

Teaching Mathematics in University Education through Internet

Javier BILBAO, Eugenio BRAVO, Purificación GONZÁLEZ,
Elena MARTÍNEZ

*Department Applied Mathematics, School of Engineering, University of the Basque Country
Alda.Urquijo, s/n 48013 – Bilbao
e-mail: {mapbilaj, mapbrsee, mapgosap}@bi.ehu.es*

Received: July 2003

Abstract. A project is being developed with the main goal of creating an application of web server which allows students to use Mathematica software from their computers without the necessity of installing this software and of programming the necessary algorithms. The project consists of a set of mathematical models programmed with Mathematica and stored in a web server, which has been installed in a computer of our department. At the moment, it is possible to access to the web pages of the application from any computer of the intranet of our School.

Key words: e-learning, mathematica, server platform.

1. Introduction

The growth of the web technologies and the simplicity to manage the so called user-friendly interface makes these technologies widely used, for example in the e-learning field. As a result of this growth, the need to supply computational capacity to the web pages has come up.

Nowadays, a wide range of software tools to face up that necessity has arisen. Among these, we chose webMathematica. It is a tool that allows us to include iterative calculations, visualizations and graphics in a web page by means of the integration of Mathematica software and web server technology.

The main reason for this choice is because we teach mathematics in a School of Engineers with various math subjects with more than 1 500 students (some of them travel for more than an hour from their homes to the university). So, despite having various laboratories where students can use computers with math software (as Mathematica), occasionally there are not enough computers for all the students. Besides, almost all students have a computer with Internet connection in their houses. Then, we made a survey asking the possibility and agreement of studying part of the subjects through Internet. The answer was mainly in favor of the proposal. Therefore we have set a project in motion to teach mathematics through Internet (Chalmers, 2002).

One of the aspects that we can emphasize of the web technology, and consequently one of the reasons to develop this project, lies in the fact that the use of both webMa-

themática and Mathematica is transparent for the user, to such an extent that it is not necessary that the user has any knowledge of these tools. Indeed, you do not even need Mathematica software.

We try that students can get all the computational power of a math software with the simple interface supplied by the web technologies. Obviously, for this proposal we use web servers and the software that will be mentioned later.

The main goal of this project is to create a server application that allows students to use Mathematica software from any computer. In this way, they avoid installing this commercial software and they do not need to program the necessary algorithms.

By doing so, any student can have access to the calculation power, to the graphic representation and to use functions libraries through the simple interface that is supplied by a web browser. It is a didactic application and allows students to focus on the results obtained by way of the algorithms that they have to study.

The project consists of a set of mathematical models programmed with Mathematica (Wolfram, 2001) and stored in a web server, which has been installed in a computer of our department. At the moment, it is possible to access to the web pages of the application from any computer of the intranet of our School to work with solving models for non-linear equations, numeric integration, gaussian elimination and graphics in an interactive way since this application allows to vary the calculation parameters. Fig. 1 shows an screen of the application.

The integration of Mathematica and web server technology (i.e., access to the software from multiple locations) are in the agreement with Mathematica license conditions.

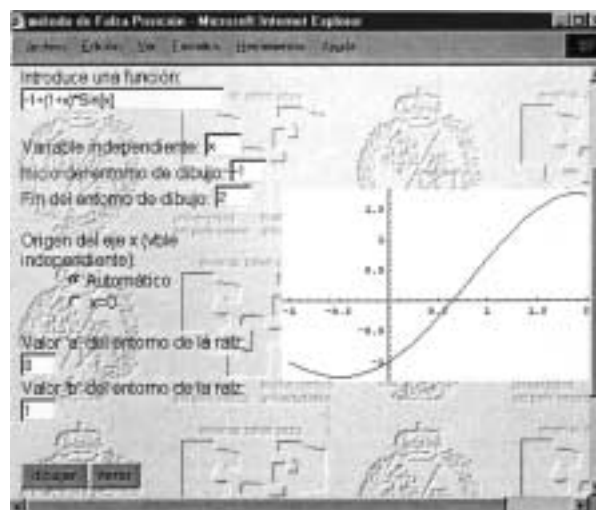


Fig. 1. Screen of application.

2. Involved Software

The involved software of the application is divided in two big groups. The necessary sets in the client's terminal will be first detailed. After that, the server platform will be described.

2.1. Client Platform

A web browser installed in a computer connected to the server is usually enough to implement the client platform, that is to say, the technology is in the same net environment.

The browsers that we can use are Internet Explorer, Netscape Explorer, Mozilla, Amaya (Quint and Vatton, 1997), etc. However, it may be necessary the installation of certain version (or superior) according to the applications of webMathematica that will be developed. In general, any version is valid for this project except in the case when MathML (Buswell *et al.*, 1999) language is used for the graphic representation of expressions and mathematical symbols without resort to images that impoverish the final result and slow down the time for downloading the web page. The problem of the use of MathML is that it is not supported by all of the browsers.

One of the browsers that supports the MathML language is Amaya. It was specifically created for the interpretation of MathML but it has a worse performance in multimedia aspects than browsers as Explorer or Netscape.

From version 0.9.9 on, Mozilla is able to supply the expected result. Nevertheless, it is not a very widely used browser among the various software platforms.

The ideal solution for us would be the use of browsers as Internet Explorer or Netscape, because it would avoid the change of the browser in the client platforms (since both browsers are the most used in that kind of environments). However, in order that the MathML can be interpreted, the version 7.0 of Netscape is needed. In the case of Internet Explorer, it is possible to install a free plug-in named IBM Techexplorer (IBM, 2003) which allows to visualize documents with MathML (even with TeX or LaTeX files).

2.2. Server Platform

This platform needs a set of software components to be installed, as stated below.

First of all, the installation of a web server (for example, Apache) is necessary. This web server takes care of receiving requests of clients by means of the http protocol and of being related to Mathematica in order to return the expected results as web page form. Some web servers do not support the servlets (an executable Java program that normally answers to a recalling from a Html page). Then, a servlet container, as Tomcat or Jrun, has to be installed. The peculiarity of these both servlet containers is that they include standalone web servers in such a way that we can use them without an external server, as Apache server. Thus, the requirements for software installation in the server platform decrease. In our particular case, the chosen solution is the installation of Tomcat (Apache, 2002) in a Windows 2000 server system in such a way that we have the services of a web server and a servlet container at the same time.

In order that the servlet container is able to access and to work with the servlets, it is necessary that the software server platform has at least the JRE (Java Runtime Environment) or even the JDK, a more complete and more powerful environment but also more complex to install.

2.2.1. *Java Runtime Environment*

The JRE is the minimum standard Java platform to execute Java-written applications. Therefore, it is necessary that servlet containers know how to translate the Java code of the servlets.

There is the chance of supplying this Java platform from the JDK package but JRE has certain advantages:

- The license of JRE allows to distribute this package with the developed software and, on the other hand, the JDK is not able to be redistributed.
- JRE is smaller than JDK and contains the necessary software to execute the application that is described so the installation and distribution of the software is much more simple.
- With Windows platforms, JRE installs automatically some software components in the default path of the operative system. So the developer does not have to worry about it.

2.2.2. *Mathematica Software Installation*

The Mathematica software installation consists of inserting the CD-ROM of Mathematica and follow consecutively the installation instructions.

An important question is the license contract. The licenses have to be updated since when they expire Mathematica does not work and, therefore, all the application stops working.

2.2.3. *J/link*

J/link (Wolfram, 2002a) is a product that integrates Mathematica and Java. It allows to call to a Java application from Mathematica in a transparent way. It also allows to access to the kernel of Mathematica from a Java program.

J/link works thanks to Mathlink, which is the protocol that sends data and commands among Mathematica and other software. Thus, J/link supplies a higher level interface than Mathlink and, therefore, this last one goes unnoticed for users and programmers.

Thanks to that peculiarity, Java turns into a simple method to write programs that work with Mathematica.

2.2.4. *Server MSP*

The server MSP is the responsible for processing requests and returning answers to the client. The installation of this component depends on the software server platform, that is to say, on the operative system, and consequently configuration parameters will have to be established.

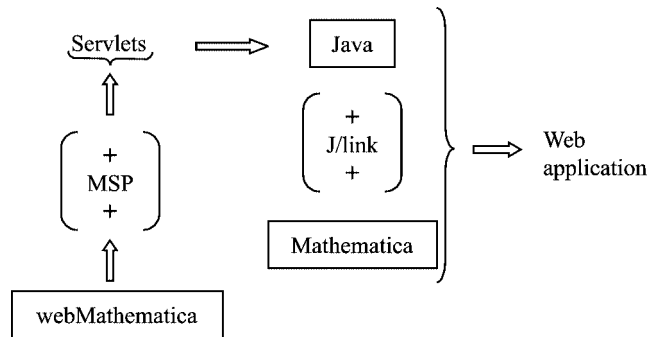


Fig. 2. Flow chart of the application.

2.2.5. WebMathematica Installation

WebMathematica (Wolfram, 2002b) is the application that supplies the programmer with a high level interface to the technology of MSP servlets. In this way, it is not necessary that the programmer establishes connections or captures of exceptions (since webMathematica realizes these operations transparently) and he/she just programs the scripts. The installation of the web application depends on the servlet container that has been installed.

Besides, webMathematica is the choice for adding interactive calculations to the web. This technology enables to create web sites that allow users to compute and visualize results directly from a web browser.

Based on the proven Java Servlet technology, webMathematica is fully compatible with Mathematica and state-of-the-art dynamic web systems (Wolfram, 2003a). Fig. 2 shows the flow chart of the application.

3. Request Processing

This operation has the next steps (Wolfram, 2003b):

- Make a request: the web browser sends a Http request to the web server. This request contains a reference to a specific MSP script, with its variables and values.
- Send the request: the web server processes the operation, for example an authentication, and sends the requests to the MSP server.
- Start of Mathematica session: The MSP server selects a Mathematica session among a set of pre-initialized sessions, in order to give service to the request. Any variable and its values are sent to this session, which takes care of loading the MSP script to execute it.
- Process page: the Mathematica session loads the MSP script and processes all labels called Mathlets. After that, the session will return the result.
- Session release: the MSP server accepts the answer of Mathematica and adds all necessary Http flags in order to return this answer to the web browser. Then, the MSP server configures the current Mathematica session as available session.

- Result return: the web server returns the answer to the web browser of the client.
- Answer receipt: the browser accepts the Http answer, which can use applets, plugins or any other characteristic. Alternatively, the answer could be formatted in MathML, TeX or Mathematica notebook.

4. Involved Technologies

4.1. *Html*

MSP technology allows a web site to contain Html pages that are associated to Mathematica commands through webMathematica. Therefore, the MSP scripts require to program in Html.

The codification that these MSP scripts have in Html takes care of the graphic and visual appearance of the document in Internet. Its technical function merely confines to the use of forms in order to send the information to the server.

Briefly, the Html language, Hyper Text Markup Language, can be defined as a language to format documents and allows any kind of computer to translate a document in the same way. It lies in a text file that contains small labels, which indicate to the browser how it can visualize the page in the client's terminal.

Besides of the graphic functions of this program language, it has another interesting functionality that is the basis of the platform client-server. This functionality is represented by the forms. The forms are the document areas where the user can introduce data or any other input (text boxes, buttons, etc.) in order to send it to the server. The forms are defined by means of the label `<form>`, which indicates the file whom the information introduced by the user will arrive. The information is passed when the user presses the button that is marked as submit (the button responsible for operating with these entries).

The integration of the Html and MSP technologies is implemented by means of the insertion in the .html file of the label `<%Mathlet expr%>` (Wolfram, 2003c). The responsible for collecting the information obtained through these labels is the MSP technology. After that, it will pass the labels to Mathematica software, which will replace each mathlet label with the result of the evaluation of its content. In this way, a new document to return to the client is made.

4.2. *MSP*

Basically, a MSP script is processed as a part of a Http transaction. The client sends a request to the server (by means of the submit buttons) and the server sends an answer that may be another Html page or any other thing as an image, Mathematica notebook or XML.

The basic component of the MSP technology is the named MSP servlet, which is the responsible for processing requests and for returning answers to the client. The servlet uses a different way for each request so it can attend different requests at the same time.

First, it determines the name of the script. This operation is permitted by the file with the web.xml configuration that establishes instructions to mapping, which redirects the requests that contain 'MSP' in the URL to the MSP servlet. In this way, the URL `http://host/webMathematica/MSP/Script` is mapped to the MSP servlet in order to find the MSP script that has to take care of generate the answers because it has the word MSP.

After locating the MSP script that has to be loaded, the servlet asks a session to the kernel of Mathematica among the set of pre-initialized sessions. This pre-initialization allows the system to work with different sessions at the same time and increases the speed of the answer. All the variables that have been sent with the request are sent to the Mathematica session with their values.

When a session is started, the kernel of Mathematica scans the script in order to locate the mathlet labels and to evaluate their content. If the result of the label processing is a string, then this string is returned to be included in the answer Html page. If it is not a string, the result is formatted with the command `MSPFormat`.

If a `MSPReturn` is evaluated, the mathlet processing finishes and the result is returned to the MSP servlet.

After the result is returned by the kernel of Mathematica, the MSP servlet calls to the kernel in order to determine the kind of the format of the answer, which may be text or images or Mathematica notebooks or another one.

If the answer is an image (or various images), they will be stored in a file and a label `` is generated. This label references the image returned as output. For example, ``. The images are stored in the default directory.

Fig. 3 shows the operation of MSP servlets.

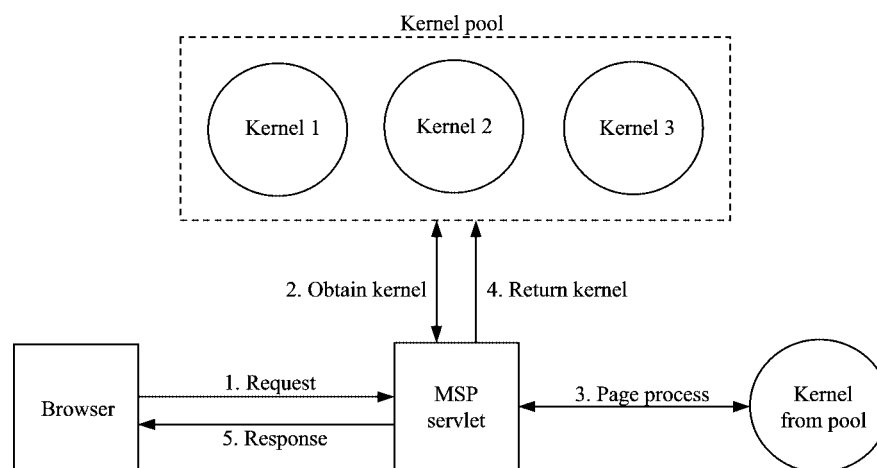


Fig. 3. Operation of MSP servlets.

4.2.1. MSP Programming

Due to the technology advance rate, the programming language being used nowadays for setting up a webMathematica application is JSP but when the project described in the article began, the only supported way was based upon MSP.

MSP technology allows an HTML page to contain Mathematica commands so the web server can interact with this application in order to offer the client the correspondent output to his request. It is not the goal of the article to enter into a full description of HTML and its characteristics as it is already well known. Nevertheless, it is needed to be taken into account that to integrate Html and MSP a specific tag is needed. It is called 'mathlet tag' and it is formatted into the following expression: `<%Mathlet expr %>`.

MSP technology will deliver the information collected by the above mentioned expression to Mathematica so it can calculate the output and substitute the mathlet tag for the correspondent answer.

As a describing example, the following lines will be attached:

```
<%Mathlet
If[{$$origen= "auto",MSPBlock[{$$input,$$ind,$$inicio,$$fin},
MSPShow[Plot[$$input,{ $$ind,$$inicio,$$fin}, PlotStyle->RGBColor[0,0,1]]],
MSPBlock[{$$input,$$ind,$$inicio,$$fin},
MSPShow[Plot[$$input,{ $$ind,$$inicio,$$fin},AxesOrigin->{$$inicio,0},
PlotStyle->RGBColor[0,0,1]]]]]
%>
```

The purpose of these lines is just to plot the function entered by the user in one of the textboxes of the form.

The mathlet tag informs the web browser not to interpret its content as a normal html tag. When one of the SUBMIT buttons in the form is pressed, the information is sent to the web server who, once it realizes the mathlet tag, sends the whole content to the MSP server, the one in charge of connecting with the Mathematica application.

The commands beginning with MSP which can be seen in the example above, are evaluated by the MSP server while the ones inserted between the branches [] will be delivered to Mathematica for its evaluation. The later will substitute the Mathematica commands by its result, once evaluated (in the server), and will deliver the output to the MSP server who will attend to the commands wrapping the Mathematica ones, to offer its own output to the web server.

Applying this explanation to the example described in the lines above, Mathematica will plot (due to the plot command) a function entered by the user in a textbox called 'input' and in the colour determined by the Mathematica parameter PlotStyle. Then, the MSP server will format that output using the MSPShow that allows graphical results to be included into an html page.

webMathematica capabilities do not stop in delivering graphical results but it can offer a wide range of solutions to change a static html into a more dynamic learning tool.

Another particularity offered by this tool is the animation of the obtained output. To make it easier to understand, a part of the program will be added here.

```

<% Mathlet
Get["Graphics`Legend"];
If[$$button=="animar", MSPBlock[{$$input,$$ind,$$inicio,$$fn},
Get["MSP"]; p1 = Plot[$$input, {$$ind,$$inicio,$$fn},
PlotStyle -> RGBColor[0, 0, 1]];
$ShowAnimationOptions = {"AnimationDisplayTime" -> 1};
MSPShowAnimation[ Table[Show[{p1, Graphics[{PointSize[0.0187],
RGBColor[1,0,0], Point[{Evaluate[$$ptos[[i]],0}]}]}], {i, 1, $$iterMaxAnim}]]],
If[$$origen=="auto", MSPBlock[{$$input,$$ind,$$inicio,$$fn},
MSPShow[Plot[$$input, {$$ind,$$inicio,$$fn},
PlotStyle->RGBColor[0, 0, 1]]],
MSPBlock[{$$input,$$ind,$$inicio,$$fn},
MSPShow[Plot[$$input, {$$ind,$$inicio,$$fn}, AxesOrigin->{$$inicio,0},
PlotStyle->RGBColor[0, 0, 1]]]]]]
%>

```

There are many commands utilized in this example that will not be commented due to the lack of space for it and not being that the goal of the article.

The main MSP command used is *MSPShowAnimation* who will receive from Mathematica a set of graphics that will be shown one after another in a rapid slide to give the user a sense of animation. The Mathematica command *Table* is the one in charge of presenting an array of objects and *Show* will turn the graphics into objects.

It may seem quite easy to program but to offer the reader a clear vision of what is being discussed about, a more complex example will be added in the following lines.

```

<% Mathlet
If[$$button=="iterar",MSPBlock[{$$input,$$ind,$$a,$$b},
OtroBuclebis[f_ ,xx_ ,aa_ ,bb_ ]:=Module[{i,a,b,error,resultado,x0,x1},
i=0;error=1;x0=b; resultado={}; a=aa; b=bb;
While[{i<111 && error>10(-6), x1=a-(f/.xx->a)*(b-a)/((f/.xx->b)-(f/.xx->a))};
If[{(f/.xx->x1)*(f/.xx->a)<0,b=x1,a=x1};error=Abs[x1-x0];
x0=x1;i++; (*we need to construct a table and return it as a result*)
AppendTo[ resultado,{i,N[x1], N[error]}]];
resultado/.r_Real ->ToString[r,InputForm]];
TableForm[OtroBuclebis[$$input,$$ind,$$a,$$b],TableSpacing->{0,2},
TableHeadings->{None,{"#","VALUE","ERROR"}}]]]
%>
<% Mathlet
If[($$button=="iterar" || $$button=="animar"),
MSPBlock[{$$input,$$ind,$$a,$$b},
TableForm[OtroBuclebis[$$input,$$ind,$$a,$$b],TableSpacing->{0,2},
TableHeadings->{None,{"#","a","b","ERROR"}}]]]
%>

```

Anyway, MSP pages functionality can be complemented by adding JavaScript programming, applets, JSP, etc. so the built up application disposes of a good dynamism and interactivity.

4.3. JavaScript

JavaScript (Netscape, 2003) is a program language, integrated in Html, useful to operate with web browsers and documents that can be open in these browsers. This language adds dynamism to the web pages since it is able to access locally, without any process of the server, to the objects that compose the web document in order to modify or work with them. JavaScript can react to events, read html and change its content, analyze data and check their worth before sending them to the server, etc.

5. Alternatives

The technologies referred to the program languages and other software elements that we have used in this application are an example of the possibilities of webMathematica. Taking into account that the aim of these web pages is the learning of Mathematics, we have used certain program languages and techniques that do not have to be the only solution to be used.

Because of this ability to adapt to different techniques and software, the webMathematica is a tool very versatile and one of the reasons to be chosen by us.

5.1. Some Examples

There is an enormous variety of technologies (Sellers, 2003) that can be used to develop this application. Some of them are:

- CGI scripts: they allow a program to execute with each request of the client.
- Active Server Pages (ASPs): it is a program language to develop the dynamic web content. It is being checked to use with the webMathematica platform.
- Plug-ins for servers.
- Perl: it is a language used inside the CGIs or with the plug-ins of servers.
- Java servlets and Java Server Pages (JSPs) (Sun, 2003): the JSPs supply to Java servlets a convenient and advisable way to return Html.
- MathML (Harris, 2002).
- JavaScript.
- Applets: they are Java programs that can be executed in an adapted browser. They are less integrated with Html than JavaScript but they are simpler to implement and can work with powerful Java technology.

6. Assessment of the Project

As it is mentioned above, it is a project in motion to teach mathematics through Internet. This idea has been very well received by the students, so the project will have continuity.

Almost all the students qualified the experience as well or very well, and the group (less than 10%) that put a regular mark to the project agreed that the major problem was kind of some obstacle in the connection to the server, specially at the beginning of the project.

7. Conclusions

A project for Mathematics e-learning is being developed by means of webMathematica. It works perfectly in the intranet of our School, so any student can connect to the server where the application is installed and learn various chapters of the math subjects. It is not necessary to install any mathematical software or program any algorithm of the Mathematica software in the computer that the student uses to e-learn.

References

- Apache Jakarta Project (2002).
<http://jakarta.apache.org/tomcat/>
- Buswell, S., *et al.* (1999). MathML specification.
<http://www.w3.org/TR/REC-MathML/>
- Chalmers, Goteborg University (2002). Mathematical Sciences.
<http://www.math.chalmers.se/~chf/sefi/abstracts.pdf>
- Harris, J. (2002). Implementing MathML in Mathematica. Wolfram Research.
<http://www.mathmlconference.org/2002/presentations/harris/>
- IBM (2003).
<http://www-3.ibm.com/software/network/techexplorer/>
- Netscape (2003).
<http://devedge.netscape.com/central/javascript/>
- Quint, V., and I. Vatton (1997). An Introduction to Amaya, W3C Note.
<http://www.w3.org/TR/NOTE-amaya>
- Sellers, D. (2003). webMathematica now includes Mathematica itself. *Bio-IT World News Magazine*.
- Sun Microsystems (2003).
<http://java.sun.com/products/jsp/>
- Wolfram Research (2001). Mathematica.
<http://www.wolfram.com/products/mathematica/index.html>
- Wolfram Research (2002a). Mathlink.
<http://www.wolfram.com/solutions/mathlink/jlink>
- Wolfram Research (2002b). webMathematica.
<http://documents.wolfram.com/webmathematica>
- Wolfram Research (2003a). Mathlink-jlink.
<http://www.outbacksoftware.com/mathematica/mathlink-jlink.html>
- Wolfram Research (2003b). webMathematica.
<http://www.wolfram.com/products/webmathematica/technology/>
- Wolfram Research (2003c). webMathematica.
http://documents.wolfram.com/webmathematica/v2/index_8_6.html

J. Bilbao obtained the degree in electrical engineering from University of the Basque Country, Spain, in 1991. At present he is PhD in applied mathematics and a professor at the Department of Applied Mathematics of that university.

E. Bravo obtained the degree in electrical engineering from University of the Basque Country, Spain, in 1991. At present he is PhD in applied mathematics and a professor at the Department of Applied Mathematics of that university.

P. González obtained the degree in electrical engineering from University of the Basque Country, Spain. At present she is PhD in applied mathematics and a professor at the Department of Applied Mathematics of that university and a head of the department.

E. Martínez is a last year student at the School of Engineering of Bilbao.

Matematikos mokymas universiteto studijose internetu

Javier BILBAO, Eugenio BRAVO, Purificación GONZÁLEZ, Elena MARTÍNEZ

Šiuo metu Baskų universitete plėtojamo projekto pagrindinis tikslas – sukurti tokią taikomąją žiniatinklio serverio programą, kuri studentams leistų savo kompiuteriuose naudotis kompiuterine programa *Mathematica* taip, kad nereikėtų nei specialiai jos įdiegti, nei specialiai programuoti reikiamų algoritmų. Nagrinėjamos tokios serverio programos ypatybės, pagrindžiami privatumai. Daugiau dėmesio skiriama pačiai projekto idėjai paaiškinti, o ne konkretiems sprendimams. Projektą sudaro žiniatinklio serveryje laikomas matematinių modelių, suprogramuotų naudojantis *Mathematica*, rinkinys. Kol kas šiuos žiniatinklio puslapius galima pamatyti naudojantis tik universiteto įstaigos vidinio tinklo kompiuteriu.