A WEB-BASED E-LEARNING SYSTEM FOR TEACHING MATHEMATICS AND SIMULATION WITH MATLAB

G. Zauner[1,2], G. Schneckenreither[2], F. Judex[2], F. Breitenecker[2]

[1]"Die Drahtwarenhandlung"- Simulation Services, Vienna, Austria;
[2]Vienna University of Technology, Institute for Analysis and Scientific Computing

Corresponding Author: G. Zauner, "Die Drahtwarenhandlung"- Simulation Services, Vienna, Austria, Neustiftgasse 57-59, 1070 Wien, Austria; guenther.zauner@drahtwarenhandlung.at

Abstract. The work presented in this paper deals with the development of a web-based e-learning system for teaching engineers in technical and applied mathematics as well as modelling and simulation of continuous and hybrid systems. The first task and starting point of this system was the discussion, how to combine teaching in mathematics and applications with learning an adequate widespread programming language.
The decision was then to use MATLAB because of its ability to solve algebraic problems as well as symbolic computation by using the *symbolic math toolbox* and also offering the feature of an interface for web server applications. The user interface is developed in PHP and has a modular structure for fast example implementation and good usability. The structure of the implementation and surrounding on Vienna University of Technology are explained in detail.

1 Motivation

Since 2006 the Vienna University of Technology offers a Moodle [7] based e-learning platform called TUWEL in combination with the standard web based lecture administration. This tool can be used for general organization but has not enough features for implementation of dynamical examples. That is why the department for analysis and scientific computing decided to implement an e-learning system for mathematicians and engineers in the year 2006. After a summary of the state of technology we decided to use the web server environment included in MATLAB 2006a. This decision was made because MATLAB [1, 6] is not only a computer algebra package, but has additional tools for symbolic computation and it is wide spread in industry and research.

An additional reason for the MATLAB way of a web interface was that the students should also learn programming their own code. For all of the implemented examples, the students can look at the solution in MATLAB, download the code and work with it on their own PCs.

2 Background of MATLAB Web Server

In general, as the examples are all realized in the MATLAB Release 2006a, the MATLAB web server application [4, 6] needs an additional web server to act as server in client/server web architecture. In our case we use an Apache web server [5]. The interface used for input or input/output representation is defined by standard HTML (Hypertext Markup Language) [2] frames, which interact with MATLAB via a CGI – script as shown in Figure 1.

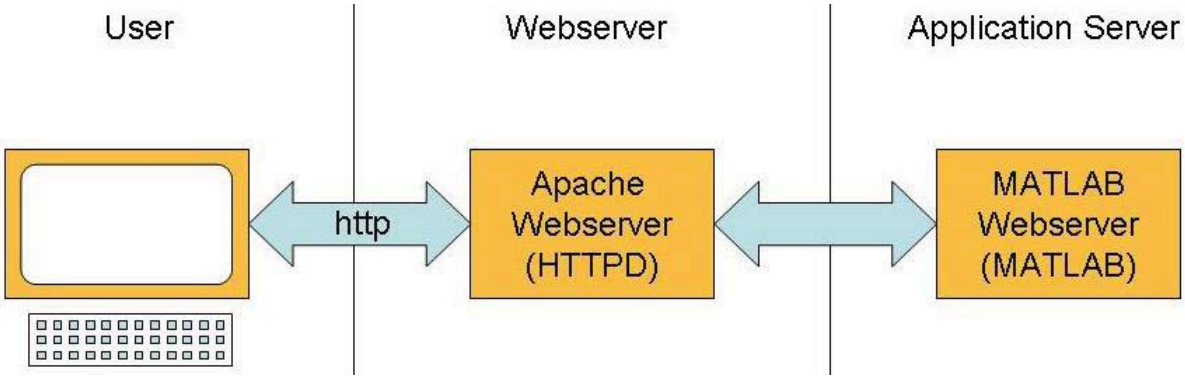

Figure 1 Architecture of the client ↔ MATLAB web server communication

In our case we decided to use PHP and interconnect the files directly with the web server. This has the following benefits:

- The system becomes more stable, because we have only two layers left instead of three.
- After defining the structure once, the whole system acts in modular concepts, which means that we can adapt examples and add new ones, without any code writing in PHP or HTML.

To get a general reusable system we have to define a global concept, capable to support the most important features for education in mathematics, modelling and simulation. Thus, we will get some restrictions in graphical representation and/or textual representation, but on the other hand a well defined structure supports easy model implementation and illustrating special content of teaching, which is not that easy explained in the common way at a blackboard [3].

3 The System Parts

To get an usable e-learning system not only the architecture of the solution system has to be planned, also the concept has to fit into the already existent tools at university and the input output framework and the system where the students get additional information, can log in and the lecturers can administer the users have to be designed.
At the here presented system this splits into the following parts:

- additional documentation and user administration,
- MATLAB implementation of the examples, and
- the PHP Interface.

3.1 Additional Documentation and User Administration

In the year 2006 Vienna University of Technology started the handling of the new e-learning platform TUWEL (Technische Universität Wien E-Learning), which is based on Moodle, an open source project. The system is using the same authentification system as TUWIS++ (Technische Universität Wien Informationssystem), the main electronic system of Vienna University of Technology. The new system has gateways to the electronic lecture administration and thereby can import student date of persons who are announced at the lesson as well as the possibility for data and mark export. This is done to anticipate parallel system structures. The structure of TUWEL (Figure 2) is quite intuitive and thereby students do not need extra explanation to get used to the important features.

TUWEL as designated e-learning platform has the problem that because of its range of usage the system does not fit that good to the technical terms needed for developing a numerical based mathematical e-learning platform. Furthermore user administration, document management for all included lectures, forums and calculation of dynamical systems with a computer algebra package on one system with up to 15000 users are not practicable. That is why we started a search for the state of the art and decided to use a MATLAB system. The lecture dependent general information, user administration, forums for the students and communication with the instructor as well as further information is organized in TUWEL. Each lecture has its own closed part where the students get all the information, can upload their solutions of extra examples. The instructors evaluate the work and give marks. The lecture surrounding an the MATLAB server are physically separated.

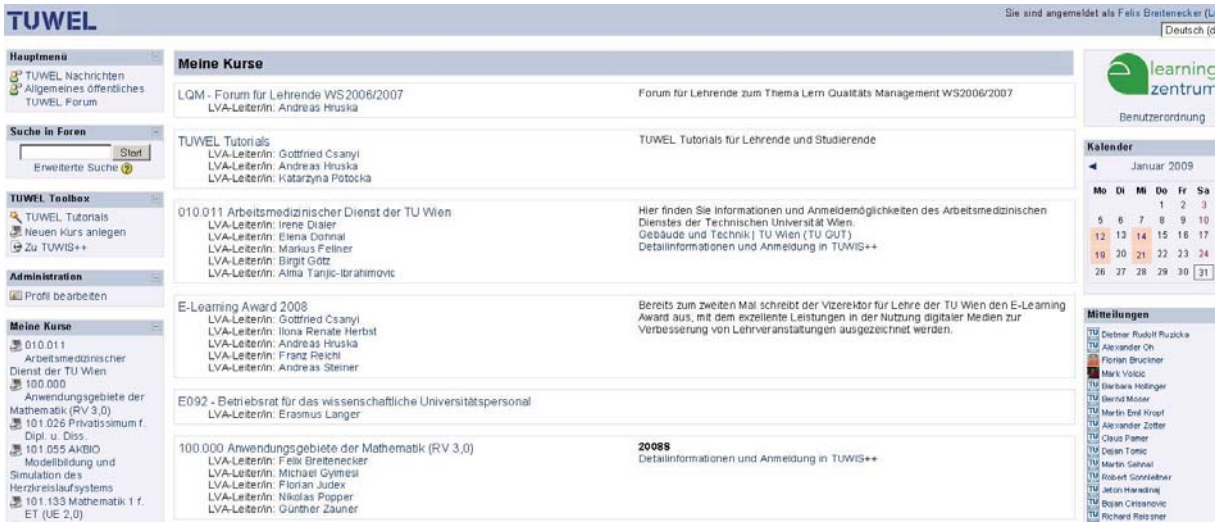

Figure 2 The main page of the TUWEL system. In the middle part all the courses a student is announced to are listed.

3.2 MATLAB Implementation of Examples

Source code of all (MATLAB) implementations is aimed to be consistent throughout the e-learning system. However, some examples involve ODEs, zero-crossing detection or SIMULINK models while others don't. Generally, graphical output of an application must be rendered to an image in order to be displayed on the user's web-browser and ODE-solvers require the right-hand side of the equation to be defined as a (separate) function are listed at the end of the m –file as an extra function, no further m – files are allowed.

Otherwise we could not guarantee that the whole source code is listed for the students. The basic skeleton of a MATLAB program on our remote teaching system is shown in Table 1.

function handler	`function retstr = example_001(instruct)`

transformation of input variables	`%% SERVER-SPECIFIC CODE %%`
	`a = str2num(instruct.var1);`
	`b = str2num(instruct.var2);`
	`%% END OF SERVER-SPECIFIC CODE %%`

calculations/program	`c = a*b;` `ode45(odefunction,...);`
or call to SIMULINK model	`sim simulinkmodel;`

pre-plotting commands	`%% SERVER-SPECIFIC CODE %%`
	`Pic = figure('visible','off');`
	`%% END OF SERVER-SPECIFIC CODE %%`

specific plotting commands	`...`

finalising plotting commands	`%% SERVER-SPECIFIC CODE %%`
	`wsprintjpeg(Pic,instruct.mlimgfilename);`
	`retstr = 'IMAGE';`

or switching to textual output	`retstr = 'TEXT';` `%% END OF SERVER-SPECIFIC CODE %%`
right side of ODE function	`out = odefunction(a,b,c)`

or additional functions	`...`
	`end % of function odefunction`
	`end % of function example_001`

Table 1 This table shows the source code of a MATLAB file stored on the server. Input values are passed through the web interface and as output a browser ready *.jpeg image or a text string is created.

The sole difference to a normal MATLAB program concerns modification of plotting commands and extraction of input parameters from a special input object. As a consequence, our programs can easily be used in a standard stand alone version of MATLAB and the source code of the exercise is readable all the time.

3.3 The PHP Interface

Unlike individual examples for a web application that are performed to show one special effect the goal was to define a surface that can deal with a lot of applications in an adequate way. To get the same format for all examples we decision was made to split into several files. So we got a well defined register for each chapter and including this structure an example list, again organized in several input files.

The single examples consist of a file including the example headline, one file with a general description in HTML style and a file with the parameter interface and the name of the *.m – file. The structure of this file is listed in detail in Table 2.

One level higher the so called chapters are organized. The structure can be seen in Figure 3. Chapters include a list of examples that can be chosen by left mouse click out of a list in the upper right part of the interface. Furthermore it includes the list of the valid users. This list consists of users that are organized on the top level of the system. The data from this list is organized using data from the TUWEL interface. In addition the chapter structure includes a global headline and a file including the general information text. These texts are also formatted using basic HTML commands. The size, colour and alignment is predefined in the PHP script and assures that the system looks the same in all applications.

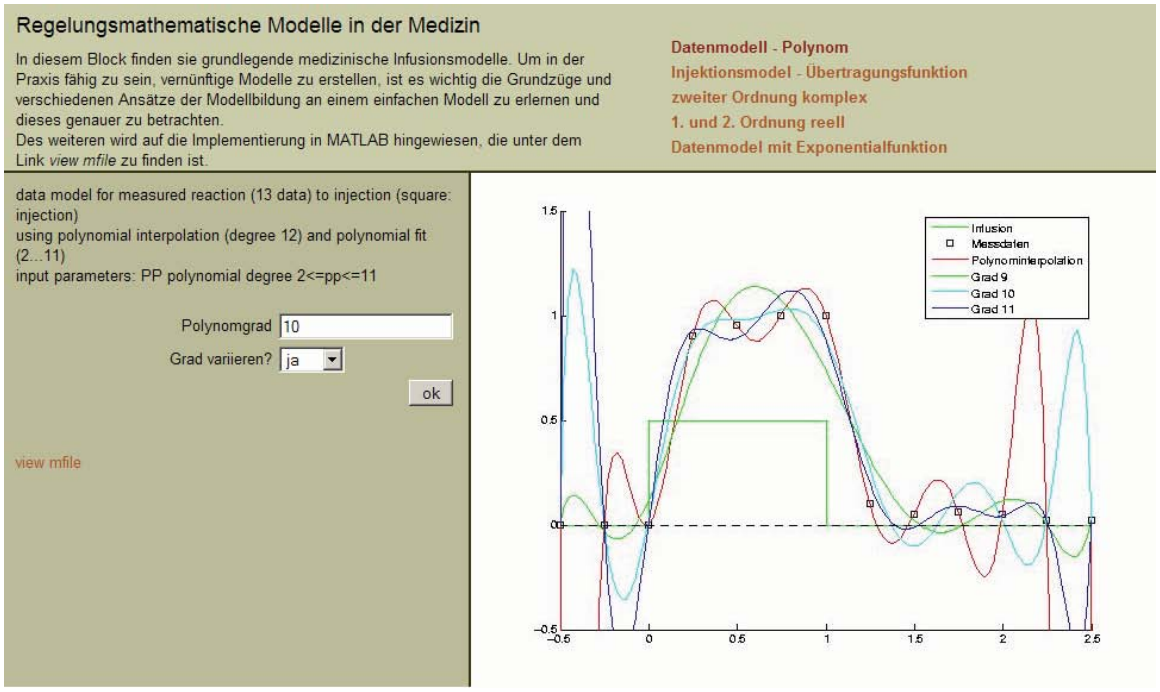

Figure 3 Screenshot showing the structure of the web interface

As mentioned before one part of standardization used in our web interface is the input mask. In an extra file the PHP code for defining the appropriate *.m – file, the parameter list, the explanation text on the web interface, the type and default value of each input parameter as well as the default values and in case of numerical values the boundaries.

The definition of the boundaries in this interface makes the standard MATLAB files shorter, easier readable und thereby the readability of the code increases. This helps the students to focus on the important parts of the implementation and guides them to get the knowledge for implementing akin examples as homework or project together with another student.

The following types of input variables are implemented:

- Integer variable
- Floating point number
- string input and
- combo box.

The usage of integers and floating point numbers is clear. Strings are in our basic analysis and linear algebra examples normally used to define vector or matrix inputs for the MATLAB interface. Another application is to get MATLAB commands or functions as input in MATLAB style. These functions are in most cases used as basic function for symbolic operations like integration, differentiation or limit value calculations.

The way of implementation of the parameter interface for all four different types is shown in Table 2 below. As can be seen, the input structure is very clear and only few lines of code are necessary.

opening tag for PHP	`<?php`
defining the main *.m - file	`$pageVars['ml_mfile']='example_001';`
	`$form = new Form();`
source for a combo box with four choices and the variable name: “parameter to optimize:”	`$form->addField(new ComboField('var1', 'paramter to optimize:, 1,array(1 =>'TI',2 =>'KI',3=>'TB',4 =>'KB')));`
source for a floating point number called “starting value:” with default value 0.1 and the boundaries 0.0001 and 60	`$form->addField(new FloatField('var2', 'starting value:', 0.1, 0.0001, 60));`
defining an integer field with name “Number	`$form->addField(new IntegerField('var3', 'Number`

of steps:" and the same syntax for boundaries as the float field	`of steps:', 5, 1, 30));`
syntax for a string input; only a default value is necessary	`$form->addField(new TextField('var4,'Answer:', 'Optim.'));`
	`$pageVars['form'] = &$form;`
closing tag	`?>`

Table 2 Structure of the PHP interface for input variable definitions, including the coupled *.m – file, the variable names, their default values and the boundaries.

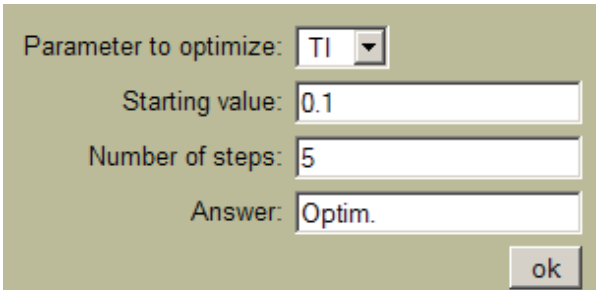

Figure 2 Parameter interface as defined in the PHP code in Table 2

The source code in Table 2 above results in the following parameter interface (Figure 2) and shows all four types of input variables implemented in the system. The main benefits of this structure are that the lecturers can implement additional variables fast and that the type check and the check for boundary conditions are outside MATLAB which leads to a better readable code. Another positive task is that the interface has always the same style.

4 Reorganization of Lectures

The advanced modelling and simulation lectures can now, after introducing the MATLAB web server system be reorganized. The disunion of theory and practical part are more and more ceased because the time needed to show a n example implementation on the web system takes only a fraction of time comparing to the way they where used before.
Also the presenting of the examples gets easier, because of the structured model and data storage using TUWEL and the easy connection. No laptop with special software is needed any more in the seminar rooms to show the behaviour of a dynamical system, only a web browser is necessary. All the models are implemented in some kind of virtual laboratory, where all students can work with the models at any time of the week.

	students	
	local	web based
experiment: standard	attendance lab	remote laboratory
experiment: web based	local simulation	virtual laboratory

Table 3 local versus web based simulation laboratory

The main benefits of so called virtual laboratories are the higher flexibility for the students as well as for trainers. Another positive aspect is the multi user actuation and the simple maintenance of the examples on the server. This aspects lead to an economization in resources for the normal working process.

Nevertheless we have to keep in mind that the starting investigations are quite high and additional features coming up with better technologies every year are time consuming during implementation and planning of the new concept for fully developed e-learning strategies. To sum this arguments up we see, that the workflow is shifted from normal presenting to concept planning and model implementation – the work does not decrease, but the quality of the apprenticeship labs can increase significantly.

5 Conclusion and Outlook

Since starting the e-learning project with MATLAB web server at Vienna University of Technology in summer 2006 about 320 web server based examples dealing with mathematics, numerical algorithms and simulation organized in 25 chapters have been implemented.

They are now part of the normal lecture structure especially in mathematics for surveying and mapping as well as in the advanced lectures in modelling and simulation of dynamical and structural dynamical systems. The MATLAB web server is used to explain numerical problems, computer numeric, as well as how to implement dynamical problems (especially the ARGESIM Comparisons, mechanical problems and dynamical physiological problems) and SIMULINK examples.

The main drawback of the implemented system is the restriction of output features which come together with calculation only on server side. In the current work this problem is solved by calculation of MATLAB movies for interesting parameter sets and transforming this MATLAB internal notation into a web animation. Because of calculation time this can not be done on demand for arbitrary parameter sets. Another drawback is the instability of the MATLAB web server by testing code including errors. Errors can take the whole web server down. This is the reason why new examples are always tested on workstation before porting it to the server. Until now only two plot commands regarding 3D options could be found that work on the single workstation but not on the MATLAB web server and therefore it is no real problem.

Besides the drawbacks the benefits:

- always the same style and thereby easy handling for the user,
- only a browser at client side necessary, no JAVA applets or something similar needed,
- the students are forced to learn MATLAB by “playing” with the system, which leads to better learning results,
- fast implementation of already given MATLAB source code into a standardized system,
- due to license agreements learning with a tool the students can get for cheap.

overbalance.

The development of the system is still going on. In the moment the workgroup of modeling and simulation is porting the system from the MATLAB web server 2006a version to a normal MATLAB engine. This is done because *Mathworks*, the developing company of MATLAB, does not support the web server any more. An additional benefit of the new architecture is that more instances of MATLAB can work in parallel (one per core), which leads to better performance. Although a content management system will be implemented for example management and to get more options regarding language of the explanation texts, parameter names and the composition of examples in the chapters.

6 References

[1] F. Breitenecker, N. Popper, G. Zauner:
"Structural Features in Simulation Systems - Evolution and Comparison";
Proc. 20th European Modeling and Simulation Symposium EMSS2008, A. Bruzzone, F. Longo, M. Piera, R. Aguilar, C. Frydman (Hrg.); Univ. Calabria, (2008), ISBN: 978-88-903724-0-7; S. 399 - 407.

[2] M. Lubkowitz:
“Webseiten programmieren und gestalten – HTML, CSS, JavaScript, PHP, Perl, MySQL, SVG.“
Galileo Press, Bonn 2003, ISBN-10: 3898423131.

[3] G. Schneckenreither, F. Breitenecker, N. Popper, G. Zauner (2006):
„Cellular Automata for SIR-type Epidemics“

[4] G. Zauner, N. Popper, F. Breitenecker:
"A PHP/MATLAB based E-Learning System for Education in Engineering Mathematics and in Modelling and Simulation"; "Proc. EUROSIM 2007 - 6th EUROSIM Congress on Modelling and Simulation", B. Zupancic, R. Karba, S. Blazic (Hrg.); ARGESIM / ASIM, Vienna (2007), ISBN: 978-3-901608-32-2;

[5] http://httpd.apache.org/h

[6] http://www.mathworks.com/

[7] http://www.moodle.org/