PROGRAMMING ONE'S OWN STAND-ALONE TEACHING AIDS FOR STUDENTS OF MATHEMATICAL SUBJECTS

Michal Novák
Brno University of Technology, Czech Republic
E-mail: novakm@feec.vutbr.cz

Abstract

The article encourages teachers to program their own applications – mathematical teaching aids. It discusses the issue of teaching aids for students in mathematical subjects from the point of view of their accessibility, i.e. especially what software equipment students are required to use in order to actually use the aids. A number of aids require that either specialized mathematical software such as Maple, Matlab or Mathematica is installed at the client computer – i.e. that the student bought a license of the software, or that specialized software is installed at the university server(s). The article focuses on situations when neither of the conditions is possible. Comments on programming teaching aids which can be used with minimum requirements at the client side (typically Internet connection only) are included. First, the overview of possibilities is given. Later, the issue of PHP based applications is explored. The article includes examples of existing teaching aids prepared by a non-programmer.
Key words: *e-learning, programming, teaching aids, teaching mathematics.*

Introduction

Regardless of a country modern teaching of mathematics relies on and makes use of mathematical software. The software can be used in a number of contexts: a pupil / student may use it in his / her free time, it may be introduced to a pupil / student by a friend or a chance, the software may be used at some special classes (usually at lower grades of education) or – at universities – practical computer-aided classes may form a significant part of time allocated to a given subject. Teachers have two options when using software for didactic purposes: either *use existing programmes* or *program one's own applications.*

Even though programming one's own applications may seem unnecessary as most – if not all – areas of school mathematics have already been covered by professional software, there are still some issues in favour of using one's own applications. First of all, there is the *language barrier* as most of the complex professional software such as Maple or MATLAB have not been localized into many languages. This results in low efficiency when using the software by non-native speakers as pupils / students are often required to use software, the language of which (e.g. help files) they do not fully understand. For details on this topic cf. e.g. (Langerová, 2006), (Novák & Langerová, 2006) or references of these articles. Second, it is the issue of *didactic content*. When programming one's own application, the programmer – teacher may shape it exactly the way he / she needs or wants. The didactic content and message may be communicated more efficiently this way than in large and complex software, which is always a compromise for all possible target groups. Third, licensing of such applications is usually very simple as they are either freeware or open source products. Using such software is therefore very cheap.

The major drawback of programming one's own application is that a number of non-programmers believe programming is a difficult task. Apart from advocating the use of one's own applications, I am going to challenge the view of programming as a difficult task.

Some classifications of teaching aids

When classified from the point of view of accessibility, which is no doubt an important aspect for every student (especially of combined or distance forms of study), mathematical teaching aids can be divided into two disjoint categories:

- *software-dependent files* such as MATLAB m-files, Maple worksheets, Mathematica files, Excel spreadsheets, etc. The common feature of this type of teaching aids is that they can be run from within the application only, i.e. their user is required to have access to their native application.
- *stand alone applications* programmed using a variety of programming languages such as C, C++, Java, PHP, graphic user interfaces (GUI) of various programmes including Maple.NET, special MATLAB toolboxes, etc.

When classified from the point of view of software requirements, mathematical teaching aids can be divided into the following disjoint categories:

- aids, which *require* the user *to modify the configuration of the computer* the user is actually using when working with the teaching aid (i.e. of the client computer),
- aids, which *do not require any modification of the client computer configuration*.

When classified from the point of view of access restrictions, mathematical teaching aids can be divided into the following disjoint categories:

- aids, *the access to which is restricted in some way*, e.g. by licensing schemes restricting free and unlimited use. The access to teaching aids is also often limited to students of a given university / faculty upon entering a login / password combination.
- aids, *the access to which is not restricted in any way*.

The above classifications overlap to some extent. Before using a software-dependent file, the respective software must usually be installed (thus the client computer configuration must be changed). In a number of cases, stand-alone applications must also be installed prior to use. Access to both software-dependent files and stand-alone applications is usually restricted under various licensing schemes.

Professional software vs. gadgets

All mathematical software widely used at universities of technology – Maple, MATLAB, Mathematica – has one common flaw: the programmes are "too professional", which results in their lack of didicaticity as far as many mathematical concepts are concerned. The programmes are able to compute a normal form a matrix, multiply two matrices or compute their determinants in a split of a second even for huge matrices. They find derivatives, integrals, etc. However, all they show is a *result*, not the *process* of reaching it. Sometimes special commands, often in the form of tutors, such as Maple DiffTutor `()` command, which attempt showing even the process, are included but these are rare and scarcely used in classes for a number of reasons including language barrier – cf. e.g. (Novák, 2007, chap. 2) or (Langerová, 2006).

PROBLEMS
OF EDUCATION
IN THE 21st CENTURY
Volume 5, 2008

On the other hand, one-purpose gadgets, or simple applications, which do not aspire to be all-encompassing, can focus on the didactic part of the concept or skill in question. Therefore, the debate on what is more useful for a student – whether professional software such as Maple, MATLAB or Mathematica or a well-prepared set of one-purpose applications – is a legitimate and an inspiring one.

A typical student (of both attended and combined / distance forms of study) – or even a typical user in general – considers three aspects when any kind of software is regarded: *functionality meeting needs*, *accessibility* and *price*. Software which does everything the user currently needs and which can be accessed from any computer without any restrictions or costs is naturally everybody's ideal. Given the above classifications, stand-alone applications, which do not require any modifications of the client computer, and the access to which is not restricted in any way, are closest to this ideal (provided their functionality meets the students' needs).

A choice of programming languages

As far as stand-alone applications are concerned, these may be programmed in a number of programming languages or ways:

- using a Maple.NET, MATLAB compiler toolbox or other environments,
- using Java or variants of C such as C++, C#, etc.
- using plain (X)HTML or (X)HTML combined with JavaScript, i.e. the teaching aids can exist in the form of client-side web pages,
- using PHP or ASP or other dynamically generated web pages technology,
- other stand alone applications.

Maple.NET and MATLAB compiler toolbox teaching aids are Java based applications. Moreover, they can be used once the programmer's workplace has appropriate – rather expensive – licences. Apart from some Java- or C-based or "other stand alone" aids, all the above types of applications require an active Internet connection; apart from some C-based or "other stand alone" applications, they require a web browser. Most of the "other stand alone applications" require installation at the client computer.

Java based applications usually require that Java Runtime Engine is installed at the client computer. Therefore, from this point of view, most of the above programming languages / ways falls within the category of aids, which require the user to modify the configuration of the client computer. This may not be a problem when the student is using his / her own computer and is willing to install additional software or plug-ins. However, when using somebody else's computer or a computer in an Internet café, etc. some of the above type of aids may be difficult to use. This is especially true for all Java based applications or stand alone applications requiring installation. On the other hand, plain (X) HTML or (X) HTML combined with JavaScript gives too little possibilities for turning mathematical ideas into practice.

Java vs. PHP

As far as Java is concerned, the java.com website (http://www.java.com/) states the following under its homepage link What is Java:

> *Java allows you to play online games, chat with people around the world, calculate your mortgage interest, and view images in 3D. These applications, written in the Java programming language and accessible from your browser, are called "applets". Corporations also use Java applets for intranet applications and other e-business solutions.*

More details on Java can be found in (Sun Microsystems, 2007).

As far as PHP is concerned, the PHP website (http://www.php.net/) states the following on its homepage and under faq > Preface links:

> *PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. ... PHP, which stands for "PHP: Hypertext Preprocessor" is a widely-used Open Source general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Its syntax draws upon C, Java, and Perl, and is easy to learn. The main goal of the language is to allow web developers to write dynamically generated web pages quickly, but you can do much more with PHP.*

Mathematical functions are included in both Java and PHP. Unlike PHP, Java has a graphic user interface (GUI), which means that its applets can make use of 2D or even 3D graphic. Therefore, Maple or MATLAB simulations may be exported in the form of Java applets. Furthermore, visual applets such as those displaying graphs of user entered functions may be programmed easily using Java. Institute of Mathematics, FME BUT website (*MATH Online*) features a nice collection of such applets; cf. e.g. the *Graphs of functions of one variable* or *Parametrical curves* applets of Mathematics I.

While the GUI is the main advantage of Java over PHP, PHP is a technology of dynamically generated web pages, i.e. there are no applets being *embedded* into the web page. This gives the programmer greater control over page formatting and other web design issues.

Mathematical functions included in Java and PHP include the usual range of common algebraic operations including division modulo m, evaluating elementary functions, manipulating multidimensional arrays (i.e. ability to access elements of matrices), conversion between various number systems, generating random numbers as well as the knowledge of usual mathematical constants. Since Java and PHP are programming languages, all the usual features of programming languages such as cycles, if – then decisions, etc. may be used. This means that a number of even complex mathematical problems can be treated in either Java or PHP. Unfortunately, neither of the languages is capable of performing symbolic differentiation or integration – these remain the domain of professional mathematical software.

There is enough open-source or freeware software for both Java and PHP developers. For details refer to the java.com (http://www.java.com) or PHP (http://www.php.net) websites.

Examples of existing applications[1]

A number of linear algebra applications can be found on the Internet. Examples include applications at the Institute of Mathematics, FME BUT website (MATH Online), (Gauss Elimination), (Gaussova eliminace) or (Novák, M. 2007, algebra). The topics of the above mentioned applications include Gaussian elimination, performing addition and multiplication of matrices and counting determinants.

The calculus applications found on the Internet are mainly applications visualising the concepts of limit, derivative and integral, such as applets available at (Laval, n.d.) web pages. Some applications attempt at computing symbolic derivatives – such as (Novák, 2007, calculus), which is a PHP based application – or integrals – such as (Kříž & Šrot, 2003), which is in fact a Maple interface.

Other university mathematics applications mostly deal with numerical methods. This is natural given the algorithmic nature of numerical computations. Examples can be found at the Institute of Mathematics, FME BUT website (MATH Online).

Programming one's own application

Following applications have been programmed in PHP by the author, who is not a professional programmer: (Novák, 2007, June), (Novák, 2007, July), (Novák, 2008, January), (Novák, 2007,

1 The below selection includes applications with either English interface or applications programmed by Czech authors. This has been motivated by the language of this article and the author's origin only. Naturally, there exists a number of applications in many non-English languages.

PROBLEMS
OF EDUCATION
IN THE 21st CENTURY
Volume 5, 2008

February). Their topics are included in almost any standard algebra and calculus university courses anywhere in E.U. Sources of information needed for programming included the PHP website (http://www.php.net), which has been localized into a number of languages, (Castagnetto, 2004), (Holzer, 2003), (Staníček, 2003) – which are all either English books translated into a number of languages or books which have equivalents in almost any language, and (HTML 4.01), (XHTML 1.0) specifications and the *Markup Validation Service* of the World Wide Web Consortium available at http://validator.w3.org/. It is to be noted that no other specific sources of information are needed even if a person is new to both PHP and accompanying technologies such as (X) HTML or JavaScript.

Depicted below are screenshots of two of the applications.[2] In both cases the screen is divided into frames, most of which are blank pages by default. As far as (Novák, 2008, February) is concerned (see Figure 1), the user starts with Matrix A and Matrix B fieldsets. Once the Generate button has been pressed, a form in which coefficients can be written is unfolded, then the respective matrix is displayed as seen in Figure 1. When both matrices have been entered, the buttons of algebraic operations are displayed; when a given active button is pressed, the result is displayed. If the user leaves a mouse cursor on the element of the resulting matrix (62 in Figure 1), the way of computing the element is highlighted.

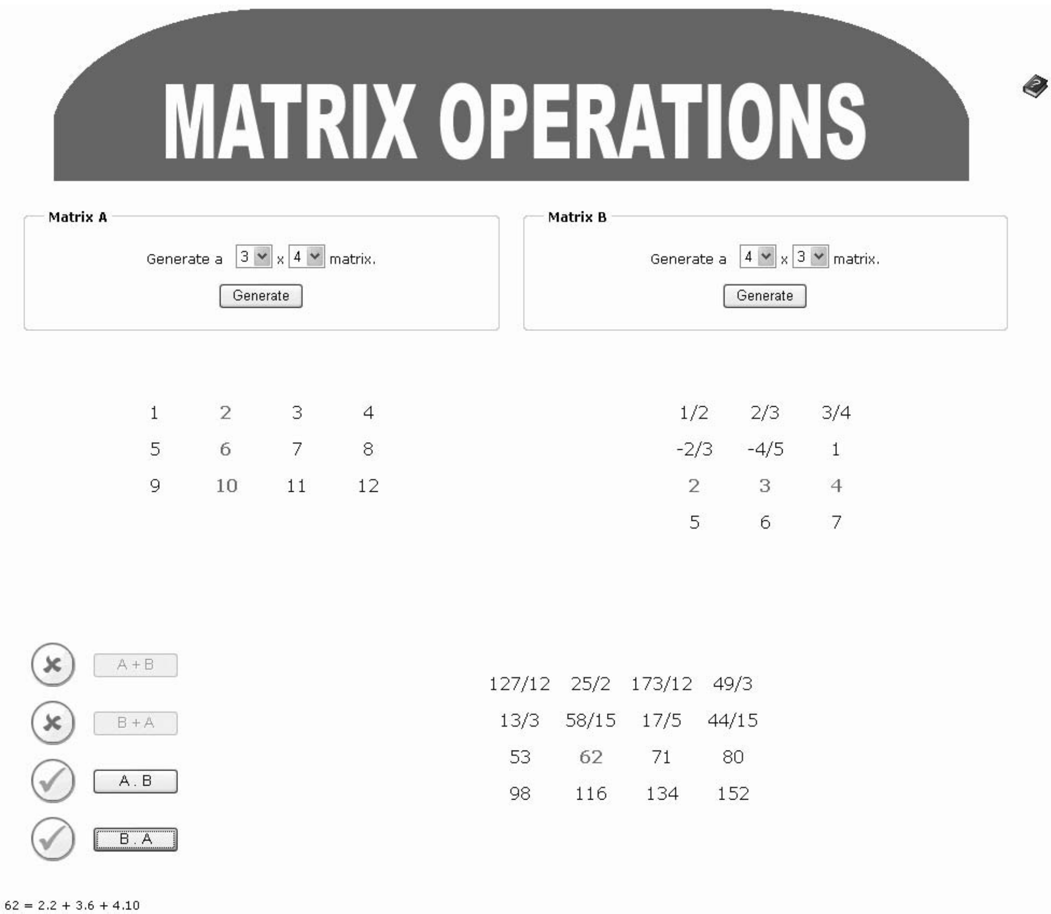

Figure 1. Screenshot of the (Novák, 2008, February) application.

The process of computing the determinant in (Novák, 2008, January) is similar. First a matrix is generated then coefficients can be entered. Further – based on the matrix order – either direct computation can be performed or the user is prompted to choose lines or columns for the Laplace

2 English versions of the applications have been depicted for publication. However, preparing multiple language versions of a single application is rather simple in PHP.

expansion. In Figure 2 the determinant of a 4 x 4 matrix is computed, i.e. Laplace expansion is simple. On contrary, Figure 3 shows the application response if Laplace expansion chosen by the user resulted in laborious computation.

On-line help (notice the book icon) as well as context help (e.g. notice the line explaining the meaning of 62 in Figure 1) are included in both of the applications. The combination of PHP and JavaScript and CSS makes formatting and dynamical highlighting consistent while easy to implement.

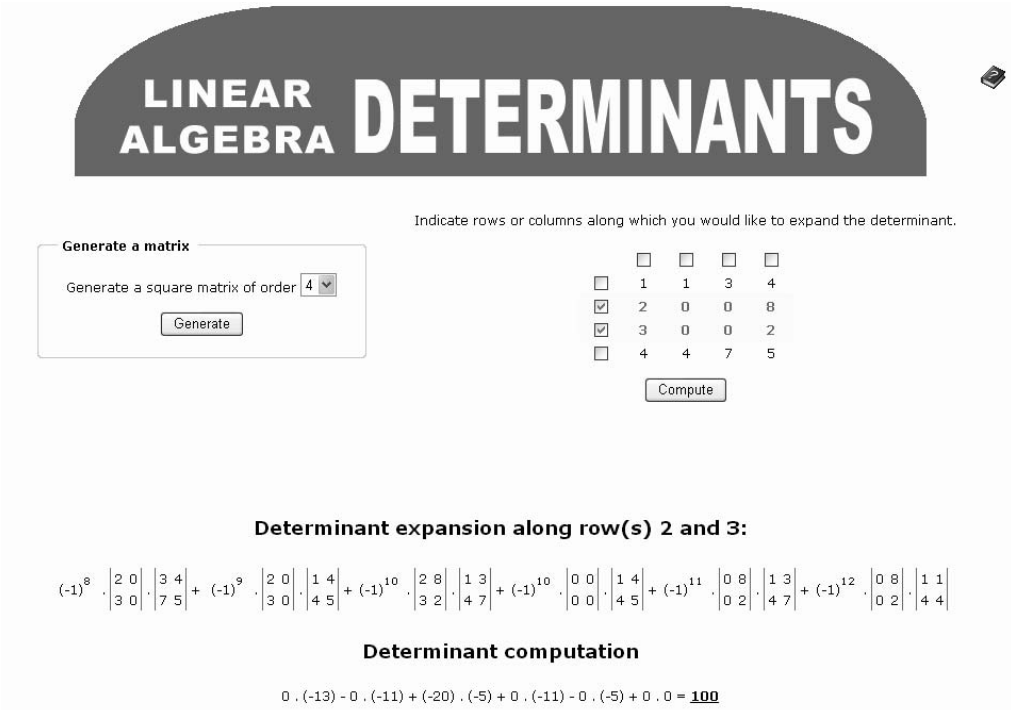

Figure 2. Screenshot of the (Novák, 2008, January) application – cropped.

1

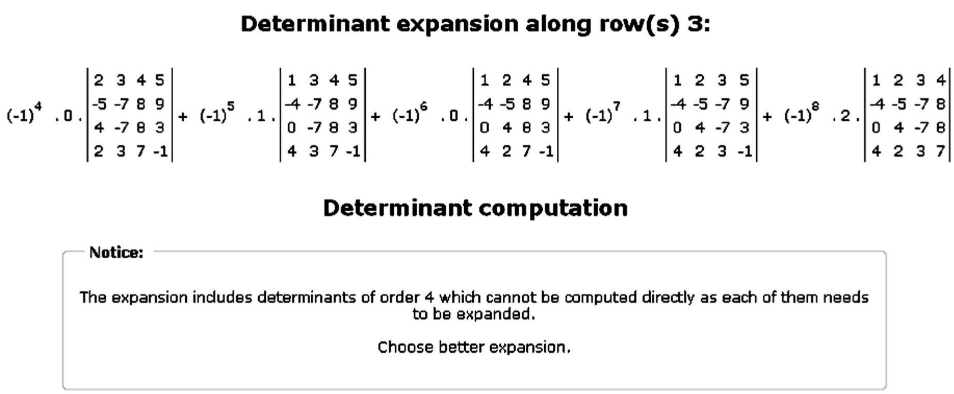

Figure 3. The user attempted to expand determinant of order 6 along one row only.

PROBLEMS
OF EDUCATION
IN THE 21st CENTURY
Volume 5, 2008

Both PHP and Java are languages powerful enough to handle a number of tasks, teaching aids for which a teacher is likely to need. Moreover, there is one great advantage in programming one's own teaching aids – the author may have them exactly the way he / she wishes (provided the task is performable with the available tools). (Ma-tri-ca.narod.ru, n.d.) may have felt that working with generally rectangular matrices is not necessary. Similarly, the author of (*Gauss Elimination*) might have believed that there is no didactic need in performing the Gaussian elimination of a rectangular matrice or that displaying the result only is sufficient. On the other hand, when programming (Novák, 2007, July) I focused on the *process* of practicing the Gaussian elimination algorithm. The similar holds for (Ma-tri-ca.narod.ru, n.d.) vs. (Novák, 2008, January) and computing determinants.

Furthermore, the author has full control of the visual aspect of the teaching aid. Compare the visual side of all the three above-mentioned applications, which deal with similar topics.

Conclusion

Programming one's own stand alone application with mathematical content is not as difficult as it seems. Once the topic is decided, there are at least two technologies that may be used: Java and PHP. A variety of mathematical concepts and skills can be implemented into Java or PHP based applications. In this way the teacher can communicate the didactic content and message the way he / she intends. Furthermore, unlike with professional software with English-only interface, there is no language barrier when using such applications.

References

Castagnetto, J. & al. (2004). *Programujeme PHP profesionálně.* [translated from *Professional PHP Programming.*] Brno: Computer Press.

Gauss Elimination. Retrieved March 5, 2007, from http://www25.brinkster.com/denshade/GaussElimination.html

Gaussova Eliminace. Retrieved March 5, 2007, from http://e-learning.tul.cz/elearning/obr/system/matlab/246/583/gauss2_vstupni.html

Holzer, S. (2003). *JavaScript profesionálně.* Brno: Mobil Media, a.s.

Institute of Mathematics, FME BUT (2007): *MATH online.* Retrieved March 5, 2007, from http://mathonline.fme.vutbr.cz/COURSES/sc-1002-sr-2-a-231/default.aspx

Kříž, P. & Šrot, K. (2003). *Integrace s programem Maple.* [En: *Integrating with Maple.*] Retrieved March 5, 2007, from http://cgi.math.muni.cz/~xsrot/int/uvod.cgi?cnt=yes

Langerová, P. (2006). Mathematical terminology in English: its place at FEEC BUT and students' knowledge. In *Proceedings of 5. matematický workshop s mezinárodní účastí.* Brno: VUT v Brně. [CD-ROM].

Laval, P. B. (n.d.). *Testing Java Applets.* Retrieved March 5, 2007, from http://science.kennesaw.edu/~plaval/tools/index.html

Ma-tri-ca.narod.ru (n.d.): *The Matrix Calculator.* Retrieved March 5, 2007, from http://matri-tri-ca.narod.ru/en.index.html#1

Novák, M. (2007, June). *Basics of differential calculus: derivatives.* Retrieved March 5, 2007 from http://www.umat.feec.vutbr.cz/~novakm/derivace_en/

Novák, M. (2008, January). *Linear algebra: matrix operations.* Retrieved March 5, 2007 from http://www.umat.feec.vutbr.cz/~novakm/determinanty/en/

Novák, M. (2008, February). *Linear algebra: matrix operations.* Retrieved March 5, 2007 from http://www.umat.feec.vutbr.cz/~novakm/algebra_matic/en/

Novák, M. (2007, July). *Lineární algebra: práce s maticemi.* [En: *Linear algebra: matrices.*] Retrieved March 5, 2007 from http://www.umat.feec.vutbr.cz/~novakm/linearni_algebra/

Novák, M. (2007). *Matematický slovník jako pomocný studijní materiál.* [En: *Mathematical dictionary as a supplementary teaching aid.*] Brno: Masarykova Univerzita. Přírodovědecká fakulta. (RNDr. Thesis).

Novák, M. & Langerová, P. (2007) *English-Czech / Czech-English dictionary of mathematical terminology.*

Retrieved March 5, 2007 from http://www.umat.feec.vutbr.cz/~novakm/slovnik_matematicke_terminologie/index_en.html

Novák, M. & Langerová, P. (2006) Raising efficiency in teaching mathematics in non-English speaking countries: an electronic bilingual dictionary of mathematical terminology. In: *Proceedings of 3rd International Conference on the Teaching of Mathematics at the Undergraduate Level.* Istanbul: TMD (Turkish Mathematical Society), 2006. [CD-ROM].

Sun Microsystems (2007): *Learn About Java Technology.* Retrieved March 5, 2007, from http://www.java.com/en/about/

Staníček, P. (2003): *CSS: Kaskádové styly. Kompletní průvodce*. [En: *CSS: Cascading style sheets. Complete guide.*] Brno: Computer Press.

Adviced by Helena Durnova,
Brno University of Technology, Czech Republic

Michal Novák Assistant teacher at Faculty of Electrical Engineering and Communication, Brno University of Technology.
Technická 8, 616 00 Brno, Czech Republic.
Phone: +420-541143135
E-mail: novakm@feec.vutbr.cz
Website: http://www.umat.feec.vutbr.cz/~novakm/