conduct research on many aspects related to using CAS in mathematics teaching.

However, this is another story (see [3]).

## 10 References

1. Buchberger B., Should Students Learn Integration Rules?, SIGSM Bulletin, Vol. 24, Nr. 1, January 1990, pp. 10-17.
2. Hardy D., Walker C., Doing Mathematics with Scientific Workplace & Scientific Notebook, MacKichan Software Inc., 2005.
3. Majewski M., MuPAD for the Classroom— a Discussion on Using Computer Algebra Systems in Teaching Mathematics, Proceedings of ATCM 2007, Taiwan, December 2007.
4. Majewski M., Getting Started with MuPAD, Springer Verlag, Berlin-Heidelberg-New York, 2005, pp. 265.
5. Mathforum geometry software list <http://mathforum.org/geometry/geometry.software.html>
6. Maple web site: <http://www.maplesoft.com/>
7. Mathematica web site <http://www.wolfram.com/>
8. MuPAD Education Group web site <http://www.sciface.com/education/>
9. mathPAD Online magazine <http://www.mathpad.org>
10. SNB forum <http://forum.mackichan.com:81/~mackichan>
11. MacKichan Software Inc. web site <http://www.mackichan.com>
12. Sciface Software GmbH web site <http://www.sciface.com>

## 11 Supplementary files

1. MuPAD file for creating the fern fractal `fern.mn`
2. MuPAD file for the animated surface, figure 4, `animsurf.mn`