MathDox Manual

R. Verrijzer, M. Spanbroek

June 8, 2007

Contents

1 Introduction

MathDox is an XML based format for interactive mathematical documents. MathDox documents can be transformed to interactive mathematical web pages using the Mathox Player. Although MathDox can be used as an interactive information source for any topic, it is tailored for material containing interactive mathematics. But even if the interactivity that the MathDox Player offers is ignored, the MathDox tools still allow for easy publishing of scientific documents or papers on the web. Any MathDox document will benefit from the ease with which mathematical formulae can be rendered in web browsers, but MathDox really shows its potential where it concerns demonstrating the workings of an algorithm, testing readers' skills with exercises or explaining new concepts with dynamic, on-screen, calculations. Interactivity offers the reader of the MathDox document the possibility to test and experience the document.

MathDox uses OpenMath for semantic representation of mathematics and allows the use of programming constructs and web-services to their interactive potential.

The MathDox Player is freely available under the LGPL[28] license. The software packages that are used by the MathDox Player are also available under an open source license. The MathDox Player can be downloaded at http://mathdox.org.

This manual is built up as follows. The next section explains how to install the MathDox Player. Section 3 explains the technical details of the MathDox format and explains how the MathDox Player works. The last section contains a small tutorial on authoring a MathDox document, and provides a number of examples of MathDox tools in use.

2 Installing the MathDox Player

In this section the download, compilation and deployment process of the MathDox Player is described. First the prerequisites are discussed followed by download instructions, configuration, compilation, deployment and information about OpenMath Phrasebooks.

2.1 Prerequisites

In order to run the MathDox Player a few software packages need to be installed first. In this section all packages are mentioned and if necessary extra remarks concerning installation are made. One should have the third-party software installed prior to installing the MathDox Player, if it is not stated as optional.

- Java version 1.5 (5.0). Follow instructions at http://java.sun.com, make sure also to configure the Java plug-in in your browser.
- Apache Tomcat version 5.5. (JBoss or any other Java Servlet container will work as well.) Follow instructions at http://www.tomcat.apache.org.
- Apache Ant. Download and install version 1.6.5 or higher from http://ant.apache.org.
- eXist 1.0.1. This package is optional. By default the filesystem is used as source location for MathDox documents. It is however possible to store the documents in a XML database like eXist and use the benefits a database has to offer. Download eXist from http://exist.sourceforge.net. Follow the instructions to obtain a exist.war file and copy this file to the `webapps` directory of your Tomcat installation, or the deployment directory of another Java Servlet container of choice. It is also possible to run exist outside a Java container. Refer to the eXist documentation for more information on the package or on its configuration.
- A Computer Algebra System, with an OpenMath Phrasebook and its accompanying service program. These are not needed but highly recommended when one wants to perform (more complex) computations. Follow the instructions given in section 2.5.

2.2 Download

Two versions of the MathDox Player are available for download. The latest stable release can be found at `http://mathdox.org/player`. After downloading the zip file, the file should be unzipped. Alternatively, the latest development version of the MathDox Player can be checked out from subversion. One first needs to install the version control utility Subversion (svn)[26]. Using the command line tool svn in Linux or Windows, checking out the needed package is done as follows.

```
svn checkout https://mathdox.org/svn/repos/public/mathdoxplayer
```

No username or password is required. Configuration of the MathDox Player is only needed if the default configuration parameters are not suitable. Follow the instructions in the next section to see how to do this.

2.3 Configuration

The MathDox Player has some settings that can be changed. These settings can be found in the `properties.xml` file located in the configuration folder. Each setting (or property) has the following form:

```
<property
  name='org.mathdox.storage.url'
  value='oxf:/player/manual'
  as='xs:string'
/>
```

The following properties can be configured:

- org.mathdox.storage.url Location of MathDox documents, this points by default to the manual that is included with the MathDox player.
- org.mathdox.phrasebook.default Default OpenMath Phrasebook used to send OpenMath queries too.
- org.mathdox.phrasebook.mathematica.host Location of Mathematica only needed when Mathematica is used.
- org.mathdox.phrasebook.mathematica.port The port where Mathematica can be reached.

- org.mathdox.phrasebook.maxima.host Location of Maxima, only needed when Maxima is used.
- org.mathdox.phrasebook.maxima.port The port where Maxima can be reached.
- org.mathdox.phrasebook.gap.host Location of GAP, only needed when GAP is used.
- org.mathdox.phrasebook.gap.port The port where GAP can be reached.
- org.mathdox.phrasebook.wiris.endpoint The location where the WIRIS OpenMath Phrasebook can be reached.

2.4 Compilation and deployment

At this point it is assumed that you have either a development version from Subversion (see section 2.2) or the zip file and this file has been unzipped. It is also assumed that any changes to the settings have been done. The next step is to compile the MathDox Player and deploy it in a Java Servlet container. This can be done with the help of the build tool Ant, by means of the following steps:

- Type `ant war`, at the top level of the MathDox Player directory. Keep in mind that building a war file may take several minutes to finish. Afterwards the mathdoxplayer.war file can be found in the `build/war` directory.
- Copy the mathdoxplayer.war file into the deploy directory of your Java Servlet container, like JBoss or Tomcat.
- (Re)start your Java servlet container.

2.5 OpenMath Phrasebooks

The MathDox Player can use a Computer Algebra System (CAS) to evaluate mathematical expressions. An OpenMath Phrasebook is used to access the CAS with OpenMath queries. Phrasebooks for Maxima, GAP and Mathematica have been included in the Mathdox Player, and need therefore not be installed. However, each phrasebook depends on a so-called 'service' program. The service program enables communication between an OpenMath Phrasebook and a CAS, and should therefore be installed and running before using a CAS from the MathDox Player. The Maxima Service, GAP Service and Mathematica Service can all be downloaded from the http://mathdox.org website. Installation instructions are available in the `INSTALL` file inside the downloaded package.

3 MathDox architecture

3.1 The MathDox format

The MathDox standard combines several different XML-based languages into one new format. Each language is used in the field where its strengths are. The combination of these strengths forms the MathDox specification. MathDox also offers the use of web-services. See section 3.2 for more details. The parts of MathDox are:

- DocBook
- OpenMath
- MONET
- XForms
- Jelly
- XInclude

Since all these parts work together, it is quite important for an author to know at least a bit about each of these XML languages. The functionality of these languages within MathDox will now be briefly discussed.

3.1.1 DocBook

DocBook is used as a structural format for MathDox. DocBook, standardized by OASIS[5], is both an open standard and XML-based. It offers the document markup for MathDox and is therefore responsible for the structure of the text in a MathDox document, such as paragraphs, enumerations, tables and more. MathDox is an extension of DocBook version 5.0. Existing

DocBook tools also help translating MathDox documents to LaTeX, PDF, HTML or other formats. Interactivity is, however, not supported in LaTeX or PDF. Documentation and tutorials about DocBook can be found at [2], [3] and [4].

3.1.2 OpenMath

Ambiguity is undesired in any interaction with a user, especially in mathematics. That is why a non-ambiguous representation of the mathematics is needed. But it is not just the user who needs to be able to understand the mathematics, the computer needs to understand it also. In most cases the user might still be able to grasp the meaning from the loosely defined context, but computer software will have a much bigger problem in this respect. LaTeX and MathML presentation are often used formats to represent mathematics, but these formats do not hold the semantic meaning of the mathematics as required. For this reason OpenMath is used within the MathDox system.

OpenMath[6] has been developed in a series of projects that began in 1993. OpenMath gives a meaning to mathematics instead of just telling how to present the mathematics. OpenMath is also extendable with new Content Dictionaries (CDs). These CDs are library files which define new symbols and their correlation to other OpenMath entities. This makes it possible to let OpenMath evolve in new directions when needed. However in most cases an author should have enough with the standard content dictionaries. An overview of the standard OpenMath CD's and the symbols that they contain can be found at the OpenMath website[6].

Although OpenMath is not designed to render mathematics on the screen, it can still be used to do so. An OpenMath expression which needs to be presented on the screen will be translated by the MathDox system first into MathML and then to LaTeX. The LaTeX expressions will then be rendered onto the screen using either the browser fonts or images. This latest step is performed by jsMath[13], a software package specialized in rendering LaTeX expressions in web pages. This also implies that as long as the goal is just to present the mathematics, without performing any computations on the mathematics, both LaTeX and MathML can be used as well.

Computations are normally done with the use of OpenMath queries and through an OpenMath Phrasebook[24]. OpenMath Phrasebooks translate OpenMath expressions to a Computer Algebra System specific language, send it off to the CAS, and translate the result back to OpenMath. Even though each OpenMath Phrasebook targets a specific CAS, all OpenMath Phrasebooks communicate in OpenMath, making it relatively easy for a MathDox document author to switch from one OpenMath Phrasebook to another. At the moment Mathematica[23], Maple[22], GAP[19], Maxima[20] and WIRIS[21] have a (partially implemented) OpenMath Phrasebook. More information about OpenMath can be found at [6] and [7].

3.1.3 MONET

The MONET project was a two-year investigation into mathematical web services funded by the European Commission. One of its goals was to construct a broker architecture for Computer Algebra Systems. The MONET project made a specification of such a broker. The MathDox Player has made an implementation of this broker and offers its use for Computer Algebra System queries. At the moment the supported CAS systems are GAP, Mathematica, Maxima and WIRIS. If the CAS is not set up or configured properly, an error is given. It is also possible to indicate whether to use the native language of the CAS or to use OpenMath. This can be selected for both input and output. WIRIS only supports OpenMath mode. An example can be seen in Section 4.1.3.

3.1.4 XForms

Any interactive system needs input from its users to base its reaction back to the user upon. The MathDox Player is not different, it needs user input to determine the reaction of the system. Information entered by the user can, for instance, be used for adapting explanations, answering exercises or start queries. Changing values will allow readers to observe the results of the (on-screen) calculations for different values. In Section 4.1.4 an example of this feature will be given.

For dealing with user input the MathDox system uses XForms. XForms is a next generation of HTML forms. It is XML based and can do everything HTML forms can do and more without the need for the author to know JavaScript[16]. This makes creating interactivity within MathDox documents easier, especially because JavaScript is subjected to many different dialects on the available browsers.

XForms uses the Model-View-Controller pattern. This separates the data (the model), the presentation (the view) and the conditions and restrictions (the controller) in separate components. The set of conditions and restrictions tell web browsers how to react to entered data, effectively implementing interactivity into web pages generated by MathDox documents. Data entered in the web page will be collected into an XML document and will be sent as such to the server on a possible submit action. However the data can also be used on the web page itself without any submit.

At the moment XForms is not yet properly supported by all browsers. However XForms is included into the XHTML 2.0 specifications and work towards inclusion of XForms in web-browsers is being done. For now the used XForms in MathDox are translated by the MathDox Player into HTML forms supplemented with some JavaScript. More information about XForms can be found at [8] [9] [10].

3.1.5 Jelly

For specifying and fine-tuning the reactions of the MathDox system to user input, an author needs programming constructs. MathDox includes Jelly[11] for this purpose. Jelly is a JSP-like [15] XML-language, and has been developed as an Apache project[14]. Jelly can be used for conditional statements, loops, variables, calls to Java objects and calls to web-services.

3.1.6 XInclude

MathDox Documents do not need to be a single big file. There are cases in which it is advantageous to split documents into smaller parts. The resulting set of parts will increase possiblities for reuse and support maintainability. XInclude[12] is used to include and group together XML components which are kept in different files. XInclude is a W3C standard.

3.2 Web-services

It is possible to invoke web-services from MathDox documents. In this way MathDox documents can be enriched with features otherwise not implemented in the MathDox Player. One example is the use of Computer Algebra Systems. Another example from the WebALT project is to use it for natural language translation. Of course other third party services may be used in a similar way. Web-services can be called with a Jelly expression.

3.3 Inner workings of the MathDox Player

In this section a brief explanation is given on how the MathDox system works. The software running on the server, which makes MathDox documents accessible over the web is called the MathDox player. Similarly to to way in which a web-server ensures that HTML documents stored on disk can be browsed using a web-browser, the MathDox Player ensures that MathDox documents can be browsed. Because web-browsers do not understand the MathDox XML format, the MathDox Player needs to do some additional work; translate the MathDox document into browser-understandable HTML.

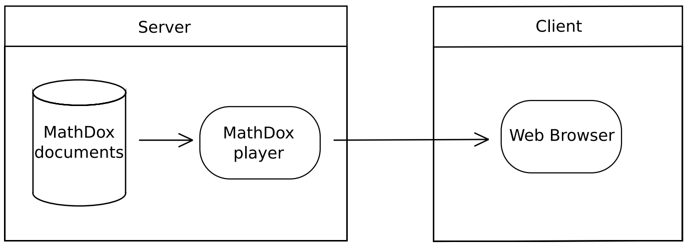


Figure 1: MathDox Player overview

MathDox documents are converted to interactive web pages by means of multiple translations within the MathDox Player. Each translation handles a part of the MathDox language. For instance there is a step that translates the DocBook structure into an HTML structure, and another step will translate the XForms elements into HTML and JavaScript. An overview of all these steps is presented in Figure 2.

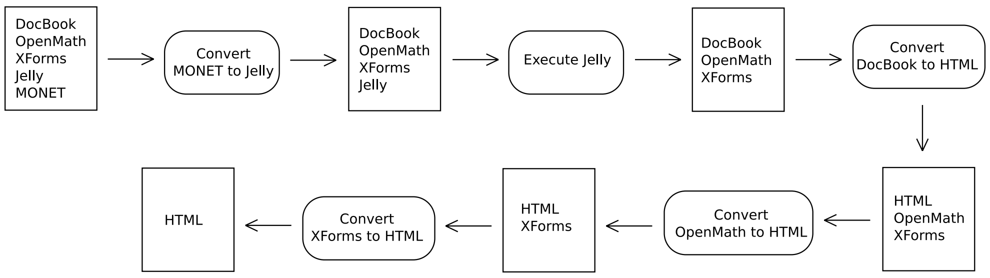


Figure 2: Translation steps in the MathDox player

MONET queries are handled by the MathDox Player by transforming them into Jelly code first. For instance, when a MONET query is encountered that requests evaluation of the expression $a * a = a^2$ by a Computer Algebra System, then the generated Jelly code will contain the statements for calling a specific Computer Algebra System that is configured for use with the MathDox Player, and that can handle the evaluation. The actual translation of MONET into Jelly is handled by an XSLT transformation. XSLT[17] is a language especially suited for describing transformation of one XML format into the other, and it is used extensively by the MathDox Player. After the MONET queries have been translated into Jelly code, all Jelly code is executed. This includes also the Jelly code that was already present in the MathDox document. Next in line is the DocBook conversion. DocBook is translated into HTML, again with the use of XSLT. Also OpenMath is translated with the help of XSLT. It is first translated into MathML and next to LaTeX. Subsequently, a JavaScript program jsMath[13] is used for the final translation to either a suitable font installed in the user's browser, or to images with the same appearance if the font is not present at the user's browser. The final step is converting XForms to HTML. This step is executed by Orbeon Forms[1]. It translates XForms to JavaScript and HTML.

It is important to keep this chain of events in mind when one is writing a MathDox document. Communication and cooperation between different XML-based languages can sometimes be a bit difficult or even impossible. For instance Jelly is executed before XForms; for this reason it is possible to insert Jelly variables into XForms but not the other way around.

4 MathDox tutorial and examples

The combination of different XML-techniques grants an author a high degree of freedom in the creation of MathDox documents. In this chapter some MathDox document code and a few more complex examples without code are given.

4.1 MathDox tutorial

In this section some code examples will be shown. The different XML-techniques are discussed and each new example will be a bit more complex than the previous one. However one will not learn everything possible with MathDox from this section. On purpose the examples are kept simple and easily comprehensible. The reader who needs more information is refered to more specific documentation or tutorials on the subject[8] [9] [10].

4.1.1 Structure

```
<article>
  <title/>
</article>
```

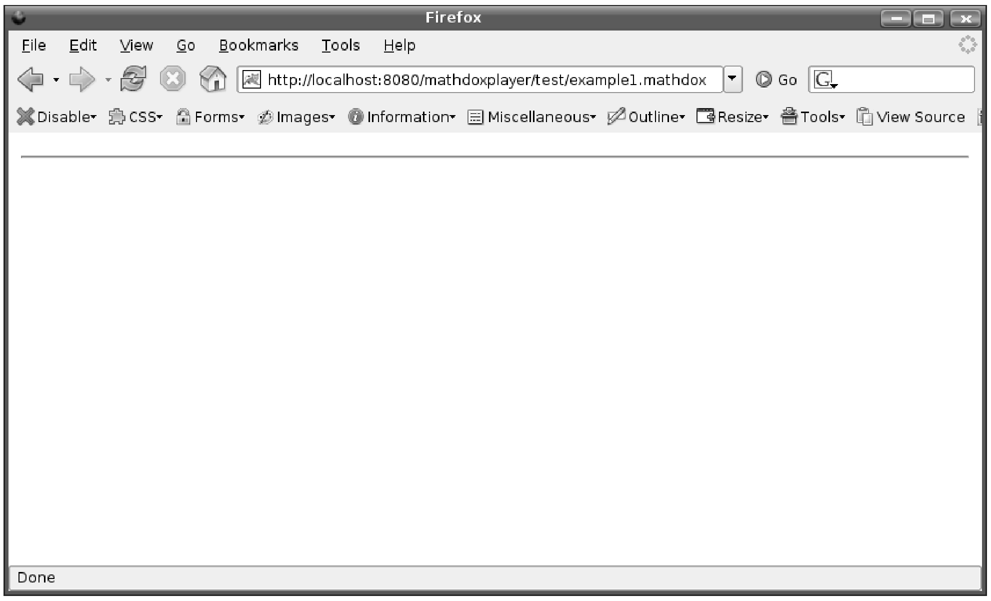
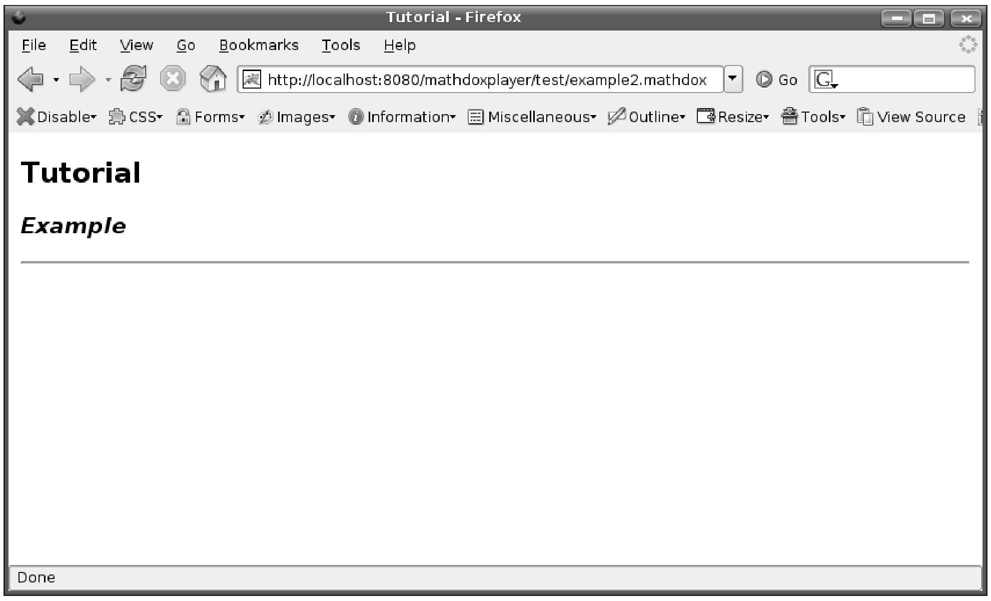

Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example1.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

Figure 3: The tiniest MathDox example

The structure of a MathDox document is taken care of entirely by DocBook. The smallest MathDox document possible is shown above. It is just a DocBook <article> tag containing an empty <title> tag and nothing more. This document can be made more meaningful by adding DocBook, XForms, Jelly, Monet or XInclude tags. This is shown in the next few steps. Feel free to experiment with any of the tags of the included, and previously named, XML-formats.

```
<article>
  <title>Tutorial</title>
  <subtitle>Example</subtitle>
  <chapter>
    <para/>
  </chapter>
</article>
```

In this next step some more DocBook tags have been added. The title, subtitle and chapter are self-explanatory. The para tag is a tag for paragraphs and can be filled with anything from just plain text to OpenMath and Jelly expressions. In the next step it will contain some Jelly code.


Tutorial - Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example2.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

Tutorial

Example

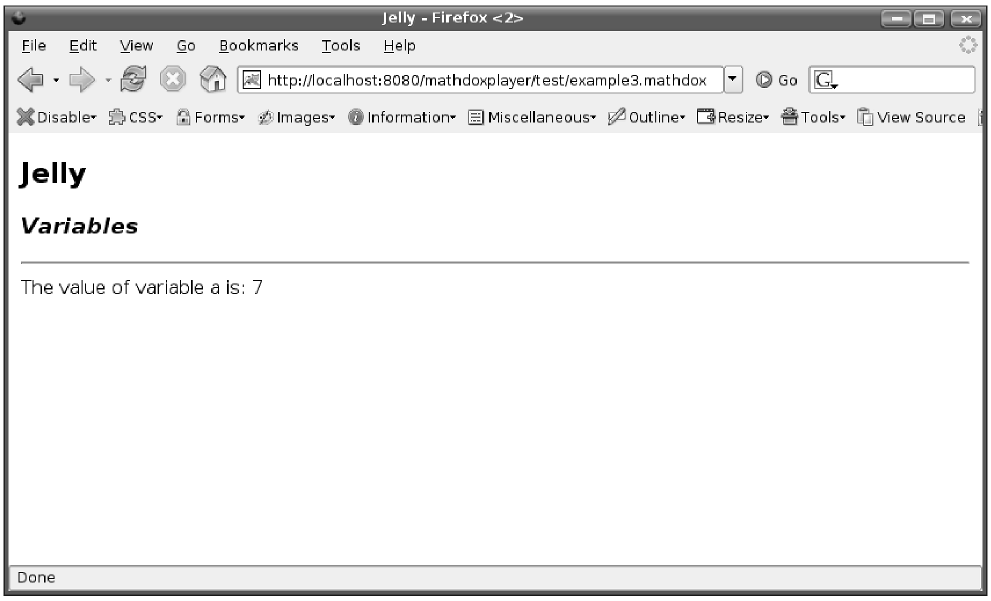
Figure 4: A MathDox example with a minimal structure

4.1.2 Programming constructs

Programming constructs needed in MathDox documents are provided by Jelly. A few examples are shown in relation to the use of variables in Jelly. Note that Jelly is not limited to the use of variables or a while loop. Please refer to the Jelly[11] site for more information on the possibilities of Jelly.

```
<article xmlns:c="jelly:core">
  <title>Jelly</title>
  <subtitle>Variables</subtitle>
  <para>
    <c:set var='a' value='7'/>
    The value of variable a is: ${a}
  </para>
</article>
```

Some Jelly statements have been inserted into the para tag. These statements demonstrate the use of a variable. In the <c:set> tag, the value of the variable a is is set to 7. The `${a}` statement indicates how the value of a can be called. This is the most effective way of using a variable. In the next example a Fibonacci algorithm has been implemented with the use of Jelly. This example includes some calculations and a while loop with a conditional


Jelly - Firefox <2>
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example3.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

Jelly

Variables

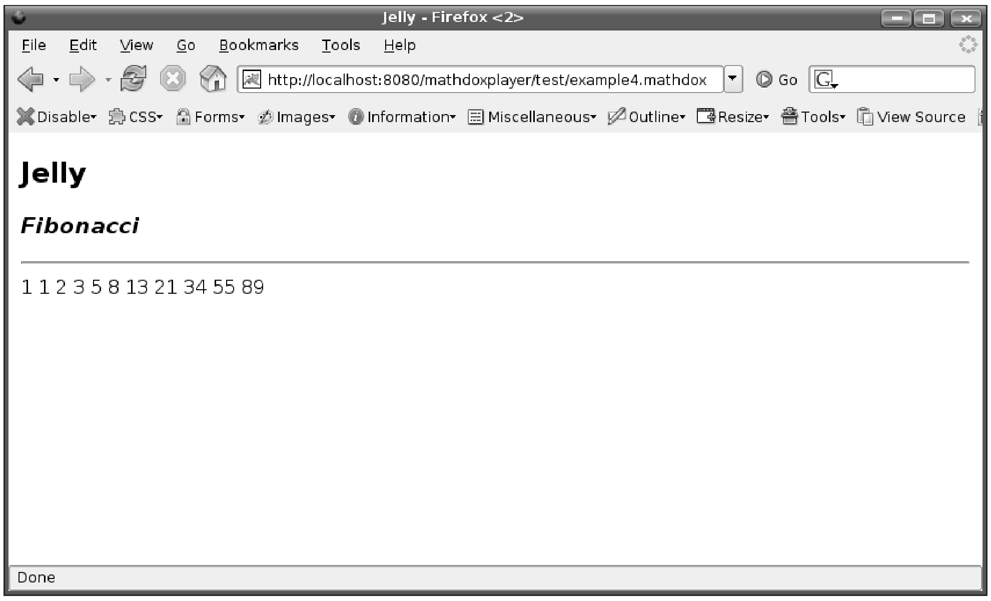
The value of variable a is: 7

Figure 5: Jelly variable in MathDox

statement. The jelly code will have the same functionality as the belkow stated pidgin code

```
a:=1;
b:=1;
while a<100 do
  print a;
  c := a + b;
  a := b;
  b := c;
od
```

```
<article xmlns:c="jelly:core">
  <title>Jelly</title>
  <subtitle>Fibonacci</subtitle>
  <para>
    <c:set var='a' value='1'/>
    <c:set var='b' value='1'/>
    <c:while test='${ a &lt; 100 }'>
      ${a}
      <c:set var='c' value='${ a+b }'/>
      <c:set var='a' value='${ b   }'/>
      <c:set var='b' value='${ c   }'/>
    </c:while>
  </para>
</article>
```


Jelly - Firefox <2>
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example4.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

Jelly

Fibonacci

```
1 1 2 3 5 8 13 21 34 55 89
```

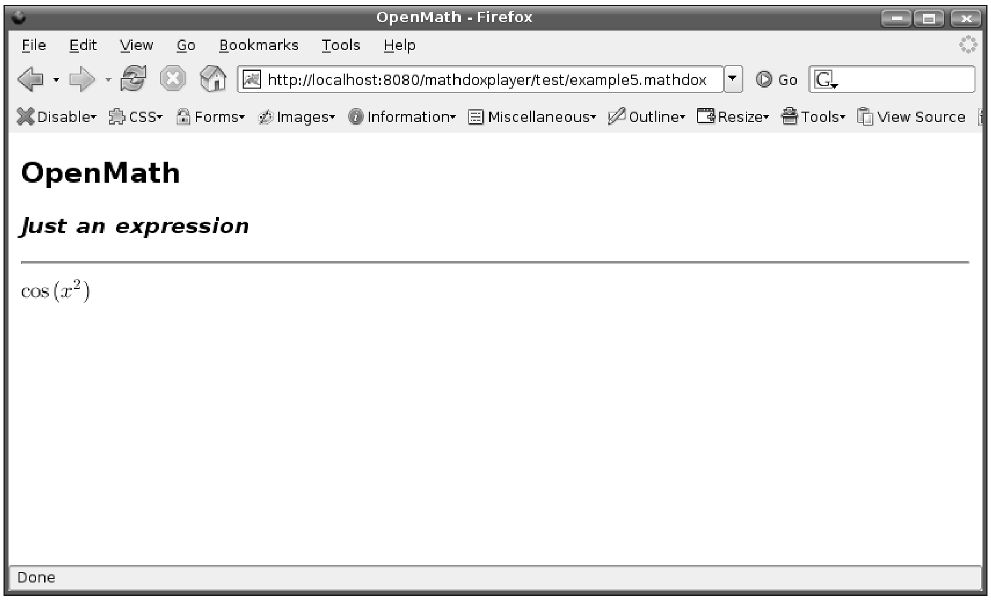
Figure 6: A Fibonacci example

Above an implementation is given of the Fibonacci algorithm. Note that the $<$ (less than) character in Jelly expressions needs to be escaped to prevent confusion with XML-tags.

4.1.3 Mathematics

MathDox supports OpenMath for mathematical expressions. Here is an example of a MathDox document that contains an OpenMath expression.

```
<article>
  <title>OpenMath</title>
‘  <subtitle>Just an expression</subtitle>
  <para>
    <OMA>
      <OMS cd='transc1' name='cos'/>
      <OMA>
        <OMS cd='arith1' name='power'/>
        <OMV name='x'/>
        <OMI>2</OMI>
      </OMA>
    </OMA>
  </para>
</article>
```


OpenMath - Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example5.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

OpenMath

Just an expression

$\cos(x^2)$

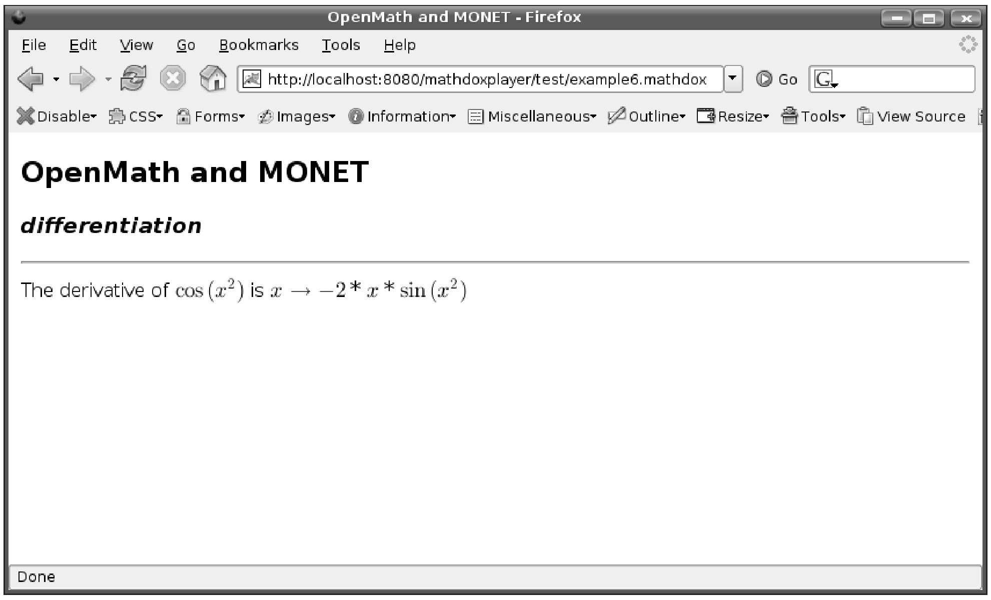
Figure 7: OpenMath in MathDox

The OpenMath code above is used to represent the expression $\cos(x^2)$. It contains the `cos` symbol from the `transc1` Content Dictionary (CD), and the `power` symbol from the `arith1` CD. These CDs are standard OpenMath CDs as mentioned in section 3.1.2. Later we will also see examples of OpenMath expressions being used for computations and editing.

The next example uses OpenMath, Jelly code, and a MONET expression.

```
<article xmlns:monet='http://monet.nag.co.uk/monet/ns' xmlns:x="jelly:xml" >
  <title>OpenMath and MONET</title>
  <subtitle>differentiation</subtitle>
  <para>
    <x:parse var="om-expression" trim="true">
      <OMA>
        <OMS cd='transc1' name='cos'/>
        <OMA>
          <OMS cd='arith1' name='power'/>
          <OMV name='x'/>
          <OMI>2</OMI>
        </OMA>
      </OMA>
    </x:parse>
    The derivative of
    <x:copyOf select='$om-expression/*'/>
    is
    <monet:query>
```

```
      <monet:classification>
        <monet:directive-type href='http://mathdox.org/phrasebook/mathematica#eval'/>
      </monet:classification>
      <monet:body>
        <monet:output>
          <OMA>
            <OMS cd='calculus1' name='diff'/>
            <OMBIND>
              <OMS cd='fns1' name='lambda'/>
              <OMBVAR>
                <OMV name='x'/>
              </OMBVAR>
              <x:copyOf select='$om-expression/*'/>
            </OMBIND>
          </OMA>
        </monet:output>
      </monet:body>
    </monet:query>
  </para>
</article>
```


OpenMath and MONET - Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example6.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

OpenMath and MONET

differentiation

The derivative of $\cos(x^2)$ is $x \to -2 * x * \sin(x^2)$

Figure 8: A MONET call

The expression $\cos(x^2)$ we saw earlier, has now been inserted into a Jelly variable that is able to contain XML. Inserting XML-code into a Jelly variable needs some parsing. This is done by the `<x:parse>` tag. The aim is to reuse the XML stored in the variable as XML again instead of a value as in previous examples. To this end the XML-contents of the variable needs to be copied from the variable and into the XML structure of the document. This is done by means of the `<x:copyOf>` tag. In this way one and the same expression can be printed to the screen now and be used for calculations later as it will be used as input for the MONET query. This MONET query adds some more OpenMath around the contents of the XML-containing variable to tell a CAS to differentiate the function $x \mapsto \cos(x^2)$. The result is then printed to the screen as well.

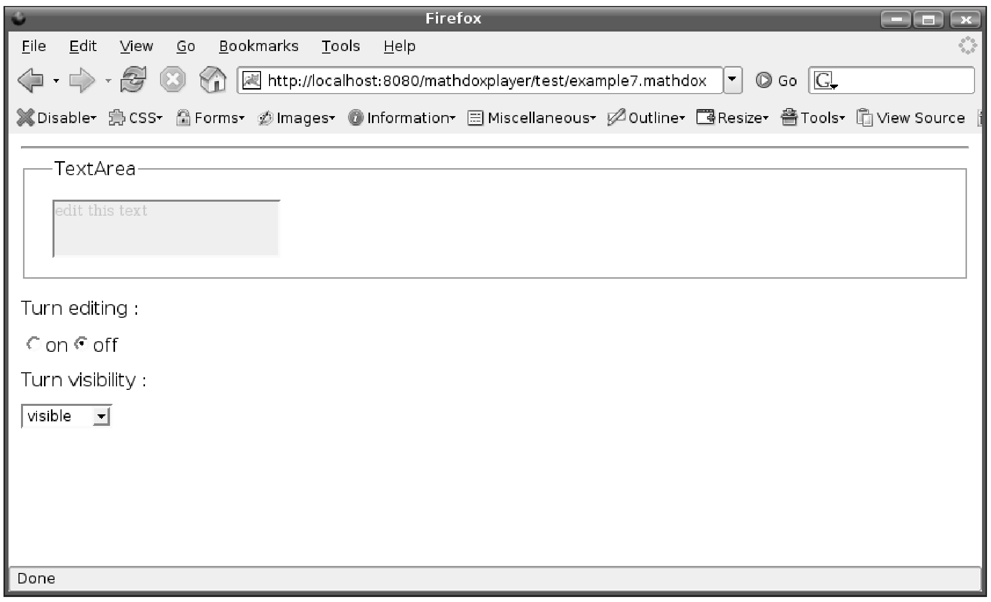
4.1.4 XForms

XForms in MathDox documents adds a means for communication with the reader of the MathDox document. It allows for many form elements which can be used for user-input, but it can also be used with a set of rules to bind entered data to specific conditions, or disable editing in some form elements or remove form elements all together from the web page, or the other way around. The next example shows how this works.

```
<article xmlns:xforms='http://www.w3.org/2002/xforms'>
  <xforms:model>
    <xforms:instance>
      <model>
        <textarea>edit this text</textarea>
        <editable>off</editable>
        <visible>true</visible>
      </model>
    </xforms:instance>
    <xforms:bind
      nodeset='textarea'
      readonly='/model/editable="off"'
    />
  </xforms:model>

  <xforms:group ref='visible[.="true"]'>
    <xforms:label>TextArea</xforms:label>
    <xforms:textarea ref='/model/textarea'  />
  </xforms:group>

  <para>Turn editing :</para>
  <xforms:select1 ref='editable' appearance='full'>
    <xforms:item>
      <xforms:label>on</xforms:label>
      <xforms:value>on</xforms:value>
    </xforms:item>
    <xforms:item>
      <xforms:label>off</xforms:label>
      <xforms:value>off</xforms:value>
    </xforms:item>
  </xforms:select1>
  <para>Turn visibility :</para>
  <xforms:select1 ref='visible'>
    <xforms:item>
      <xforms:label>visible</xforms:label>
      <xforms:value>true</xforms:value>
    </xforms:item>
    <xforms:item>
      <xforms:label>invisible</xforms:label>
      <xforms:value>false</xforms:value>
    </xforms:item>
  </xforms:select1>
</article>
```


Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example7.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

TextArea

edit this text

Turn editing :

( ) on (•) off

Turn visibility :

visible

Figure 9: XForms and MathDox documents

This MathDox document shows three input elements in a web page. The first, a textarea, can be made readonly by the second control and made to disappear by the third. The example above is divided into four parts. The first is an internal model in which all variables and selections of the user are stored and which is send to the MathDox Player on a submit event. In this part there is also a bind rule, which can be used to impose restrictions on the XForms control and elements. This particularly bind element turns the readonly flag of the textarea (part two) on or off based on the model's editable element. The predefined values in the model are the default values. The other parts of the example contain the XForms controls. These are a textarea, radioboxes (select1) and a dropdown menu (again a select1).

These controls are pointed to elements from the model and contain initially the same values as the elements of these forms.

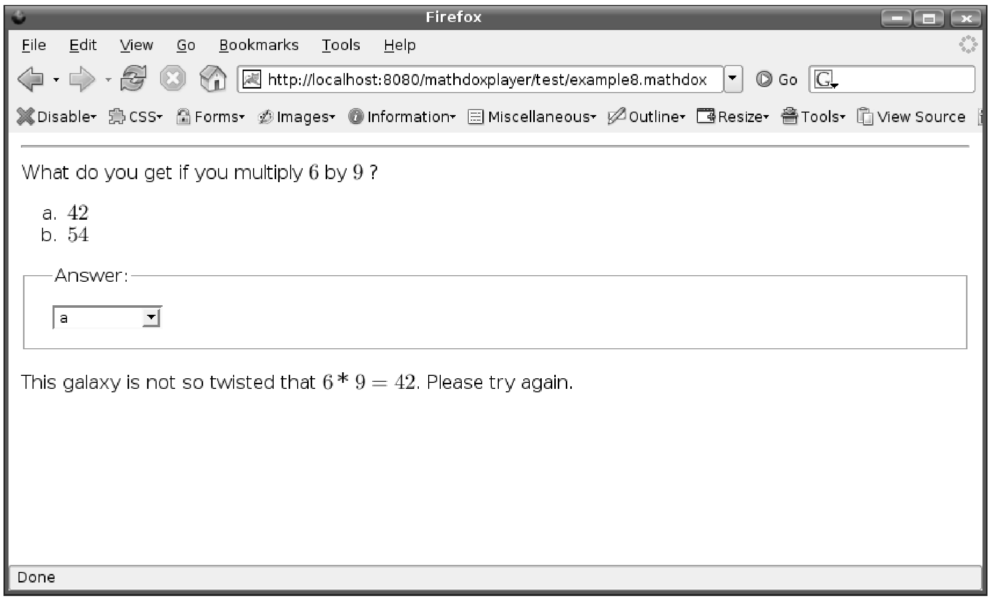
```
<article xmlns:xforms='http://www.w3.org/2002/xforms'>
  <xforms:model>
    <xforms:instance>
      <model><answer/></model>
    </xforms:instance>
  </xforms:model>

  <para>
    What do you get if you multiply
    <OMOBJ><OMI>6</OMI></OMOBJ> by <OMOBJ><OMI>9</OMI></OMOBJ> ?
  </para>

  <orderedlist numeration="loweralpha">
    <listitem>
      <OMOBJ><OMI>42</OMI></OMOBJ>
    </listitem>
    <listitem>
      <OMOBJ><OMI>54</OMI></OMOBJ>
    </listitem>
  </orderedlist>

  <xforms:group ref='answer'>
    <xforms:label>Answer:</xforms:label>
    <xforms:select1 ref=".">
      <xforms:item>
        <xforms:label>no answer</xforms:label>
        <xforms:value>no answer</xforms:value>
      </xforms:item>
      <xforms:item>
        <xforms:label>a</xforms:label>
        <xforms:value>1</xforms:value>
      </xforms:item>
      <xforms:item>
        <xforms:label>b</xforms:label>
        <xforms:value>2</xforms:value>
      </xforms:item>
    </xforms:select1>
  </xforms:group>

  <xforms:group ref="answer[ . = '1' ]">
    <para> This galaxy is not so twisted that
      <OMOBJ>
        <OMA>
          <OMS cd="relation1" name="eq"/>
          <OMA>
            <OMS cd="arith1" name="times"/>
            <OMI>6</OMI>
            <OMI>9</OMI>
          </OMA>
          <OMI>42</OMI>
        </OMA>
      </OMOBJ>. Please try again.
    </para>
  </xforms:group>
  <xforms:group ref="answer[ . = '2' ]">
    <para>Your answer,
      <OMOBJ><OMI>54</OMI></OMOBJ>
      is correct!
    </para>
  </xforms:group>
</article>
```


Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/test/example8.mathdox
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source
Done

What do you get if you multiply 6 by 9 ?

a. 42
b. 54

Answer:

a

This galaxy is not so twisted that $6 * 9 = 42$. Please try again.

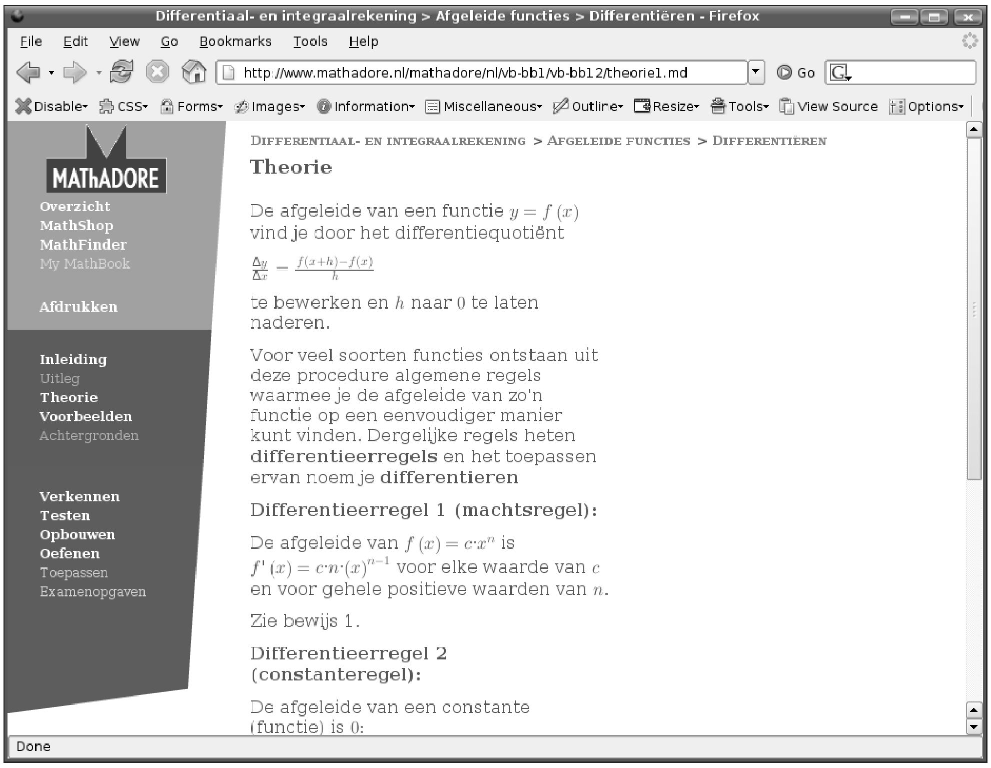
Figure 10: A MathDox Exercise

This MathDox document is a multiple choice question. The reader is asked to answer a simple question by choosing the right answer between two options. Depending which answer was chosen by the reader a feedback is displayed, telling the reader if the answer was correct. This example is constructed out of 5 parts. At the top the document starts with a XForms model, which is used to store the answer of the reader of the MathDox document. Directly underneath the question is posed, followed by a list of possible answers to chose from. The fourth block gives the user a drop down menu which allows the reader to choose. The answer is stored in the model which was found in the first block. In the last block it is determined which feedback should be shown to the reader. The MathDox Player tries to find the reference to the answer element which contains a '1' (indicating that answer a was chosen) in the instance in the model, defined at the top of the document. If it is found then the xforms:group line becomes active and the included lines within this element are shown. Otherwise the answer must be '2' activating a different feedback in the same way.

More MathDox code examples can be found at the MathDox website[18].

4.2 Examples

In the previous section several MathDox code listings could be found, in this section some more complex examples of what MathDox can do are given. The listings of these examples would become too large to show in this document, for that reason they where omitted.


Differentiaal- en integraalrekening > Afgeleide functies > Differentiëren - Firefox
File Edit View Go Bookmarks Tools Help
http://www.mathadore.nl/mathadore/nl/vb-bb1/vb-bb12/theorie1.md
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
MAThADORE
Overzicht
MathShop
MathFinder
My MathBook
Afdrukken
Inleiding
Uitleg
Theorie
Voorbeelden
Achtergronden
Verkennen
Testen
Opbouwen
Oefenen
Toepassen
Examenopgaven
Done

DIFFERENTIAAL- EN INTEGRAALREKENING > AFGELEIDE FUNCTIES > DIFFERENTIEREN

Theorie

De afgeleide van een functie $y = f(x)$ vind je door het differentiequotiënt

$\frac{\Delta y}{\Delta x} = \frac{f(x+h)-f(x)}{h}$

te bewerken en h naar 0 te laten naderen.

Voor veel soorten functies ontstaan uit deze procedure algemene regels waarmee je de afgeleide van zo'n functie op een eenvoudiger manier kunt vinden. Dergelijke regels heten **differentieerregels** en het toepassen ervan noem je **differentieren**

Differentieerregel 1 (machtsregel):

De afgeleide van $f(x) = c{\cdot}x^n$ is $f'(x) = c{\cdot}n{\cdot}(x)^{n-1}$ voor elke waarde van c en voor gehele positieve waarden van n.

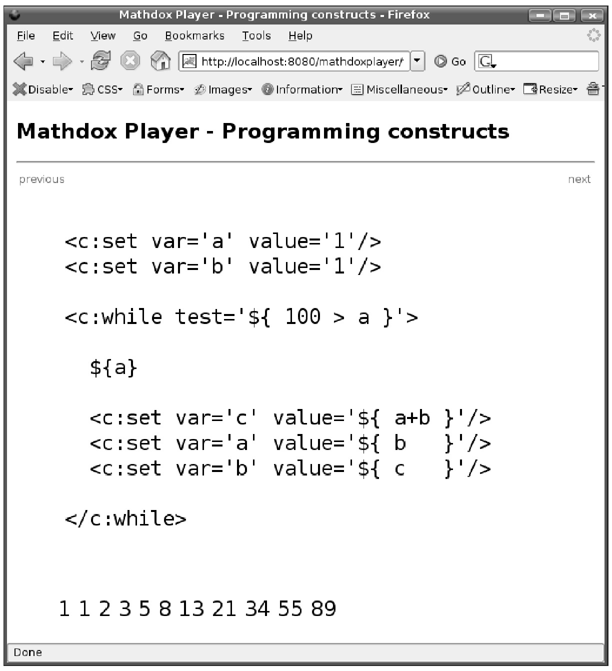
Zie bewijs 1.

Differentieerregel 2 (constanteregel):

De afgeleide van een constante (functie) is 0:

Figure 11: An example in which differentiation is explained

Figure 11 shows how MathDox documents can be used to represent mathematics in a (Dutch) web page. Web pages of this kind can be used for a lot of different purposes, the example here is used to teach high school student the basics of differentiation.


Mathdox Player - Programming constructs - Firefox
File Edit View Go Bookmarks Tools Help
http://localhost:8080/mathdoxplayer/
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize
previous
next
Done

Mathdox Player - Programming constructs

```
<c:set var='a' value='1'/>
<c:set var='b' value='1'/>

<c:while test='${ 100 > a }'>

  ${a}

  <c:set var='c' value='${ a+b }'/>
  <c:set var='a' value='${ b   }'/>
  <c:set var='b' value='${ c   }'/>

</c:while>
```

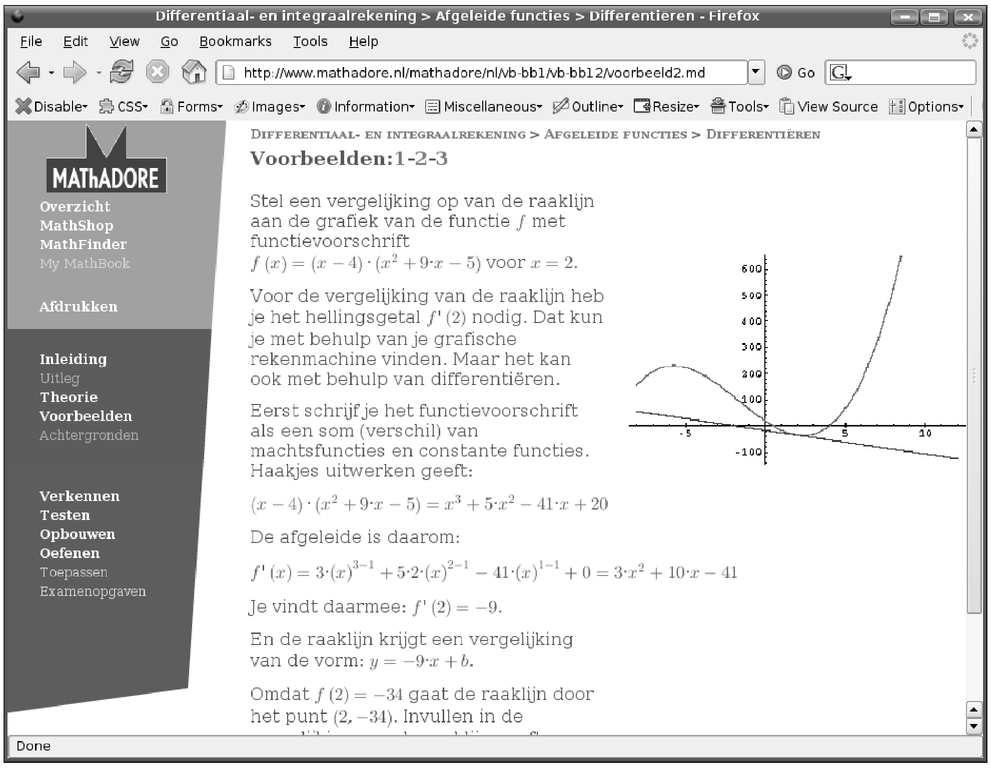
```
1 1 2 3 5 8 13 21 34 55 89
```

Figure 12: The Fibonacci algorithm coded and demonstrated

In the previous section a listing was shown which would produce the Fibonacci sequence. In figure 12 an adapted version is shown. In this version the while loop as well as the sequence itself are shown together in the web page.

Figure 13 shows an example about tangient lines (again in Dutch). Note that the values in this example are randomly chosen. Each time the page is reloaded (reload button on the browser, or reload link on the page) new values are chosen. These values are incorporated into the explanation and into the graph elsewhere on the page. The graph is drawn by a Computer Algebra System. Users, in this case students, can carefully examine the theory behind differentiation, and request as much examples with different values as they would like.

Exercises can be designed in such a way that they consist of multiple steps. As such they are an example of programmable reactions in MathDox. These exercises can, for instance, respond to wrong answers by either giving a hint, a reference to theory, or by simplifying the question. The kind of exercise that directs readers based on answers to questions to more specific questions


Differentiaal- en integraalrekening > Afgeleide functies > Differentiëren - Firefox
File Edit View Go Bookmarks Tools Help
http://www.mathadore.nl/mathadore/nl/vb-bb1/vb-bb12/voorbeeld2.md
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options
MAThADORE
Overzicht
MathShop
MathFinder
My MathBook
Afdrukken
Inleiding
Uitleg
Theorie
Voorbeelden
Achtergronden
Verkennen
Testen
Opbouwen
Oefenen
Toepassen
Examenopgaven
600
500
400
300
200
100
-100
-5
5
10
Done

DIFFERENTIAAL- EN INTEGRAALREKENING > AFGELEIDE FUNCTIES > DIFFERENTIËREN

Voorbeelden:1-2-3

Stel een vergelijking op van de raaklijn aan de grafiek van de functie f met functievoorschrift $f(x) = (x-4)\cdot(x^2+9{\cdot}x-5)$ voor $x = 2$.

Voor de vergelijking van de raaklijn heb je het hellingsgetal $f'(2)$ nodig. Dat kun je met behulp van je grafische rekenmachine vinden. Maar het kan ook met behulp van differentiëren.

Eerst schrijf je het functievoorschrift als een som (verschil) van machtsfuncties en constante functies. Haakjes uitwerken geeft:

$$(x-4)\cdot(x^2+9{\cdot}x-5) = x^3+5{\cdot}x^2-41{\cdot}x+20$$

De afgeleide is daarom:

$$f'(x) = 3{\cdot}(x)^{3-1}+5{\cdot}2{\cdot}(x)^{2-1}-41{\cdot}(x)^{1-1}+0 = 3{\cdot}x^2+10{\cdot}x-41$$

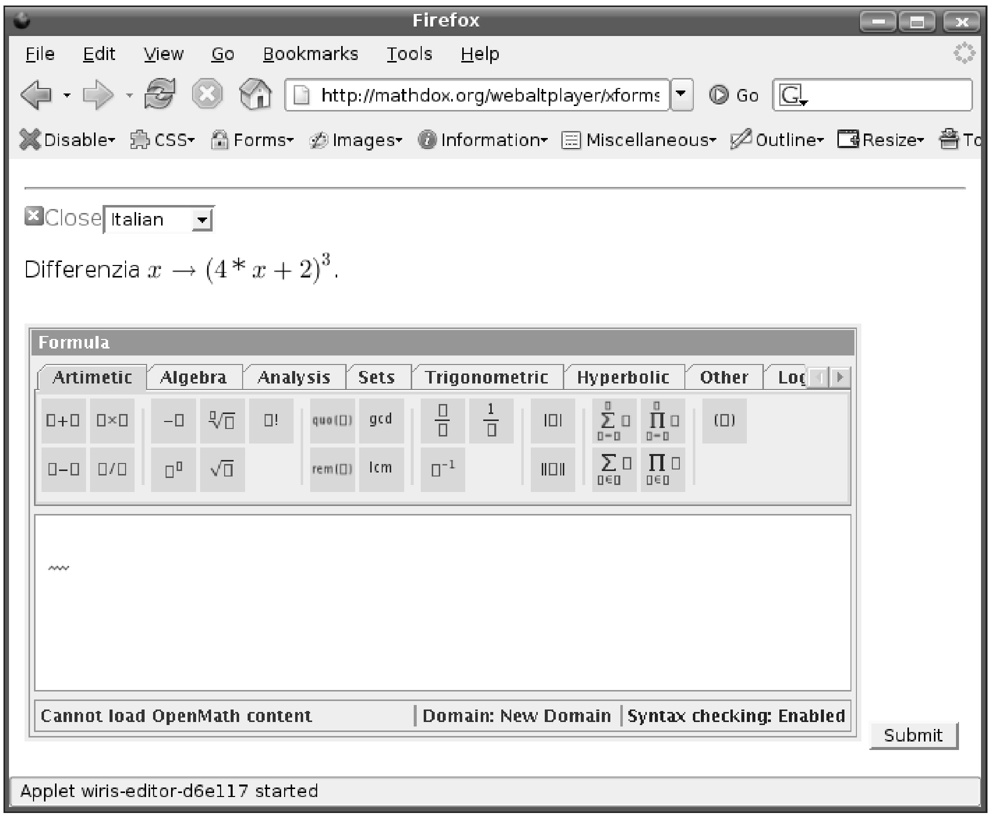
Je vindt daarmee: $f'(2) = -9$.

En de raaklijn krijgt een vergelijking van de vorm: $y = -9{\cdot}x+b$.

Omdat $f(2) = -34$ gaat de raaklijn door het punt $(2,-34)$. Invullen in de

Figure 13: An example about tangient lines

in the exercise, is called an exercise graph. A step within these exercises is called an interaction. Each interaction can contain for instance some text, a link, an image or an answer field. Both multiple choice questions (one or more valid answers) and open questions are supported. In figure 14 the first interaction of a multi-step exercise is shown. This exercise was created during the WebALT project. In this European project the goal was to achieve multilingual mathematical exercises. This goal was achieved by creating a special artificial language which is translated by a web-service to a natural language (such as English or Spanish) of choice. Figure 14 shows one of these exercises and demonstrates the ability of MathDox documents to call this web-service and incorporate the results into the document. Note that the WIRIS OpenMath Editor Applet [29], labeled 'Formula' in figure 14, is not open source and is therefore by default not included in the MathDox Player.


Firefox
File Edit View Go Bookmarks Tools Help
http://mathdox.org/webaltplayer/xforms
Go
Disable CSS Forms Images Information Miscellaneous Outline Resize
Close
Italian
Formula
Artimetic Algebra Analysis Sets Trigonometric Hyperbolic Other
gcd
lcm
Submit

Differenzia $x \to (4 * x + 2)^3$.

Cannot load OpenMath content | Domain: New Domain | Syntax checking: Enabled

Applet wiris-editor-d6e117 started

Figure 14: A WebALT exercise which demonstrates the use of programming constructs and a web-service call

References

[1] Orbeon Forms, `http://www.orbeon.org`

[2] DocBook.org, `http://docbook.org`

[3] DocBook The Definitive Guide, `http://www.oreilly.com/catalog/docbook/chapter/book/docbook.html`

[4] DocBook.org, `http://opensource.bureau-cornavin.com/crash-course/en/markup-based-on-content.html`

[5] OASIS, `http://www.oasis.org`

[6] OpenMath, `http://openmath.org`

[7] Crossroads OpenMath, `http://www.acm.org/crossroads/xrds6-2/openmath.html`

[8] XForms, `http://www.w3.org/MarkUp/Forms`

[9] Wikipedia reference on XForms, `http://en.wikipedia.org/wiki/XForms`

[10] XForms Essentials, `http://xformsinstitute.com/essentials/browse/book.php`

[11] Jelly, `http://jakarta.apache.org/commons/jelly`

[12] XInclude, `http://www.w3.org/TR/xinclude`

[13] jsMath, `http://www.math.union.edu/~dpvc/jsMath`

[14] apacheProject, `http://apache.org`

[15] JSP, `http://java.sun.com/products/jsp`

[16] JavaScript, `http://www.javascript.com`

[17] XSLT, `http://www.w3.org/TR/xslt`

[18] MathDox, `http://www.mathdox.org`

[19] GAP, `http://www.http://www.gap-system.org`

[20] Maxima, `http://maxima.sourceforge.net`

[21] WIRIS, `http://www.wirisonline.net`

[22] Maple, `http://www.maplesoft.com`

[23] Mathematica, `http://www.wolfram.com`

[24] OpenMath Phrasebooks, `http://www.mathdox.org`

[25] Ant, `http://ant.apache.org`

[26] Subversion, `http://subversion.tigris.org`

[27] XML Specification, `http://www.w3.org/TR/xml11`

[28] GNU Lesser General Public License, `http://www.gnu.org/licenses/lgpl.html`

[29] WIRIS Editor `http://www.wiris.com/content/view/20/3/lang,en/`