MuPAD — Ten Years Later

Miroslaw Majewski
e-mail: mirek.majewski@mupad.com
New York Institute of Technology
School of Arts and Sciences
Abu Dhabi
United Arab Emirates

May 17, 2007

Abstract

MuPAD is a modern, general-purpose, Computer Algebra System. It has a number of features that make it particularly suitable for teaching mathematics and experimenting with mathematical concepts.

In this paper we will describe briefly origins of MuPAD, its history, and we will demonstrate some of its features. We will concentrate on a few selected aspects of using MuPAD in the classroom and on the development of teaching materials for use in mathematics classes.

1 Introduction

Many of us have heard about Maple or Mathematica. Some of us still remember Maxima—one of the very first Computer Algebra Systems (CAS). However, not many of us are familiar with another computer algebra system known as MuPAD. In fact, many of us use MuPAD without even knowing it. For years, MuPAD, more specifically its computing engine and libraries, have been hidden in such popular mathematical software as Scientific Notebook and Scientific Workplace from MacKichan Software Inc. The scientific publisher Springer Verlag also uses MuPAD in several of their electronic publications. One of them is *DUBBEL interaktiv—Das elektronische Taschenbuch für den Maschinenbau*. Another famous electronic book using MuPAD computing engine and Scientific Notebook interface is *Calculus, Understanding Its Concepts and Methods*, published by MacKichan Inc. Recent developments show that in the nearest future, there will be more applications and books powered by MuPAD. It is also worth noting that MuPAD itself is becoming popular as a teaching tool for undergraduate mathematics. In many European schools and universities, MuPAD is used to support the teaching of mathematics, to visualize numerous mathematical facts using its impressive graphics, or to demonstrate mathematical algorithms in MuPAD's programming language.

Before discussing specific features of MuPAD, let us remind ourselves what computer algebra systems are and how we use them.

Most CAS use command-line instruction. This means that we have to type in a command or a sequence of commands and execute them by pressing the [Enter] key. The result, in the form of a formula or a graph, is then displayed below. We can then type in further commands, or modify the existing ones, press [Enter], get new results, and so on. From this description it might seem that using CAS is a very mechanical process. However, we still have to use our brains to analyze the obtained results and decide what commands are required to proceed with our calculations. Below I show two basic commands and the output obtained by executing them. The code comes from MuPAD.

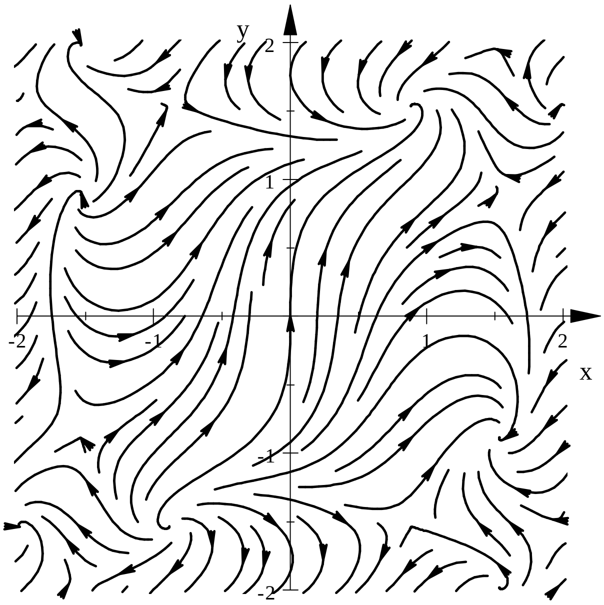
Let us begin with a command for calculating the integral of ax^n, for $x = 1..100$.

```
int(a*x^n, x=1..100)
```

$$\begin{cases} a \cdot \ln(100) & \text{if} \quad n = -1 \\ \frac{a \cdot 100^{n+1}}{n+1} - \frac{a}{n+1} & \text{if} \quad n \neq -1 \end{cases}$$

With the next command, we plot the streamlines of a vector field given by the parametric equations $f(x, y) = (\sin(x^2 + y^2), \cos(x^2 - y^2))$.

```
plot(plot::Streamlines2d(sin(x^2+y^2), cos(x^2-y^2),
     x = -2..2, y = -2..2,
     MinimumDistance = 0.1,
     TipLength = 2*unit::mm
  )
)
```



Many scientist use CAS as electronic laboratories, where they develop a sequence of commands representing a chemical, engineering or other process, and then use these sequences of commands for different sets of input values. In this way they are able to simulate possible outcomes of a process in a specific situation. Such simulation chains can be very long, sometimes spanning thousands of lines of code.

2 A very brief history of MuPAD

MuPAD is the youngest Computer Algebra System. Its origins are based on a number of students' projects from Paderborn University, and frequently the students who contributed in its early stages later become members of the development team.

It is rather difficult to list all the facts in the history of MuPAD, and I believe not all of them are important enough to be included in this short review. I will concentrate on a few milestones that could be interesting to an average reader.

The beginning of MuPAD goes back to the year 1989, when Professor Benno Fuchssteiner from Paderborn University and his student Waldemar Wiwianka started a research project on handling large data generated by algorithms used to investigate the structure of nonlinear systems. The first students taking part in this project were Oliver Kluge and Karsten Morisse. Their joint Master's thesis was the first successful outcome of the project. At the same time, in MuPAD's early stages, Gudrun Oevel, another student of Professor Fuchssteiner, developed foundations of MuPAD graphics. Since then, MuPAD has become a major research project, carried out by researchers and students at Paderborn University.

In 1997, MuPAD became a commercial product. Its producer is *SciFace GmBH,* a computer company in Paderborn, with Dr. Oliver Kluge as manager. It is important to mention that a strong group of researchers, called the *MuPAD Research Group*, is continually developing and implementing new algorithms for the system and works on expanding its functionality. It is also worth mentioning that MuPAD has a number of features that distinguish it from other Computer Algebra Systems. These include its open concept, where all users can expand the functionality of the program by developing their own libraries, the object-oriented concept applied to all MuPAD elements, as well as its excellent graphics and animation tools.

In 2007, after ten years of existence as a commercial product, MuPAD has become a major tool for teaching mathematics at many universities and schools all over the world. In particular, it is well known in Europe and Japan. Let us see what mathematics educators appreciate the most in this modern program.

3 Modern interface and accessible help

In the last few years, MuPAD's interface, the notebook, and its help tools were redesigned to make them as user-friendly and ergonomic as possible.

Although the interface is similar to the one used in older versions, it is significantly different — there are new icons, a new sidebar, a new look, and new functionality. Most of the former functions remain in the old places, but many things were simplified or were made more user-friendly. For instance, in the past, saving graphics, animation or video required some knowledge about file formats, image resolution, keyframe frequency, etc. In the new MuPAD, the export wizard makes exporting such files as easy as possible by suggesting file formats and their properties, the number of frames in an animation, etc. This way, creating a movie with an animated graph of a function in 3D is a very simple and straightforward task.

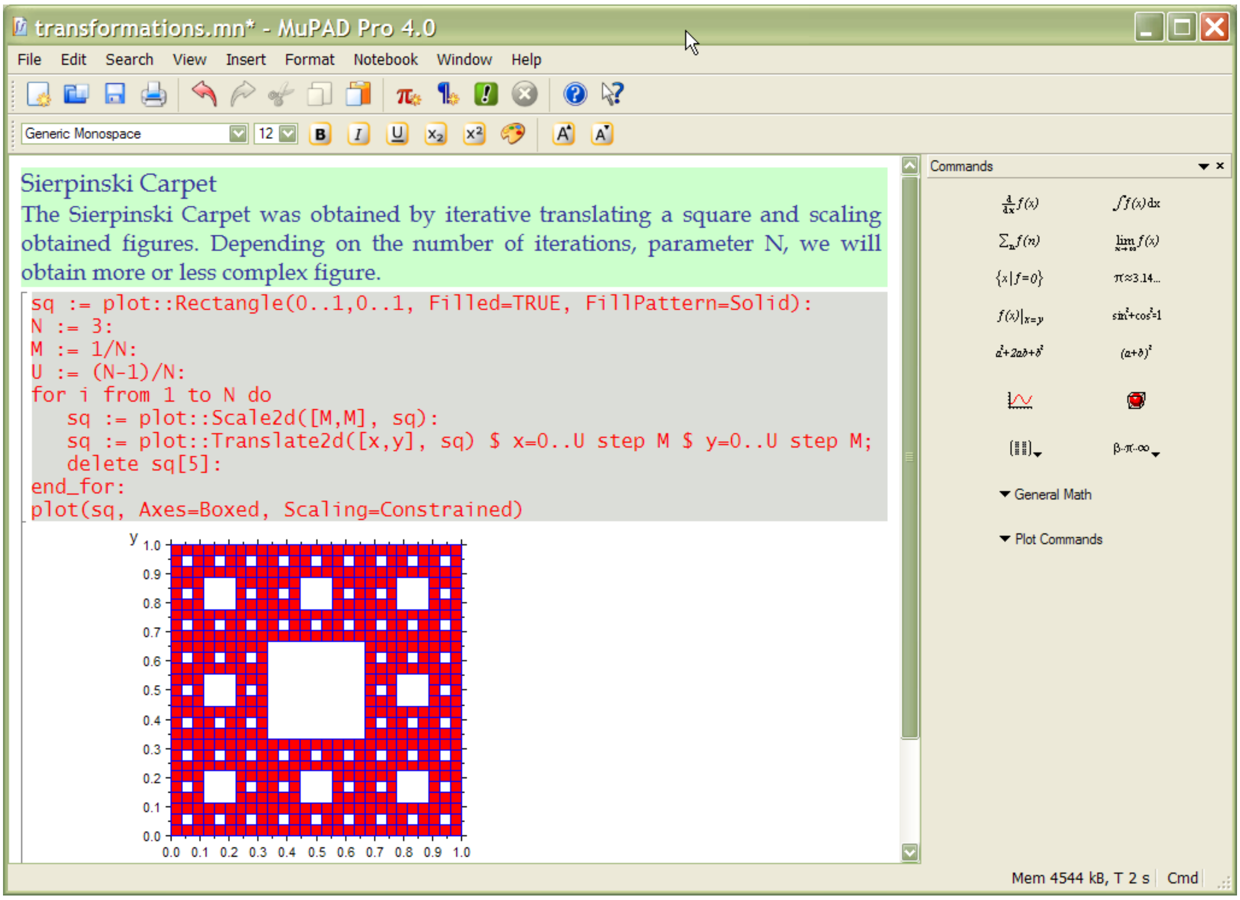


Figure 1: *Fig.1 The new interface of MuPAD 4.x*

Another thing worth noticing is the sidebar. This utility appeared recently, in MuPAD 4.0 (see fig.1). This is where we will find shortcuts to many MuPAD functions, procedures and plot classes.

Another important change is the new look of the notebook and its format. The notebook editing functionality has been expanded in many ways. As before, we have three types of regions: text, input and output. However, in each of these regions, we can apply formatting like in any good word processor. Therefore, we have many of these nice features like text wrapping, fonts, colors, or tables. This concept of document formatting resembles that of Scientific Workplace. For instance, the line width is reformatted depending on the screen size, users can change the character size globally, etc. Preparing different versions of the same notebook is now a very simple task. We can make the same notebook with formatting convenient for working on the computer screen, with large characters for a classroom presentation, or using small characters for printing purposes. Text regions may contain tables, formulas, mathematical characters, and pictures.

One thing worth emphasizing is the integration of Virtual Camera (VCam) with the notebook. We no longer need to open graphical objects in Virtual Camera in order to manipulate them. A single click on the plot in the notebook automatically switches to Virtual Camera. The new menu and icons appear on the interface and the sidebar changes to the graphics objects browser. We can then do with our plot everything that was possible in Virtual Camera before. We will talk more about Virtual Camera and graphics concepts in MuPAD in one of the later sections.

One other thing that we will notice right away is the new help. The help pages look like MuPAD notebooks, all the graphics embedded there are alive, we can manipulate them, preview animations, etc. MuPAD's help is printable, and we can print each section of it. We can copy any part of the help file and paste it into a MuPAD notebook, recalculate MuPAD commands and produce plots. We can paste the content of the help file into any word processor and format it the way we need.

4 The open concept

The open concept in MuPAD this is something that most creative mathematics teachers appreciate and use quite often. Like in other CAS, almost all of MuPAD's mathematical knowledge is located in its libraries. The libraries in MuPAD contain about 2000 procedures, and continue to grow. Each version of the program adds many new procedures, mostly developed on users' requests. However, if we think that there is something specific that we urgently need, we can produce our own new procedures and place them in our own library or expand one of the existing libraries. This can be done in many ways. Our new library can be in a separate file and imported into any MuPAD notebook, or it can be inserted into a notebook as hidden code. For example, some the teachers, when using MuPAD with students, develop notebooks covering selected aspects of their course. Very often, they hide more complex code in the start up commands section of the notebook (see menu File>Properties>Start Up Commands). This code is executed automatically after opening the notebook and each command declared there can be used in the notebook like any built-in command. Students will not see the code, but they will be able to use it. This way students will not be confused by strange looking, and sometimes very difficult to understand code.

Another very important feature of the recent version of MuPAD is the format of the files where the whole interface is defined. All these files are in XML format, which can easily be opened in any XML editor or even in Windows Notepad, if we do not have any better tool. In these files, we can redefine the whole menu of the program, add or remove commands from the side bar, change the set of icons used by the program, etc. This way, educators can make a customized version of MuPAD that is convenient for teaching a particular topic or subject. For example, a teacher of a particular course can customize MuPAD for use in his course by creating, on the commands bar, a new menu with the commands that he needs for his course (see fig. 2). This way, students do not need to learn the syntax of these commands, and they are able to use them as soon as there is such a need.

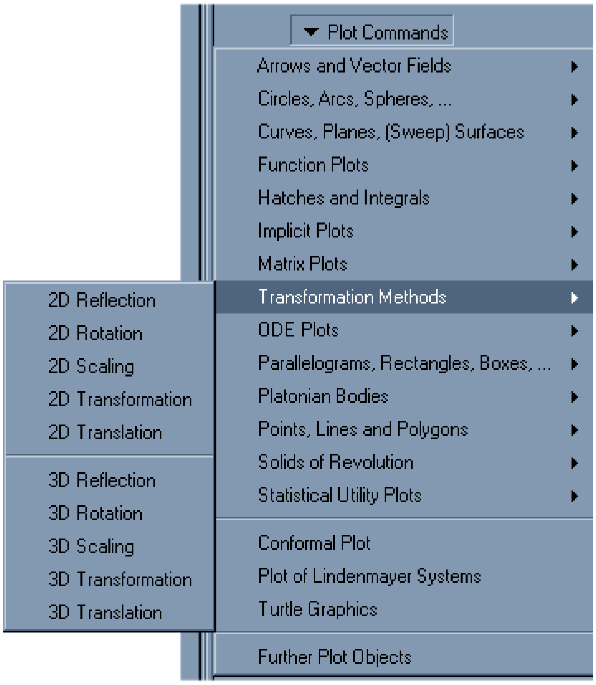


Fig. 2 Command bar in MuPAD 4.x

5 Graphics concepts

There is no doubt that computer graphics can play an important role in the teaching of mathematics. We can show graphs of functions and other mathematical objects, use animations to show how the objects change when some parameters change, and analyze what will happen if the data used to generate the graph changes, and so on.

The MuPAD graphics have always been impressive. The recent version of the program contains a completely new Virtual Camera (VCam)—the tool to render and manipulate mathematical graphs. All graphs displayed in VCam are highly interactive. We can move, rotate, and transform them in several ways. VCam gives users access to all the properties of graphs and their components (see fig.3).

In fact, VCam can do much more for us. It shows us the complete structure of the graph (top-right part of figure 3) and gives us access to all the parameters of all the objects displayed. Finally, from VCam we can export MuPAD graphs into many formats. We can produce static pictures with low resolution for the web or with high resolution for printing, animations can be exported to AVI files and used to produce a gallery of mathematical videos, 2D graphs can be saved in SVG format for web pages, and finally, any graph can be exported to JavaView format and displayed as an interactive object on the web. When exporting graphics from VCam, we do not need to know much about files and formats. The VCam's export wizard guides us through the entire process and suggests suitable formats and other parameters. This is particularly important when exporting video files with mathematical animations.

Although MuPAD graphics are organized systematically and the collection of different types of graphs is impressive, high school students may not appreciate this feature as much as university students. The number of graph types and their syntax may be too difficult

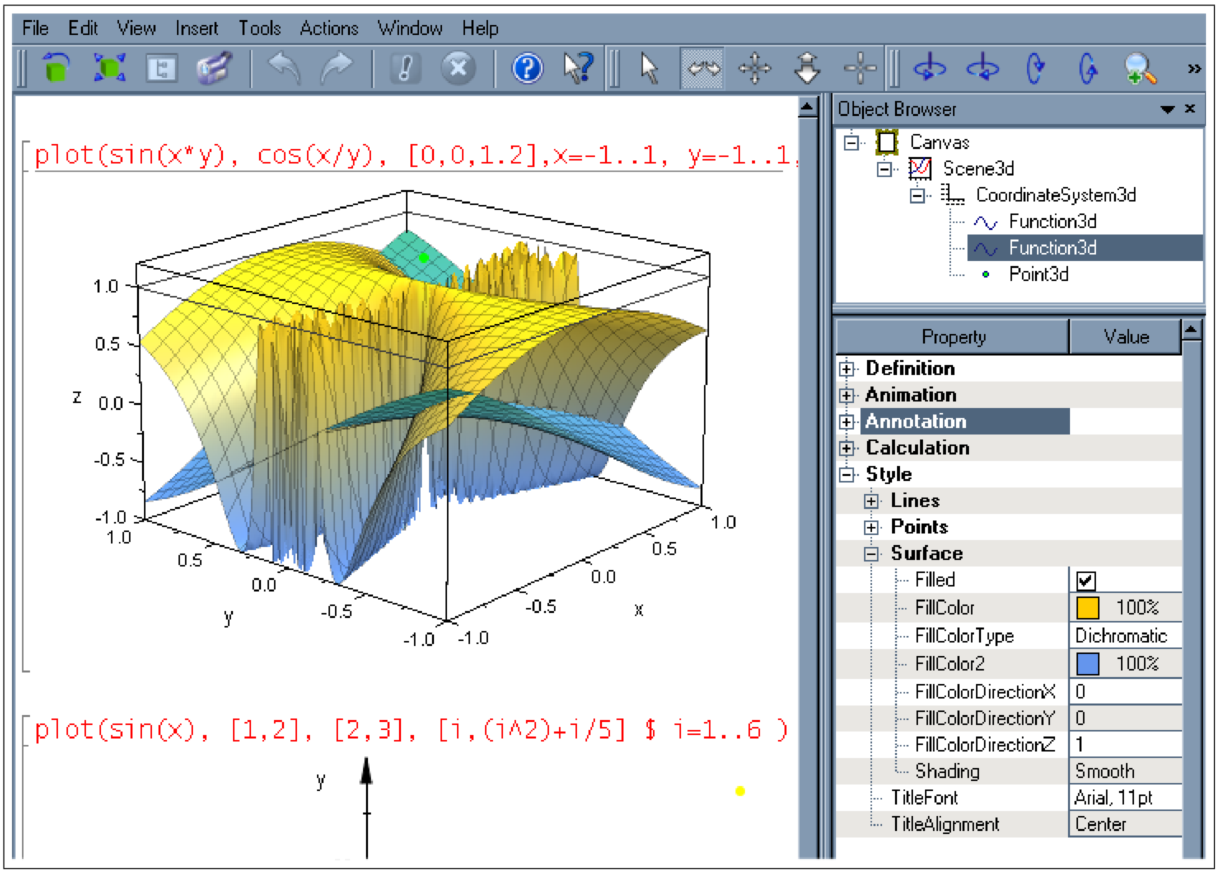
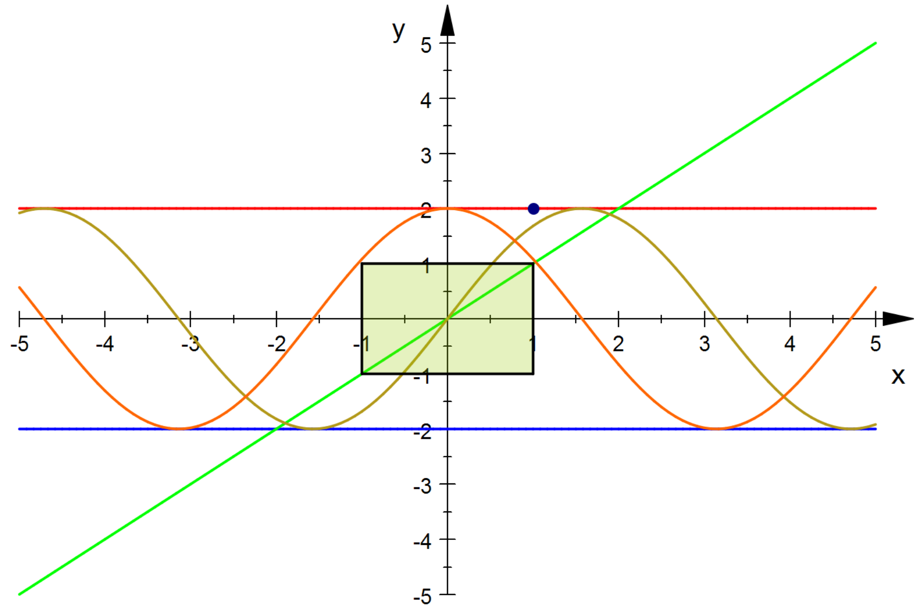


Figure 2: *Fig. 3 In VCam users have access to all properties of the graph*

for many students. Therefore, recent developments have concentrate on producing a smart plot procedure that will be able to guess what users want to plot, based on minimal data input. The statement below shows how this command works. Here, we plot two horizontal lines ($y = -2$ and $y = 2$), the two functions ($y = 2\sin x$, and $y = 2\cos x$), point $[1, 2]$, and a polygon connecting the points $[-1, -1], [-1, 1], [1, 1], [1, -1]$. We use here the absolute minimum of data needed to obtain such a plot.

```
plot(-2, 2, x, 2*sin(x), 2*cos(x), [1,2],
  {[[-1,-1],[-1,1], [1,1], [1,-1], [-1,-1]], Filled}
);
```



A very similar result using the traditional syntax in MuPAD would require at least the following set of commands:

```
f1 := plot::Function2d(-2):
f2 := plot::Function2d(2):
f3 := plot::Function2d(x):
f4 := plot::Function2d(2*sin(x)):
f5 := plot::Function2d(2*cos(x)):
p1 := plot::Point2d([1,2]):
poly1 := plot::Polygon2d(
   [[-1,-1],[-1,1],[1,1],[1,-1]], Filled=TRUE, Closed=TRUE
):
plot(f1, f2, f3, f4, f5, p1, poly1)
```

The new smart plot will be a part of MuPAD's next update. However, its early version is already used for developing some educational resources.

6 MuPAD and Symbolic Geometry

While teaching geometry, many teachers prefer to use Dynamic Geometry software like Cabri or Geometer's Sketchpad (GSP). This is the software where we can develop geometric objects in a similar manner like with pencil, compass and ruler on a paper. Additionally, in Cabri or GSP we can use the mouse pointer to move geometrical objects, change their orientation, etc. Unfortunately this type of interaction with graphs in CAS is not possible. In VCam, we can interact with geometry objects provided that we are not changing their location in the scene. For instance, if we create a line with a point on it in MuPAD, we will be able to move the line, as well as the point and the whole coordinate system together. However, we will not be able to move the point along the line.

Geometry Expressions, a recently developed software by Saltire Software, is a new computer tool where users can create and interact with geometry objects in a way similar to other Dynamic Geometry programs. Geometry Expressions combines some features of Computer Algebra Systems and Dynamic Geometry software. We can consider it as a starting point for a new class of mathematical programs—Symbolic Geometry software.

Geometry Expressions can represent its calculations in the MuPAD format. We can then paste such output into MuPAD, proceed with further calculations and paste the obtained results back to Geometry Expressions This way, dynamic geometry becomes available to MuPAD users.

7 MuPAD for the classroom

There are still not many printed or electronic resources to help teachers in using MuPAD. There are very few books showing how to use MuPAD. Two of them, written by the author of this article, are available on the market (see [4] and [5]).

For years, mathematics teachers using MuPAD in their classes have been developing interactive notebooks with MuPAD where students can perform experiments with high school or university mathematics topics. This development was never organized in a systematic way, and such loose examples have been used mostly locally, sometimes exchanged between teachers.

The need for a more organized collection of MuPAD teaching resources is the driving force behind the latest project — *MuPAD for the classroom*. This is going to be a long-term project to develop hundreds of MuPAD notebooks covering all the major areas of undergraduate mathematics. The first part of this project, *Precalculus*, is in its early development. It is expected that the precalculus notebooks will be ready for use sometime in December 2007. The whole project *MuPAD for the classroom* is a good opportunity to test how MuPAD will fit into undergraduate mathematics teaching in a more organized context.

8 Final comments

Although in recent years MuPAD has already proven itself to be a valuable tool in teaching mathematics, recent developments show that in the next few years, mathematics educators may expect further improvements of the program, which will make its use in the classroom easier and more transparent. We may also expect that, after years of software development, teachers will start getting printed and electronic resources that will make teaching and learning mathematics with MuPAD more exciting and effective.

9 References

1. Beitz W., Grote K.-H, *DUBBEL interaktiv—Das elektronische Taschenbuch für den Maschinenbau*, 2002.

2. Hardy D., Walker C., Richman F., Wisner R., *Calculus, Understanding Its Concepts and Methods*, MacKichan Inc., 2005.

3. Majewski M., *Getting Started with MuPAD*, Springer Verlag, 2005.

4. Majewski M., *MuPAD Pro Computing Essentials*, Springer Verlag, second edition, 2004.