Making Items in Möbius part B

Januari 2023

Making Items in Möbius part B

Contents

List of Figures

1 Möbius Making test items part B

Update febr. 2022

1.1 Information and downloads

- **Home page Metha Kamminga**
  More information about courses en news about Möbius on the web site of Metha Kamminga: https://www.methakamminga.nl.
- This manual is a part of a set:

- Making test items part A (without formulas)
- Making test items part B (with formulas)
- Making test items part C (special features)
- Randomizing

For general information about the Content Manager and the all over view of an item, look in the manual of Making test items part A.

1.2 Formulas with Möbius

1.2.1 Introduction

The Möbius system is very powerful for using formulas. Any time you can use the underlying computer algebra system Maple.
The use of formulas is now independent of Java by the new programming based on html5.

- You are able to use formulas in the text of the question, in the feedback, in hints and everywhere where the computer algebra system Maple is ready to generate these formulas.
- It is even possible to test students with blank answer fields where they have to enter a formula. For formulas it is very interesting to randomize each question. The Möbius system is very powerful with randomization not only formulas but also for text, pictures etcetera. Look in the manual *Randomizing*.
- Be on the lookout, because there are a lot of different settings of the formulas in the answer fields. Try out your questions very well before giving them to the students. In the next sections you will get a lot of hints to solve problems in entering formulas in the answer fields.
- Finally it is important that students will get some training to fill in the blanks with a formula in Maple-syntax.

1.2.2 Setting of the expression type

1.2.2.1 Introduction at the several settings for formulas

Students are very indecisive with typing formulas and are dubious about the Maple-syntax. As a creator of items you can make it easier for your students by giving them some tools to check.
The computer algebra system Maple will help to make a perfect *Grading Code* to grade the students answer (response). You will match the response with the correct answer. There are several ways to do that with different kind of types of formulas. See in section *Hints for the Grading Code (page 46)*.

In different ways the student can type the formula as an answer in a so called *Free Response Area* embedded in a question.
This document will give you an introduction to different kinds of Free Response Areas with benefits and disadvantages of these.

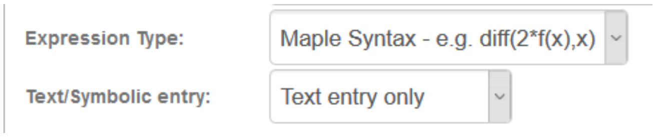
In general there are two question types suitable for testing formulas:
The quesion type *Maple graded* and the question type *Mathematical Formula*.
The settings for entering formulas in these question types are different as you will see in the next sections.

1.2.2.2 Question type Maple graded. Setting Maple syntax and Text entry only

In the question type *Maple graded* with the setting *Maple syntax* the student types the formula with the official Maple syntax, with lots of possibilities. Character combinations are seen as one variable. More mathematical objects as matrices, differential equations, integrals are prossible. Working with subscript, working with functions etcetera is also possible with this setting *Maple syntax*. The student must have a good knowledge of the correct syntax. The system is pretty precisely and case sensitive. But it has a lot of benefits in the field of math and science, from simple questions to complex questions in applications.

In the *Maple graded* question type are two possibilities for the formula settings: *Maple syntax* en *Formula.*
There is a huge difference between these two settings as you will see in the examples of the next section.

Start with the following setting:



Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

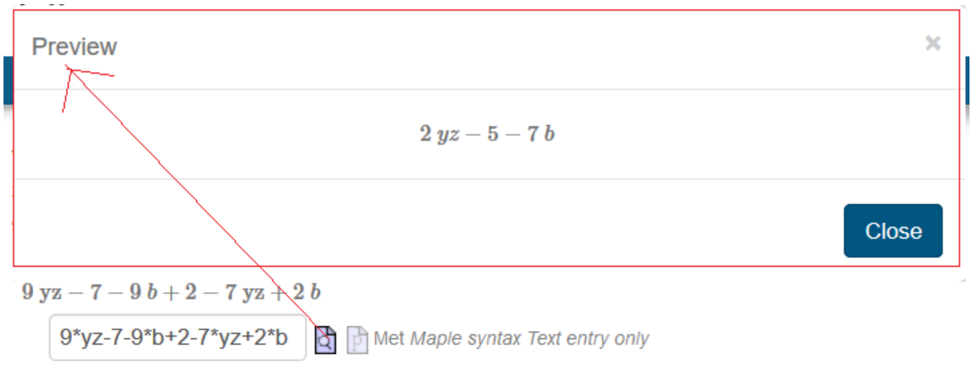
Figure 1.1: Formula settings of the Maple graded question type

This setting with *Maple syntax* and *Text entry only* is the most reliable and has very poor disadvantages.

The student clicks on *Preview* (little magnifying glass) to check his input formula in the answer field.

NOTE: You ask to simplify a formula. The student enters the not simplified formula and with *Preview* he sees the simplification because het system automatically simplifies it. This is mostly not what you want.
There is a way to prevent automatic simplification by programming something in the *Custom Previewing Code*.
See in *(page 39)*.

HINT: If you don't program extra code in the *Grading Code* you will probably see that the system sees the students response which is not simplified as correct.



$2\,yz - 5 - 7\,b$

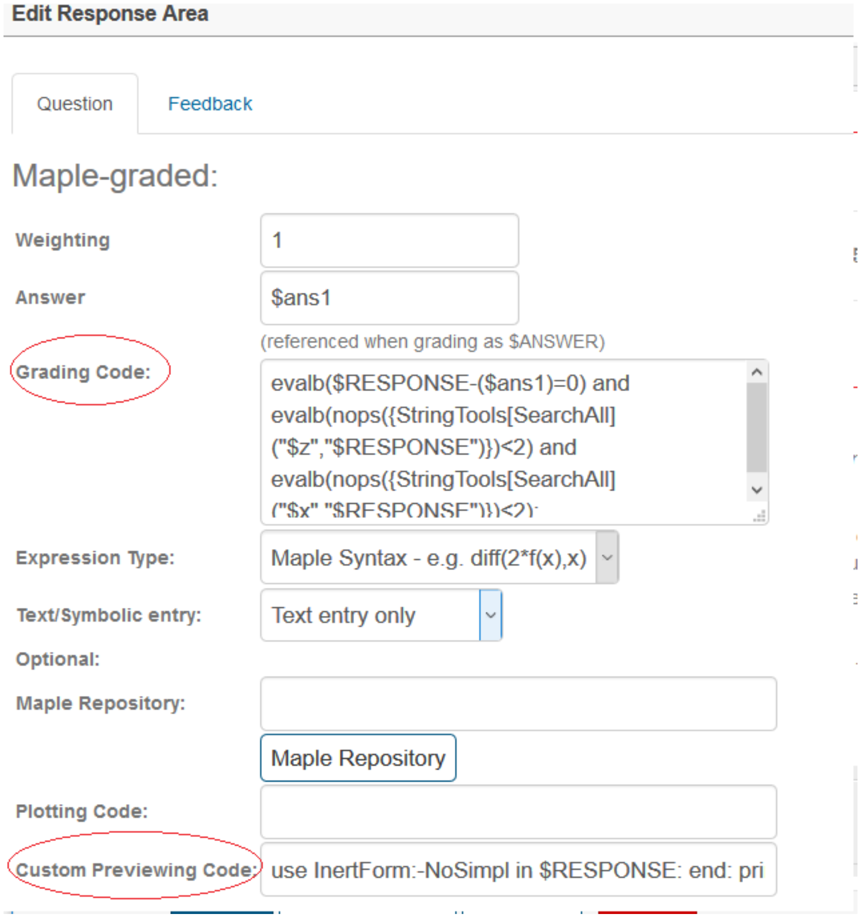
$9\,\mathrm{yz} - 7 - 9\,b + 2 - 7\,\mathrm{yz} + 2\,b$

9*yz-7-9*b+2-7*yz+2*b

Met *Maple syntax Text entry only*

Figure 1.2: The Preview for checking the input

- The student clicks on *Preview* and sees the simplified formula. Mostly you don't want him to see that simplification. Precautions for that you will put extra code in the *Custom Previewing Code*. See *(page 42)*.
- With the *Grading Code* you will program the restrictions for the formula that you will grade as correct. See *(page 39)*.



Maple-graded:

Weighting: 1

Answer: $ans1

(referenced when grading as $ANSWER)

Grading Code:

```
evalb($RESPONSE-($ans1)=0) and
evalb(nops({StringTools[SearchAll]
("$z","$RESPONSE")})<2) and
evalb(nops({StringTools[SearchAll]
("$x","$RESPONSE")})<2)
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code:

Custom Previewing Code: use InertForm:-NoSimpl in $RESPONSE: end: pri

Figure 1.3: Edit Response Area of Maple graded question type

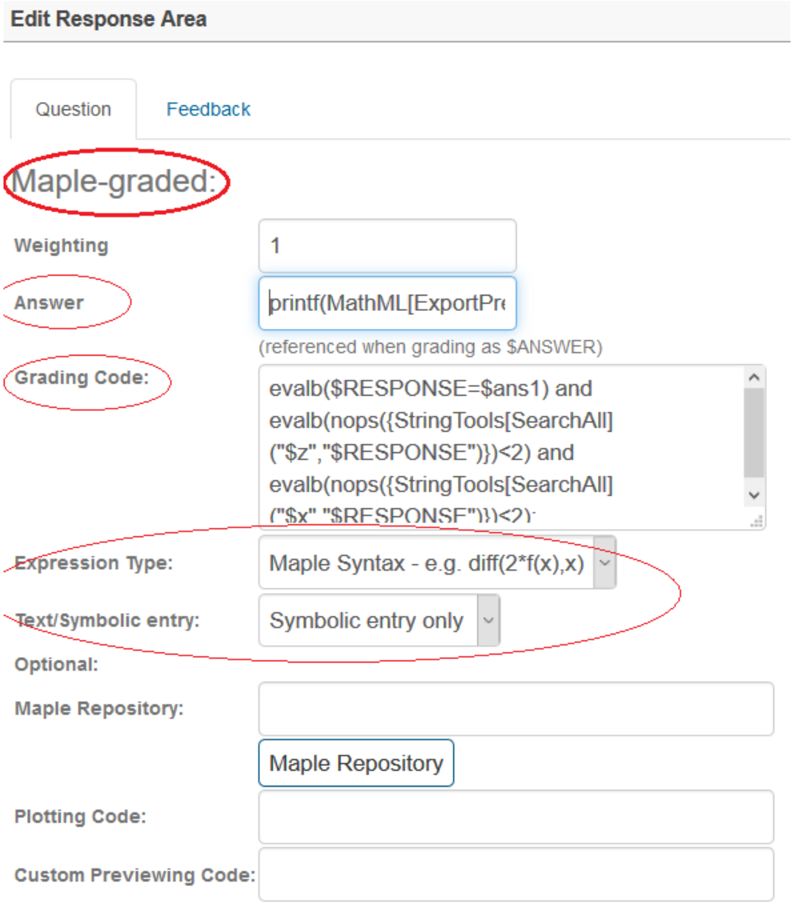
In the figure you see the comprehensive code of the *Grading Code* and the *Custom Previewing Code*.

HINT: In case the answer is too wide for the response area (text input); then put an extra code in the *Custom CSS* of the question. All the answer fields of this kind of the question you can set on a certain width (for example 200 or 250 pt.).

```
.response input {width:250pt;}
```

1.2.2.3 Question type Maple graded. Setting Maple syntax and Symbol entry only

In the *Edit Response Area* of the *Maple graded* question type, in the *Expression Type* you opt again for *Maple Syntax* and additionally you opt for *Symbol entry only*. If you do so, then the student will see an editor to fill in his answer.



Weighting: 1

Answer: printf(MathML[ExportPre

(referenced when grading as $ANSWER)

Grading Code:

```
evalb($RESPONSE=$ans1) and
evalb(nops({StringTools[SearchAll]
("$z","$RESPONSE")})<2) and
evalb(nops({StringTools[SearchAll]
("$x","$RESPONSE")})<2)
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Symbolic entry only

Optional:

Maple Repository:

Plotting Code:

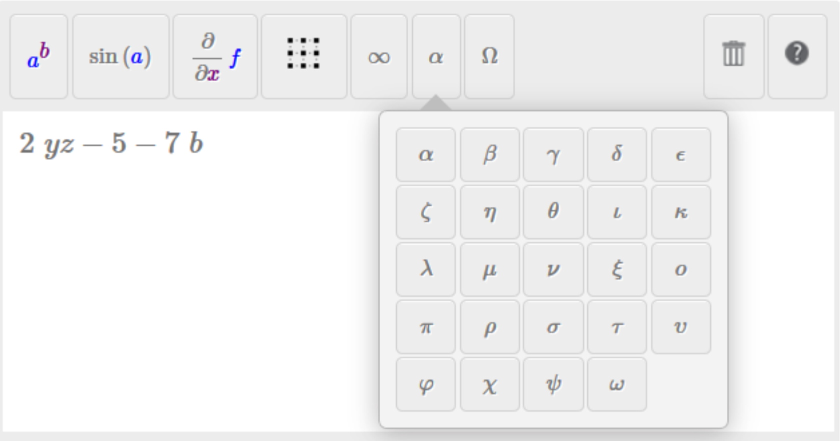
Custom Previewing Code:

Figure 1.4: Maple Syntax Symbolic entry only

In the figure you see that the same comprehensive *Grading Code* is used to prevent the not simplified answer will be graded as correct.
No need for a *Custom Previewing Code* because the editor is itself a preview. No chance for automatical simplifying.

HINT: Because the student does not type the operators like * and ^ and brackets you can offer a 2D formula (MathML) for purposes of the feed back (presenting the correct answer after grading) in the *Answer* field by programming the following:

```
printf(MathML[ExportPresentation]($ans));
```



$2\,yz - 5 - 7\,b$

Figure 1.5: Maple syntax in Symbol Mode with Editor

In the figure you see the Editor offered in the setting *Maple syntax* with *Symbol Mode.* Also character combinations are possible as well subscript and superscript, integrals, differentials, matrices and so on. In the figure you see the Greek palette for undercast.

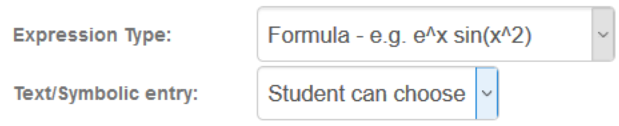
Some hints for the use of the Editor:

- The student has to exercise with this editor.
- For multiplication (times), the student enters * (which is seen as a dot in the editor) or typing a space (which is seen as a invisible times).
  2 x with or without space is seen as a multiplication, but ab is not seen as multiplication and also not b7 until you put a space in between: a b or you type a*b or b*7. It gives lots of benefits in applications.
  Like the expression $M\,(a + b)$ which is meant as a multiplication type a space just in front of the bracket open M (.....)
  On the other hand without space or asterix it is seen as a function. It is also possible to check functions with this question type *Maple graded*.
- Be aware that the symbols only can used separately. The combination Δx does not work for grading for example. In that case type it with a space.
- If the formula is destroyed use the bin (right above) to start over to prevent underlying codes will mess up the formula.
- All basic functions will be entered with the argument between brackets like sin(x), exp(x) and ln(x) and so on.
- **HINT**: The response area with editor of the Maple-syntax has often a unnecessary big height. Use the following code in the *Custom CSS* of the question to make the field less high for all the editors in the whole question.
  ```
  .mwEquationEditor {height:100px !important}
  ```

1.2.2.4 Question type Maple graded. Setting Formula

In the question type *Maple graded* you can opt for the setting *Formula* for entering the formula on a special way.

Additional information is important. Use this setting only in special cases, because this setting is very weak and the preview is very bad with lots of brackets.

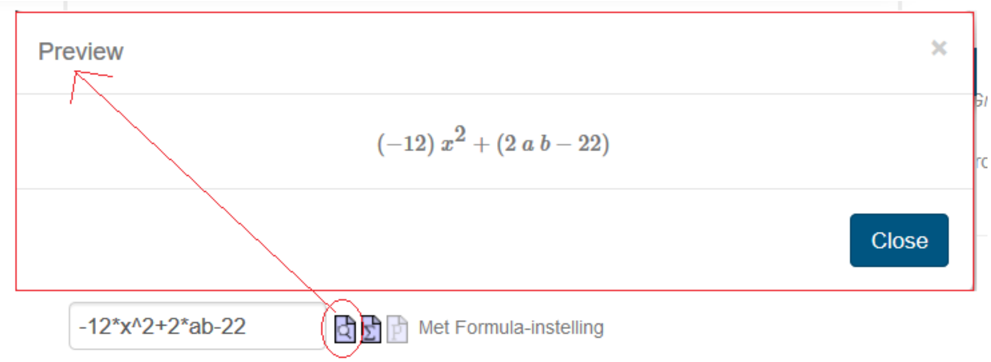


Expression Type: Formula - e.g. e^x sin(x^2)

Text/Symbolic entry: Student can choose

Figure 1.6: Expression type Formula

This is what the student sees with previewing a simple formula. The student does not type brackets and in the preview he sees lots of brackets.



Preview

$(-12)\,x^2+(2\,a\,b-22)$

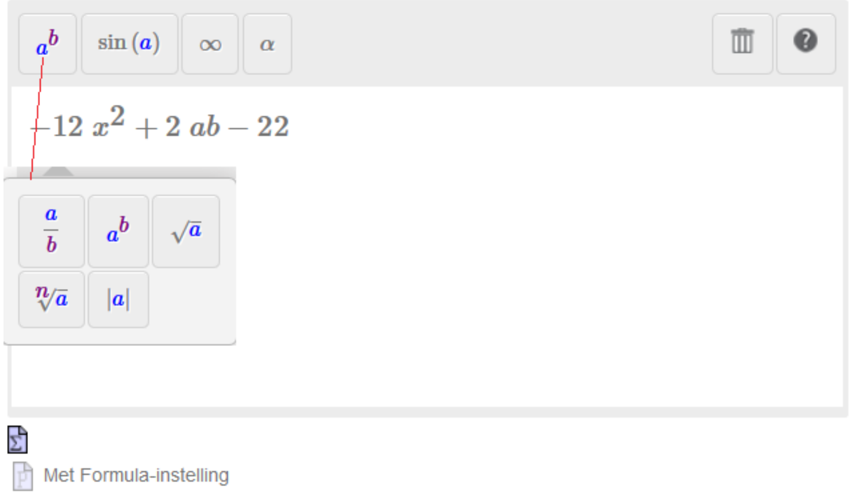
-12*x^2+2*ab-22

Met Formula-instelling

Figure 1.7: De Preview with Formula setting

- Additionally the expression ab is not seen als a character combination. The system transforms the input automatically to a multiplication.
  So the system is not able to grade formulas with character combinations, subscript and so on.
- It is possible the student can change the inputfield from text input to symbol mode (with an editor) by the Sigma-button.

After pressing the Σ-button, the question comes up with the editor.



$-12\,x^2+2\,ab-22$

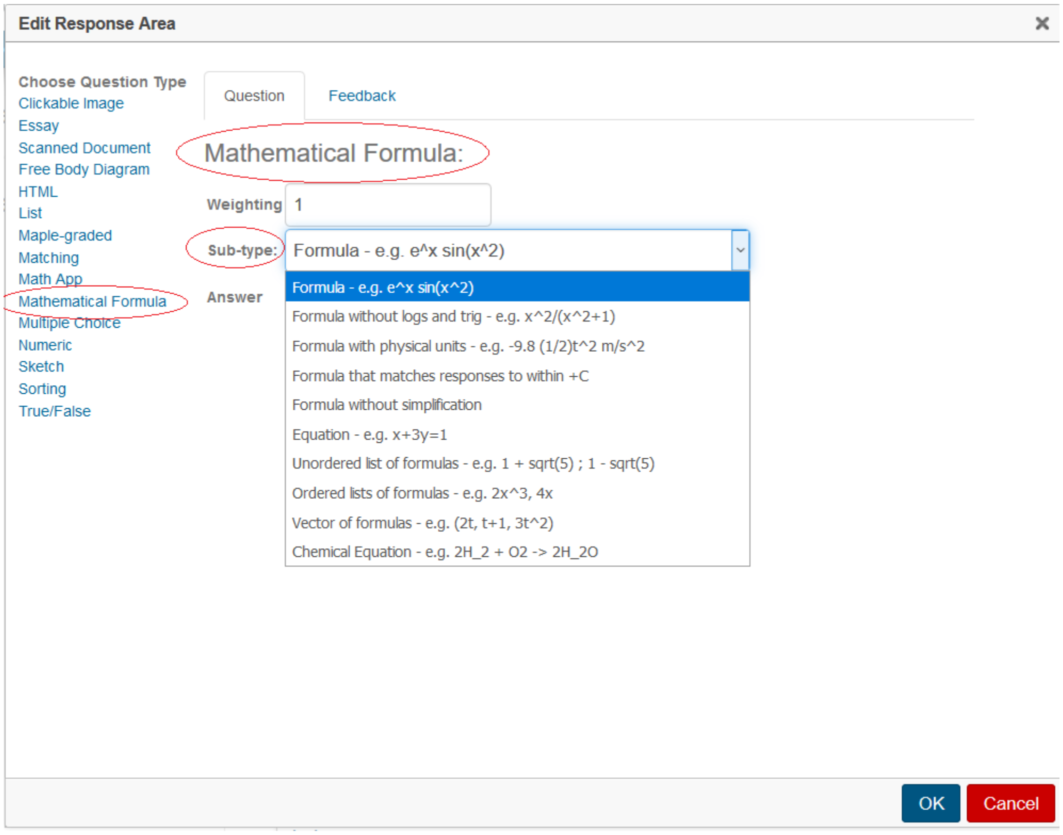
Met Formula-instelling

Figure 1.8: The editor with Formula setting

The editor offered with the setting *Formula* is not very reliable and less comprehensive in comparison with the editor offered with the setting *Maple Syntax*! So again a reason to not use the setting *Formula!*

1.2.2.5 Question type Mathematical Formula. Setting Formula- e.g.e^x sin(x^2)

The question type *Mathematical Formula* is really inappropriate for formulas. With the question type *Maple graded* the setting with *Formula* was weaker than with the setting *Maple Syntax*. With the question type *Mathematical Formula* with subtype *Formula* you will do another step behind with the possibilities for formulas.


Edit Response Area

Choose Question Type
- Clickable Image
- Essay
- Scanned Document
- Free Body Diagram
- HTML
- List
- Maple-graded
- Matching
- Math App
- Mathematical Formula
- Multiple Choice
- Numeric
- Sketch
- Sorting
- True/False

Question | Feedback

Mathematical Formula:

Weighting: 1

Sub-type: Formula - e.g. e^x sin(x^2)

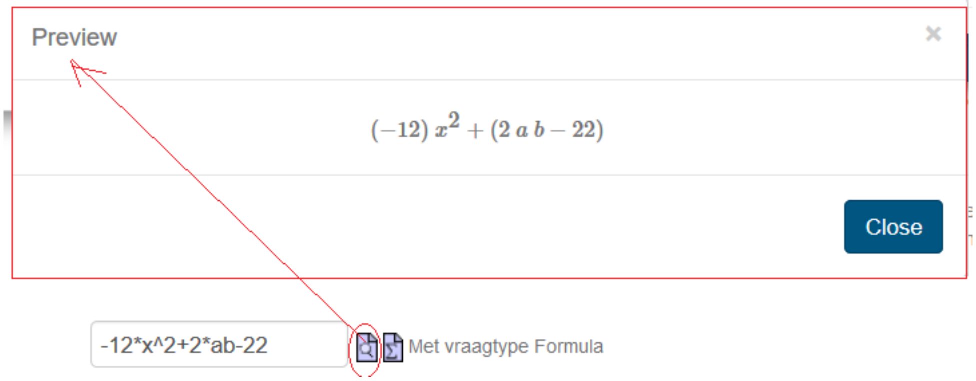
Answer

- Formula - e.g. e^x sin(x^2)
- Formula without logs and trig - e.g. x^2/(x^2+1)
- Formula with physical units - e.g. -9.8 (1/2)t^2 m/s^2
- Formula that matches responses to within +C
- Formula without simplification
- Equation - e.g. x+3y=1
- Unordered list of formulas - e.g. 1 + sqrt(5) ; 1 - sqrt(5)
- Ordered lists of formulas - e.g. 2x^3, 4x
- Vector of formulas - e.g. (2t, t+1, 3t^2)
- Chemical Equation - e.g. 2H_2 + O2 -> 2H_2O

Figure 1.9: Sub-types of the Mathematical Formula questiontype

In the picture you see that there is a set of sub-types which are pretty interesting to use for other things than formulas.

In the figure below you see that the *Preview* of normal formulas is pretty bad with all those brackets.

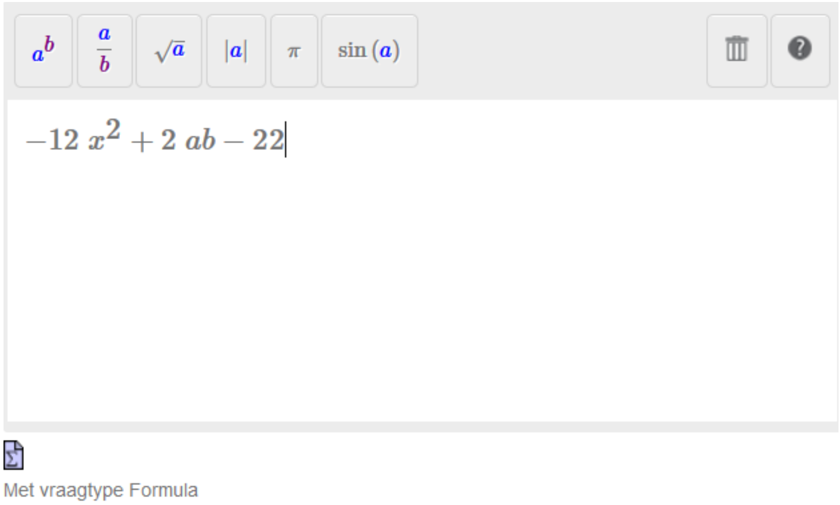

Preview

$$(-12)\,x^2 + (2\,a\,b - 22)$$

-12*x^2+2*ab-22 Met vraagtype Formula

Figure 1.10: Preview of de Mathematical Formula question type

The character combination ab is also seen as a multiplication.
In this case the student himself is able to toggle with the Σ-button from text-mode to symbol-mode.
In the editor the student will see even less possibilities with buttons as you can see in the figure below.



$-12\,x^2 + 2\,ab - 22$

Met vraagtype Formula

Figure 1.11: Editor of the question type Mathematical Formula

The buttons of this editor are not expandable. These buttons are all the buttons available for this editor.
Character combinations in the formulas are not possible in this editor.

1.2.2.6 Important hint

Important hint: The setting with **Formula** for testing formulas in the question type *Maple graded* we will **discourage strongly!**

We will even more discourage strongly the question type **Mathematical Formula** with sub-type **Formula**.
You are not able to test formulas with character combinations with this setting! You will put the students on the wrong leg with this so called "user friendly" settings.
The student has to learn what is the difference between f*x, fx, f(x) and f x.

These things are all seen as *f* times *x* by the system.
Also the *Preview*-button gives absolutely no reliable reproduction of the input. With entering cosx the *Preview*-button will show the formula as: $\cos(x)$. And sin2x is seen as sin(2) *x*. How terrible!

For the rest the *Preview* will show a surplus of brackets.
No programming code for grading is available to check the students answer.

1.2.3 Preparing formulas

1.2.3.1 Introduction

For the formulas needed in the text of the question on the screen you will use HTML-code or MathML-code or even LaTeX. This code is translated by the browser and will provide a proper presentation of your formula on the screen
There are a few ways to generate the MathML-code or LaTeX-code.

For examples with HTML see *Formules with HTML (page 11)* and for LaTeX in section *LaTeX (page 25)*.

It is possible to prepare formulas in the *Algorithm* and use them in the text of the question or in the feedback or in hints by calling them with their algorithmic variable.

Formulas you will present on the screen have to follow the international conventions about spacing italic-or-not etcetera. See also https://www.methakamminga.nl/MapleTA/formules.pdf.

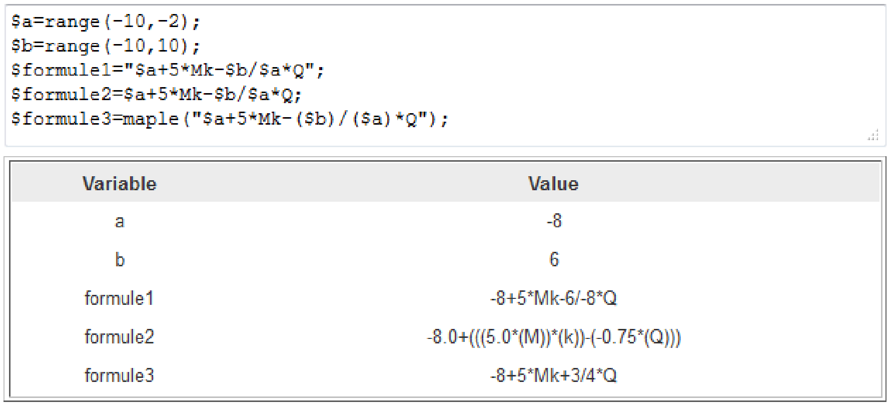
In the next sections you will find several ways to generate MathML-code.

1.2.3.2 Preparing formulas and numbers in the Algorithm

Formulas

The formulas with which you will calculate, you will prepare them in the *Algorithm*.

But be aware: there are different ways to perform these formulas. Look in the next figure.



```
$a=range(-10,-2);
$b=range(-10,10);
$formule1="$a+5*Mk-$b/$a*Q";
$formule2=$a+5*Mk-$b/$a*Q;
$formule3=maple("$a+5*Mk-($b)/($a)*Q");
```

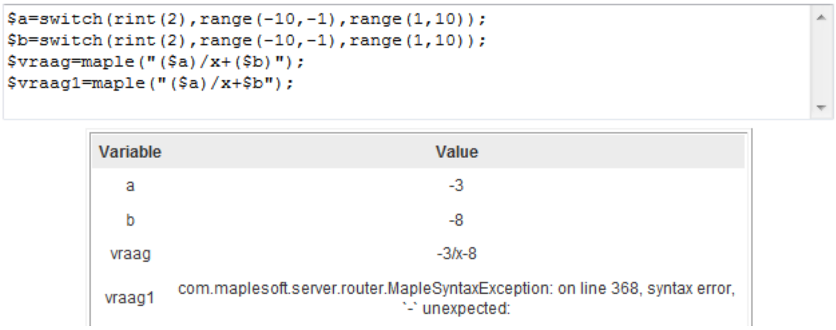
Variable	Value
a	-8
b	6
formule1	-8+5*Mk-6/-8*Q
formule2	-8.0+(((5.0*(M))*(k))-(-0.75*(Q)))
formule3	-8+5*Mk+3/4*Q

Figure 1.12: Formulas preparing in Algorithm

In the figure above you see the following:

- The variable $formule1 is made with quotes around it in order to reproduce exactly what you mean like the character combination Mk and times with * etcetera. But there is a disadvantage with the minus-sign where you see -6/-8. When it is coming to calculation, better to use brackets around the algorithmic variable $a if this is possible a negative variable. The fraction is not simplified.
- The variabele $formule2 is made without quotes around it. The result is awful and the character combination Mk is seen as a multiplication M*k. The fraction 6/8 is changed in decimals and also the other numbers are converted into decimal numbers. Without quotes around the formula it is also not possible to define equations with =-sign.
- The variable $formule3 is the most elegant one to calculate with.
  With maple you can define lots of formulas without problems and maple will calculate exact. There is only one disadvantage. You are at the mercy of the automatic simplifying maple mostly does.
  The code to let maple do it is the following:
  maple("......");
  The character combination Mk is seen as one variable and the fraction is automatically simplified but not converted to a decimal number (unless the variables $a and $b are defined as decimal numbers).

HINT: Take care of brackets around the variables $a and $b if they possibly are negative and in the middle of a formula. Mind that maple does not understand double operators.



```
$a=switch(rint(2),range(-10,-1),range(1,10));
$b=switch(rint(2),range(-10,-1),range(1,10));
$vraag=maple("($a)/x+($b)");
$vraag1=maple("($a)/x+$b");
```

Variable	Value
a	-3
b	-8
vraag	-3/x-8
vraag1	com.maplesoft.server.router.MapleSyntaxException: on line 368, syntax error, `-` unexpected:

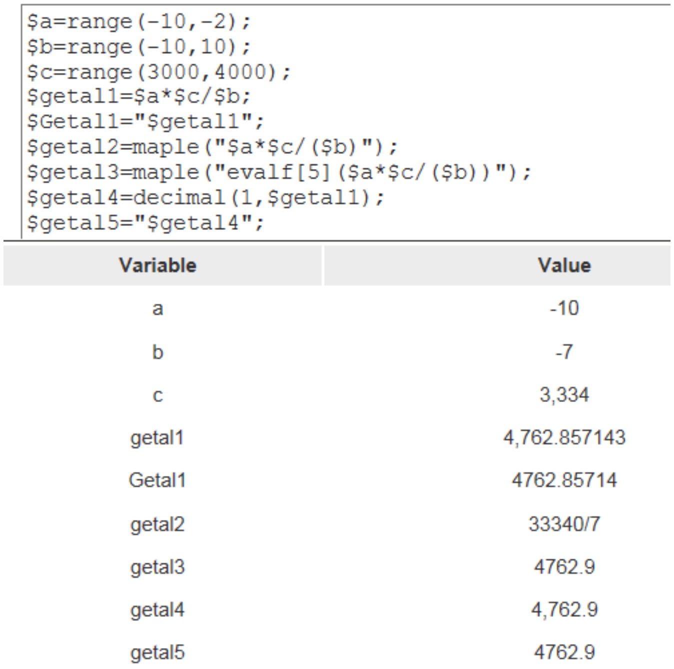
Figure 1.13: Negative values of the variables

With this error-message with `-` unexpected, then you will know that it is important to put brackets round possible negative variables.

Numbers

The trick with the quotes is also very useful with big numbers where you will see thousand separators if you don't use quotes.

See the difference in the following ways to perform numbers.



```
$a=range(-10,-2);
$b=range(-10,10);
$c=range(3000,4000);
$getal1=$a*$c/$b;
$Getal1="$getal1";
$getal2=maple("$a*$c/($b)");
$getal3=maple("evalf[5]($a*$c/($b))");
$getal4=decimal(1,$getal1);
$getal5="$getal4";
```

Variable	Value
a	-10
b	-7
c	3,334
getal1	4,762.857143
Getal1	4762.85714
getal2	33340/7
getal3	4762.9
getal4	4,762.9
getal5	4762.9

Figure 1.14: The performance of numbers

In the figure above you will see how to get rid of the 1000s separator:
You can do it with maple or put quotes around the number.
In maple it is possible to set the number of (5) significance (not the decimals) with evalf[5](..).

Different ways to set the decimals as follows in the *Algorithm* section:

- Decimals with `numfmt` (numeric according to format)
  `$test1=numfmt("#.00",20.9);` gives as result 20.90 (numeric according to format).
  `$test2=numfmt("#0.000",3/4);` gives as a result 0.750.
- Use maple to round:

`$test3=maple("Float(round(100*20.9),-2)");` gives as a result 20.90.
`$test4=maple("Float(round(1000*3/4),-3)");` gives as a result 0.750.

- Let the system do the rounding itself in the following way with `decimal`:
  `$test5=decimal(2,20.90123);` gives 20.9
  `$test6=decimal(2,20.9123);` gives 20.91
  NOTE: This way of rounding does not always give the desired result.
  There is another alternative for better rounding with the following command:
  `$test7=decimal(3,5*1.7^4);`
  gives the **wrong rounding**: 41.76 of the number: 41.7605
  `$test8=maple("MapleTA:-Builtin:-decimal(3,5*1.7^4)");`
  gives the **correct rounding** 41.761 of the number 41.7605.

1.2.3.3 Formulas in the text of the question

Small formulas in the text of the question type LaTex between \(........\).

There is a small Latex manual in section LaTeX *(page 25)* .
After typing the formula in LaTeX press the *Source*-button and then again to see the formula in the text of the question.

Double click on this formula, you will be able to make changes.

For quick symbols there is a special button *Quick Symbols.*

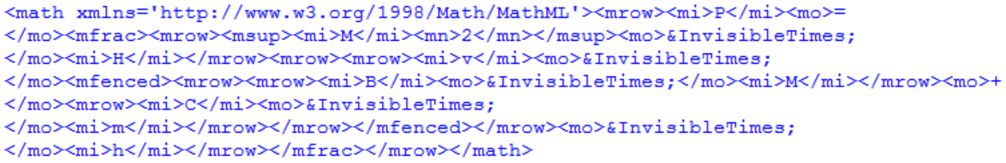
HINT: In the manual part A you will see a lot of hints with short formulas with HTML-code.

HINT: Small formulas in the text, hints or feedback: better not use the Equation Editor but use LaTeX (for example \(\alpha\). See section *(page 25)*

1.2.3.4 What is MathML-code

The underlying code for formulas in the text of your question or feedback is the MathML-code. Möbius supports the MathML in an elegant way. You never type MathML-code yourself.

An example of a coded formula is the following which will be converted by the browser and shows a pretty-print-formula:



```
<math xmlns='http://www.w3.org/1998/Math/MathML'><mrow><mi>P</mi><mo>=
</mo><mfrac><mrow><msup><mi>M</mi><mn>2</mn></msup><mo>&InvisibleTimes;
</mo><mi>H</mi></mrow><mrow><mrow><mi>v</mi><mo>&InvisibleTimes;
</mo><mfenced><mrow><mrow><mi>B</mi><mo>&InvisibleTimes;</mo><mi>M</mi></mrow><mo>+
</mo><mrow><mi>C</mi><mo>&InvisibleTimes;
</mo><mi>m</mi></mrow></mrow></mfenced></mrow><mo>&InvisibleTimes;
</mo><mi>h</mi></mrow></mfrac></mrow></math>
```

Figure 1.15: MathML-code

This code is translated by the browser and you will see the following formula on the screen

$$P = \frac{M^2\,H}{v\,(B\,M + C\,m)\,h}$$

As I said you will never type, read or even see this code.
There are several ways to generate the MathML-code. Explanation how to generate the code is in the next section.

1.2.3.5 Four methods to genarate MathML-code

To genearate the MathML-code in Möbius in four ways.

- In the *Algorithm* with means of the maple-command (most effective).
  `$displayvraag=maple("printf(MathML[ExportPresentation]($a/x+$b))";`

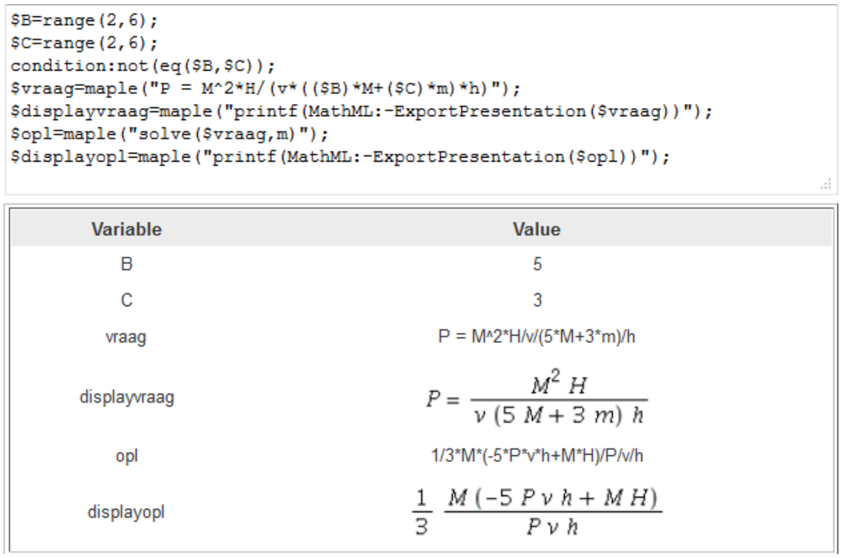
For more information about this method see section *Programming MathML-code in the Algorithm (page 12)*

- In the *Algorithm* by means of the mathml-function of Möbius itself.
  `$displayvraag=mathml("$a/x+$b");`
  For more information about this method see section *Programming MathML-code in the Algorithm (page 12)*
- In the *Algorithm* by means of LaTeX. You can prepare and test the formula by typing LaTeX-code between quotes in the *Algorithm* as follows:
  `$displayvraag="\(\frac{$a}{x}+$b\)";`
  It is also possible to type this code (without quotes) directly in the text of the question.
  For more information about this method see a section with a short LaTeX manual *(page 25)*
  .
- With the *Equation Editor* is the last option. This is the last resort because LaTeX can do it everything even better.
  The MathML-code made by the editor is very confusing and is not very convenient.
  An advantage is the possibility to paste MathML-code from another application (e.g. MathType) into this *Equation Editor*. For more information about MathType see *(page 24)*.

1.2.3.6 Programming MathML-code in the Algorithm

Preparing formulas in MathML-code is very easy in the *Algorithm*.
First define the formula of the question as $vraag. With this formula you will do the calculation in the *Algorithm*.
With the maple command you can convert this formula to MathML as follows:

`maple("printf(MathML:-ExportPresentation($vraag))");` or
`maple("printf(MathML[ExportPresentation]($vraag))");`



```
$B=range(2,6);
$C=range(2,6);
condition:not(eq($B,$C));
$vraag=maple("P = M^2*H/(v*(($B)*M+($C)*m)*h)");
$displayvraag=maple("printf(MathML:-ExportPresentation($vraag))");
$opl=maple("solve($vraag,m)");
$displayopl=maple("printf(MathML:-ExportPresentation($opl))");
```

Variable	Value
B	5
C	3
vraag	P = M^2*H/v/(5*M+3*m)/h
displayvraag	$P = \frac{M^2\ H}{v\ (5\ M+3\ m)\ h}$
opl	1/3*M*(-5*P*v*h+M*H)/P/v/h
displayopl	$\frac{1}{3}\ \frac{M\ (-5\ P\ v\ h+M\ H)}{P\ v\ h}$

Figure 1.16: Formula and MathML preparation in Algorithm

In the figure above you see the equation $vraag. It is a real equation and with the maple command `solve` you can solve this equation (maple does the calculation).
The variabele $displayvraag actually is the MathML-conversion and the browser translates it to a 2D formula. You will never see the MathML-code itself.
The solution is called $opl and in the same way you can make the MathML-code of this very easy by typing:

```
$displayopl=maple("printf(MathML:-ExportPresentation($vraag))");
```

This algorithmic variable you can use in the feedback or somewhere else by calling it by its name.

Check you formulas in the *Algorithm* by clicking on *Refresh algorithm preview.*

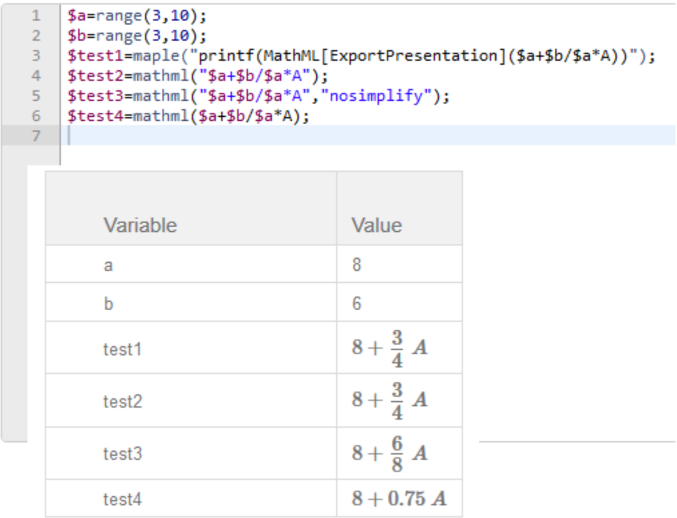
Second way to make MathML-code is with the command: `mathml("...")`like you will see in the following.
This method is rather primitive and has only few possibilities.

```
$a=range(2,5);
$b=range(3,5);
$vraag="$a/x+$b";
$displayvraag=mathml("$vraag");
```

Try also `$displayvraag = mathml("$vraag","nosimplify");`
to prevent simplifying.

BE AWARE: With the conversion of a formula into MathML-code you will get a formule with the purpose to show on the screen and NOT to calculate with! With the variable $vraag you can calculate in the *Algortihm*.
But with the variable $displayvraag (MathML-coded formula) you are not able to calculate.
It is smart to give names with "display" in it to recognize MathML-coded formulas.

Look into the following for ideas to have formulas on the screen with the MathML-conversion:



```
$a=range(3,10);
$b=range(3,10);
$test1=maple("printf(MathML[ExportPresentation]($a+$b/$a*A))");
$test2=mathml("$a+$b/$a*A");
$test3=mathml("$a+$b/$a*A","nosimplify");
$test4=mathml($a+$b/$a*A);
```

Variable	Value
a	8
b	6
test1	$8+\frac{3}{4}A$
test2	$8+\frac{3}{4}A$
test3	$8+\frac{6}{8}A$
test4	$8+0.75\,A$

Figure 1.17: Different ways to generate MathML-code

In the figure you see it is not always required to let maple do the conversion into MathML by the call
`maple("printf(MathML[ExportPresentation](...))");`
Möbius can do it for you with `mathml("...")` and is mostly quick and easy for simple formulas with no character combinations. With the extra option "nosimplify" in the command mathml("...") it will prevent the automatic simplifying.

Dropping the quotes in the `mathml` command, you will see decimal numbers despite the variables $a and $b are not decimal numbers.

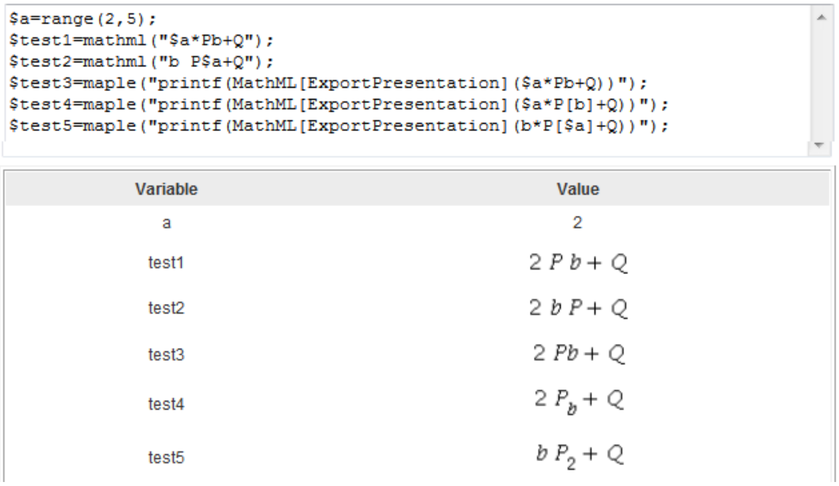
HINT: Here you don't see brackets round the variables in $a+$b/$a*A because it is not absolutely necessary.
Here the variables are all positive in a range between 3 and 10.

But if the variables are possibly negative then it is wise to put brackets round them. Se also in *Figure 1.16 (page 12)*. For more information see section *Formulas in Algorithm (page 9)*.

HINT: The function `mathml("...")` of the system itself is less powerful than the method with the maple command: `maple("printf(MathML[ExportPresentation](...))")`. With `mathml("...")` it is impossible to handle character combinations, subscript and inequations.

The maple command `maple("printf(MathML[ExportPresentation](...))")` is more powerful and is able to convert character combinations and a lot more like subscript, Greek, integrals, inequations differentiation and even more like matrices and vectors etcetera.

Be aware of character combinations! Impossible with `mathml("...")` because that will be converted to a multiplication with space between the characters as you will see in the next figure.

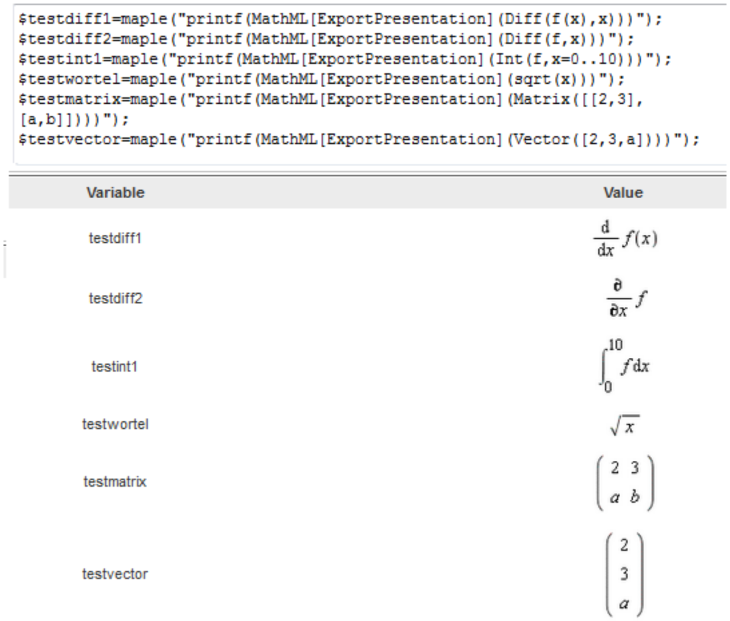


```
$a=range(2,5);
$test1=mathml("$a*Pb+Q");
$test2=mathml("b P$a+Q");
$test3=maple("printf(MathML[ExportPresentation]($a*Pb+Q))");
$test4=maple("printf(MathML[ExportPresentation]($a*P[b]+Q))");
$test5=maple("printf(MathML[ExportPresentation](b*P[$a]+Q))");
```

Variable	Value
a	2
test1	$2\ P\ b + Q$
test2	$2\ b\ P + Q$
test3	$2\ Pb + Q$
test4	$2\ P_b + Q$
test5	$b\ P_2 + Q$

Figure 1.18: MathML with subscript and character combinations

In the following figure you will see more examples of formulas where the method to generate MathML with maple is more powerful:

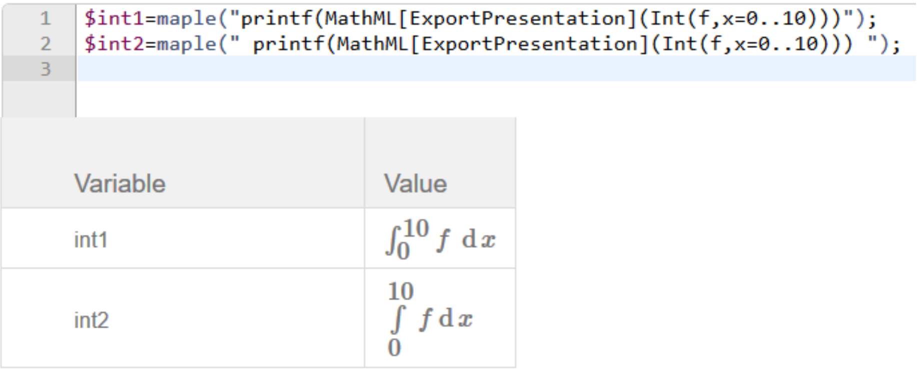


```
$testdiff1=maple("printf(MathML[ExportPresentation](Diff(f(x),x)))");
$testdiff2=maple("printf(MathML[ExportPresentation](Diff(f,x)))");
$testint1=maple("printf(MathML[ExportPresentation](Int(f,x=0..10)))");
$testwortel=maple("printf(MathML[ExportPresentation](sqrt(x)))");
$testmatrix=maple("printf(MathML[ExportPresentation](Matrix([[2,3],
[a,b]])))");
$testvector=maple("printf(MathML[ExportPresentation](Vector([2,3,a])))");
```

Variable	Value
testdiff1	$\frac{\mathrm{d}}{\mathrm{d}x} f(x)$
testdiff2	$\frac{\partial}{\partial x} f$
testint1	$\int_0^{10} f\,\mathrm{d}x$
testwortel	$\sqrt{x}$
testmatrix	$\begin{pmatrix} 2 & 3 \\ a & b \end{pmatrix}$
testvector	$\begin{pmatrix} 2 \\ 3 \\ a \end{pmatrix}$

Figure 1.19: MathML-Formules made by Maple

HINT: Be aware of spaces next to the quotes using `printf`. Sometimes it makes a difference if you will use spaces or not:
`maple(" printf(MathML[ExportPresentation](...) ) ")`
For example look at the integral sign and the boundaries of the integral.

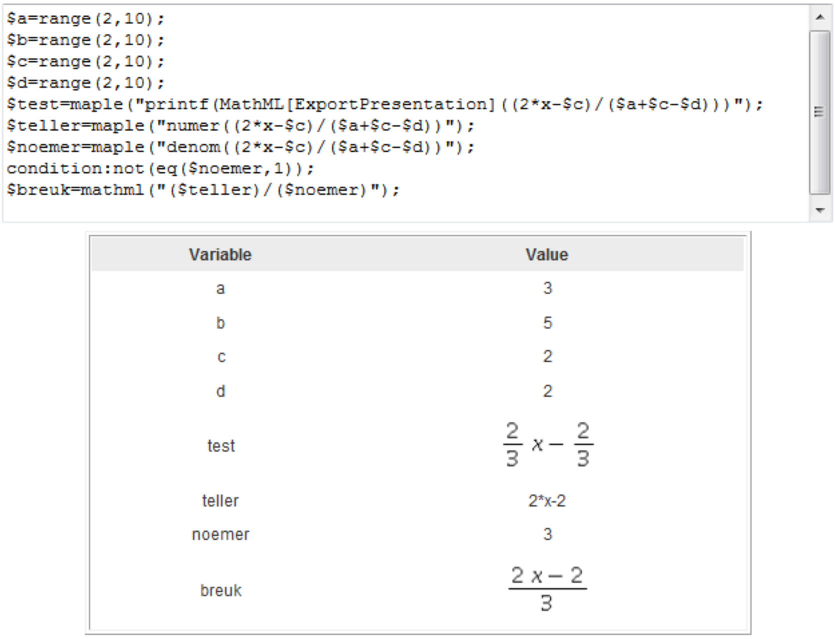


```
$int1=maple("printf(MathML[ExportPresentation](Int(f,x=0..10)))");
$int2=maple(" printf(MathML[ExportPresentation](Int(f,x=0..10))) ");
```

Variable	Value
int1	$\int_0^{10} f\ \mathrm{d}x$
int2	$\overset{10}{\underset{0}{\int}} f\,\mathrm{d}x$

Figure 1.20: Space with printf

HINT: Use a *combination* of more components with `mathml("...")` and `maple("printf(MathML[ExportPresentation](...) )")` as a trick like the following:



```
$a=range(2,10);
$b=range(2,10);
$c=range(2,10);
$d=range(2,10);
$test=maple("printf(MathML[ExportPresentation]((2*x-$c)/($a+$c-$d)))");
$teller=maple("numer((2*x-$c)/($a+$c-$d))");
$noemer=maple("denom((2*x-$c)/($a+$c-$d))");
condition:not(eq($noemer,1));
$breuk=mathml("($teller)/($noemer)");
```

Variable	Value
a	3
b	5
c	2
d	2
test	$\frac{2}{3}x-\frac{2}{3}$
teller	2*x-2
noemer	3
breuk	$\frac{2x-2}{3}$

Figure 1.21: MathML as a combination of methods

In the figure you see it is not easy to make a fraction with MathML without simplification.
Try to combine different methods for coding MathML.
In the *Algorithm* you keep full control by checking the effect of your code. Finally you can use LaTeX to make the formula if other methods fail.

HINT: There is an extra possibility nowadays to prevent automatic simplification:
See in section *(page 16)*.

HINT: Another possibility is to combine two separate formulas is the formula you see in the figure.
Use × ([AltGr][=]) or the middot with the button for Special Characters if you want to multiply the two formulas:

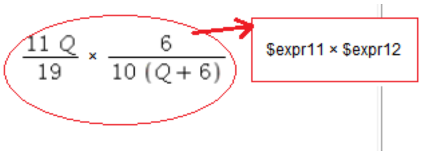


Figure 1.22: MathML preparing in parts

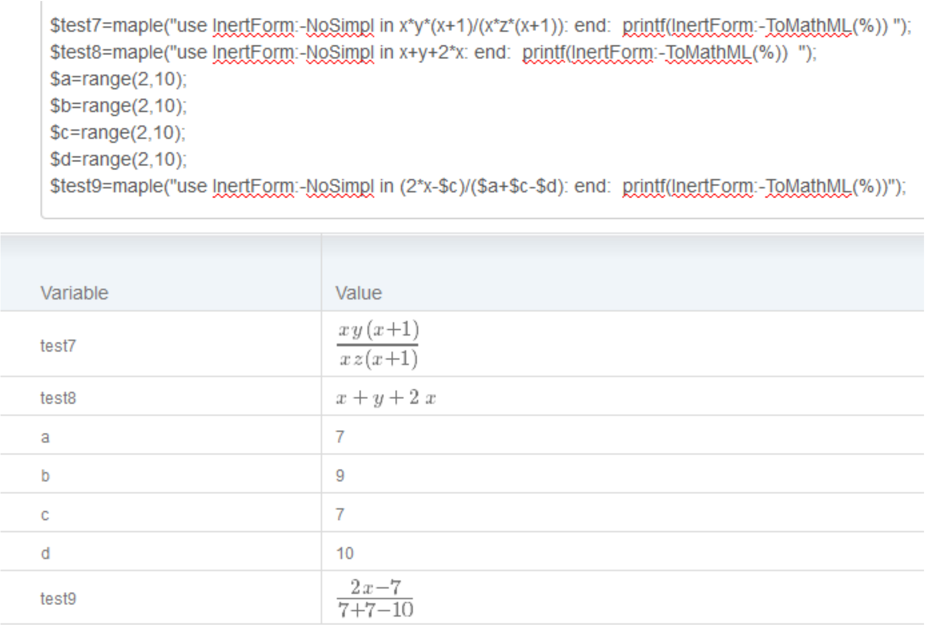
HINT: There are lots of solutions for problems by using LaTeX-code. Look in the section about LaTeX *(page 25)*.

1.2.3.7 Preventing automatic simplification

There are different ways to prevent automatic simplification of a formula you want to show in the text field of your question.

- With the package InertForm:
  In case of fractions and polynomials you often see the simplification Maple does with the conversion to MathML.
  Using the package InertForm, Maple will not simplify the formula. Try out some things in the *Algorithm* with the following:

```
$test7=maple("use InertForm:-NoSimpl in x*y*(x+1)/(x*z*(x+1)):
 end: printf(InertForm:-ToMathML(%))");
```



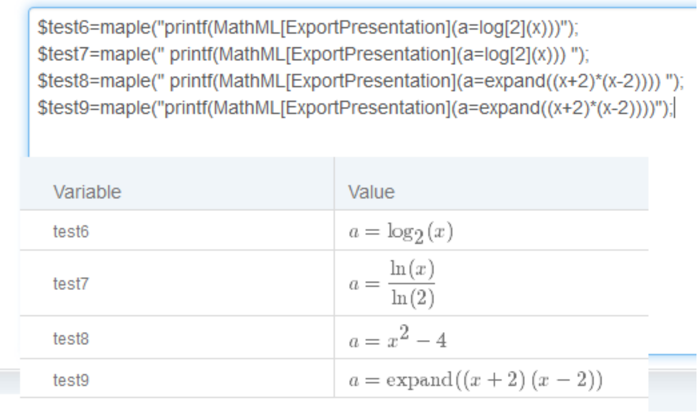
```
$test7=maple("use InertForm:-NoSimpl in x*y*(x+1)/(x*z*(x+1)): end:  printf(InertForm:-ToMathML(%)) ");
$test8=maple("use InertForm:-NoSimpl in x+y+2*x: end:  printf(InertForm:-ToMathML(%))  ");
$a=range(2,10);
$b=range(2,10);
$c=range(2,10);
$d=range(2,10);
$test9=maple("use InertForm:-NoSimpl in (2*x-$c)/($a+$c-$d): end:  printf(InertForm:-ToMathML(%))");
```

Variable	Value
test7	$\frac{x\,y\,(x+1)}{x\,z(x+1)}$
test8	$x + y + 2\,x$
a	7
b	9
c	7
d	10
test9	$\frac{2x-7}{7+7-10}$

Figure 1.23: Prevent automatic simplification

- As we said earlier, the trick with the space in combination with the quotes and the command `printf`.
  A Maple command will not be executed if you place the quotes directly next to the command. But if you use an extra space between the quotes and the `printf` command, you will see the Maple command will be executed.

```
$test6=maple("printf(MathML[ExportPresentation](a=log[2](x)))");
$test7=maple(" printf(MathML[ExportPresentation](a=log[2](x)))
 ");
$test8=maple(" printf(MathML[ExportPresentation](a=expand((x
+2)*(x-2)))) ");
$test9=maple("printf(MathML[ExportPresentation](a=expand((x
+2)*(x-2))))");
```



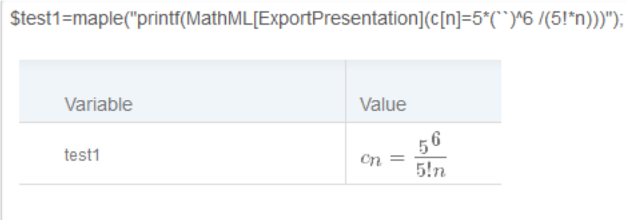
```
$test6=maple("printf(MathML[ExportPresentation](a=log[2](x)))");
$test7=maple(" printf(MathML[ExportPresentation](a=log[2](x))) ");
$test8=maple(" printf(MathML[ExportPresentation](a=expand((x+2)*(x-2)))) ");
$test9=maple("printf(MathML[ExportPresentation](a=expand((x+2)*(x-2))))");
```

Variable	Value
test6	$a = \log_2(x)$
test7	$a = \frac{\ln(x)}{\ln(2)}$
test8	$a = x^2 - 4$
test9	$a = \mathrm{expand}((x+2)(x-2))$

Figure 1.24: Use quotes with Maple's MathML conversion

HINT: The presentation of the logarithm with subscript, like most calculation displays, is very common: $\log_4(20)$.

- Trick with **backward quotes** is easy to let Maple do what you want and prevent it from automatic simplification.
  With these backward quotes Maple is not able to do the calculation as you can see in the following.
  For example:
  `$test1=maple("printf(MathML[ExportPresentation](c[n]=5*(``)^6 / (5!*n)))");`
  This will look like $\frac{5^6}{5!\,n}$. (But with LaTeX it goes easier.)



```
$test1=maple("printf(MathML[ExportPresentation](c[n]=5*(``)^6 /(5!*n)))");
```

Variable	Value
test1	$c_n = \frac{5^6}{5!n}$

Figure 1.25: MathML and quotes

To prevent the calculation of 5^6 by Maple you let Maple do the calculation of a space to the power of 6 which is not possible.
With 5*``^6 ( of 5*(``)^6) you put Maple on the wrong leg.

It is indeed possible to do the MathML convertion of 5^6 with `mathml` but in that case the ! is not supported and also subscript is not supported.

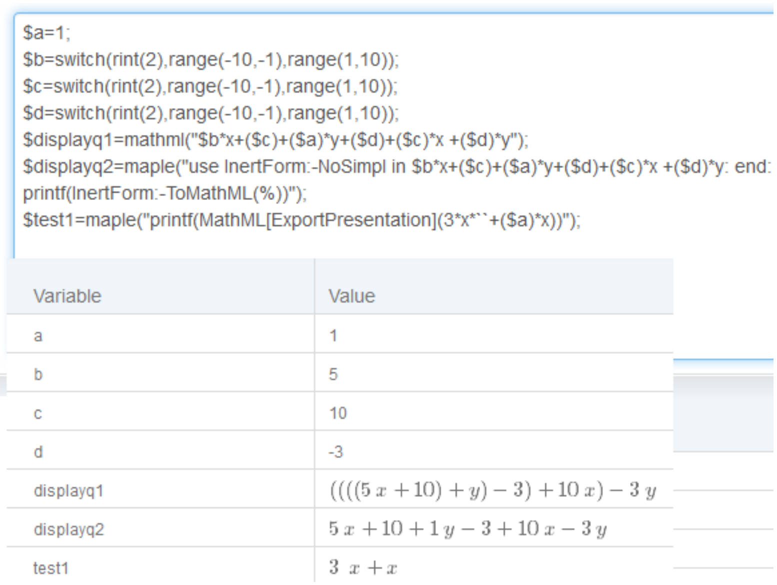
Use no spaces with the quotes in `"printf` ....because you don't want the evaluation of 5!

- Four ways to **prevent automatic simplification**:

- With mathml("...")

Not recommanded. This way is very poor.

- With the InertForm package in Maple:
  `$displayq2=maple("use InertForm:-NoSimpl in $b*x+($c)+($a)*y +($d)+($c)*x +($d)*y: end: printf(InertForm:-ToMathML(%))");`
  Disadvantage is the number 1, you will see $1\,y$ in the formula. A work around is not including the number 1 in the algorithmic variable.
- Working with single quotes in Maple:
  ``$test1=maple("printf(MathML[ExportPresentation](3*x*`` +($a)*x))");``
  Here you see that Maple is not able to evaluate a multiplication with a space. Put the space where it will do no harm for example in front of the + sign.
- As a last escape use LaTeX. Disadvantage of working with algorithmic variables which can be as well positive as negative. For more information see section *(page 25)*.



```
$a=1;
$b=switch(rint(2),range(-10,-1),range(1,10));
$c=switch(rint(2),range(-10,-1),range(1,10));
$d=switch(rint(2),range(-10,-1),range(1,10));
$displayq1=mathml("$b*x+($c)+($a)*y+($d)+($c)*x +($d)*y");
$displayq2=maple("use InertForm:-NoSimpl in $b*x+($c)+($a)*y+($d)+($c)*x +($d)*y: end:
printf(InertForm:-ToMathML(%))");
$test1=maple("printf(MathML[ExportPresentation](3*x*``+($a)*x))");
```

Variable	Value
a	1
b	5
c	10
d	-3
displayq1	$((((5\,x+10)+y)-3)+10\,x)-3\,y$
displayq2	$5\,x+10+1\,y-3+10\,x-3\,y$
test1	$3\;\,x+x$

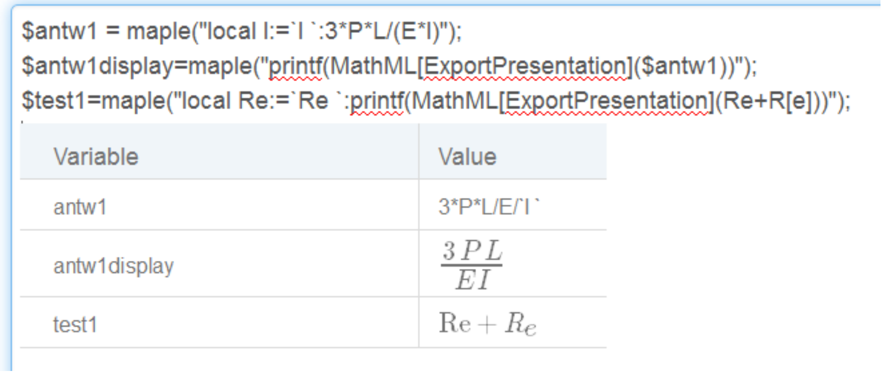
Figure 1.26: MathML and quotes

HINT: To prevent "Re" in Maple to be seen as the real part of an expression, use quotes like `Re ` (including the space!!!).

Then it is not seen as the real part of... But you will see the result in italic. (See *Figure 1.27 (page 20)*).
Otherwise use LaTeX to prepare the formula for the text of the question. See *(page 25)*.

HINT: The capital I in Maple stands for the imaginary unit and will be performed in roman and not italic.

Do you want the character *I* to use as a variable, then you have to communicate that it is not the imaginary unit.
With back quotes you can opt for first giving the command ``local I=`I ` ``.
Mind the space and look at the result in the figure below.



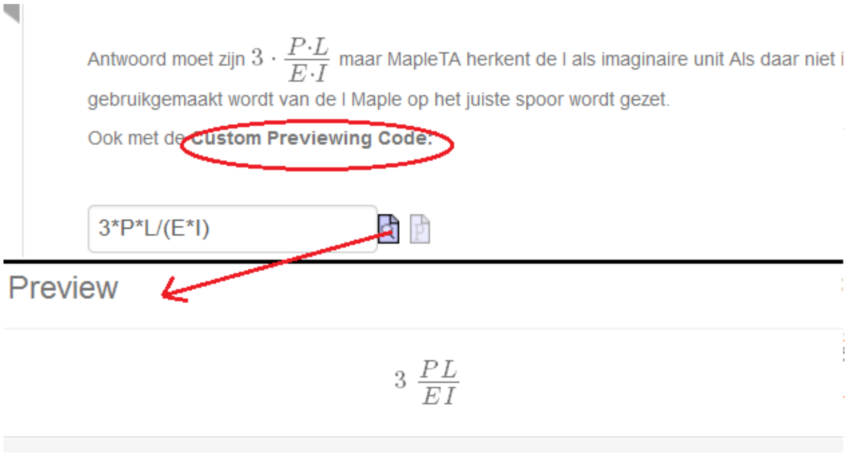
```
$antw1 = maple("local I:=`I `:3*P*L/(E*I)");
$antw1display=maple("printf(MathML[ExportPresentation]($antw1))");
$test1=maple("local Re:=`Re `:printf(MathML[ExportPresentation](Re+R[e]))");
```

Variable	Value
antw1	3*P*L/E/`I `
antw1display	$\frac{3\,P\,L}{E\,I}$
test1	$\mathrm{Re}+R_e$

Figure 1.27: Imaginary unit

For example: `$test=maple("local I=`I `:printf(MathML[ExportPresentation](Y=C+I))");`

HINT: Use the *Custom Previewing Code* in the *Maple graded* question type to prevent from automatic simplification if the student presses *Preview*.
In the rubrick *Custom Previewing Code* of the *Maple graded* question type, you can put the following:
`local I=`I `:printf(MathML[ExportPresentation]($RESPONSE));` (mind the space).



Antwoord moet zijn $3 \cdot \frac{P \cdot L}{E \cdot I}$ maar MapleTA herkent de I als imaginaire unit Als daar niet i
gebruikgemaakt wordt van de I Maple op het juiste spoor wordt gezet.
Ook met de **Custom Previewing Code:**

3*P*L/(E*I)

$$3\,\frac{P\,L}{E\,I}$$

Figure 1.28: Custom PlottingCode

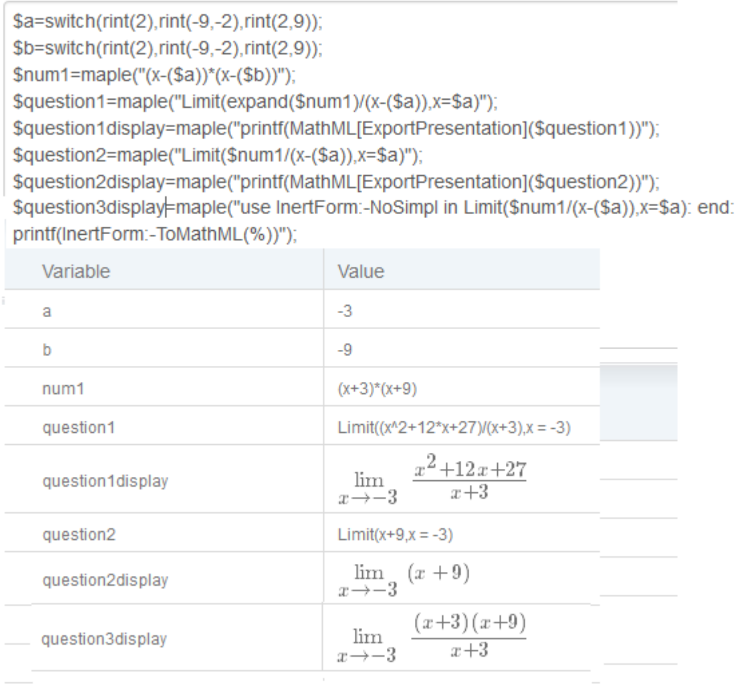
A few more examples preparing formulas using in the feedback.

Formulas not to be simplified: you make them with the InertForm package in Maple:

```
$a=switch(rint(2),rint(-9,-2),rint(2,9));
$b=switch(rint(2),rint(-9,-2),rint(2,9));
$num1=maple("(x-($a))*(x-($b))");
$question1=maple("Limit(expand($num1)/(x-($a)),x=$a)");
$question1display=maple("printf(MathML[ExportPresentation]
($question1))");
```

```
$question2=maple("Limit($num1/(x-($a)),x=$a)");
$question2display=maple("printf(MathML[ExportPresentation]
($question2))");
```

```
$question3display=maple("use InertForm:-NoSimpl in Limit($num1/(x-
($a)),x=$a): end: printf(InertForm:-ToMathML(%))");
```



```
$a=switch(rint(2),rint(-9,-2),rint(2,9));
$b=switch(rint(2),rint(-9,-2),rint(2,9));
$num1=maple("(x-($a))*(x-($b))");
$question1=maple("Limit(expand($num1)/(x-($a)),x=$a)");
$question1display=maple("printf(MathML[ExportPresentation]($question1))");
$question2=maple("Limit($num1/(x-($a)),x=$a)");
$question2display=maple("printf(MathML[ExportPresentation]($question2))");
$question3display=maple("use InertForm:-NoSimpl in Limit($num1/(x-($a)),x=$a): end:
printf(InertForm:-ToMathML(%))");
```

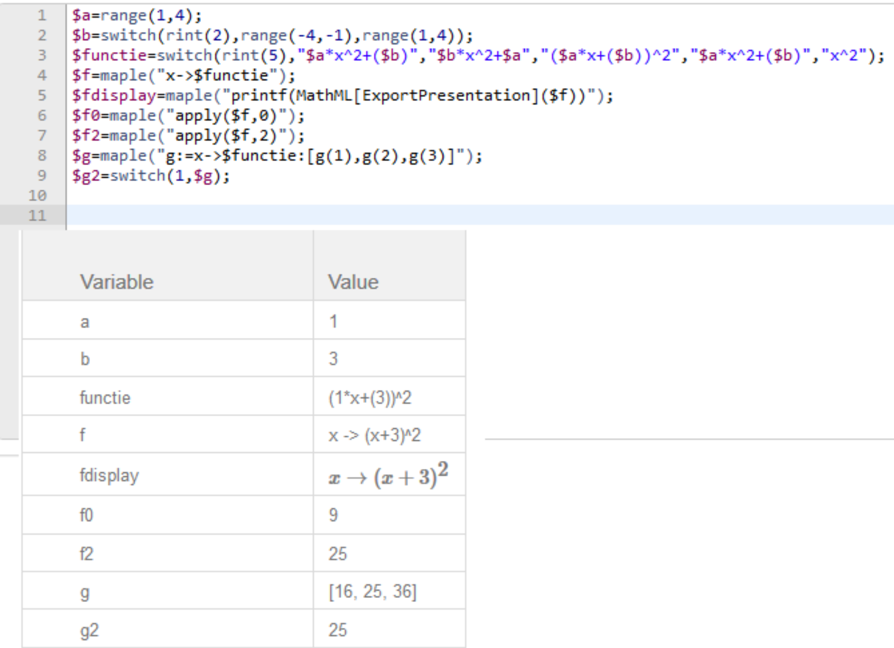
Variable	Value
a	-3
b	-9
num1	(x+3)*(x+9)
question1	Limit((x^2+12*x+27)/(x+3),x = -3)
question1display	$\lim_{x \to -3} \frac{x^2+12x+27}{x+3}$
question2	Limit(x+9,x = -3)
question2display	$\lim_{x \to -3} (x+9)$
question3display	$\lim_{x \to -3} \frac{(x+3)(x+9)}{x+3}$

Figure 1.29: Formulas with quotes and inert form

1.2.3.8 Defining a function and generate values

HINT: A nice way to **define a function** and use it in the *Algorithm* to generate values is the following:

```
$a=range(1,4);
$b=switch(rint(2),range(-4,-1),range(1,4));
$functie=switch(rint(5),"$a*x^2+($b)","$b*x^2+$a","($a*x
+($b))^2","$a*x^2+($b)","x^2");
$f=maple("x->$functie");
$fdisplay=maple("printf(MathML[ExportPresentation]($f))");
$f0=maple("apply($f,0)");
$f2=maple("apply($f,2)");
$g=maple("g:=x->$functie:[g(1),g(2),g(3)]");
$g2=switch(1,$g);
```



```
$a=range(1,4);
$b=switch(rint(2),range(-4,-1),range(1,4));
$functie=switch(rint(5),"$a*x^2+($b)","$b*x^2+$a","($a*x+($b))^2","$a*x^2+($b)","x^2");
$f=maple("x->$functie");
$fdisplay=maple("printf(MathML[ExportPresentation]($f))");
$f0=maple("apply($f,0)");
$f2=maple("apply($f,2)");
$g=maple("g:=x->$functie:[g(1),g(2),g(3)]");
$g2=switch(1,$g);
```

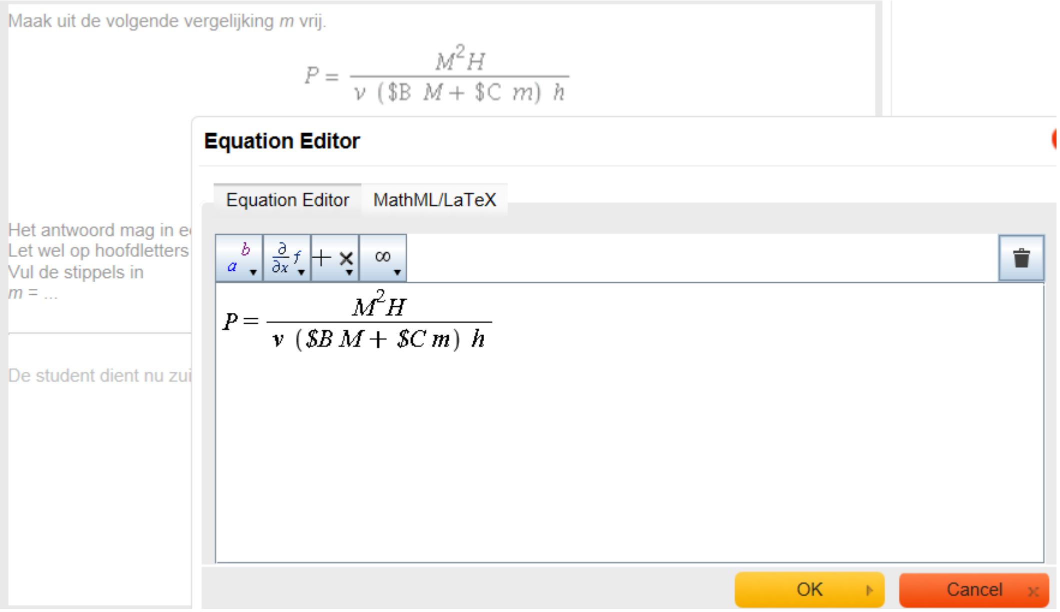
Variable	Value
a	1
b	3
functie	(1*x+(3))^2
f	x -> (x+3)^2
fdisplay	$x \to (x+3)^2$
f0	9
f2	25
g	[16, 25, 36]
g2	25

Figure 1.30: Defining a function

In the figure you see how to define a function *f* with a procedure (with arrow).
Genarate the values with the Maple command `apply`.
Another way is to stay within Maple and generate in the same session a list with values like function *g*.

1.2.3.9 MathML with the Equation Editor

Most of the time you will prepare the formula in the *Algorithm*. But there is also a way to make the formula with an editor by clicking on the Sigma-button in the text of the question (or *Hints* or *Feedback*).
A dialog box with tool-buttons opens to make your formula.



Maak uit de volgende vergelijking *m* vrij.

$$P = \frac{M^2 H}{v\ (\$B\ M + \$C\ m)\ h}$$

Het antwoord mag in e
Let wel op hoofdletters
Vul de stippels in
m = ...

De student dient nu zui

Equation Editor

$$P = \frac{M^2 H}{v\ (\$B\ M + \$C\ m)\ h}$$

Figure 1.31: The Equation Editor

In the figure you see that you can use algoritmic variables (dollar sign). But take care that these are mostly numbers and you want to see numbers in roman. That is the great disadvantage with the editor.

For the rest there are a lot possibilities to use special signs in the expression.

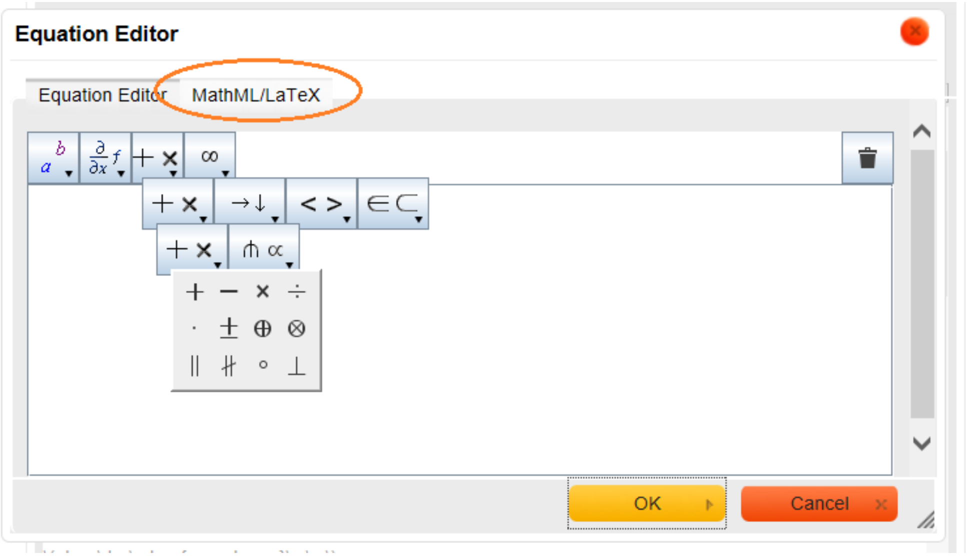


Figure 1.32: Equation Editor in Möbius

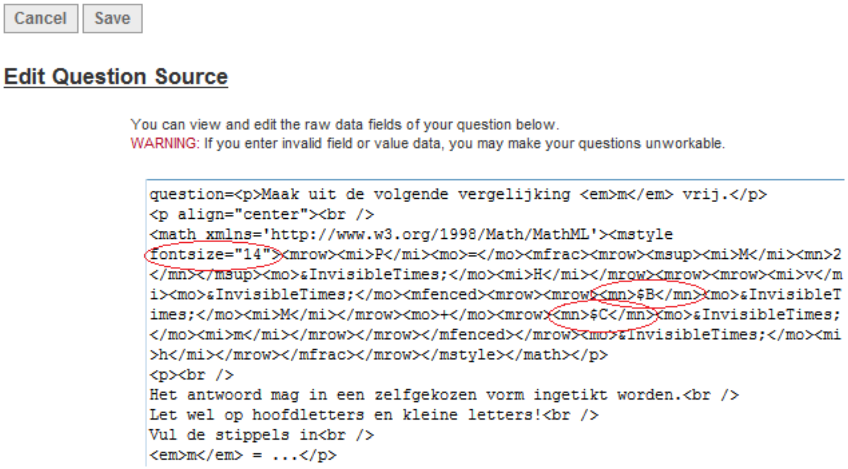
HINT: With the tab MathML/LaTeX in the *Equation Editor, Figure 1.31 (page 22)* you can change the MathML-code if you understand the MathML language. You can change font and color and make algoritmic variables not italic but roman.

It is also possible to paste MathML code made another program like MathType or Maple.

HINT: If you do some modifications in the code in the tab MathML/LaTeX of the Editor, press directly on the OK button.

HINT: Modifications in the code is easier to do in the source of the question itself. Open the question with *Edit Source*. In this source code of the question you see clean code and to modify this code is very easy. For example add some things like: `fontsize="14"` or italic or not italic to change the tags `<mi>...</mi>` in `<mn>...</mn>`, see in the figure.
For example <mi>sin</mi> change it in <mn>sin</mn> and in this case for example <mi>$C</mi> changing in <mn>$C</mn>.

<mi> means mathematical italic and <mn> means mathematical normal.



Edit Question Source

You can view and edit the raw data fields of your question below.
WARNING: If you enter invalid field or value data, you may make your questions unworkable.

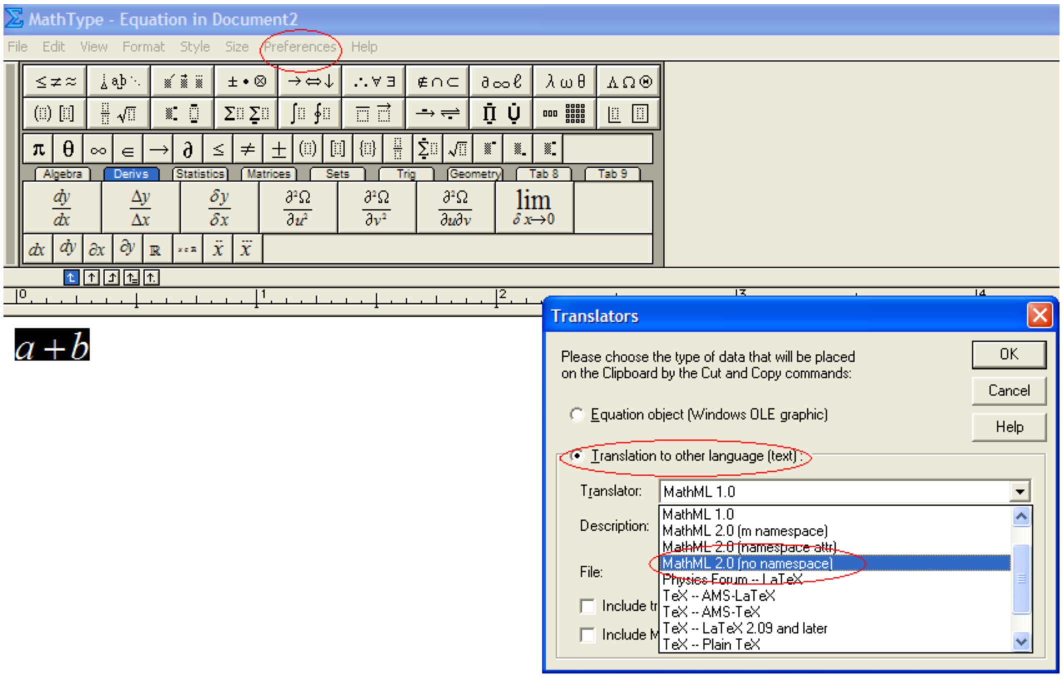
```
question=<p>Maak uit de volgende vergelijking <em>m</em> vrij.</p>
<p align="center"><br />
<math xmlns='http://www.w3.org/1998/Math/MathML'><mstyle
fontsize="14"><mrow><mi>P</mi><mo>=</mo><mfrac><mrow><msup><mi>M</mi><mn>2
</mn></msup><mo>&InvisibleTimes;</mo><mi>H</mi></mrow><mrow><mrow><mi>v</m
i><mo>&InvisibleTimes;</mo><mfenced><mrow><mrow><mn>$B</mn><mo>&InvisibleT
imes;</mo><mi>M</mi></mrow><mo>+</mo><mrow><mn>$C</mn><mo>&InvisibleTimes;
</mo><mi>m</mi></mrow></mrow></mfenced></mrow><mo>&InvisibleTimes;</mo><mi
>h</mi></mrow></mfrac></mrow></mstyle></math></p>
<p><br />
Het antwoord mag in een zelfgekozen vorm ingetikt worden.<br />
Let wel op hoofdletters en kleine letters!<br />
Vul de stippels in<br />
<em>m</em> = ...</p>
```

Figure 1.33: Modifying MathML code in the source code of the question

HINT: Better to use LaTeX! (See section *(page 25)*.)

1.2.3.10 MathML-code with MathType

Special formulas made in MathType are easy to implement in Möbius. To prepare your MathType application do the following:



Translators

Please choose the type of data that will be placed on the Clipboard by the Cut and Copy commands:

○ Equation object (Windows OLE graphic)

● Translation to other language (text) :

Translator: MathML 1.0

Description:

File:

□ Include tr

□ Include M

MathML 1.0
MathML 2.0 (m namespace)
MathML 2.0 (namespace attr)
MathML 2.0 (no namespace)
Physics Forum -- LaTeX
TeX -- AMS-LaTeX
TeX -- AMS-TeX
TeX -- LaTeX 2.09 and later
TeX -- Plain TeX

Figure 1.34: With MathType generating MathMLcode

Start the application MathType and go to *Preferences* and opt for *Translators*.

Set the following policies: see *Figure 1.34 (page 24)*.
With the radio button opt for *Translation to other language* and opt for the *Translator: MathML -- WebEQ compatible* or *MathML 2.0 [no namespace]*. No need to check the checkboxes below .

If you make a formual in MathType just select and copy and paste it in the *Equation Editor* of Möbius in the tab MathML

1.2.3.11 LaTeX

LaTeX is now supported in Möbius. Even in the *Equation Editor* the LaTeX-code is supported.

It is also possible to use the LaTeX-code directly in the text but preparing this code in the *Algorithm* is very convenient.
Type LaTeX directly in the text and you can make changes later. Doubbleclick on the formula and you will get into the Editor. In the LaTeX tab you can make your changes.

In the line of the text start a formula with backslash and bracket: \(and type the LaTeX-code. At the end you close with \).
The code in between the system will be recognized as LaTeX and it will be converted.

\(..........\)

If you want a formula centered in the text, use \[....\] is stead of normal brackets.

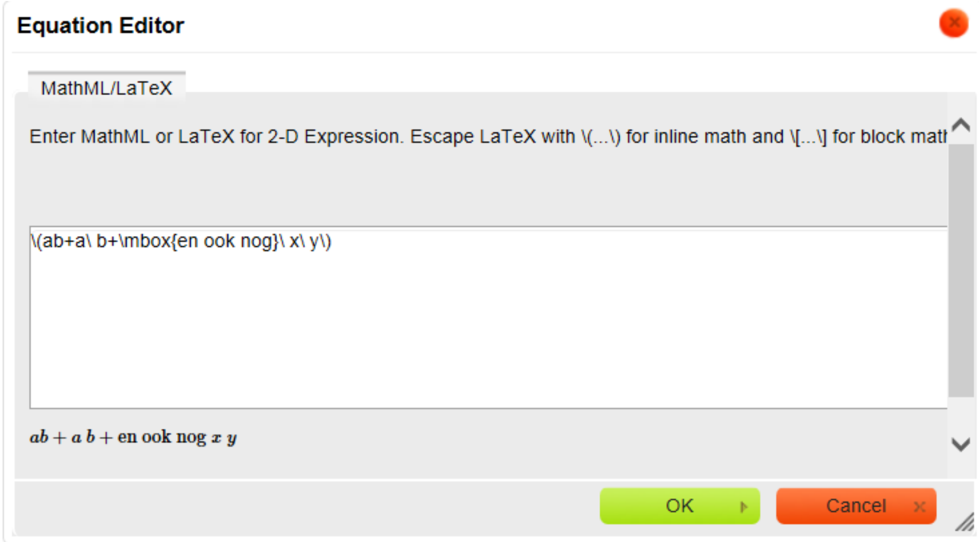
\[........\]

This way has as result a better performance of the formula for subscript of integrals, sums and so on.

(See *Figure 1.50 (page 31)* limits and summation).
But centralize a formula is also possible with the formula in line and then centralize the text.
You can type the LaTeX-code directly in the text with the special brackets with backslash. But you can also use the Editor by clicking on the Sigma button. Go directly to the LaTeX-tab and you see immediately the result of your typing.

If you don't have any knowledge of the LaTeX code, then you can find here a short manual. But the code is also generable on the following web site:

https://www.codecogs.com/latex/eqneditor.php



Equation Editor

Enter MathML or LaTeX for 2-D Expression. Escape LaTeX with \(...\) for inline math and \[...\] for block matt

```
\(ab+a\ b+\mbox{en ook nog}\ x\ y\)
```

$ab + a\ b + \text{en ook nog}\ x\ y$

Figure 1.35: LaTeX with the Equation Editor

HINT: Modifications in the LaTeX-code you can do by clicking on the formula and in the LaTeX-tab you can change things. Or go to the source code of the questions to make the changes.

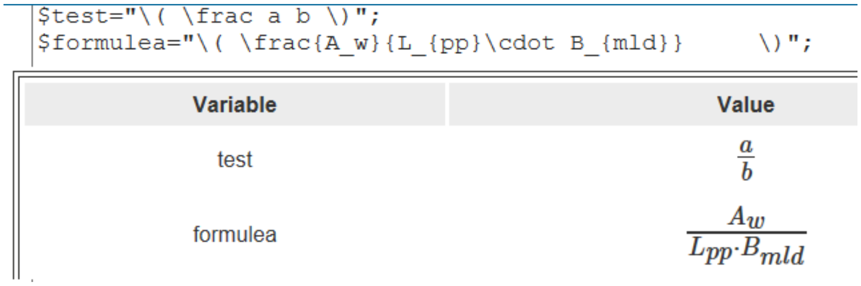
HINT: Recommendation is to prepare the formula in the *Algorithm* as follows:

```
$test="\( \frac a b \)";
```

for a small fraction, even better is `\(\frac{a}{b}\)`.

```
$formulea="\( \frac{A_w}{L_{pp}\cdot B_{mld}} \)";
```

In the figure you will see the result:



```
$test="\( \frac a b \)";
$formulea="\( \frac{A_w}{L_{pp}\cdot B_{mld}}     \)";
```

Variable	Value
test	$\frac{a}{b}$
formulea	$\frac{A_w}{L_{pp}\cdot B_{mld}}$

Figure 1.36: Example with LaTeX

HINT: Put the parts of the formula between braces {...} like the denominator of a fraction.

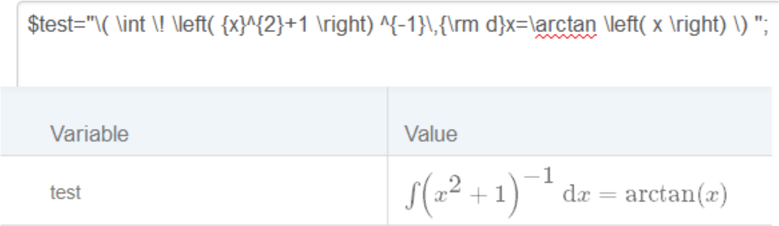
HINT: If you have the computer algebra system Maple, you can generate the LaTeX code as follows:

```
> latex(Int(1/(x^2+1), x) = int(1/(x^2+1), x));

\int \! \left( {x}^{2}+1 \right) ^{-1}\,{\rm d}x=\arctan \left( x
 \right)
```

With copy and paste this output in Möbius and place it between the brackets: \(..........\) in the text or even in the *Algorithm*.
With the use of \left (and \right) the brackets will be as big as needed in the formula.
The sign \! will skip a space or makes it smaller.

An example of the use in the *Algorithm* you can see in the figure.



```
$test="\( \int \! \left( {x}^{2}+1 \right) ^{-1}\,{\rm d}x=\arctan \left( x \right) \) ";
```

Variable	Value
test	$\int \left(x^{2}+1 \right)^{-1} \mathrm{d}x = \arctan(x)$

Figure 1.37: LateX preparing in the Algorithme

HINT: Lots of information about LaTeX you can find on the web with many special characters and tools for your formula.

https://en.wikibooks.org/wiki/LaTeX/Mathematics
http://w2.syronex.com/jmr/tex/texsym.old.html

or

http://nl.wikibooks.org/wiki/LaTeX/Wiskundige_formules

http://garsia.math.yorku.ca/MPWP/LATEXmath/latexsym.html

1.2.3.11.1 Variables, spaces and text

Variables in LaTeX are automatically in italic.

Half spaces between two variables you will get with \, (backslash and comma).

A **whole space** you will get with backslash and space (space with the key board).

A **medium space** with \: (backslash and colon).

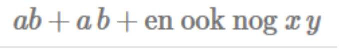
A **big space** with \; (backslash and semicolon).

A **negative space** with \! (to skip the automatic space in the formula).

A piece of **text in roman** with \mbox{...} or \ text{...}.
This text is default in the font Times New Roman.
Within the curly brackets {..} you can type normal spaces for text.

Try it out yourself with the following:

```
\(ab+a\,b+\mbox{en ook nog}\ x\, y\)
```



$ab+a\,b+\mbox{en ook nog}\ x\, y$

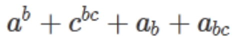
Figure 1.38: LaTeX with variables and spaces and text

1.2.3.11.2 Superscript and subscript

Superscript is easy to do with ^ and subscript with _ (underscore).

Try out yourself with the following:

```
\(a^b+c^{bc}+a_b+a_{bc}\)
```



$a^b+c^{bc}+a_b+a_{bc}$

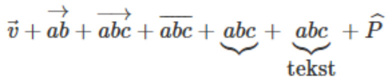
Figure 1.39: LaTeX with superscript and subscript

Mind the curly brackets {..} for more extensive superscript and subscript.
You will also see these superscript and subscript later with integrals and summation.

1.2.3.11.3 Vectors and underlines

In the following examples you see how to decorate your formula.

```
\(\vec{v}+\vec{ab}+\overrightarrow{abc}+\overline{abc}+
\underbrace{abc}+\underbrace{abc}_{\mbox{tekst}}+\widehat{P}\)
```



$\vec{v}+\vec{ab}+\overrightarrow{abc}+\overline{abc}+\underbrace{abc}+\underbrace{abc}_{\mbox{tekst}}+\widehat{P}$

Figure 1.40: LaTeX with vectors and underlines and over lines

Try out the dots above your variable with:

```
\( \dot a+\ddot b \)
```

$\dot{a} + \ddot{b}$

1.2.3.11.4 Vectors and matrices

A few examples to make column vectors and matrices with LaTeX:

A small vector with two dimensions you can make with the binomium: binom:

```
\( \binom{4}{3}+\binom{a_1}{a_2,a_3} + {32 \brack 5 } + {a \brace
 b } \)
```

Another possibility is with **pmatrix** (normal brackets) or with **bmatrix** (with square brackets like [..].

Make a matrix with one or more columns.

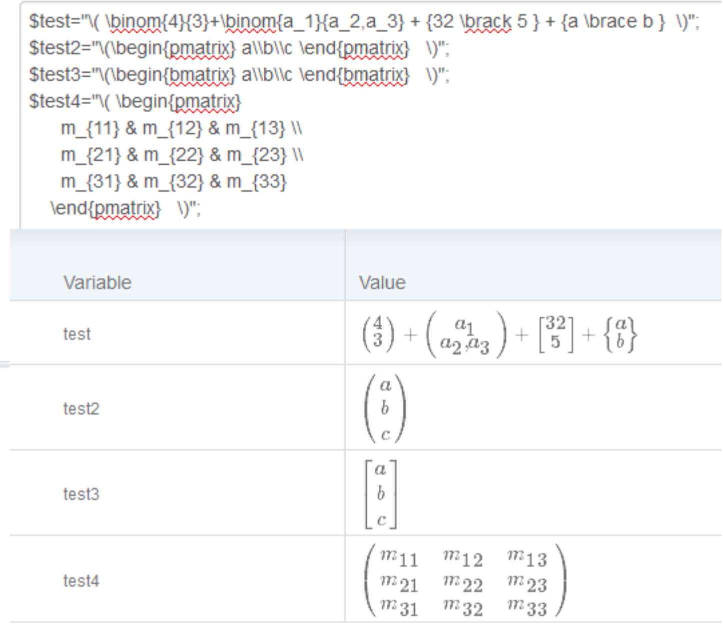
With **vmatrix** you will get the bars for modulus like |..| .

Try out the following in the *Algorithm*:

```
$test3="\(\begin{bmatrix} a\\b\\c \end{bmatrix} \)";
```

```
$test4="\( \begin{pmatrix}
      m_{11} & m_{12} & m_{13} \\
      m_{21} & m_{22} & m_{23} \\
      m_{31} & m_{32} & m_{33}
    \end{pmatrix}   \)";
```

The &-sign provides the lining up.



```
$test="\( \binom{4}{3}+\binom{a_1}{a_2,a_3} + {32 \brack 5 } + {a \brace b }  \)";
$test2="\(\begin{pmatrix} a\\b\\c \end{pmatrix}   \)";
$test3="\(\begin{bmatrix} a\\b\\c \end{bmatrix}   \)";
$test4="\( \begin{pmatrix}
     m_{11} & m_{12} & m_{13} \\
     m_{21} & m_{22} & m_{23} \\
     m_{31} & m_{32} & m_{33}
   \end{pmatrix}   \)";
```

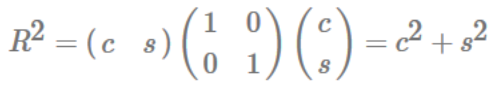
Variable	Value
test	$\binom{4}{3}+\binom{a_1}{a_2,a_3}+\left[{32 \atop 5}\right]+\left\{{a \atop b}\right\}$
test2	$\begin{pmatrix} a\\b\\c \end{pmatrix}$
test3	$\begin{bmatrix} a\\b\\c \end{bmatrix}$
test4	$\begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$

Figure 1.41: Column vectors and matrices with LateX

Another example:

```
\(\begin{equation}
R^2 =
\begin{pmatrix} c & s \end{pmatrix}
\begin{pmatrix} 1 & 0\\ 0 & 1 \end{pmatrix}
\begin{pmatrix} c \\ s \end{pmatrix}
= c^2 + s^2
\end{equation}\)
```

With the result:



$$R^2 = \begin{pmatrix} c & s \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} c \\ s \end{pmatrix} = c^2 + s^2$$

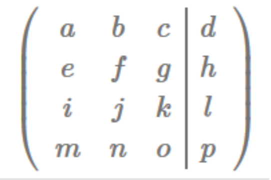
Figure 1.42: Matrixequation

Mind the line up with &.

Augmented matrix you can make with array:

```
\(\left(\begin{array}{ccc|c}
a & b & c & d \\
e & f & g & h \\
i & j & k & l \\
m & n & o & p
\end{array}\right)\)
```

With the result:



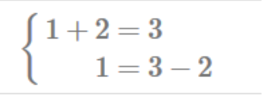
$$\left(\begin{array}{ccc|c} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{array}\right)$$

Figure 1.43: Augmented matrix

1.2.3.11.5 Sets of equations (brackets and lining up)

An example of a set of equations:

```
\(\left\{ \begin{align*}
  1 + 2 &= 3\\
  1 &= 3 - 2
\end{align*}\right.\)
```



$$\left\{ \begin{aligned} 1 + 2 &= 3 \\ 1 &= 3 - 2 \end{aligned} \right.$$

Figure 1.44: Sets of equations and lining up

The curly bracket at the end you can skip by `\right.` (so with a dot and nothing behind).
The &-characters provide the lining up.

The asterix next to align if you don't want labels (numbers) for the formulas.

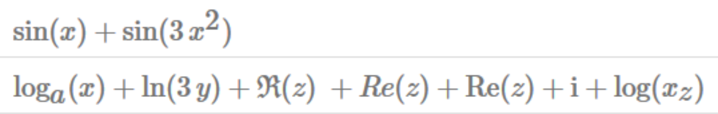
1.2.3.11.6 Functions and commands italic and not-italic

The well-known functions like sin, cos, log etcetera are automatically in roman and not italic if you put the backslash in LaTeX in front of it. A recommendation is to put always brackets round the argument of the function for better understanding.
A few examples of these functions you will see here.

```
\(\sin(x)+\sin(3x^2)\)
```

```
\(\log_{a}(x)+\ln(3\,y)+\Re(z)\ +Re(z)+\mbox{Re}(z) +\mathrm{i} +\log(x_{z})\)
```

With the following result:



$$\sin(x)+\sin(3\,x^2)$$

$$\log_a(x)+\ln(3\,y)+\Re(z)\;+Re(z)+\mathrm{Re}(z)+\mathrm{i}+\log(x_z)$$

Figure 1.45: LaTeX with functions and commands

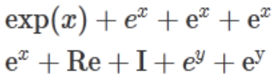
You see the small space between 3 and y in ln(3y). This is by using \, (backslash comma) to get a half space.
For example the Euler's number e you want to see it in roman, and not italic. Use for example \mathrm{e} it means mathematics roman or \mbox{} . The curly brackets will keep all together.

A few examples to copy for exercise;

```
\(\exp(x)+e^x +\mbox{e}^x+ \mathrm{e}^x\)
```

```
\( \mathrm{e}^{x} +\mathrm{Re} + \mathrm{I} +e^y +\mathrm{e}^{\mathrm {y}} \)
```

with the following result:



$$\exp(x)+e^x+\mathrm{e}^x+\mathrm{e}^x$$

$$\mathrm{e}^x+\mathrm{Re}+\mathrm{I}+e^y+\mathrm{e}^{\mathrm{y}}$$

Figure 1.46: LaTeX with italic and not italic

1.2.3.11.7 Fractions and square roots

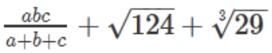
A fraction you make with \frac{}{}.
Between the first curly brackets the numer and the second curly brackets will keep the denominator together. No komma in between.

The nth root is by \sqrt[n]{}. Everything between the curly brackets belongs to the root and you will see a bar over the full argument.

For the square root you can skip [2] so `\sqrt{}` will do:

```
\( \frac{abc}{a+b+c}+\sqrt{124} +\sqrt[3]{29} \)
```

With the result as follows:



$$\frac{abc}{a+b+c}+\sqrt{124}+\sqrt[3]{29}$$

Figure 1.47: LaTeX with fractions and square roots

1.2.3.11.8 Inequations and special characters

HINT: To type an inequation it is important not to use the keyboard but the official code for these. Mind the backslash.

For example \le means less equal.

>= \geq
<= \leq
< \lt
> \gt
∞ \infty
U \cup

∩ \cap

HINT: Some errors may occur when migrating content using the characters from the keyboard like < and > .
So better is to use `\lt` etcetera.

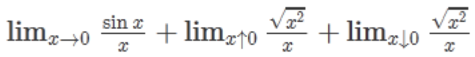
1.2.3.11.9 Limits, summation and products

For the **limit** you use the following LaTeX code: `\lim{x \to 0}` .
Behind that you make the formula (with brackets if you want).
For left and right limit you have the following code: `\lim{x \uparrow 0}` and `\downarrow` etcetera.

Look in the example for the performance of it:

```
\( \lim_{x \to 0} \frac{\sin{x}}{x} +\lim_{x \uparrow 0}
 \frac{\sqrt{x^2}}{x} +\lim_{x \downarrow 0} \frac{\sqrt{x^2}}{x}
 \)
```

With the following result:



$\lim_{x \to 0} \frac{\sin{x}}{x} +\lim_{x \uparrow 0} \frac{\sqrt{x^2}}{x} +\lim_{x \downarrow 0} \frac{\sqrt{x^2}}{x}$

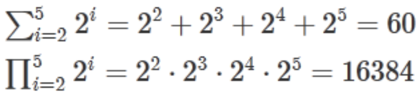
Figure 1.48: LaTeX with limits

The **sum and product** with subscript and superscript for the boundaries.

```
\(\sum^5_{i=2} 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60\)
```

```
\(\prod^5_{i=2} 2^i = 2^2 \cdot 2^3 \cdot 2^4 \cdot 2^5 = 16384\)
```

With the following result:



$\sum^5_{i=2} 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60$
$\prod^5_{i=2} 2^i = 2^2 \cdot 2^3 \cdot 2^4 \cdot 2^5 = 16384$

Figure 1.49: LaTeX with sum and product

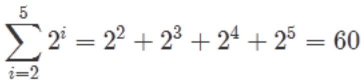
Mind the central dot with `\cdot` for multiplication.
Three dots you can make with `\cdots`.

The dot with the keyboard is for the decimal numbers.

To display your formula **centralized in the line** you want to see a better performance for the subscript and superscript of the summation. In that case you start and end the LaTeX-code not with \(but with \[and at the end with \] instead of \).

```
\[\sum^5_{i=2} 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60\]
```

With result:



$$\sum^5_{i=2} 2^i = 2^2 + 2^3 + 2^4 + 2^5 = 60$$

Figure 1.50: LaTeX with centered formula

HINT: For the formula in line for a better performance of the superscript and subscript with an extra option in the code with `\displaystyle {..}` as you can try out yourself in the following formula.

```
\(\displaystyle{\sum^5_{i=2}\! 2^i} = 2^2 + 2^3 + 2^4 + 2^5 = 60\)
```

1.2.3.11.10 Differentiation and integration

The LaTeX code for integration is \int. With subscript and superscript you can give the boundaries of the integral.
Behind the function you need some space in front of the differential. Use the backslash comma or the backslash space.
Because the d-operator of the differential d*x* is not italic you have to code it with math roman with `\mathrm{d}`.

Try out yourself the following codes:

```
\(\int_{a}^{b} f(x) \,\mathrm{d}x\)
```

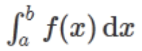
```
\(\int_a^b f(x) \ \mathrm{d}x\)
```

And for a better performance of the superscript and subscript use \displaystyle{} round your formula in line.

```
\(\displaystyle{\int_{a}^{b} f(x) \,\mathrm{d}x}\)
```

Or even better with limits for the bouderies of the integral:

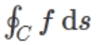
```
\(\int \limits_{a}^{b} f(x) \,\mathrm{d}x\).
```



$\int_a^b f(x)\,\mathrm{d}x$

Figure 1.51: LaTeX with integrals

A contour integral with `\(\oint_C f \ \mathrm{d}s\)` .



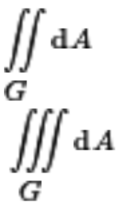
$\oint_C f \ \mathrm{d}s$

Figure 1.52: LaTeX with contour integral

Multiple integrals:

```
\( \displaystyle{\iint\limits_{G} \mathrm{d}A}\)
```

```
\( \displaystyle{\iiint\limits_{G} \mathrm{d}A}\)
```



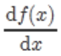
$\displaystyle\iint\limits_{G} \mathrm{d}A$

$\displaystyle\iiint\limits_{G} \mathrm{d}A$

Figure 1.53: multiple integrals

Differentiate by means of a fraction: \frac{}{} . Take care that the differential operator is not italic:

```
\(\frac{\mathrm{d}f(x)}{\mathrm{d}x}\)
```



$\frac{\mathrm{d}f(x)}{\mathrm{d}x}$

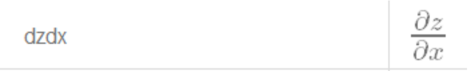
Figure 1.54: LaTeX and differentating

The symbol for infinity ∞ is with \infty.

The middot you make with met \cdot

The official notation for the partial derivative you see here prepared in the *Algorithm*:

```
$dzdx="\( \frac{\partial z}{\partial x} \)";
```



dzdx $\frac{\partial z}{\partial x}$

Figure 1.55: partial differentiating

HINT: Bigger integration sign with \displaystyle{}.

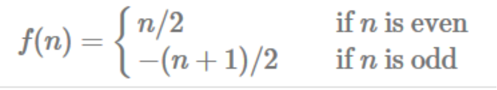
```
$displayvraag="\( \displaystyle{\int \frac{1}{\sqrt{1-$A\ x^2}} \
 \mathrm{d}x}\) ";
```

1.2.3.11.11 Piecewise functions

```
"\( f(n) =
  \begin{cases}
    \frac{n}{2}       & \quad \text{if } n \text{ is even}\\
    -\frac{n+1}{2}  & \quad \text{if } n \text{ is odd}
  \end{cases}
\)"
```

With the code `\quad` you can make some space between formula and text.

The &-character is for lining up things.



$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ -(n+1)/2 & \text{if } n \text{ is odd} \end{cases}$

Figure 1.56: Piecewise function

1.2.3.11.12 Brackets and arrows

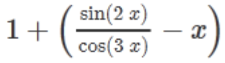
If you want brackets around something, you can just do that with (...) and square brackets with [...]. Also |...| you make them with your keyboard.
But for the curly brackets \{...\} you will need backslashes because curly brackets without backslash are meant to hold things together.

However, if you want these brackets to automatically scale with the formula, then work with tags like:

```
\left(.....\right) or \left \{ ......\right\}
```

The following formula shows the effect of scaling the brackets:

```
\(1+\left(\frac{\sin(2\ x)}{\cos(3\ x)}-x\right)\)
```



$$1+\left(\frac{\sin(2\ x)}{\cos(3\ x)}-x\right)$$

Figure 1.57: LaTeX and brackets

Very big brackets you can make with `\bigl(f \bigr)`

HINT: For only the left hand side of the curly brackets: `\left\{.....\right.` , that means in stead of the right hand side curly bracket type a dot.

Arrows

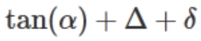
See https://latex-tutorial.com/arrow-latex/

```
\Rightarrow
\longleftrightarrow
\rightarrow
```

1.2.3.11.13 Greek characters

All greek characters mind the backslash in front of it.

```
\(\tan(\alpha)+\Delta+\delta\)
```



$$\tan(\alpha)+\Delta+\delta$$

Figure 1.58: LaTeX and special characters

1.3 Question type Maple graded

1.3.1 Introduction

The question type *Maple graded* is one of the most important question types to grade formulas. It offers many possibilities for asking open-ended questions with randomized formulas that are assessed for accuracy by means of Maple. A lot is possible with programming the grading using the *Grading Code*. For detailed information about the possibilities in the *Grading Code* see the relevant paragraph *Hints for de Grading Code (page 46)*.

Advice: If you want to grade formulas, use mainly this question type and preferably not the question type *Mathematical Formula*.
With every question type (not just *Maple graded*) there is always the option of using random variables, so that by making one test item you actually create a whole family of test items.
It is about keeping the organization of the question as simple as possible, so that modifications can be done easily (also by others).
The *Algorithm* section is an important element in the organization of the question.
Sometimes adjusting one formula in the *Algorithm* section can create a whole new family of questions if the rest of the question organization is coherent and well put together.

In general, the grading of the formula (response) entered by the student can be easily controlled by checking the difference between the correct answer and the response of the student. The computer algebra system Maple runs in the background to do the underlying calculation of this match and that's why the question type is called *Maple graded.*
For example, if the difference between the correct answer and the student's response is 0, the formula entered is basically correct.

However, there are many more possibilities for matching the correct answer with the student's response, depending on the question.
See for examples in section Hints for the *Grading Code (page 46)*.

HINT for the response area *Text entry only*:
If the answer is very long, enter the following code (starting with a point) into the *Custom CSS* of the question to widen the answer field.

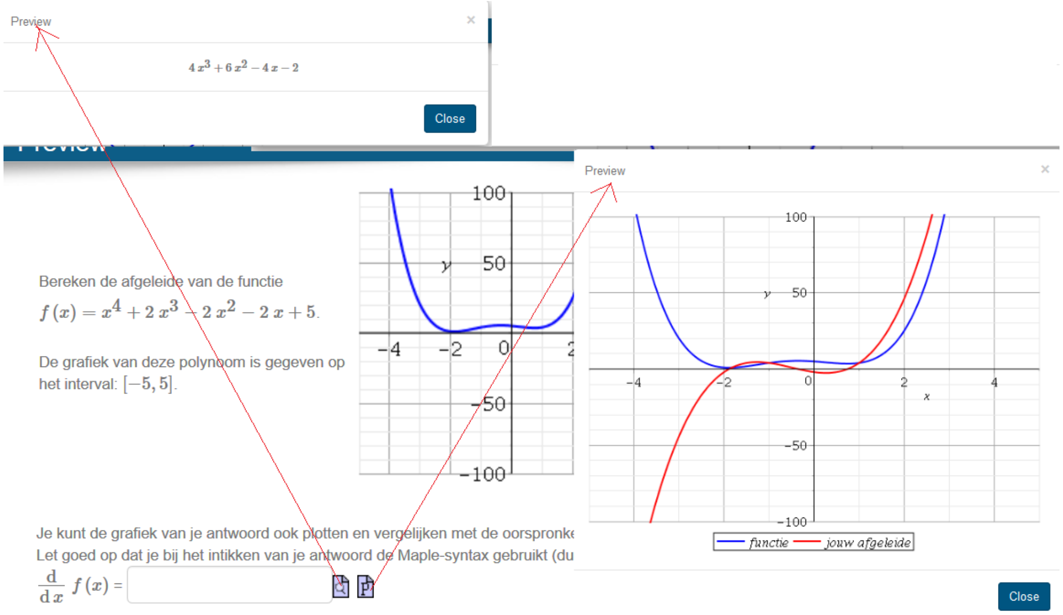
```
.response input {width:250pt;}
```

HINT for the response area *Symbolic entry only*:
If the answer is not too big, enter the following code (starting with a point) into the *Custom CSS* of the question to make the inputfield smaller.

```
.mwEquationEditor {height:100px !important}
```

1.3.2 General structure of the Maple graded question type

We'll start with an example of a *Maple graded* question type of your answer field (response area).



Preview

$4x^3+6x^2-4x-2$

Close

Bereken de afgeleide van de functie

$f(x) = x^4+2x^3-2x^2-2x+5$.

De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$.

Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronke
Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (du

$\frac{d}{dx}\,f(x) =$

Preview

Close

Figure 1.59: Example of a Maple graded question type

The figure above shows a question with a formula and a graph. In itself this is nothing special, as this can be achieved in any question type in terms of the presentation of formula and graph in the text of the question. The formula and associated graph are both randomized, so that every time the question is opened, a different function appears along with the associated graph.

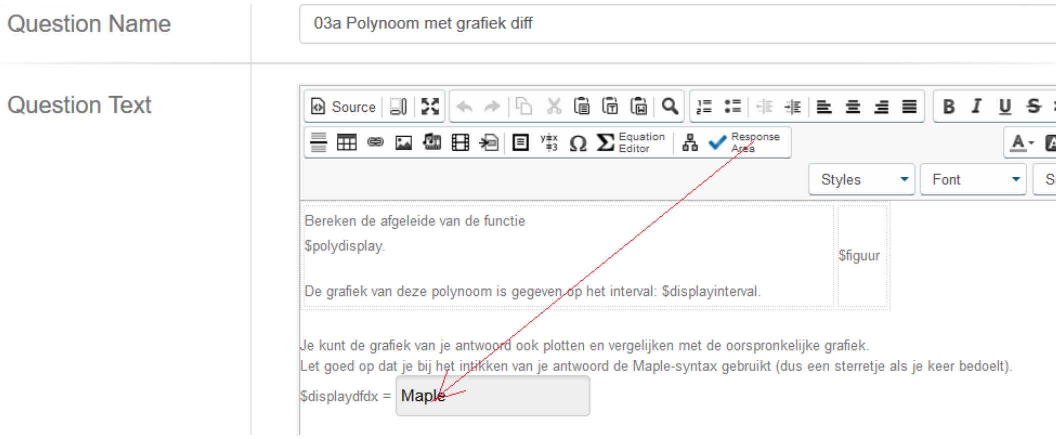
What is special about this question type is the answer field where the student can type in a formula.
Three settings are possible for this answer field. In the situation above *Figure 1.59 (page 35)* the student has to enter *Maple syntax*. For the student a button P is available with which he can call up a graph if necessary, like here the graph of his own answer together with the original graph. In the following you will see how to put the settings of this answer field (response area).

The student must also always click on the magnifying glass to preview the formula he has entered.

Creating a new question is done with the blue button *Create New - Question/Text*.

You will see all the individual sections of this question under each other and they can be opened successively for editing.
Start by naming the question. The student will not see this name.

In *Question Text*, you fill in the question and you enter an answer field (Response Area) with the button of the check mark in the checkbox as shown in the following figure:



Question Name: 03a Polynoom met grafiek diff

Question Text:

Bereken de afgeleide van de functie
$polydisplay.

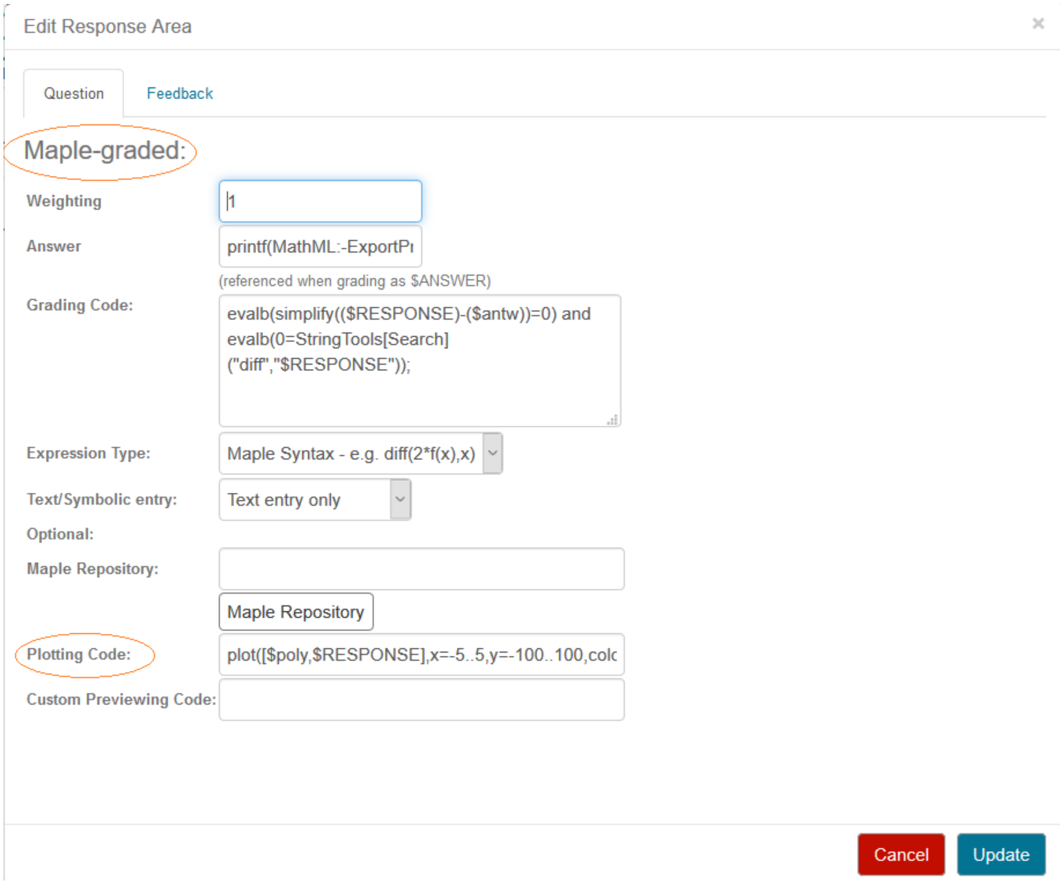
De grafiek van deze polynoom is gegeven op het interval: $displayinterval.

$figuur

Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronkelijke grafiek.
Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (dus een sterretje als je keer bedoelt).
$displaydfdx = Maple

Figure 1.60: Creating aMaple graded answer field

By adding an answer field you can choose the question type of your choice and in this case the question type *Maple graded*. You will then see a matching dialogue screen (*Edit Response Area*).



Edit Response Area

Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportPı
(referenced when grading as $ANSWER)

Grading Code:

```
evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]
("diff","$RESPONSE"));
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code: plot([$poly,$RESPONSE],x=-5..5,y=-100..100,colc

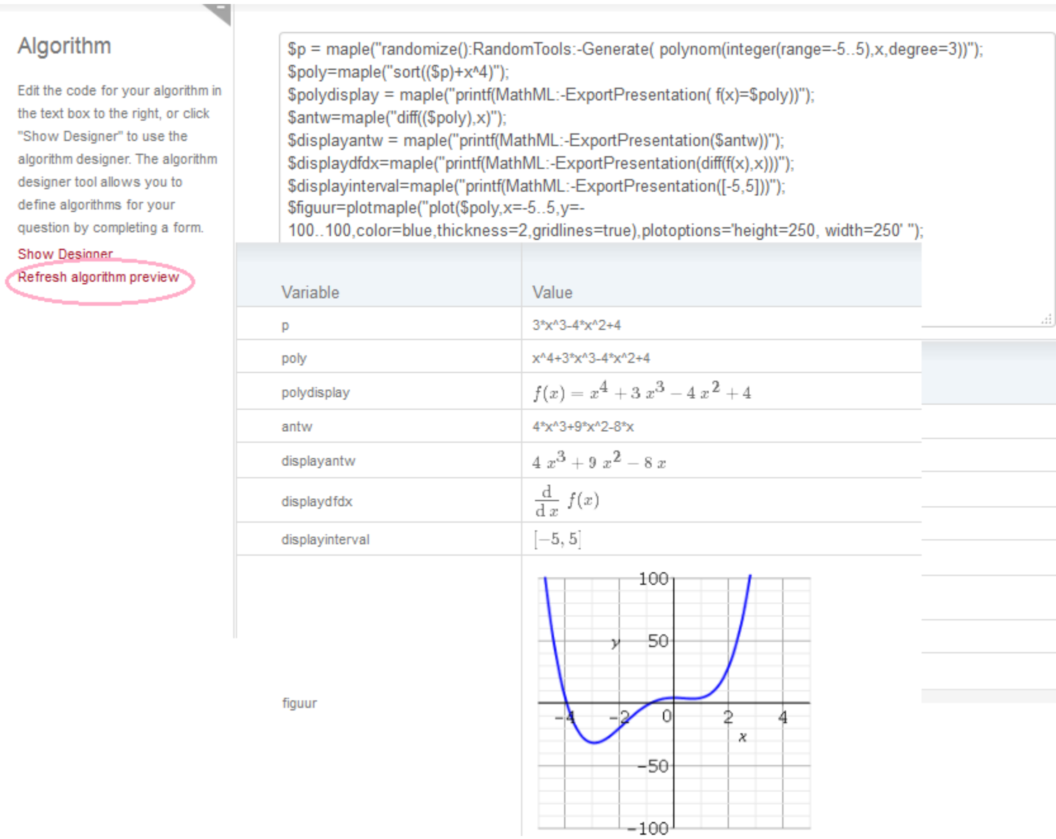
Custom Previewing Code:

Figure 1.61: Dialogbox of the Maple graded answer field

You can't fill in as much here if you don't know exactly what you want. That is why you will first make preparations in the *Algorithm* section of the question.

1.3.3 Algorithm of the question

Now first go to the *Algorithm* of the question to create the variables needed for the question. The variables you create here can be used throughout the question, in every answer field, in the Feedback and in the Question Text.... anywhere in the question where needed.
The *Algorithm* is at the heart of the question.



Algorithm

Edit the code for your algorithm in the text box to the right, or click "Show Designer" to use the algorithm designer. The algorithm designer tool allows you to define algorithms for your question by completing a form.

Show Designer

Refresh algorithm preview

```
$p = maple("randomize():RandomTools:-Generate( polynom(integer(range=-5..5),x,degree=3))");
$poly=maple("sort(($p)+x^4)");
$polydisplay = maple("printf(MathML:-ExportPresentation( f(x)=$poly))");
$antw=maple("diff(($poly),x)");
$displayantw = maple("printf(MathML:-ExportPresentation($antw))");
$displaydfdx=maple("printf(MathML:-ExportPresentation(diff(f(x),x)))");
$displayinterval=maple("printf(MathML:-ExportPresentation([-5,5]))");
$figuur=plotmaple("plot($poly,x=-5..5,y=-
100..100,color=blue,thickness=2,gridlines=true),plotoptions='height=250, width=250' ");
```

Variable	Value
p	3*x^3-4*x^2+4
poly	x^4+3*x^3-4*x^2+4
polydisplay	$f(x) = x^4 + 3\,x^3 - 4\,x^2 + 4$
antw	4*x^3+9*x^2-8*x
displayantw	$4\,x^3 + 9\,x^2 - 8\,x$
displaydfdx	$\frac{d}{dx}\,f(x)$
displayinterval	$[-5, 5]$
figuur	

Figure 1.62: The Algorithme of a Maple graded question type

In the *Algorithm* section fill in the following and by pressing *Refresh algorithm preview* the variables are refreshed and you can check things.

A variable is marked by a dollar sign.

```
$p=maple("randomize():RandomTools:-
Generate( polynom(integer(range=-5..5),x,degree=3))");
$poly=maple("sort(($p)+x^4)");
$polydisplay=maple("printf(MathML:-ExportPresentation( f(x)=$poly))");
$antw=maple("diff(($poly),x)");
$displayantw = maple("printf(MathML:-ExportPresentation($antw))");
$displaydfdx=maple("printf(MathML:-ExportPresentation(diff(f(x),x)))");
$displayinterval=maple("printf(MathML:-ExportPresentation([-5,5]))");
$figuur=plotmaple("plot($poly,x=-5..5,y=-100..100,color=blue,thickness=2,gridlines=true),plotoptions='hei
 width=250' ");
```

Using Maple, a random third degree expression is generated.

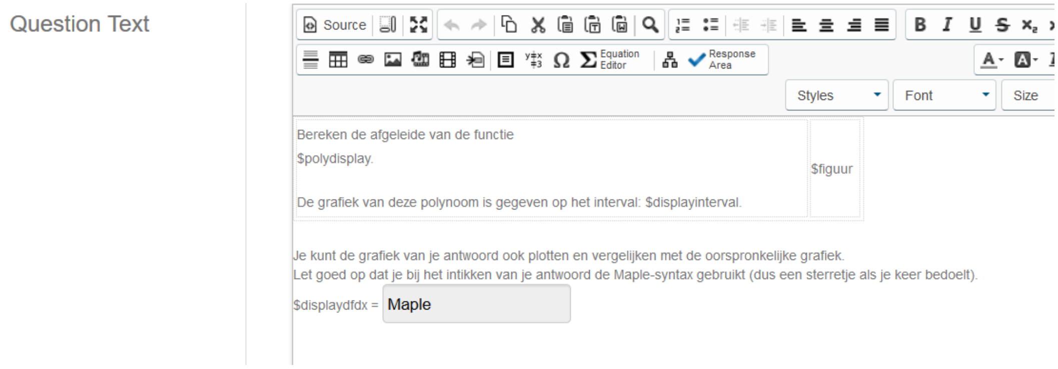
Then you add x^4 and sort the whole expression so that the given expression always starts with x^4. The final question is again MathML encoded and defined as the variable $polydisplay.

Using a Maple statement (diff), the correct answer $antw is generated and then converted in the variable $displayantw to MathML code for the *Feedback*.
See for the generating of dynamic graphs in the manual part C.

You can now use the algorithmic variables throughout the question by calling them by name with the dollar sign.

1.3.4 Question Text with a Maple graded answer field

In the *Question Text* you can put the formula and the graph as a variable in the text. Also the prepared formula $displaydfdx.



Bereken de afgeleide van de functie

$polydisplay.

De grafiek van deze polynoom is gegeven op het interval: $displayinterval.

$figuur

Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronkelijke grafiek.
Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (dus een sterretje als je keer bedoelt).

$displaydfdx = Maple

Figure 1.63: Question Tekst of the question with Response Area Maple Graded

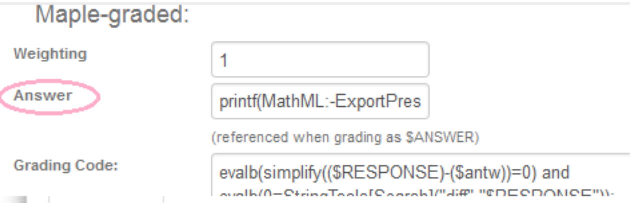
You decide where the answer field is placed and if necessary you can ask more questions and create more answer fields.

HINT: The text is placed next to the figure by means of a table without border.

HINT: Double click on the *Maple graded* answer field will open it.

1.3.5 Answer of the Maple graded answer field

In the *Answer* section, make sure to include a formula here that the student will later see as the correct answer.
Using the command `printf(MathML:-ExportPresentation($antw));` this answer will be presented as a correct answer in 2D in the feedback. But you can also just enter the formula $answer here and the feedback will show the formula that the student would have typed.



Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportPres

(referenced when grading as $ANSWER)

Grading Code: evalb(simplify(($RESPONSE)-($antw))=0) and

Figure 1.64: the correct answer in 2D-notation

In this section you should definitely enter a Maple command and end with a semicolon.
This result is automatically saved as the $ANSWER variable, and you can optionally use this variable in the *Grading Code* section to match the student's answer.
In this example, however, the MathML code is programmed in this section: `printf(MathML:-ExportPresentation($antw));`

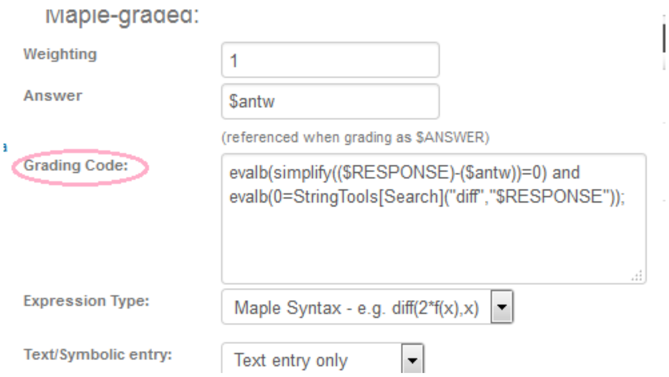
The student sees the 2D version of the correct answer on his screen in the feedback as being the correct answer in case the given answer was wrong. You cannot match the student's answer with this MathML code.

HINT: The answer you enter in the *Answer* section can only be used in the *Grading Code* as $ANSWER if it is a real formula (not as MathML code).

1.3.6 Grading Code of the Maple graded answer field

The **Grading Code** is the section where you put the Maple code in order to match the students answer ($RESPONSE) with the correct answer (mostly prepared in the *Algorithm* section as the variable $antw.
This *Grading Code* is very important in the *Maple graded* question type and offers many possibilities for the most diverse situations.

Example of a Grading Code



Maple-graded:

Weighting: 1

Answer: $antw

(referenced when grading as $ANSWER)

Grading Code:

```
evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]("diff","$RESPONSE"));
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Figure 1.65: Grading Code of the Maple graded answer field

In this case, the student's response (by definition the automatically generated variable $RESPONSE) is compared with the correct answer $antw already prepared in the *Algorithm* section. Subtracted from each other and its result simplified with the Maple command `simplify`, should yield 0.
The command `evalb` (evaluate boolean) will check (true or false).
`evalb(simplify(($RESPONSE)-($ANSWER))=0);` or better
`evalb(simplify(($RESPONSE)-($antw))=0);` (using the algorithmic variable).

The addition: `and evalb(0=StringTools[Search]("diff","$RESPONSE"));` prevents the student from using the available maple command.

HINT: In case you gave the correct answer in mathml code in the *Answer* section (*Figure 1.64 (page 38)*) it is not possible to use the variable $ANSWER in the *Grading Code*, because the correct answer in this case is a MathML code and not a formula that can be used for calculations.

HINT: The programming with `evalb(simplify(($RESPONSE)-($antw))=0);` is not sufficient here, because with the settings for *Expression type* (see section E*xpression type in the Maple graded question (page 40)*) is opt for *Maple syntax* and for *Text entry only*, implying that the student could also type Maple commands (such as diff(....)) to arrive at the correct answer. That is why you also have to program that the answer does not contain the character combination "diff".

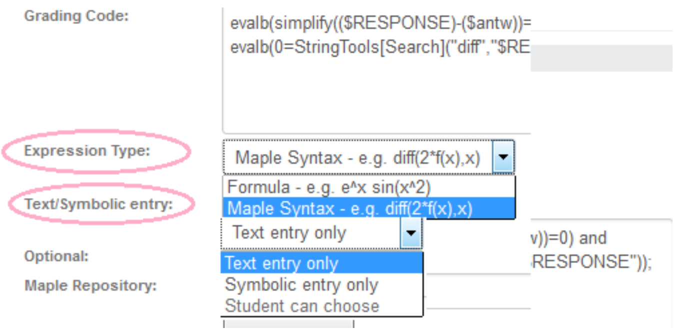
After all, the student should be able to calculate the answer himself with pen and paper. That the student is not allowed to use "diff" in his answer, you can program with the extra addition:

....and `evalb(StringTools[Search]("diff","$RESPONSE")=0)`.
It means that the string of the student's answer "$RESPONSE" is searched for the string "diff" and the result must be 0. (A number of characters between quotes is a string).
If you opt for *Symbol entry only*, the student will see an Editor where the student makes the formula with buttons. It cannot (to some extent) include Maple commands and then this extra programming is often not necessary. But with differentiate be on the look out with the *Symbol Mode*, because the student can use the button $\frac{\partial}{\partial x}$ which will lead to the correct answer. The extra restriction is therefore necessary with "diff" for both settings: *Text Mode* and *Symbol Mode*.
HINT: Please note that different *Grading Codes* are sometimes required for different settings for the Maple-syntax: *Text Mode* and *Symbol Mode.*

HINT: See for more information about the *Grading Code* in section *Tips for the Grading Code (page 46)*.

1.3.7 Expression type of the Maple graded answer field

In most cases, choose the *Maple Syntax* setting and, in general, definitely not *Formula*! The latter is in fact very weak and not always correct and cannot even handle character combinations. Also, the system has annoying errors if *Formula* is set and the student fills in something that does not conform to the syntax. So in general choose *Maple syntax*. Then choose *Text entry only* if the student has to type Maple syntax in a text field with the option for *Preview*. Or else offer the Editor at the *Symbolic entry only* setting. As a designer, you therefore determine whether or not an editor is offered to the student.



Grading Code: evalb(simplify(($RESPONSE)-($antw))= v))=0) and
evalb(0=StringTools[Search]("diff","$RE RESPONSE"));

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)
Formula - e.g. e^x sin(x^2)
Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only
Text entry only
Symbolic entry only
Student can choose

Optional:

Maple Repository:

Figure 1.66: Settings for Maple syntax in the Maple graded question type

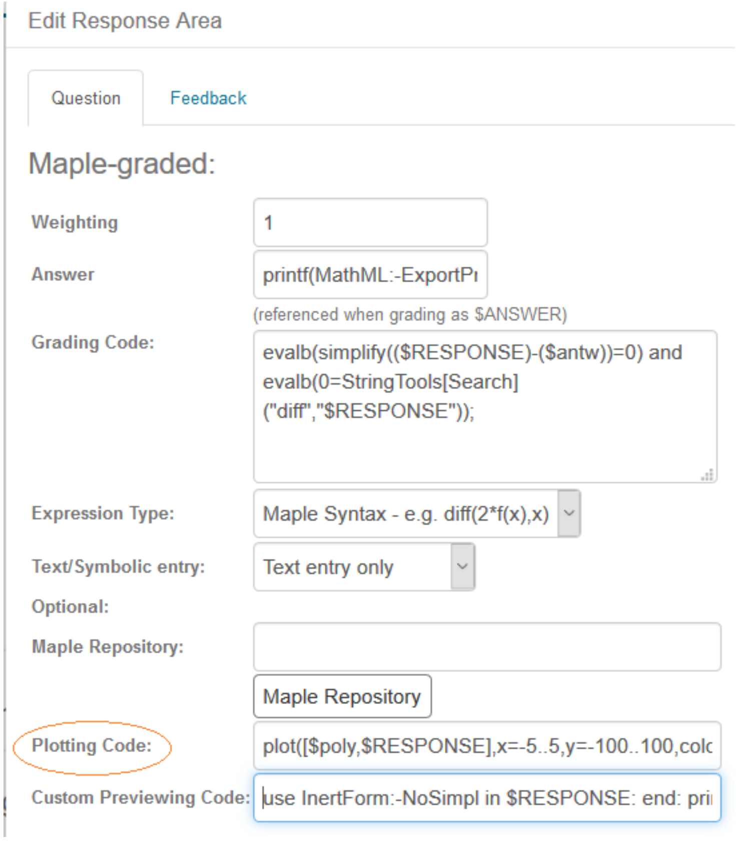
See for more information about the settings in section *Setting of the expression type (page 1)*.

1.3.8 Maple Repository of the Maple graded answer field

It is possible to predefine Maple functions and upload them in the on the server in the File Manager. In the question, you can import it and reference it. You do that when you want to leave complex jobs to Maple. You can read how that works in the *Manual part C*.

1.3.9 Plotting Code of the Maple graded answer field

At the bottom is a *Plotting Code* section. Here you can formulate one or more Maple statements (end each statement with a colon and the last statement with a semicolon) that can generate a graph when the student clicks P (*Plot*) in the question. In this command you can call variables from the *Algorithm* section and you can program the graph quite extensively.



Edit Response Area

Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportPı

(referenced when grading as $ANSWER)

Grading Code:

```
evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]
("diff","$RESPONSE"));
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code: plot([$poly,$RESPONSE],x=-5..5,y=-100..100,colc

Custom Previewing Code: use InertForm:-NoSimpl in $RESPONSE: end: prii

Figure 1.67: Plotting Code in the Maple graded answer field

In this statement for plotting you can even use the students formula ($RESPONSE) in order to show the graph of the student response as well. So the graph of the students respons and the graph of the polynom $poly will be seen by clicking P (*Plot*).

This *Plotting Code* is a maple command ending with a semicolon.

```
plot([$poly,
$RESPONSE],x=-5..5,y=-100..100,color=[blue,red],thickness=2,legend=[`functie`,`jouw
 afgeleide`],gridlines=true);
```

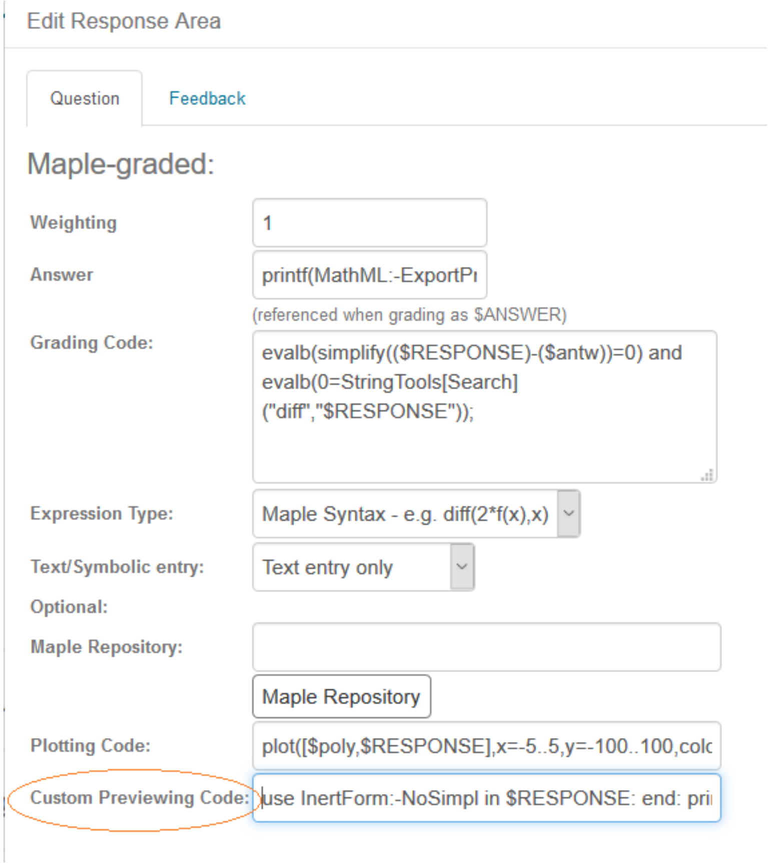
HINT: In maple-commands use no double quotes. Here you see the titel between backward quotes in stead of double quotes.

HINT: For a good use of the plotting facilities it is most useful if you set the formula settings to *Maple syntax* with *Text entry only*, but with the other settings it usually works well too. It is wise to always try it out. In most cases it also works fine if you chose *Symbolic entry only* and even with the *Formula* setting (not recommanded). The Editor translates everything neatly into *Maple syntax*.

HINT: If you do not enter anything in the *Plotting section*, the Plot button will not be active when the student is presented with the question.

1.3.10 Custom Previewing of the Maple graded answer field

In general, you do not enter anything in this section, but occasionally you will need to program something here.



Maple-graded:

Weighting: 1

Answer: printf(MathML:-ExportPı

(referenced when grading as $ANSWER)

Grading Code:

```
evalb(simplify(($RESPONSE)-($antw))=0) and
evalb(0=StringTools[Search]
("diff","$RESPONSE"));
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code: plot([$poly,$RESPONSE],x=-5..5,y=-100..100,colc

Custom Previewing Code: use InertForm:-NoSimpl in $RESPONSE: end: prii

Figure 1.68: Custom Previewing Code

If you do not enter anything, and the student clicks on *Preview* after filling in his formula, the student will see the entered formula in 2D format.
However, you are then dependent on the automatic simplification that Maple does.
If you want to stop the simplification, you can program the following:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

Try it out, but in most cases the *Preview* will show the unsimplified formula in 2D that the student has typed in.

In many other cases, you can manipulate the *Preview* to control what the student sees when pressing the *Preview* button.
See for more information about the programming of this *Custom Previewing Code* in the next section *(page 43)*.

HINT: The *Preview* button is only available if you opt for *Text entry only*.

1.3.10.1 Examples of Custom Previewing Codes:

-
  ```
  printf(MathML[ExportPresentation](Vector($RESPONSE)));
  ```
  The student enters a list for example [3,4,5]. In the *Preview* he will see the vector as a column. For more information see the section about grading with *Matrices and Vectors (page 118)*.

-
  ```
  if evalb(StringTools[Search]("abs","$RESPONSE")=0) then
   printf("Don't forget the absolute value stripes for
   the logarithm. You can do this with abs(..).") else
   printf(MathML[ExportPresentation]($RESPONSE)) end if;
  ```
  Here you check whether the student's answer may not contain abs. In that case, he will receive a message that he should not forget abs.

-
  ```
  use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
  ToMathML(%));
  ```
  Here, the Preview displays the unsimplified form that the student enters. This way you can stop the automatic simplification.

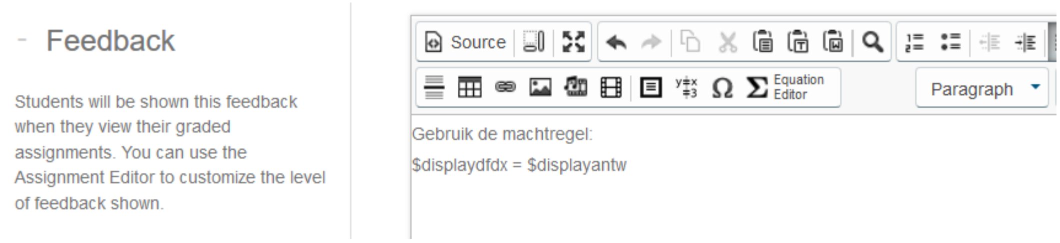
-
  ```
  if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","[")
   >= 2) or evalb(StringTools[CountCharacterOccurrences]
  ("$RESPONSE",",") > 1) then printf("Enter a two
   dimensional vector and use decimal point.") else
   printf(MathML[ExportPresentation](Vector($RESPONSE))) end if;
  ```
  Here you check whether the student enters only one vector with two elements. So only one square bracket may open and one comma may appear in the answer.

-
  ```
  if evalb(StringTools[Search]("*","$RESPONSE")=0) then printf("Don't forget to
   enter an asterix for multiplication.") else printf(MathML[ExportPresentation]
  ($RESPONSE)) end if;
  ```

-
  ```
  if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","*") <= 2)
   then printf("Don't forget to enter an asterix for multiplication.") else
   printf(MathML[ExportPresentation]($RESPONSE)) end if;
  ```

-
  ```
  if evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","=")=0)  then
   printf("Enter an equation in x and y.") else printf(MathML[ExportPresentation]
  ($RESPONSE)) end if;
  ```

1.3.11 Feedback of the whole question



- Feedback

Students will be shown this feedback when they view their graded assignments. You can use the Assignment Editor to customize the level of feedback shown.

Gebruik de machtregel:

$displaydfdx = $displayantw

Figure 1.69: Feedback for the whole question

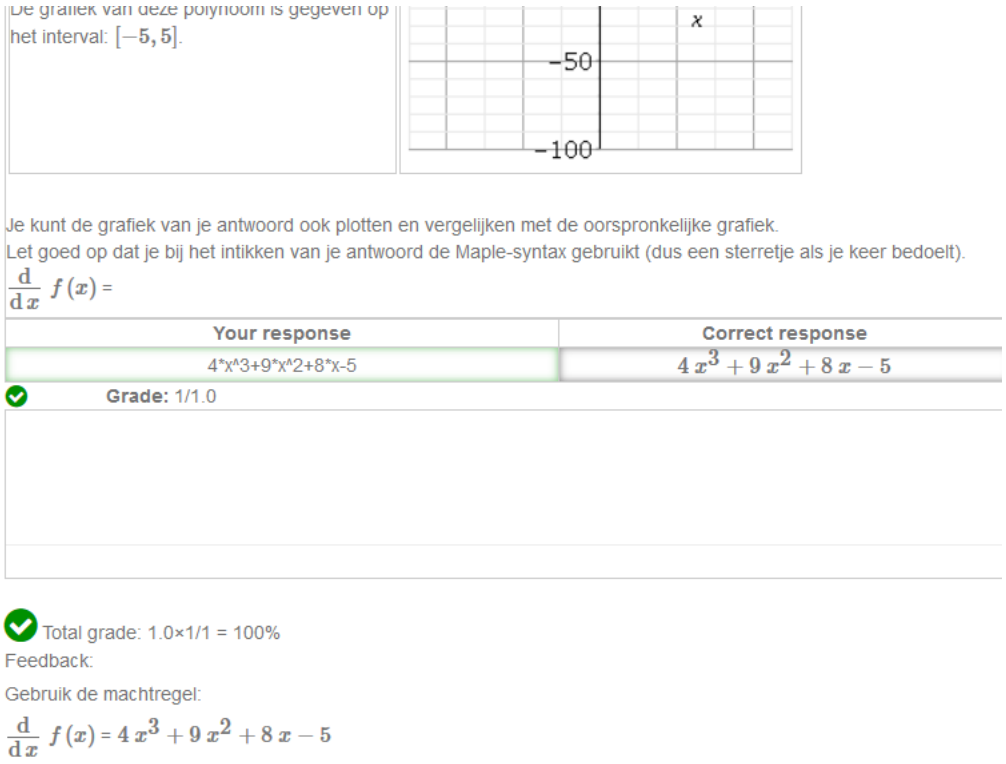
The figure shows that you can also communicate the correct answer in the *Feedback* of the question.

This correct answer has already been prepared and MathML-encoded ($displayantw) in the *Algorithm* section.

```
$displayantw=maple("printf(MathML[ExportPresentation]
($antw))");
```
or

```
$displayantw=maple("printf(MathML:-ExportPresentation($antw))");
```

The following figure shows the result with the feedback after clicking on *Grade*. The student will see this feedback in his *Gradebook*.



De grafiek van deze polynoom is gegeven op het interval: $[-5,5]$.

Je kunt de grafiek van je antwoord ook plotten en vergelijken met de oorspronkelijke grafiek.
Let goed op dat je bij het intikken van je antwoord de Maple-syntax gebruikt (dus een sterretje als je keer bedoelt).

$\frac{d}{dx} f(x) =$

Your response	Correct response
4*x^3+9*x^2+8*x-5	$4x^3+9x^2+8x-5$

✓ **Grade:** 1/1.0

✓ Total grade: 1.0×1/1 = 100%

Feedback:

Gebruik de machtregel:

$\frac{d}{dx} f(x) = 4x^3+9x^2+8x-5$

Figure 1.70: Grading of the question with feedback

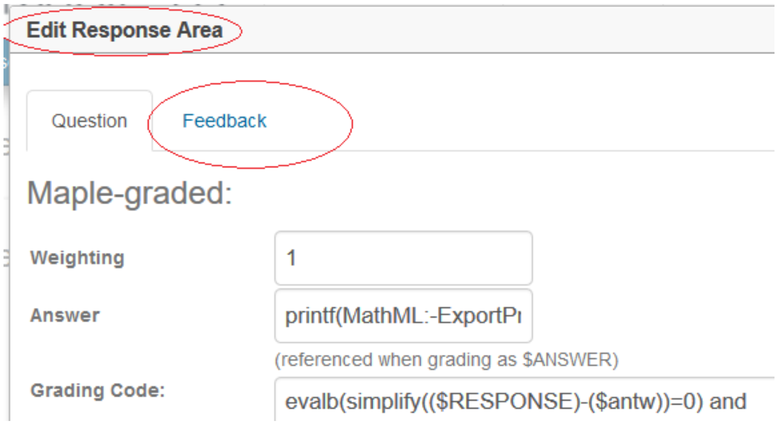
The figure above shows the question again and below it exactly what the student has entered. The formula settings were set to *Text entry only* and so text can be seen.
If you had set the settings to *Symbol entry only*, this students answer would have been displayed in two-dimensional form exactly as the student had entered it in the Editor.
Right next to \`Your response\` you see the correct answer (you programmed it in the *Answer* section with MathML).

Below the *Total grade* you see the *Feedback of the question* with the correct answer prepared in the *Algorithm*.

1.3.12 Special Feedback just for the answer field for the purpose of How did I do

If you give Assignments or Lessons in Möbius, students often have the option to click on *How did I do*.
For that situation, you can also prepare Feedback per answer field.

Open the question and double click on the answer field. You will then enter the *Edit Response Area* and you will see a *Feedback tab* at the top.
This tab is especially for Feedback of the answer field in the *How did I do* situation. This feedback is not visible after the grading, nor in the *Gradebook*.



Maple-graded:

Weighting: 1

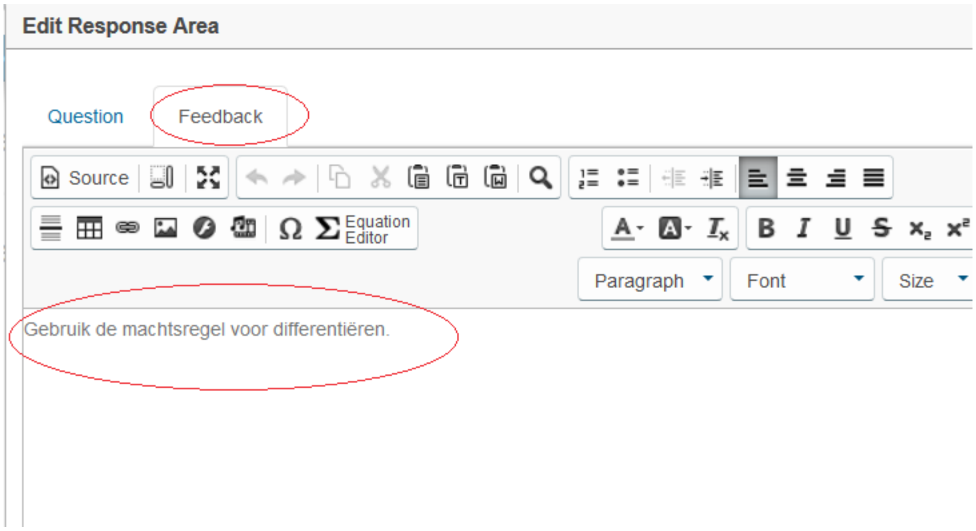
Answer: printf(MathML:-ExportPı

(referenced when grading as $ANSWER)

Grading Code: evalb(simplify(($RESPONSE)-($antw))=0) and

Figure 1.71: Tab for the specific Feedback of the answer field

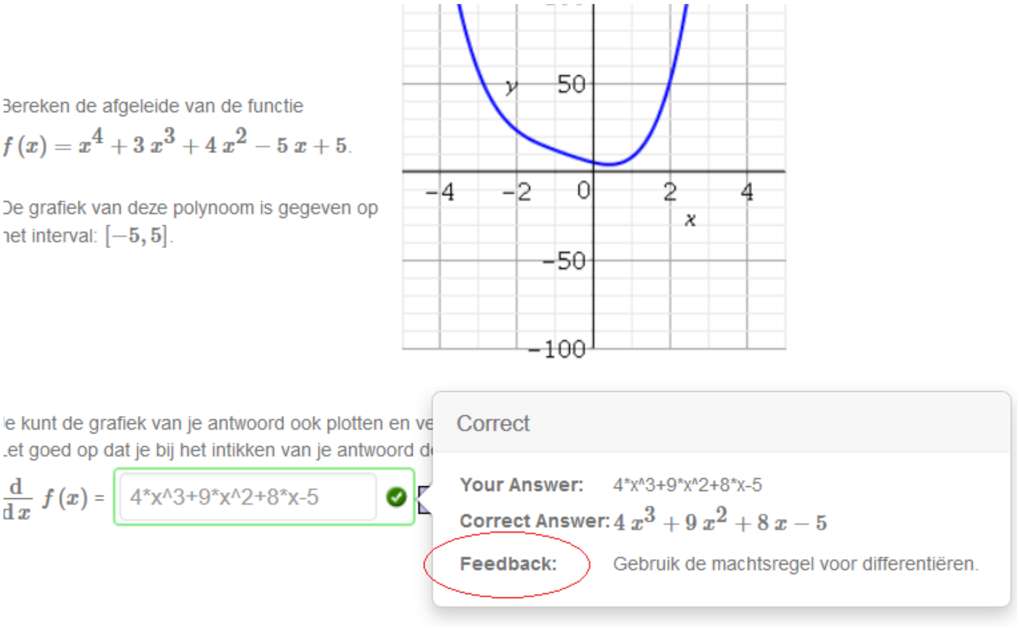
Go to the Feedback tab.



Gebruik de machtsregel voor differentiëren.

Figure 1.72: Entering Feedback for the answer field

Check your question with *Preview* and cklick on *How did I do* to see the effect.



Bereken de afgeleide van de functie

$f(x) = x^4 + 3\,x^3 + 4\,x^2 - 5\,x + 5$.

De grafiek van deze polynoom is gegeven op het interval: $[-5, 5]$.

Je kunt de grafiek van je antwoord ook plotten en ve

Let goed op dat je bij het intikken van je antwoord d

$\frac{d}{dx} f(x) =$ 4*x^3+9*x^2+8*x-5

Correct

Your Answer: 4*x^3+9*x^2+8*x-5

Correct Answer: $4\,x^3 + 9\,x^2 + 8\,x - 5$

Feedback: Gebruik de machtsregel voor differentiëren.

Figure 1.73: How did I do and Feedback

1.4 Hints for the Grading Code

1.4.1 Introduction

The *Maple graded* question type as described in section Q*uestiontype Maple graded (page 34)* is pre-eminently the question type to test formulas. There are many possibilities in this question type to adapt the programming for the grading to your liking.
The easiest way to check the student's response is the following principle:
The formula entered by the student, by definition the variable $RESPONSE, is subtracted (possibly after simplification) from the correct formula (possibly prepared in the *Algorithm*) and the result must be equal to 0.
With the *Maple graded* question type, something similar is preprogrammed by default and the *Grading Code* already contains:

```
evalb(($ANSWER)-($RESPONSE)=0);
```

The variable $ANSWER is the result of the formula that you entered as the correct answer in the *Answer* section (*Enter Maple code that evaluates to the correct answer:*)

But the following is also possible and often comes down to the same thing:

```
is(($ANSWER)-($RESPONSE)=0);
```

or

```
evalb($RESPONSE=$ANSWER);
```

The maple commands `is` and `evalb` are boolean commands with result *true* or *false*. This gives the system a definitive answer as to whether the student's answer is assessed as correct or not.

HINT: Be aware `evalb` and `is` don't always work the same as the command below in maple shows:

```
> is( 4*(2*x-5)*(2*x+5)=(4*x+10)*(4*x-10));
```

$$\mathit{true} \tag{1.1}$$

```
> evalb( 4*(2*x-5)*(2*x+5)=(4*x+10)*(4*x-10));
                                  false                              (1.2)
```

HINT: You can also program something that returns a number between 0 and 1. This can make the grading a bit more nuanced. The result 0 is wrong and the result 1 is correct. Anything in between gives a partially good rating.
See also section *Numbers with Maple graded (page 48)* and also in section *Unordered lists and ordered lists (page 93)* with examples of partial grading.

HINT: Default is $RESPONSE the students answer. And default is $ANSWER the formula of the correct answer in the section *Answer* (*Enter Maple code that evaluates to the correct answer:*)
It is not allowed to leave he section *Answer* empty in this *Maple graded* question type.
Usually you take the prepared correct answer from the *Algorithm* like $antw or something or you give the MathML coded formula. Be aware that it is not possible to use this MathML coded formula in the *Grading Code* as $ANSWER.
HINT: For the *Grading Code* it is wise to use ALWAYS the correct answer prepared in the *Algorithm* $antw like the following *Grading Code*:

```
evalb(($antw)-($RESPONSE)=0);
```

or as an alternative:

```
evalb(($antw)=($RESPONSE));
```

HINT: Also pay attention to the brackets in case the response starts with a minus sign or just always for safety.

HINT: Sometimes it is necessary to simplify the difference between the student's answer and the correct answer first and then check whether the result is zero. This is often done when, for example, the student's spelling does not matter. This is often a matter of trial and error.

```
evalb(simplify(($RESPONSE)-($antw))=0);
```

But more is possible!! In addition to reviewing the student's response formula ($RESPONSE), you can even view the **string** of the student's typed response ("$RESPONSE"). This string consists of a number of characters in succession that have been typed in exactly the same way by the student.

In the following sections a large number of possibilities are shown and explained for programming the *Grading Code* in all kinds of situations, such as equations, integrals, matrices and the like where it is not so easy to match the correct answer with that of the student by simply equating or subtracting each other and requiring zero to come out.

The philosophy is actually general: you do something with the student's answer, which is $RESPONSE by default, and then you match the correct answer to do the same.
The correct answer is always called $antw or $answer in the following paragraphs, because often this correct answer has already been created and checked in the *Algorithm* section.

HINT: The answer you program in the *Answer* section is not always useful in the *Grading Code* section, for example if you have prepared this answer in a 2D presentation, where no formula, but a Mathml code is generated by Maple. Of course, this cannot be matched in the *Grading Code* section. And there are a few more examples where the *Answer* section should be looked at critically.

So it is wise to use ALWAYS the correct answer prepared in the *Algorithm.*

The following paragraphs provide examples of the *Grading Code* in a variety of situations.

HINT: Please note that the different settings *Text Mode* and *Symbol Mode* sometimes require different *Grading Codes*.

1.4.2 Numbers with Maple graded

You can also test **numbers** with a *Maple graded* question type.
Usually for numbers you use the *numerical question type*, but sometimes there are more then one correct answers (for example $antw1 and $antw2 both are correct answers). In that case, a Maple graded question is also useful. In this case, you can add tolerance to the answer. For example, in the *Grading Code* you program the following:

```
evalb(abs(($antw1)-($RESPONSE))<0.1) or evalb(abs(($antw2)-
($RESPONSE))<0.1);
```

You can also, for example, grade the first answer totally correct and grade the second answer half correct with the following grading code:

```
if evalb(abs(($antw1)-($RESPONSE))<0.1) then 1 elif evalb(abs(($antw2)-
($RESPONSE))<0.1) then 0.5 else 0 end if;
```

HINT: Mind the brackets round $RESPONSE. It is better to always put the brackets around it.

1.4.3 When are formulas equal?

It is quite close which formulas are seen as **equal** in Maple's eyes.
In the following examples, two formulas are always judged to be unequal, while these two formulas are subtracted and after simplification yield zero.
In fact `evalb(($RESPONSE)=($antw));` is stronger then `evalb(($RESPONSE)-($antw)=0);` and this is more powerful than `evalb(simplify(($RESPONSE)-($antw))=0);`.
This last grading code is most robust and often useful if it doesn't matter at all what format the student gives the answer.
It is useful to try something out with the computer algebra system Maple if you have it available or possibly in the *Algorithm*.

HINT: Also keep in mind that in many cases the *Grading Code* does something with the $RESPONSE (the student's answer). Sometimes it is important that this $RESPONSE is entered in Maple syntax!

Simplifying fractions, for example, is difficult to check in a *Maple graded* question.
Section *Fractions (page 71)* provides a collection of tips that can help you when it comes to the *Grading Code* for fractions.

HINT: Always check whether it is possible for the student to simply retype the question in order to get a correct grading.
Or maybe the student clicks *Preview* and is presented with a simplified form that actually gives the answer away. After all, with the *Maple syntax Text entry only* setting, a simplification is often offered when clicking *Preview*.
What you need to do is explained in the section on stopping automatic simplification *(page 16)*.
However, there are tricks to getting it exactly the way you want it.

Fractions

Below are a few Maple commands to see what the Maple computer algebra system actually does with fractions of all kinds, for example, and how you can try things out yourself in the Maple program.

```
> restart; breuk1:=a/x+b;
```

$$\mathit{breuk1} := \frac{a}{x} + b \tag{1.3}$$

```
> breuk2:=simplify(breuk1);
```

$$\mathit{breuk2} := \frac{b\,x + a}{x} \tag{1.4}$$

```
> evalb(simplify(breuk1-breuk2)=0);
```

$$\mathit{true} \tag{1.5}$$

After simplification, of course, the difference between the two forms equals 0.
However, look at the following two statements:

```
> evalb(breuk1=breuk2);
```

$$\mathit{false} \tag{1.6}$$

```
> evalb(breuk1-breuk2=0);
```

$$\mathit{false} \tag{1.7}$$

The fractions are considered unequal if no simplification takes place.

Not only with numbers but also as b a multiple of a, we have a problem.
Also consider the type of expression. If the expression is written as one fraction, then the expression type is a multiplication.

```
> whattype(breuk1);
```

$$`+` \tag{1.8}$$

```
> whattype(breuk2);
```

$$`*` \tag{1.9}$$

```
> breuk3:=a/x+3*a;
```

$$\mathit{breuk3} := \frac{a}{x} + 3\,a \tag{1.10}$$

```
> breuk4:=simplify(breuk3);  breuk4a:=normal(breuk3);
```

$$\mathit{breuk4} := \frac{3\,a\,x + a}{x}$$

$$\mathit{breuk4a} := \frac{a\,(3\,x + 1)}{x} \tag{1.11}$$

```
> breuk5:=(a+3*a*x)/x;
```

$$\mathit{breuk5} := \frac{3\,a\,x + a}{x} \tag{1.12}$$

```
> evalb(breuk4=breuk5);
```

$$\mathit{true} \tag{1.13}$$

```
> evalb(breuk3=breuk4);
```

$$\mathit{false} \tag{1.14}$$

The two fractions that are the same but differ in spelling are not always seen as the same by Maple.
This has advantages if we want to force the student to simplify the fraction or to add more fractions together.

See for more information in the section *Fractions (page 71)*.

Also note the following **expressions** that are not always seen as the same by maple.

```
> evalb(a*(x-5)=a*x-5*a);
```

$$\mathit{false} \tag{1.15}$$

```
> evalb(a*(x-5)-(a*x-5*a)=0);
```

$$\mathit{false} \tag{1.16}$$

```
> a*(x-5)-(a*x-5*a);
```

$$a\,(x-5)-a\,x+5\,a \tag{1.17}$$

```
> simplify(a*(x-5)-(a*x-5*a));
```

$$0 \tag{1.18}$$

Here it can be seen that two expressions written in different ways are not always seen as the same by maple.
Also the difference of the two expressions is not by definition equal to 0 if it can yield zero **after simplification**.

HINT: What you can do with `evalb(($antw)=($RESPONSE));` is matching two expressions, two strings, two lists or two sets where that is not so easy with subtraction from each other.

Sets are equal if they have the same elements. Duplicate elements actually count for singles and the order doesn't matter.

Lists, with numbers or formulas, where the order is important, can also be compared with each other. Lists are defined with square brackets.

```
> evalb([a,b,c*x-c*y]=[a,b,(c*(x-y))]);
```

$$\mathit{false} \tag{1.19}$$

```
> is([a,b,c*x-c*y]=[a,b,(c*(x-y))]);
```

$$\mathit{false} \tag{1.20}$$

```
> evalb([a,b,c*x-c*y]=[a,b,expand(c*(x-y))]);
```

$$\mathit{true} \tag{1.21}$$

```
> is([a,b,c*x-c*y]=[a,b,expand(c*(x-y))]);
```

$$\mathit{true} \tag{1.22}$$

Above it can be seen that even the formulas in the lists have to be of the same form in order for them to be valued as "the same".
With `is`, you can often do the same as with `evalb`.

See for more information in the section about *Sets (page 64)* .

Strings. You can easily match strings. A string is a number of characters between double quotes whose order is fixed. Spaces count as character too! (For formulas, the spaces around an operator are simply ignored, otherwise a syntax error will be reported in the *Preview*, however in the translation of the $RESPONSE to the string, the spaces are always included!)

```
> evalb("string1+s^2"="string1 +s^2");
```

$$\mathit{false} \tag{1.23}$$

```
> evalb("string1+s^2"="string1+s^2");
```

$$\mathit{true} \tag{1.24}$$

```
> StringTools[Remove](" ","string1 +s^2");
```

$$\text{"string1+s^2"} \tag{1.25}$$

In the above it has been made clear that it is possible to remove, for example, spaces from a string.

See the section about *StringTools* *(page 51)* for more information.

Also two **equations** can be matched if the equations are prepared first (otherwise you will get more than one = sign in the formula).

```
> verg1:=a*(-7+x+3*a-2*x^2)=8-p; verg2:=-a*(2*x^2-3*a+7-x)=-p+8;
  evalb(verg1 = verg2);
```

$$\mathit{verg1} := a\left(-2x^2+3a+x-7\right)=8-p$$

$$\mathit{verg2} := -a\left(2x^2-3a-x+7\right)=8-p$$

$$\mathit{false} \tag{1.26}$$

```
> verg1:=expand(a*(-7+x+3*a-2*x^2))=8-p; verg2:=expand(-a*(2*x^2-3*a+7-
  x))=-p+8;
  evalb(verg1 = verg2);
```

$$\mathit{verg1} := -2ax^2+3a^2+ax-7a=8-p$$

$$\mathit{verg2} := -2ax^2+3a^2+ax-7a=8-p$$

$$\mathit{true} \tag{1.27}$$

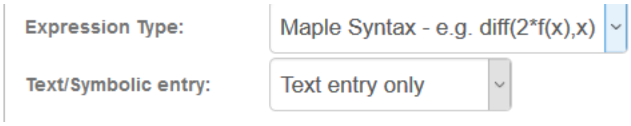
Basically what happens when matching two equations whether they are the same or not, the left and right sides are compared. They have to be in the same "form" (i.e. factorized or expanded or something like that), but the order doesn't matter. So it is important to try something out.
HINT: But this is rather primitive. Better to look in the section about *Equations* *(page 85)* for better way of grading equations.

1.4.4 Working with StringTools

Most formulas in the form of **expressions** can be easily matched with the *Grading Code*.

Maple doesn't always see the expressions as equal, but subtracting and then simplifying always leads to good results.
With the *Grading Code* `evalb(simplify(($RESPONSE)-($antw))=0);` the grading of an ordinary formula is then very flexible with regard to the student's notation. (This does not apply to matrices, sets and equations etcetera.)
Sometimes additional programming is required if you want to force some form of an expression. You can actually look at the formula the student is typing on two ways.

1) You look at the $RESPONSE as already explained.
2) You are viewing **the string** of the student's response which is "$RESPONSE". So literally what the student has typed.
You have a number of tools available for this, for example, count the number of certain characters that occur in the string "$RESPONSE" and do something with them in the *Grading Code*.
Maple includes a package (StringTools) of commands that can work with strings. A string is nothing more than a number of characters in a row and everything between double quotes.
So the student's response can easily be made into a string by just putting double quotes around it.
This works best if you have set the formula entry settings to *Maple Syntax and Text entry only*.



Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Figure 1.74: Formula settings with Maple Syntax

In section *Analysing the Response (page 61)* a number of hints are given to find out exactly what the string of the student's response looks like at the different settings as shown in the figure above.
It then looks at what exactly the student has typed and how that is translated into a string.
An example could be that the number of times the character q in the typed answer of the student is counted. So you can check with the boolean command evalb whether that number is smaller than 2 or not (true or false).

```
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","q")<2);
```

This means: count the number of occurrences of q in the string of the student's response and check with evalb whether this is less than 2.

You can also "search" for a particular character in the string of the student's answer using Search from the StringTools package.

```
evalb(StringTools[Search]("/","$RESPONSE")=0);
```

This means: find where a slash occurs in the string of the response. The result will be a number that indicates where the first slash appears in the string and if the result is zero, then there will be no slash in the string.
HINT: So this command can only be used to check whether the character does not occur or occurs at least once by demanding:
`evalb(StringTools[Search]("/","$RESPONSE")>0);`There are no more options.
You can also check with this command whether certain **sequence of characters** may not occur in the string.

```
evalb(StringTools[Search]("factor","$RESPONSE")=0);
```

This means that the sequence of characters that make up the word "factor" may or may not appear in the string of the student's response (true or false). The Search command returns a number if the character combination does occur, namely where its beginning in the string is found. If it returns zero, the character combination does not occur. (You can optionally test that it occurs at least once by requiring >0.)

HINT: Still, in a sense you can count the number of times a character combination occurs within a string. It goes as follows:
You search the string for the specific character combination with `SearchAll`. The result is a row of numbers that indicates where the first character of this character combination is located in the string.

Then you **make a set of it** by putting curly braces around it and ask for the number of operands in this set with the command `nops`. So you know how many times this character combination occurs in the string.
Below are a few lines in Maple to try out.

```
> StringTools[SearchAll]("pq","a+p*b+pq+q^2-1/q+1/pq");
                                   7, 20                                  (1.28)
> nops({StringTools[SearchAll]("pq","a+p*b+pq+q^2-1/q+1/pq")});
                                     2                                    (1.29)
```

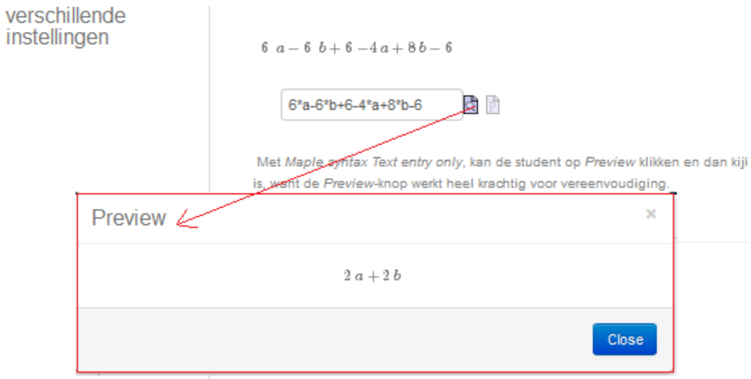
So by programming the next line in the *Grading Code* you can ensure that, for example, the combination x^2*y^2 occurs less than 2 times in the string of the student's answer. (Of course you still have to check whether the answer in its entirety is equal to the correct answer.)

```
evalb(nops({StringTools[SearchAll]("x^2*y^2","$RESPONSE")})<2);
```

HINT: You can also take variables in this command, even if the variable consists of more characters, for example:

```
evalb(nops({StringTools[SearchAll]("$x","$RESPONSE")})<2);
```

In the following example you can see at different settings (*Text entry only* of *Symbol entry only*) asking for a simplification of a formula.



verschillende instellingen

$6\ a-6\ b+6-4a+8b-6$

6*a-6*b+6-4*a+8*b-6

Met *Maple syntax Text entry only*, kan de student op *Preview* klikken en dan kij
is, want de *Preview*-knop werkt heel krachtig voor vereenvoudiging.

Preview

$2\,a+2\,b$

Figure 1.75: Automatic simplify

The above command asks to simplify the given formula.

HINT: The *Preview* feature on *Maple syntax Text entry only* is so powerful it foretells simplification as you can see in section *Figure 1.75 (page 53)* .

In such a case there are two possibilities.
1) You use the *Custom Previewing Code* containing the following statement:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

See section *Prevent Automatic symplify with Preview (page 58)*.

2) You use the *Symbol entry only* setting, so that the student is offered an Editor where no *Preview* button is available. After all, the Editor itself is the preview.
In either case, you can check with the *Grading Code* for the number of specific characters in the student's answer. See also section *The Analysis of the Response (page 61)*.

In the programming of the questions *Grading Code,* as you can see in section *Simplify Figure 1.75 (page 53)* it is of course not sufficient to check whether the answer of the student, subtracted from the correct answer, yields zero. After all, then the student can simply retype the question to get a correct grade. Similar terms should be combined here. In the *Grading Code* you can program that the number of times that a character (combination) occurs, for example, must be less than 2. The student's answer is then first made into a string by putting double quotes around it: "$RESPONSE" and then the number of characters can be counted with the command `CountCharacterOccurrences`. In this way, the form in which the student has to give the answer can be enforced.
However, this only applies to single characters. The `Search` or `SearchAll` commands are more useful here because the variable $x and $z are alternating different characters or character combinations.
Below you see the *Grading Code* for the additional requirement that the character combination $z may appear less than 2 times and likewise other characters or character combinations. (In the *Algorithm* the variable $z for example was:
`$z=switch(rint(6),"a","b","x^2","yz","p","qr");`.)

```
evalb($RESPONSE=$ans2) and evalb(nops({StringTools[SearchAll]
("$z","$RESPONSE")})<2) and evalb(nops({StringTools[SearchAll]
("$x","$RESPONSE")})<2);
```

HINT: In these cases, therefore, do not use the *Formula* setting. After all, Formula cannot handle character combinations. Furthermore, the *Preview* of the setting with *Formula* will definitely look bad.
Advice: never use the *Formula* setting unless there is no other option. See for an example in section *Fractions (page 71)* in *Figure 1.94 (page 77).*

HINT: Another last resort is to check that the student is typing the answer exactly as you want. You simply check exactly whether the student is typing the prescribed answer. In the *Algorithm* you then prepare the "string answer" checking whether it has the desired form for example with the algorithmic variable:

```
$stringanswer=maple("convert($answer,string)");
```

In the *Grading Code* you will program as follows:

```
evalb(convert($RESPONSE,string)=$stringantwoord);
```

However, if you think the student might be able to type spaces in their answer, please program the following in the *Grading Code* to be safe:

```
stringresponse:=StringTools[Remove]
(IsSpace,convert($RESPONSE,string)): evalb(stringresponse=
$stringantwoord);
```

or

```
stringresponse := StringTools[Remove](IsSpace, "$RESPONSE");
  evalb(StringTools[SubString](stringresponse, 1..-1)=$stringantwoord);
```

This means that you first remove all spaces from the student's answer and then check whether the result is exactly as you had prepared it.
The `Remove` command from the StringTools package can also extract other characters from the string. And a space is also a character so the following could also have been done:

```
StringTools[Remove](" ",convert($RESPONSE,string));
```

HINT: Also look in Maple for the Maple commands `searchtext` and `searchText` which can search a string case insensitive and case sensitive respectively. This command is outside the StringTools package.
However, the commands discussed above are more than enough.

If you want to do even more with strings, see the *Randomization* manual which has a special section on strings.

HINT: Suppose that when simplifying fractions, you want to see the answer as one fraction. The correct answer is generated by Maple, but in the feedback the student may see the correct answer as (6/5)/(x^2*y) while you want to be correct: 6/(5*x^ 2*y).

You can then enter the string you want the student to type in the correct answer in the *Maple graded* question. The command `printf` removes the quotes. So for example enter in the *Answer* section: `printf("6/(5*x^2*y)");`

Another way to get one fraction in the feedback is to convert the answer to mathml with `printf(MathML[ExportPresentation]($antw))` in the *Answer* section.

HINT: You get the same at the square root in the feedback.
Make a variable in the *Algorithm* for example $antw="sqrt(3)"; (So not with Maple because then you will get 3^(1/2)).
You can just use this variable in the *Grading Code.* As correct answer in the *Answer* section do not fill in $antw but `printf("$antw");` (This command removes the quotes and writes exactly what is between quotes, nothing more.)
Another way to avoid getting 3^(1/2) in the feedback is to write the answer to mathml with `printf(MathML[ExportPresentation]($antw))` in the *Answer* section.

1.4.5 Exponents, logarithm and Pi

The exponential function

It is important to distinguish between arithmetic inside maple and outside maple.

- **Inside maple** is the exponential function e^x defined as exp(x) and e is not recognized as Eulers number.
- **Outside maple**, exp(...) is the same as e^(...) and can therefore be used interchangeably.
  In the *Algorithm* you see that both variables wil give as a result 2.72 where you don't use maple.
  ```
  $test3=decimal(2,e);
  $test4=decimal(2,exp(1));
  ```

HINT: In the *numeric question type* the student is not allowed to use exp(...) or e^(...) even when *Accept arithmetic* is checked.

Within the *Maplegraded question type*, the correct answer must be defined with exp(..) and the *Grading Code* is easy as usual.
In both situations of the settings *Maple Syntax Text entry only* and *Symbolic entry only* the student must also enter exp(...) if he means $e^{(\ ...\ )}$.

HINT: For the setting *Formula* in the *Maple graded question type* (you know this is not recommended), the student can both enter exp(...) and e^(...) for a correct grading.
Also the *Mathematical Formula question type* (you know this question type is not recommended), the student can both enter exp(...) and e^(...) for a correct grading.

HINT: If you want the student typing e^(...) as well as exp(...) for a correct grading, while the correct answer is exp(...), you can take the following *Grading Code* in both situations of the setting *Maple Syntax Text entry only* and *Symbol entry only.* (The editor has no special button for e.)

```
resp:= "$RESPONSE":
```

```
fixed_resp:= StringTools[SubstituteAll](resp, "e^", "exp"):
evalb(($ans)-(parse(fixed_resp))=0);
```

It means that you first replace e^ with exp everywhere in the string of the answer.
This gives you a new string of which you make a real formula to calculate with the help of parse.
The fact that StringTools is used here for substitution is done because ordinary substitution does not work with e^: a character followed by an operator.

If the exponent consists of only one character

In the situation that e^x should be seen as exp(x) in the correct answer and therefore if both have to be graded correctly, you cannot use the above code because that would change e^x to expx without brackets.

Suppose exp(x) and e^x both need to be graded as correct, you can take:

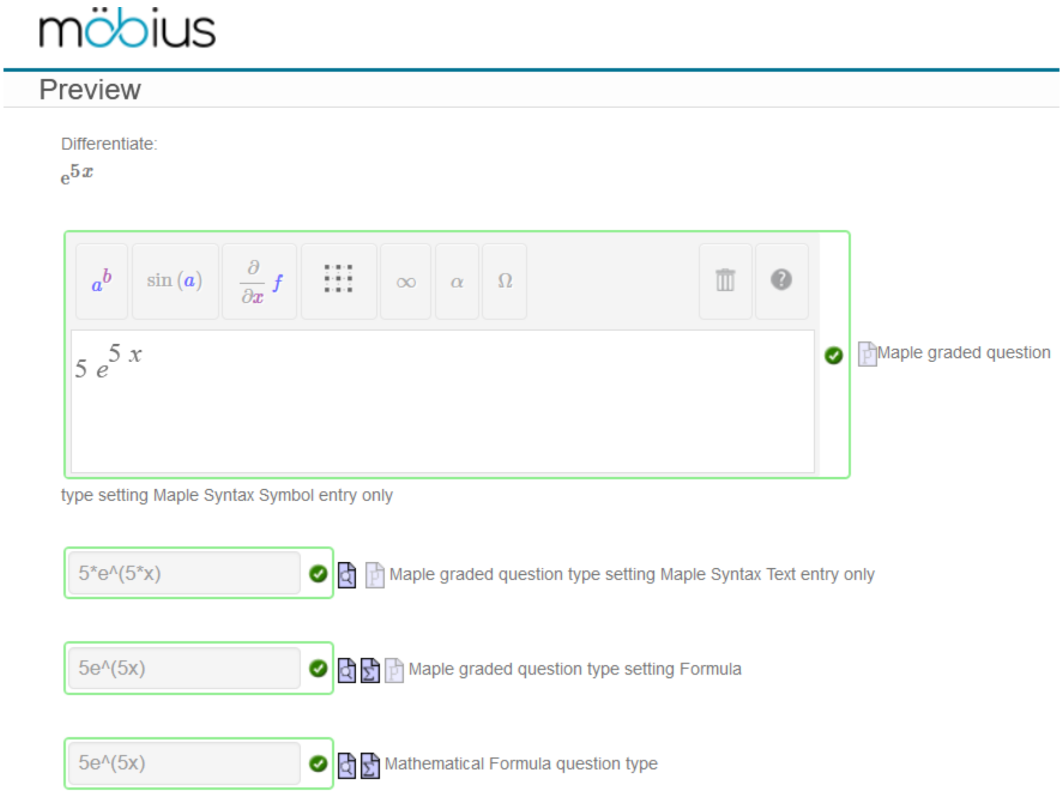
```
evalb(subs(e^x=exp(x),$RESPONSE)-exp(x)=0);
```

Here the string and the command parse are not necessary. You can simply work with substitution and replace e^x with exp(x) in the student's answer.

HINT: For the setting *Formula* of the *Maple graded* question type the e^ is automatically converted to exp and the following *Grading Code* wille be sufficient:

```
evalb(($ans)-($REPSONSE))=0);
```

The disadvantage is that at the *Formula* setting the student does not have to type asterix. So this question type is not recommanded.



Differentiate:

e^{5x}

$5\ e^{5\ x}$ ✓ Maple graded question type setting Maple Syntax Symbol entry only

5*e^(5*x) ✓ Maple graded question type setting Maple Syntax Text entry only

5e^(5x) ✓ Maple graded question type setting Formula

5e^(5x) ✓ Mathematical Formula question type

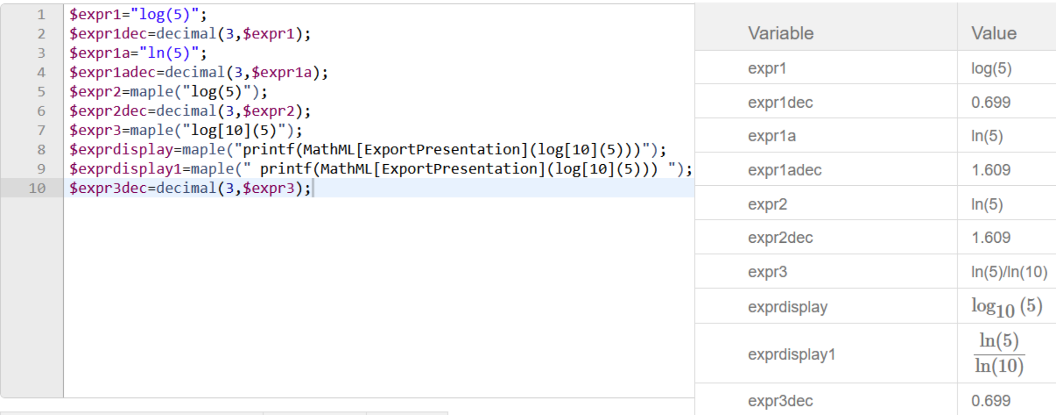
Figure 1.76: How to work with the exponential function

Logarithm

It is important to distinguish between arithmetic inside maple and outside maple.

- **Inside maple** there is no difference between $\log(x)$ and $\ln(x)$. Both are the natural log. The Brigg logarithm in maple you enter as log[10](x).
- **Outside maple** there is a difference! Then $\log(x)$ has the meaning of the Briggs logarithm: $\log_{10}(x)$ and $\ln(x)$ is the natural logarithm.

So pay close attention to whether you work inside or outside maple when you use log as you can see in the figure below.



```
$expr1="log(5)";
$expr1dec=decimal(3,$expr1);
$expr1a="ln(5)";
$expr1adec=decimal(3,$expr1a);
$expr2=maple("log(5)");
$expr2dec=decimal(3,$expr2);
$expr3=maple("log[10](5)");
$exprdisplay=maple("printf(MathML[ExportPresentation](log[10](5)))");
$exprdisplay1=maple(" printf(MathML[ExportPresentation](log[10](5))) ");
$expr3dec=decimal(3,$expr3);
```

Variable	Value
expr1	log(5)
expr1dec	0.699
expr1a	ln(5)
expr1adec	1.609
expr2	ln(5)
expr2dec	1.609
expr3	ln(5)/ln(10)
exprdisplay	$\log_{10}(5)$
exprdisplay1	$\frac{\ln(5)}{\ln(10)}$
expr3dec	0.699

Figure 1.77: Logarithm inside and outside maple

HINT: Look at the trick with the space for the performance of the MathML conversion of the formula $exprdisplay1 by maple.

HINT: For the *Numeric question type* you work outside maple so log(5) means the Briggs logarithm and ln(5) means the natural logarithm.

The number π

It is important to distinguish between arithmetic inside maple and outside maple with the number π.

- **Inside maple,** only the number denoted by Pi is the constant 3.14... . The character pi is simply a variable.
- **Outside maple** both Pi and pi are understood as the constant 3.14....

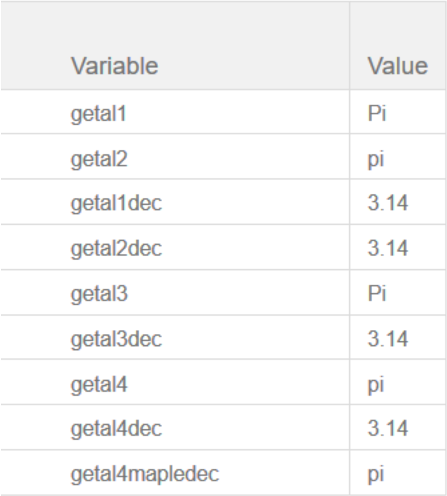
That is, in the *Numeric* and *Mathematical Formula* question types, it doesn't matter whether you use Pi or pi.

So as soon as you are in the question type *Maple graded*, there is a distinction, because then you are within maple with everything (*Grading Code* and correct answer, etc.)

In the *Algorithm* you will see some examples:

```
$getal1="Pi";
```

```
$getal2="pi";
$getal1dec=decimal(2,$getal1);
$getal2dec=decimal(2,$getal2);
$getal3=maple("Pi");
$getal3dec=decimal(2,$getal3);
$getal4=maple("pi");
$getal4dec=decimal(2,$getal4);
$getal4mapledec=maple("evalf($getal4)");
```



Variable	Value
getal1	Pi
getal2	pi
getal1dec	3.14
getal2dec	3.14
getal3	Pi
getal3dec	3.14
getal4	pi
getal4dec	3.14
getal4mapledec	pi

Figure 1.78: Difference between Pi and pi

HINT: It is useful if you take measures in the *Grading Code* of the *Maple graded* question type to approve both Pi and pi.
You then substitute all pi by Pi in the student's answer and then you match with the correct answer.

```
evalb($antw-subs(pi=Pi,$RESPONSE)=0);
```

```
or for safety:
```

```
evalb(subs(pi=Pi,$antw-($RESPONSE))=0);
```

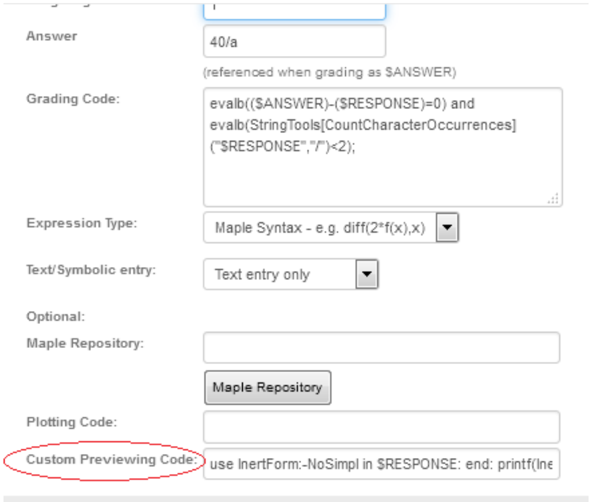
HINT: The same *Grading Code* can be used for the *Symbol entry only* setting, although the button for π is always translated with Pi if you opt for *Maple Syntax*.

1.4.6 Stop Automatic simplification for Preview

The automatic simplification that the *Preview* shows with the *Maple graded question type* in the setting *Text entry only* can largely be solved by providing it in *Custom Previewing Code*.
Maple has a new InertForm package that includes the ability to write the unsimplified formula to MathML.
So put in the *Custom Previewing Code* the following:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

The student's response is not simplified and maple converts the result into MathML
Always try to see if the desired result is achieved and otherwise you can always switch to the *Symbol entry only* setting where no *Preview* is possible.



Answer: 40/a

(referenced when grading as $ANSWER)

Grading Code:

```
evalb(($ANSWER)-($RESPONSE)=0) and
evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","/")<2);
```

Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

Text/Symbolic entry: Text entry only

Optional:

Maple Repository:

Plotting Code:

Custom Previewing Code: use InertForm:-NoSimpl in $RESPONSE: end: printf(Ine

Figure 1.79: Stop automatic simplifying in the Preview by using Custom Previewing Code

See also in section *Custom Previewing Code (page 42)* .

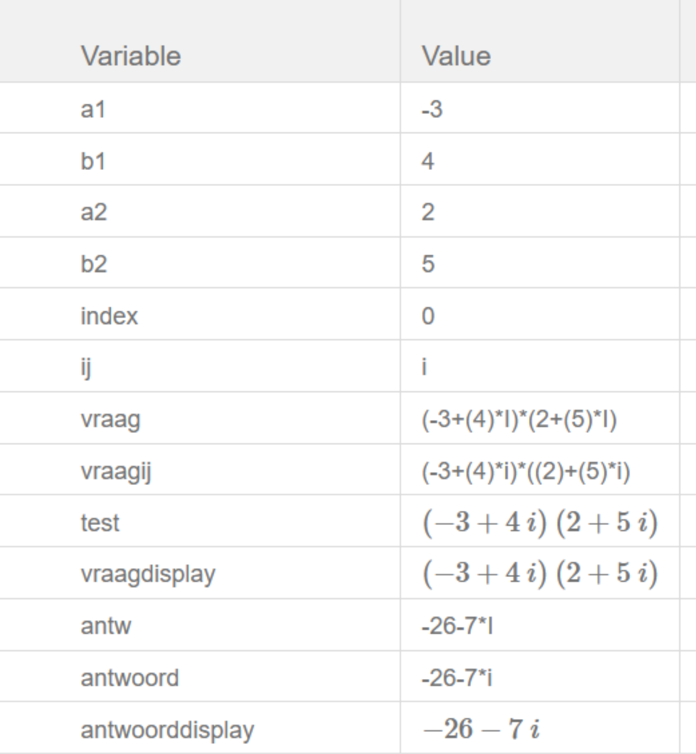
See also in section *Stop automatic simplification (page 16)* at preparing formulas in the *Algorithm*.

1.4.7 Complex numbers

The imaginary unit in Maple for calculation is the capital character I while you can make the choice between i and j for your question text and the students formula.
Let maple do the calculation with the complex numbers to get the correct answer and convert the result with I to i or j for the Imaginary unit.
It can be done with the maple command `subs` or with the statement `local I:=i:`

Start with the *Algorithm*:

```
$a1=switch(rint(2),range(-5,-1),range(1,5));
$b1=switch(rint(2),range(-5,-2),range(2,5));
$a2=switch(rint(2),range(-5,-2),range(2,5));
$b2=switch(rint(2),range(-5,-2),range(2,5));
$index=0;
$ij=switch($index,"i","j");
$vraag="($a1+($b1)*I)*($a2+($b2)*I)";
$vraagij="($a1+($b1)*$ij)*(($a2)+($b2)*$ij)";
$test=maple("printf(MathML[ExportPresentation]($vraagij))");
$vraagdisplay=maple("use InertForm:-NoSimpl in $vraagij: end:
 printf(InertForm:-ToMathML(%))");
$antw=maple("evalc($vraag)");
$antwoord=maple("local I:=$ij: $antw");
$antwoorddisplay=maple("printf(MathML[ExportPresentation]
($antwoord))");
```



Variable	Value
a1	-3
b1	4
a2	2
b2	5
index	0
ij	i
vraag	(-3+(4)*I)*(2+(5)*I)
vraagij	(-3+(4)*i)*((2)+(5)*i)
test	$(-3+4\,i)\,(2+5\,i)$
vraagdisplay	$(-3+4\,i)\,(2+5\,i)$
antw	-26-7*I
antwoord	-26-7*i
antwoorddisplay	$-26-7\,i$

Figure 1.80: Complex numbers

In the *Algorithm* you see the choice i for the imaginary unit.
The variable $vraag is for the calculation in maple. The variable $vraagij is prepared for the text of the question where you have to convert this to MathML ($vraagdisplay).

The variable $antw is the calculated answer. In the *Grading Code* you will use the correct answer $antwoord (with the i as the imaginary unit).
The correct answer converted to MathML you will use in the feedback of the question.

HINT: For the next two lines in the *Algorithm*:

```
$antw=maple("evalc($vraag)");
$antwoord=maple("local I:=$ij: $antw");
```

One line with substitution also works:

```
$antwoord =maple("subs(I=$ij,evalc($vraag))");
```

The following *Grading Code* for the *Maple graded question* type is:

```
evalb(($antwoord)-($RESPONSE)=0) and
 evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","$ij")<2);
```

In this *Grading Code* you will use $ij which is i in this case where you enforce the student to use this character only once or less.

In the *Answer* section you can give the correct answer $antwoord for the students feedback.

In the *Custom Previewing Code* you can program the following:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

Some maple commands for complex numbers:

`evalc(..)` will give the complex number in the form a+I*b. With this command `evalc` (evaluate complex) we let the program know that the unknown parameters are real numbers!

`abs(..)` will give the absolute value of the complex number.

`argument(..)` will give the argument of the complex number.

`convert(..,polar)` will give the polar coordinates of the complex number.

`conjugate(..)` wil give the complex conjugate of the complex number.

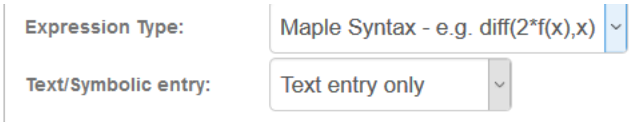
1.4.8 Expressions with the character I

If the character I appears in an expression that is not the imaginary unit, Maple has to know that. So every time you need it, ensure that Maple does not see the character I as an imaginary unit. This can be done with an extra command in the maple statement: Mind the space behind I.

- In the *Algorithm*:
  ```
  $antw="3*P*L/(E*I)";
  $antwdisplay=maple("local I:=`I
   `:printf(MathML[ExportPresentation]($antw))");
  ```
- In the *Answer* section of the *Maple Graded* question type:
  ```
  local I:=`I `:printf(MathML[ExportPresentation]($antw));
  ```
- In the *Grading Code* of the Maple Graded question type (opt for the *Expression Type* the *Maple Syntax*):
  ```
  local I:=`I `:evalb(($antw)-($RESPONSE)=0);
  ```
- *Expression Type*: *Maple Syntax*.
- In the *Custom Previewing Code*: (with setting *Text entry only*)
  ```
  local I:=`I `:printf(MathML[ExportPresentation]($RESPONSE));
  ```

1.4.9 The Analysis of the Response

In different situations, you may want to know what the student's response string looks like with a view to using StringTools to enforce any notations on the student.
In the section *Working with StringTools (page 51)* a number of examples are given of situations where scanning the student's answer is certainly useful. In the *Grading Code* you can then use it.
Only the *Maple graded* question type has the possibility of a *Grading Code.* Other question types definitely not.
If you are programming something in this *Grading Code*, it is important that you not only anticipate what the student may be entering, but also how you have the settings for the answer is important.
In the three different settings, the form of the string of the student's response ("$RESPONSE") may be different.



Expression Type: Maple Syntax - e.g. diff(2*f(x),x)

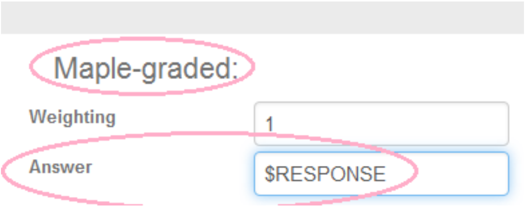
Text/Symbolic entry: Text entry only

Figure 1.81: Expression type for the Maple graded question type

The three possible settings are:
1) *Maple syntax Text entry only* **(recommanded)**
2) *Maple syntax Symbol entry only*
3) *Formula* **(not recommanded)**
We prefer not to use this last option with *Formula* for formulas, unless there is no other option.

The first setting with *Maple syntax Text entry only* is the easiest, because the string "$RESPONSE" simply shows exactly what the student has literally typed in.
Number of brackets, asterix and slash for fractions and so on is literally included in the string. However, sometimes this setting is undesirable, for example if you don't want the student to use the automatic simplification that becomes visible in *Preview*. But in most cases this can be provided for as discussed in section *Stop automatic simplification (page 58)*.
This setting also allows the student to use Maple commands. But you can prevent that by checking the string response to it using StringTools. See section *Working with StringTools (page 51)*.

We are now going to see how you can find out what the form of the "$RESPONSE" is.
This is very simple if, in a small Maple graded test question with different formula settings in the section for *Answer*, (intended for the correct answer), you now enter "$RESPONSE" instead of the real correct answer (see *Figure 1.82 (page 62)*).
In the feedback after grading the (wrong answered) question, you can see exactly how what has been typed in is translated into a string.



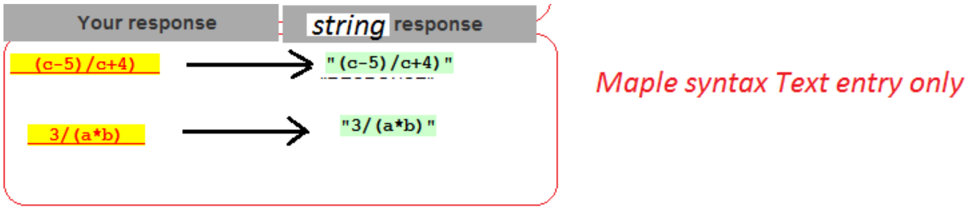
Maple-graded:

Weighting 1

Answer $RESPONSE

Figure 1.82: Getting the string of the RESPONSE

If you have created this little test question and you just fill in a wrong answer, then the feedback will literally say what you had typed in (*Your response*) and as *Correct response* it will say what kind of **string** the system makes of your typed answer. This is of course not a problem in the case of the setting with *Text entry only*, because you saw what was typed in as an answer, but in case you had done the settings with *Symbolic entry only*, it is a different story.

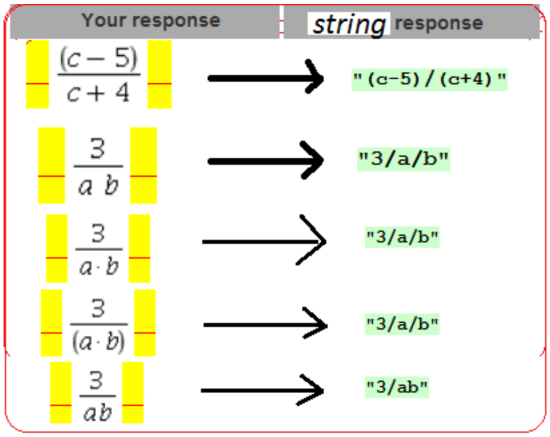


Your response	string response
(c-5)/c+4)	"(c-5)/c+4)"
3/(a*b)	"3/(a*b)"

Figure 1.83: Showing the string "$RESPONSE" with Text entry only

In the figure above, the settings with *Text entry only* are easy, because literally the string of the response always shows what the student has typed in.

In the setting of this question with *Symbol entry only*, we are at the mercy of the way in which the 2-dimensional formula, typed by the student in the Editor, is converted into a string, as can be seen in the following figure.



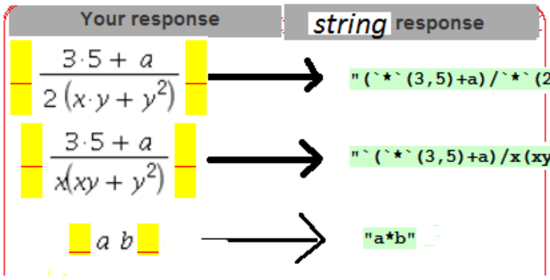
Your response	*string* response
$\frac{(c-5)}{c+4}$	`"(c-5)/(c+4)"`
$\frac{3}{a\ b}$	`"3/a/b"`
$\frac{3}{a\cdot b}$	`"3/a/b"`
$\frac{3}{(a\cdot b)}$	`"3/a/b"`
$\frac{3}{ab}$	`"3/ab"`

Maple syntax Symbol entry only

Figure 1.84: Showing the string "$RESPONSE" with Symbol entry only

In the figure above you can see that it does not matter whether you put brackets around the numer or denominator in the Editor or not, the brackets will appear in the string anyway because this is pure, unmistakable Maple syntax. It can also be seen that if you type a space in the Editor, it is correctly translated into a multiplication. For example, if the student does not type a space between two characters, this is neatly understood as a whole.

More examples with the setting *Symbol entry only*, how the typed formula is translated into a string can be seen below:



Your response	*string* response
$\frac{3\cdot 5+a}{2\left(x\cdot y+y^2\right)}$	``"(`*`(3,5)+a)/`*`(2,x*y+y^2)"``
$\frac{3\cdot 5+a}{x(xy+y^2)}$	``"`(`*`(3,5)+a)/x(xy+y^2)`"``
$a\ b$	`"a*b"`

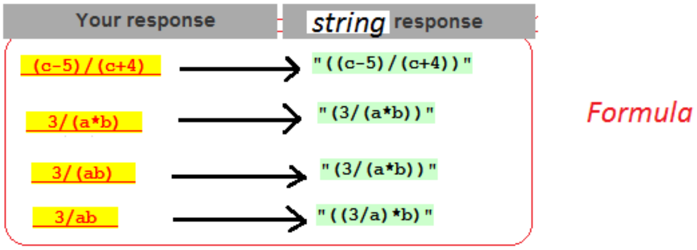
Maple syntax Symbol entry only

Figure 1.85: Getting the string "$RESPONSE" with setting Symbol entry only

The figure above shows how multiplication using an asterix is defined, but often in different ways. Also pay attention if you do not type a space just before the beginning of the brackets, that there is a function prescription similar to f(x).

It is quite possible to work with StringTools in the *Grading Code*, but with premeditation, because the number of brackets and slashes and the like will not correspond to what you might have had in mind. However, the translation to *Maple Syntax* is done in an unambiguous manner and there can be almost no misunderstanding if the student takes into account that a space between two characters is translated into a multiplication.

Finally, we show how the conversion is if the setting is set to Formula.
This so-called "user-friendly" setting is in fact NOT very user-friendly, because it is open to multiple explanations! For example, character combinations are always seen as multiplications and also translated as such into a string. It's just that you know.



Your response	string response
(c-5)/(c+4)	"((c-5)/(c+4))"
3/(a*b)	"(3/(a*b))"
3/(ab)	"(3/(a*b))"
3/ab	"((3/a)*b)"

Formula

Figure 1.86: Getting the string "$RESPONSE with setting Formula

It is noticeable that a few extra brackets are put around the formula. With this setting it is still possible to work with StringTools in the *Grading Code*, but here too you have to do that with premeditation. Counting the number of brackets will therefore be difficult. When in doubt, you can always do such a test as discussed above in *Figure 1.82 (page 62)* by typing in the *Answer* section: "$RESPONSE" and trying out things.
Further examples of the response analysis can be found in section *Square root (page 77)* in *Figure 1.95 (page 78)* and *Figure 1.96 (page 78)*.

Examples of the analysis of the response of Matrices and Vectors can be found in section *Matrices and Vectors (page 118)*.

Examples of the analysis of the response of differentials and integrals can be found in section *Integrals (page 107)*.

1.4.10 Sets

If you want to grade a row of numbers or formulas where the order is not important, it is useful to match sets. You can think of the solution set of (systems) equations or something similar.
See section *(page 89)* for a comprehensive example of this. The student then types in the different answers, separated by commas ($RESPONSE). You then create a set of them and match it {$RESPONSE} with the solution set that you first prepared in the *Algorithm*.
In case of matching sets, the student does not have to type in the curly brackets. Make sure that for the correct answer you formulate the individual answers with a comma between them and with the *Grading Code* you can use the set prepared in the Algorithm.
For example, you want to grade the set $verz={a,b,c}. The student enters: a,b,c and that is the $RESPONSE. So in the *Grading Code*, you match the set of $verz with {$RESPONSE}. Please note that you do not present the set of $verz in the *Answer* section. The student might think when he will see the correct answer that he should have entered curly brackets If necessary, work with operands of the set. (see section *(page 66)*.)

HINT: If you want the student may type the set with two prepared answers: $antw1 and $antw2, with curly brackets as well as a sequence with both a correct grading, then put in the *Grading Code* the following:

```
evalb({$antw1,$antw2}={$RESPONSE})
or
evalb({$antw1,$antw2}=$RESPONSE);
```

In the *Answer* section you put:

```
$antw1,$antw2
```

And in the *Custom Previewing Code* you fill in the following:

```
if evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE",",")=0) then printf("Separate the two answers with
 comma") else if evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","{")=0) then printf(MathML[ExportPresentation]
({$RESPONSE})) else printf(MathML[ExportPresentation]($RESPONSE))
 end if end if;
```

HINT: In sets, duplicate elements are seen as single.

HINT: An interesting alternative for sets is the question type *Mathematical Formula (page 129)* wherever it is possible to grade unordered lists as well ordered lists.

HINT: When matching sets, keep in mind that they always contain exact elements and no decimal numbers combined with integers. The numbers 3.0 and 3. and 3, for example, are not seen as the same in the context of a set. For example, the student gives a rounded integer and Maple sees this as not the same if it is an element of a set! as the command lines below in Maple show.

```
> A:={3.0,5.23,1/3};B:={3,5.23,1/3};
```

$$A := \left\{\frac{1}{3}, 3.0, 5.23\right\}$$

$$B := \left\{3, \frac{1}{3}, 5.23\right\} \tag{1.30}$$

```
> evalb(A=B);
```

$$\mathit{false} \tag{1.31}$$

```
> convert(A,rational);
```

$$\left\{3, \frac{1}{3}, \frac{523}{100}\right\} \tag{1.32}$$

```
> evalb(convert(A,rational)=convert(B,rational));
```

$$\mathit{true} \tag{1.33}$$

It is also useful to first multiply the numbers of the set by 100 and then round them:

```
> map(x->round(100*x),A);
```

$$\{33, 300, 523\} \tag{1.34}$$

Possible with the **elementwise operator (~)**:

```
> round~(100*~A);
```

$$\{33, 300, 523\} \tag{1.35}$$

```
> evalb(map(x->round(100*x),A)=map(x->round(100*x),B));
```

$$\mathit{true} \tag{1.36}$$

```
> evalb(round~(100*~A)=round~(100*~B));
```

$$\mathit{true} \tag{1.37}$$

With the first trick, by converting the decimal numbers to real fractions with `convert(...,rational)`, matching sets is fine. This command works on expressions, sets, equations, inequalities, etc. and deals with all parts of the expression, equation or set at once. An example of this can be found in the example of *(page 97)* in section *Inequalities*.

Another possibility is to first make all the numbers of the set, for example, 100 times as large and then round with the command `round`. So if you demand an accuracy of 2 decimal places, this is also a nice way of getting rid of decimals. However, then you must work with the `map` command or with the elementwise operator (~) to treat all the numbers of the {$opl} set separately. You can do the same with the set {$RESPONSE}.

```
evalb(map(x->round(100*x),{$opl}) = map(x->round(100*x),
{$RESPONSE}));
```

Also useful is working with intersect and union of sets.
For example, do you want to check a number $D1 for the number of different factors that this number contains? With the following command you can check whether the intersect of the set of factors of the correct answer and the set of the factors of the given answer is the same as that of the given answer.

```
evalb(numtheory[factorset]($D1) intersect numtheory[factorset]
($RESPONSE)=numtheory[factorset]($RESPONSE));
```

It therefore means that the set of factors of the given answer falls entirely within the set of factors of the correct answer.

```
> numtheory[factorset](258);
```

$$\{2, 3, 43\} \tag{1.38}$$

```
> numtheory[divisors](258);
```

$$\{1, 2, 3, 6, 43, 86, 129, 258\} \tag{1.39}$$

You can also check the student's answer to see if the set contains certain elements.

You can assign each of the three elements its own 'weight' with, for example, the following *Grading Code*:

```
`if`(member(-1-sqrt(2),{$RESPONSE}),0.4,0)+`if`(member(-1+sqrt(2),
{$RESPONSE}),0.4,0)+`if`(member(1,{$RESPONSE}),0.2,0);
```

1.4.11 Expressions with coefficients and operands

It's useful if you know some Maple commands to work with individual operands, coefficients, and properties of a formula.

- A test to see if a polynomial is of a certain **degree**, for example of the fourth degree in x.
  `evalb(degree($RESPONSE,x)=4);`
  In the following way, you can also check whether the degree of the student's response is the same as the degree of the correct answer $antw.
  `evalb(degree($RESPONSE,x)=degree($antw,x));`
- In the lines below in Maple you can see how a series of a function is requested and converted to a polynomial. The **coefficients** of a fifth-degree polynomial are now requested. The coefficient of x^3 can easily be retrieved with `coeff(f,x^3)`. You can use the `coeffs` command to retrieve all coefficients of the polynomial. It is then possible to turn the generated row into a set (by putting curly brackets around it) and possibly release a function on **all** elements of that set with the maple command map. The command `map` can do something at once with all elements of a matrix or set or something similar. In this case, all elements of the set are multiplied by 100 and then rounded by `round`. So you can actually view all the coefficients of a polynomial to 2 decimal places.

```
> f:=evalf(convert(series(sin(3*x),x=0),polynom));
```

$$f := 3.\,x - 4.500000000\,x^3 + 2.025000000\,x^5 \tag{1.40}$$

```
> coeff(f,x^3);
```

$$-4.500000000 \tag{1.41}$$

```
> coeffs(f);
```

$$3., -4.500000000, 2.025000000 \tag{1.42}$$

```
> map(x->round(100*x),{coeffs(f)});
```

$$\{-450, 203, 300\} \tag{1.43}$$

```
> round~(100*~{coeffs(f)});
```

$$\{-450, 203, 300\} \tag{1.44}$$

```
> f1:=evalf(-6*sin(-8)*(x+3)+3*cos(-8));
```

$$f1 := 5.936149480\,x + 17.37194834 \tag{1.45}$$

```
> coeffs(-6*sin(-8)*(x+3)+3*cos(-8));
Error, invalid arguments to coeffs
> coeffs(evalf(-6*sin(-8)*(x+3)+3*cos(-8)));
```

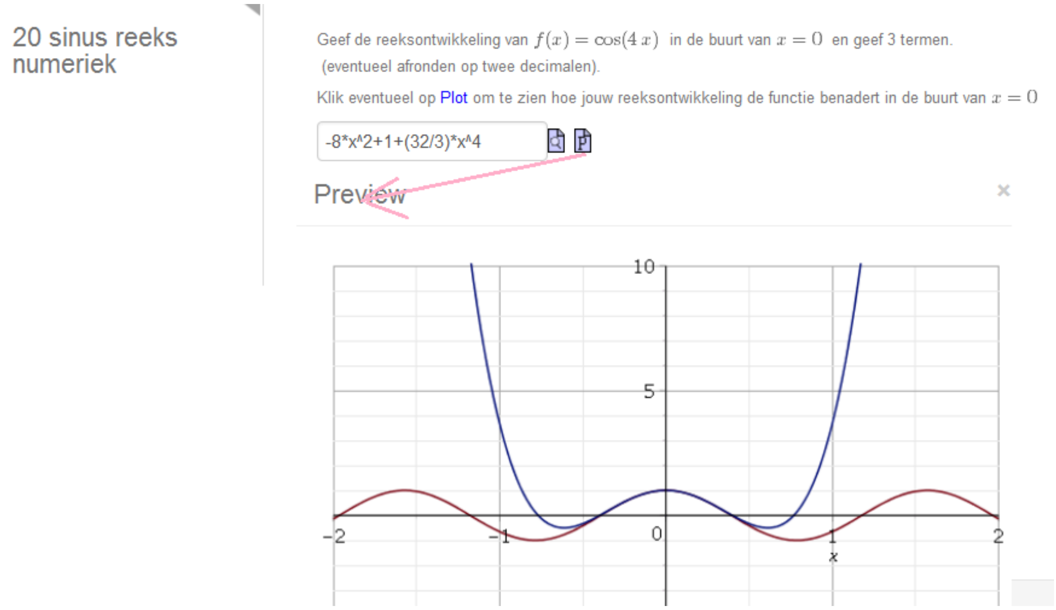
$$17.37194834, 5.936149480 \tag{1.46}$$

In the above way, you can also use the operator followed by the postfix in the form of a tilde (~) to treat all elements of, for example, a set. (The tilde is the **elementwise operator**.)
Note that you may use the `evalf` command to switch to numeric numbers if the coefficients are not easy to determine with the command `coeffs`.
An alternative is to first write the polynomial without brackets (with `expand`) otherwise retrieving the coefficients will not work.

With these commands you can easily match a polynomial where the coefficients are precisely rounded to, for example, 2 decimal places, with the correct answer, including rounding off the coefficients.

Example with series

See the following example where the series of a function is requested with three terms.



20 sinus reeks numeriek

Geef de reeksontwikkeling van $f(x) = \cos(4\,x)$ in de buurt van $x = 0$ en geef 3 termen.
(eventueel afronden op twee decimalen).
Klik eventueel op Plot om te zien hoe jouw reeksontwikkeling de functie benadert in de buurt van $x = 0$

-8*x^2+1+(32/3)*x^4

Figure 1.87: Matching polynoms with decimal numbers

In the figure above you can see that there is also an opportunity for the student to visualize his own answer together with the original function in a graph!
In the *Plotting Code* you can program the following:

```
plot([$functie,$RESPONSE],x=-2..2,-10..10);
```

See *Figure 1.61 (page 36)* how to use the *Plotting Code.*
The correct answer $reeks is prepared allready in the *Algorithm* with the Maple command: `series`.
The *Algorithm* of the question is as follows:

```
$a=range(2,4);
$f=maple("sin($a*x)");
$g=maple("cos($a*x)");
$functie=switch(rint(2),"$f","$g");
$displayfunctie=maple("printf(MathML[ExportPresentation](f(x)=$functie))");
$reeks=maple("convert(series($functie,x=0),polynom)");
$displayreeks=maple("printf(MathML[ExportPresentation]($reeks) )");
$p=plotmaple("plot([$functie,
$reeks],x=-2..2,-10..10,color=[red,blue],thickness=2,legend=[functie,reeks]),plotoptions='height=350,
 width=250'");
$x0=maple("printf(MathML[ExportPresentation](x=0) )");
```

In the *Algorithm* it can be seen that alternately the function is a sine or a cosine. Furthermore, for the purpose of the feedback, a graph $p is prepared and also the the formula of the correct answer: $displayreeks.

In the *Grading Code* you can program the following:

```
evalb(degree($RESPONSE,x)=degree($reeks,x))
and evalb(StringTools[Search]("series","$RESPONSE")=0)
and evalb(map(x->round(100*x),{coeffs($reeks)})=
 map(x->round(100*x),{coeffs($RESPONSE)})) and
 evalb(nops([coeffs($reeks)])=nops([coeffs($RESPONSE)])) ;
```

This allows you to first check whether the polynomial that the student has typed in has the correct degree. Then it is also checked whether the student does not use the Maple command `series` to give the correct answer. And finally, the set consisting of all coefficients of the student's polynomial is compared with the same set, from the correct answer $reeks. These coefficients are first multiplied by 100 and then rounded, because it is an accuracy of 2 decimal places. As an extra, it is checked whether the number of coefficients of the correct series corresponds to the number of coefficients of the series that the student enters. After all, when equating sets, double elements do not count. Of course, you can also check all coefficients one by one.

See also section *Verzamelingen (page 64)* for more information about sets.

Operands and coefficients of an expression

The **operands** of an expression you can get with the maple command `op`.
There will then be a row of expressions with successively all the operands of the expression. Sometimes it can be useful to first put the expression in order with the command `sort`, so that the list of operands is unambiguous. So you can also get a single operand out of the expression. Possibly also operands of operands (with the help of a list): for example, from the second operand the first take. The number of operands of an expression f can also be called up with `nops(f)`.

```
> f:=evalf(convert(series(sin(3*x),x=0),polynom));
```

$$f := 3.\,x - 4.500000000\,x^3 + 2.025000000\,x^5 \tag{1.47}$$

```
> op(sort(f));
```

$$2.025000000\,x^5,\ -4.500000000\,x^3,\ 3.\,x \tag{1.48}$$

```
> op(1,f);
```

$$2.025000000\,x^5 \tag{1.49}$$

```
> op([2,1],f);
```

$$-4.500000000 \tag{1.50}$$

```
> nops(f);
```

$$3 \tag{1.51}$$

```
> coeffs(-101.5*0.8^n+100);
```

$$100,\ -101.5 \tag{1.52}$$

```
> nops(-101.5*0.8^n+100);
```

$$2 \tag{1.53}$$

```
> coeff(-101.5*0.8^n+100,0.8^n);
```

$$-101.5 \tag{1.54}$$

```
> map(x->round(100*x),{coeffs(-101.5*0.8^n+100)});
```

$$\{-10150,\ 10000\} \tag{1.55}$$

Or with the elementwise operator (~):

```
> round~(100*~{coeffs(-101.5*0.8^n+100)});
```

$$\{-10150,\ 10000\} \tag{1.56}$$

An example of the use of operands in expressions can also be found in figure *Figure 1.88 (page 70)*.
More about operands you can find in section *Square roots (page 81)*.

HINT: If necessary, work in the *Grading Code* with coefficients for a quadratic expression:

```
evalb(degree($RESPONSE,x)=degree($reeks,x)) and

evalb(abs(coeff($RESPONSE,x)-(coeff($reeks,x)))<abs(coeff($reeks,x))*0.02) and
 evalb(abs(coeff($RESPONSE,x^2)-(coeff($reeks,x^2)))<abs(coeff($reeks,x^2))*0.02)
 and

evalb(abs(op(3,sort(expand($RESPONSE)))-
(op(3,sort($reeks))))<abs(op(3,sort($reeks)))*0.02);
```

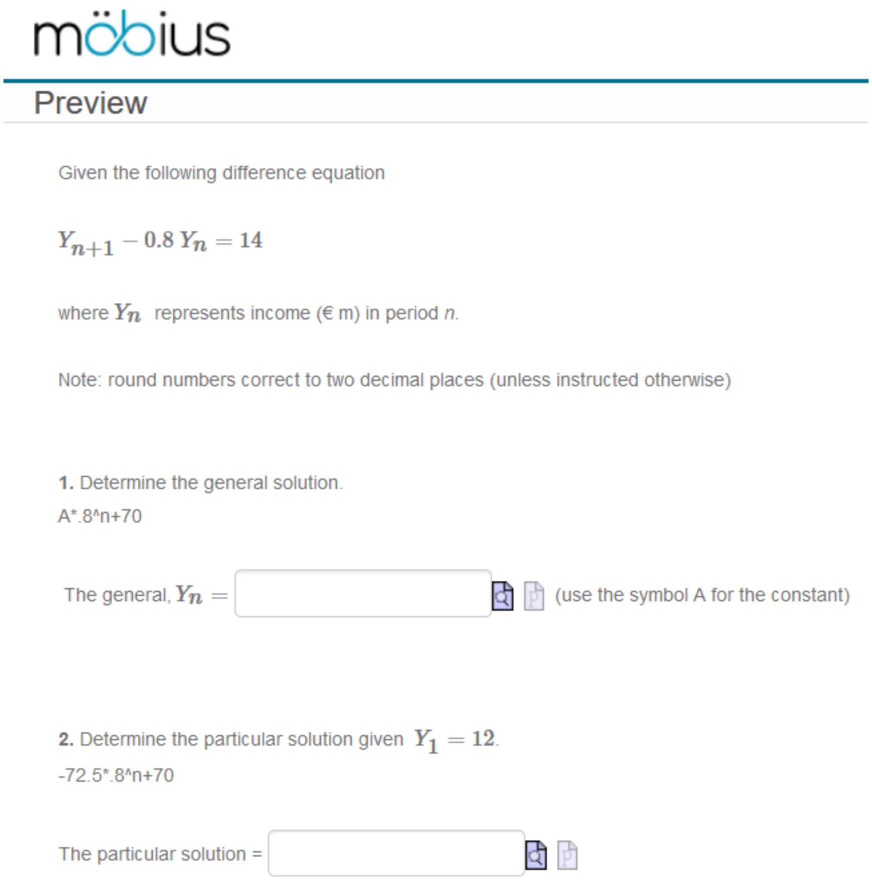
Another *Grading Code* for a **linear function**:

```
evalb(abs(coeff($RESPONSE,x)-coeff($p,x))<0.1) and

evalb(abs(($RESPONSE-coeff($RESPONSE,x)*x)-($p-coeff($p,x)*x))<0.1);
```

Working with decimal numbers

Another example of the grading of an expression whose coefficients may be decimal numbers. It is then a bit difficult to subtract the expressions from each other and set them equal to 0 if you also want to allow some margins in the coefficients.



Preview

Given the following difference equation

$Y_{n+1} - 0.8\,Y_n = 14$

where Y_n represents income (€ m) in period *n*.

Note: round numbers correct to two decimal places (unless instructed otherwise)

1. Determine the general solution.

A*.8^n+70

The general, $Y_n =$ (use the symbol A for the constant)

2. Determine the particular solution given $Y_1 = 12$.

-72.5*.8^n+70

The particular solution =

Figure 1.88: Expressions with decimal coefficients

The *Algorithm* is the following:

```
$a=decimal(1,range(0.5,0.9,0.1));
$b=range(10,20,2);
$c=range(2,20);
$d=range(12,20);
$displayeq1=maple("MathML[ExportPresentation](Y[n+1]-$a*Y[n]=
$b)");
$r=$a;
$YPi=maple("$b/(1-$a)");
$YPI=decimal(2,$YPi);
$Ygeneral=maple("A*($r)^n+$YPI");
$Aa=maple("($d-$YPi)/$a");
$A=decimal(2,$Aa);
$Ypart=maple("$A*($r)^n+$YPI");
$displaygeneral=maple("MathML[ExportPresentation](Y[n]=
$Ygeneral)");
$displayparticular=maple("MathML[ExportPresentation](Y[n]=
$Ypart)");
$y3=maple("round(eval($Ypart,n=3))");
$y20=maple("round(eval($Ypart,n=20))");
$display3=maple("MathML[ExportPresentation](Y[3]=$y3)");
$display20=maple("MathML[ExportPresentation](Y[20]=$y20)");
```

The correct answer of the **first answer field** is A*0.8^n+70. The *Grading Code* for that is:

```
evalb(coeff($RESPONSE,$r^n)=A) and evalb(abs(op(sort($RESPONSE))
[2]-($YPi))<0.02);
```

It means that the coefficient of 0.8^n of the student's answer must be equal to A. And then the difference between the second operand of the student answer and the constant $YPi must be less than 0.02.
An alternative *Grading Code* is using substitution:

```
evalb(abs(subs({A=1,n=1},$RESPONSE)-subs({A=1,n=1},
$Ygeneral))<0.01);
```

It means: Substitute A=1 and n=1 in the student's answer and in the correct answer and calculate the absolute value of the difference. That must then be smaller than, for example, 0.01.

In the **second answer field** of this question, the answer must be the following: -72.5*0.8^n+70. The right answer is $Ypart. Here the number $Aa = -72.5 may be rounded to 2 decimal places and the number $YPi = 70 may also be rounded to two decimal places (happens to come out of this as an integer).

For this answer field of question type *Maple graded* with setting *Text entry only* you can use the following *Grading Code*:

```
evalb(abs(coeff($RESPONSE,$r^n)-($Aa))<0.02) and
 evalb(abs(op(2,sort($RESPONSE))-($YPi))<0.02);
```

It means that the coefficient of $r^n (here it is 0.8^n) may deviate less than 0.02 from the correct coefficient $Aa (here it is -72.5). It goes without saying that power must have the base 0.8 ($r); there is no margin in that. Furthermore, the second operand of the expression, after it has been sorted in order, (here it is the number 70) must fall within a margin of 0.02 with the correct number $YPi.

Since it involves rounding the coefficients of a two-term to the second decimal place, you might as well have programmed with the following *Grading Code* to match the correct answer $Ypart:

```
evalb(nops($RESPONSE)=2) and evalb(map(x->round(100*x),
{coeffs($Ypart)})= map(x->round(100*x),{coeffs($RESPONSE)}));
```

Here, the number of operands is first checked (which has to do with the form in which the answer will be given).
This can possibly be done a bit more robust with
`evalb(nops($RESPONSE)=nops($Ypart))`.
Then all coefficients are multiplied by 100 and rounded and matched with each other using a set.

Avery simple **alternative Grading Code is with substitution**. Then take a handy number.

```
evalb(abs(subs(n=1,$RESPONSE)-subs(n=1,$Ypart))<0.01);
```

1.4.12 Fractions

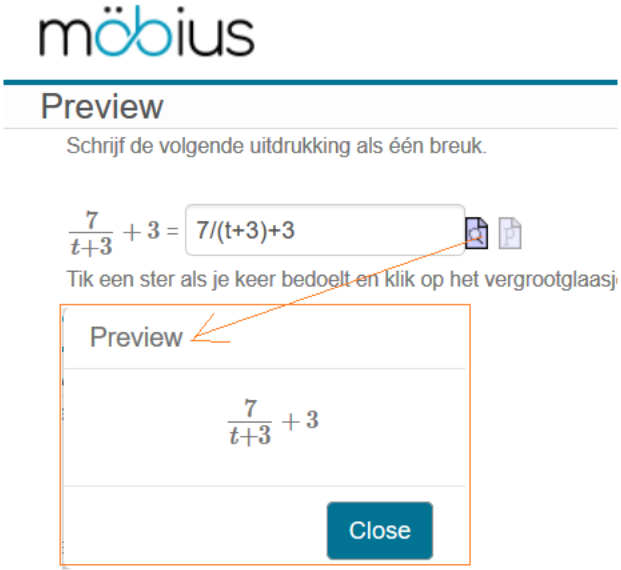
First look in section *When are formulas equal? (page 48)* where it is explained that fractions are noted differently, are not always seen as the same by Maple.

HINT: Always pay attention to whether it is possible that the student can also simply type the formula of the question to get a full grading.
Or maybe the student clicks on *Preview* and is then shown a simplified form with which the answer is in fact given away, see *Figure 1.75 (page 53)*. With the setting *Maple syntax Text entry only* the *Preview* often the simplification is showed.
However, there are plenty of tricks to get it exactly the way you want it.

Look also in section *Working with StringTools (page 51)* to enforce a certain spelling in the student's answer where it is important to know which settings you use for filling in the formula.

Below are the different options at the **different settings** (*Figure 1.81 (page 61)*).

1) With the setting **Maple syntax with Text entry only** you should be aware that the *Preview* function often simplifies the entered formula. In addition, the student can also use Maple commands at this setting. Working with *StringTools* needs extra attention here.The advantage of this setting is that the string of the response exactly takes over the input of the student. For example, you can check if the student enters one fraction by counting the number of slashes in the answer if necessary, or you can check that the student is not using Maple commands or something similar.
In the example below, we see that the *Preview* does not show the intended simplification, so we can safely work with this setting.



Preview

Schrijf de volgende uitdrukking als één breuk.

$\frac{7}{t+3}+3=$ 7/(t+3)+3

Tik een ster als je keer bedoelt en klik op het vergrootglaasj

Preview

$\frac{7}{t+3}+3$

Figure 1.89: Fraction with setting Text entry only

The *Grading Code* is:

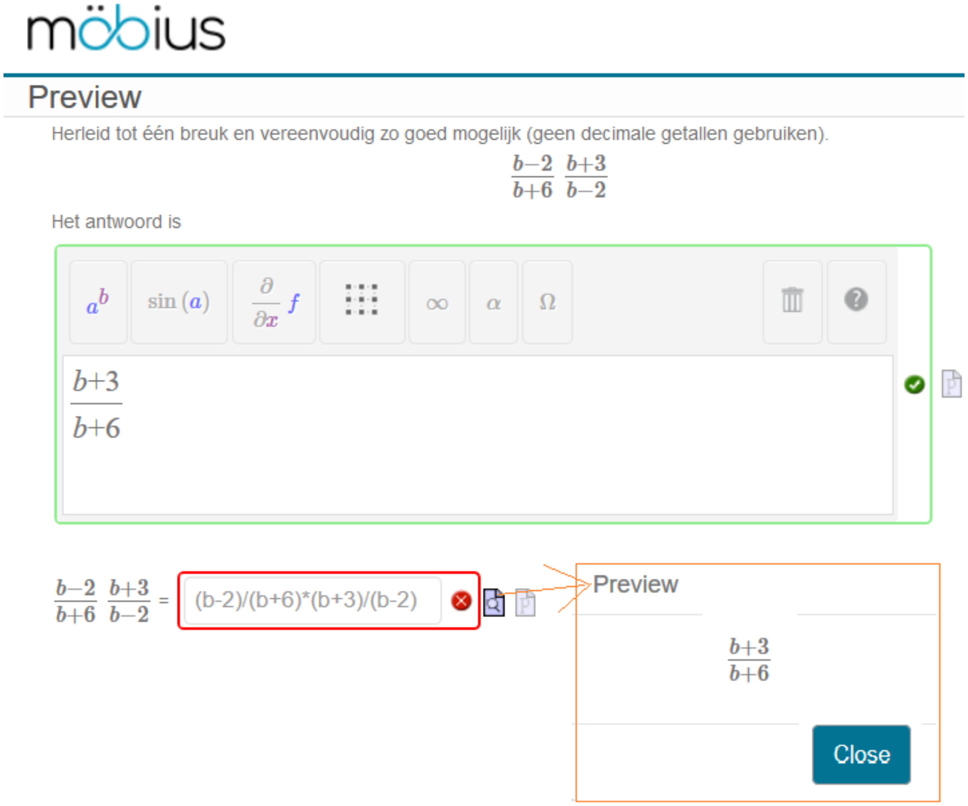
```
evalb(simplify($RESPONSE-($antwoord))=0) and
 evalb(type($RESPONSE,`*`) ) and
evalb(StringTools[Search]("simplify","$RESPONSE")=0) and
evalb(StringTools[Search]("normal","$RESPONSE")=0);
```

In this case, `evalb(simplify($RESPONSE-($antw))=0` was chosen; it is robust, but in this case it has also been ensured that the answer entered by the student is definitely in the form of one fraction by means of ``evalb(type($RESPONSE,`*`))``. In the *Grading Code* you see that the Maple commands `normal` and `simplify` are not allowed in the students answer. the correct answer is already programmed in the *Algorithm* with:
`$vraag=maple("$a/($y+$c)+$b");` and `$antw=maple("simplify($vraag)");`
The students answer as typed in the figure *Figure 1.89 (page 72)* will be graded incorrect.
HINT: Again, make sure that the automatic simplification does not take place in the *Preview* (zie *(page 58)*)

HINT: This *Grading Code* does not provide that the student in all cases does the simplification by skipping an equal factor in numer and denominator of the fraction.

2) With the setting **Maple syntax with Symbol entry only** it is important that you know how the student's input is translated into a string before you know what programming with *StringTools* is needed in the *Grading Code*. Try out something as described in section *The analysis of the Response (page 61)*.

The example below shows the difference between the options S*ymbol entry only* and *Text entry only*. With the latter makes it possible to see the automatic simplification with *Prieview*! With the *Equation Editor* of the setting *Symbol entry only* the student must fill in the simplified form.



Preview

Herleid tot één breuk en vereenvoudig zo goed mogelijk (geen decimale getallen gebruiken).

$$\frac{b-2}{b+6}\,\frac{b+3}{b-2}$$

Het antwoord is

$$\frac{b+3}{b+6}$$

$\frac{b-2}{b+6}\,\frac{b+3}{b-2} =$ (b-2)/(b+6)*(b+3)/(b-2)

Preview

$$\frac{b+3}{b+6}$$

Figure 1.90: Fractions with setting Symbol entry only

The programming in the *Grading Code* is as follows:

```
is(($antwoord)-($RESPONSE)=0) and
 evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","(")=2)
 and evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","/")<=1);
```

As you can see in section *The analysis of the Response (page 61)*, when translating to a string of the answer, entered in the *Equation Editor*, brackets around numer and denominator will always appear. In addition, in the *Grading Code* above, we enforce that only one slash (or less) is noted. However, be careful with this, there are other ways to check whether the student gives the answer in the right form.

HINT: If you prefer to offer the text entry instead of the editor, you can stop the automatic simplification with a *Custom Preview Code*. See section *(page 58)*.

HINT: To make the answer field for the editor a little less high, give the following code in the *Custom CSS* of the question: (mind the dot in front of this code.)

```
.mwEquationEditor {height:100px !important}
```

TIP: In the *Answer* section of such an answer field with editor, always give the MathML encoded formula as the correct answer!

```
printf(MathML[ExportPresentation]($antwoord));
```

The student will then always see the two-dimensional answer in the feedback. After all, he also enters the answer two-dimensionally in the Editor.

3) With the setting **Formula** it must be formulas that are not too complex and that certainly do not contain combined characters and no exponential functions. There are many reasons **not to use this setting**!

In the following example (*Figure 1.92 (page 74)*) is chosen for this setting because otherwise in the setting *Maple syntax* with *Text entry only* the students answer will be automaticly simplified in the *Preview.*
The setting *Maple syntax Symbol entry only* is pretty difficult to grade with the requirement of only one slash for the fraction. The editor translates the string as follows:

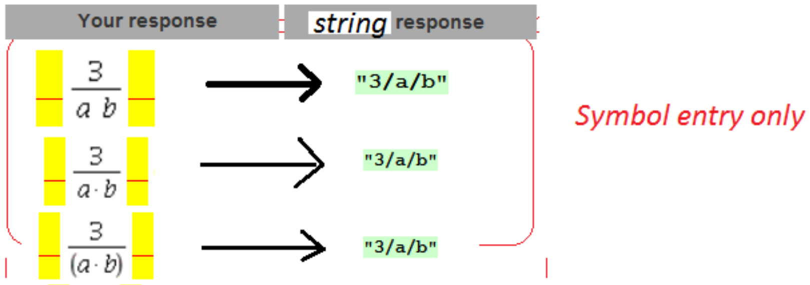
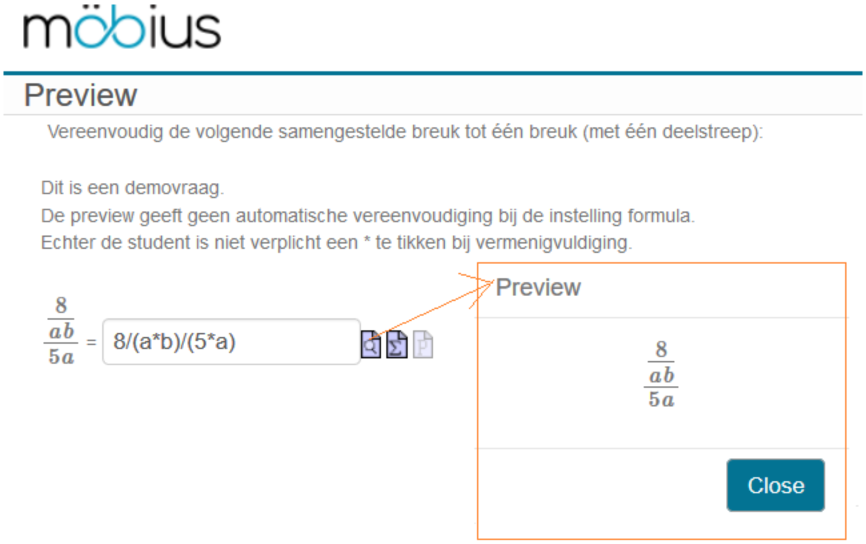

Your response
string response
$\frac{3}{a\ b}$ → "3/a/b"
$\frac{3}{a\cdot b}$ → "3/a/b"
$\frac{3}{(a\cdot b)}$ → "3/a/b"
Symbol entry only

Figure 1.91: Fractions with the Editor

In the figure below you can see that the third option to grade formulas with the *Maple graded question type* with the *Formula setting* has now been chosen. Again: only use this setting if there is no other option!



Preview

Vereenvoudig de volgende samengestelde breuk tot één breuk (met één deelstreep):

Dit is een demovraag.
De preview geeft geen automatische vereenvoudiging bij de instelling formula.
Echter de student is niet verplicht een * te tikken bij vermenigvuldiging.

$\frac{\frac{8}{ab}}{5a}$ = 8/(a*b)/(5*a)

Figure 1.92: Entering fractions as text with setting Formula

In the figure above with setting *Formula* of the *Maple graded question type* you can see that the *Preview* gives no automatic simplification.

The *Algorithm* of this example is as follows:

```
$a=range(2,8);
$b=range(2,8);
$ggd=maple("igcd($a,$b)");
condition:eq($ggd,1);
$indexL=rint(3);
$L1=switch($indexL,"a","x","p");
$L2=switch($indexL,"b","y","q");
$vraag=maple("$a/($L1*$L2)/($b*$L1)");
$vraagdisplay=mathml("$a/($L1*$L2)/($b*$L1)");
$antwoord=maple("simplify($vraag)");
$antwoorddisplay=maple("printf(MathML[ExportPresentation]($antwoord))");
```

It means the two numbers $a en $b have no common divisors (igcd =greatest common divisor of integers). So there is no possible simplification of the fraction with these numbers. The point is that the student properly lays the translation to the compound fraction and has to make one fraction of it.
The =-sign behind the fraction shows what is the main fraction bar.

The variables $L1 (in this case *a* in the compound fraction) must be taken together like $\frac{8}{5\,a^2 b}$.

In the *Grading Code* you can program the following:

```
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","/")<=1)
 and evalb(simplify(($antwoord)-($RESPONSE))=0) and
 evalb(StringTools[CountCharacterOccurrences]("$RESPONSE",
 "$L1")=1);
```

It means that only once the variable $L1 is present in the students answer to get the full grading.

So the *Grading Code* with the use of *StringTools* is also useful in the setting *Formula* of the *Maple graded question type*, but you have to pay extra attention because of the brackets.
HINT: Note that this variable consists of only one character. In case of character combinations you use the command `SearchAll` as follows:

```
evalb(nops({StringTools[SearchAll]("$x","$RESPONSE")})=1);
```

(See section *Working with StringTools (page 51)*)
It would have been difficult to demand only one slash if we can't be sure how the formula is translated to the string, as in the *Symbol entry only* setting and as seen in section *Figure 1.91 (page 74)*. You can optionally try out for yourself how the input is translated into a string in the above situation at the setting *Formula Figure 1.92 (page 74)*.
With the setting *Formula* for example the expression 8/(5a^2*b) is translated to a string as "(8/((5*(a^2))*b))".
It is therefore difficult to check the number of brackets and the number of asterixes because the program automatically adds them if the student does not type them himself.
It is also difficult in case the student switches to the editor that the number of slashes, asterixes and the like in the string of the answer no longer matches what the student enters.

CONCLUSION: With this *Formula* setting it is therefore very difficult to check for brackets and slashes with the help of *StringTools*!!
And certainly the number of multiplication characters (asterix) is difficult to check because the system simply puts extra asterixes in where the student does not type them.
Finally the conclusion is that this setting is **not recommanded**.

HINT: In the *Answer* section use 2D formula with

```
printf(MathML[ExportPresentation]($antwoord));
```

4) Working with StringTools to check the answer and **working with Inert Form** to stop the automatic simplification in the *Preview*.

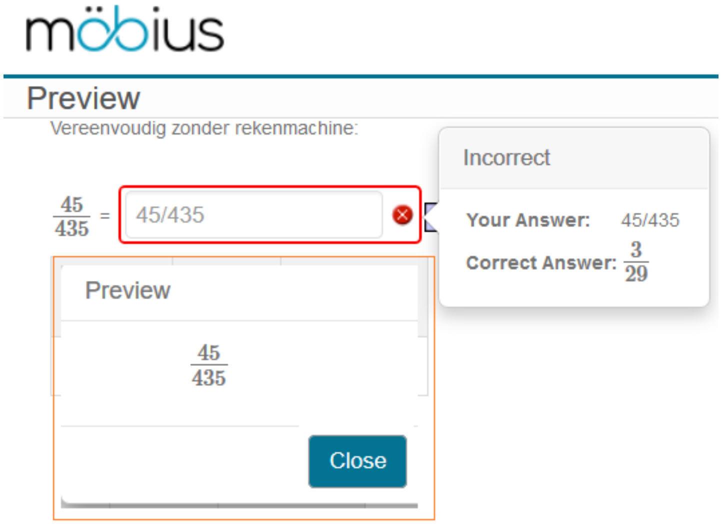


Figure 1.93: Simplifying fractions

In the *Algorithm* you program the following:

```
$A=range(1,9);
$a=5;
$B=range(1,6);
#ithprime is het zoveelste priemgetal vanaf 2.
$b=maple("ithprime($B)");
$C=range(7,12);
$c=maple("ithprime($C)");
$D=range(1,9);
$d=3;
$abd=$a*$b*$d;
$acd=$a*$c*$d;
$vraag=mathml("($abd)/($acd)","nosimplify");
$antwoord=maple("simplify($abd/$acd)");
$stringantwoord=maple("convert($antwoord,string)");
$antwoorddisplay=maple("printf(MathML:-ExportPresentation(($antwoord)))");
```

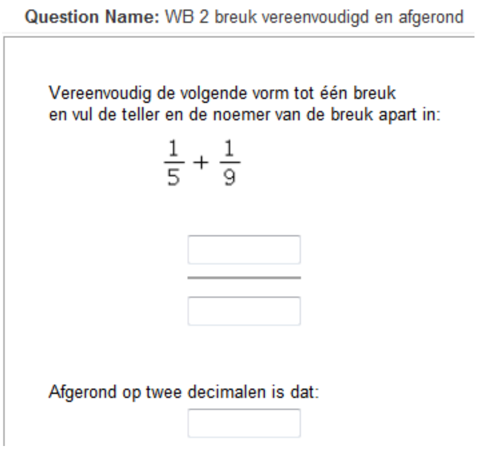
The *Grading Code* of this answer field:

```
stringrespons:=StringTools[Remove](IsSpace, "$RESPONSE"):
evalb(stringrespons=$stringantwoord);
```

In the *Custom Preview* you program the following:

```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

5) Always keep in hand that you can divide the answer into **two answer fields** (or more) as the following example shows.



Question Name: WB 2 breuk vereenvoudigd en afgerond

Vereenvoudig de volgende vorm tot één breuk
en vul de teller en de noemer van de breuk apart in:

$$\frac{1}{5}+\frac{1}{9}$$

Afgerond op twee decimalen is dat:

Figure 1.94: Fractions with two answer fields

In this example, numerator and denominator of the result must be filled in independently of each other. There is then only one answer possible per answer field, namely a number and you can simply check that with a numeric answer field. The student must then have simplified the result.

The fraction bar you can make with a small table with three rows and one column and border 0. In the second row you insert a Horizontal Line with one of the buttons. In the other rows you can put the numerical answer fields.
The two separate answers can easily be prepared in the *Algorithm*:

```
$a1=range(2,15);
$b=range(2,20);
$a=if(ne(($a1),($b)),($a1),($a1)+1);
#it means that if a1 not equal to b then a1 else a1+1.
$vraagdisplay=mathml("1/($a)+1/($b)");
$antwoord=maple("simplify(1/($a)+1/($b))");
$antwnumeriek=maple("evalf($antwoord)");
$AntwNum=decimal(2,$antwnumeriek);
$teller=maple("numer($antwoord)");
$noemer=maple("denom($antwoord)");
$antwoorddisplay=maple("printf(MathML[ExportPresentation](($antwoord)))");
$ggd=maple("igcd(($a),($b))");
#Use the greatest common divisor to see if the denominators have common factors.
$commentaar1="Make the same denominators of the fractions. In this case multiply
 the denominators.";
$commentaar2="Make the same denominators of the fractions. Mind the denominators
 have some factors in common: $gcd. Possibly simplify!";
$commentaar=if(eq($ggd,1),"$commentaar1","$commentaar2");
```

In the *Algorithm* above you can see that the question itself was created with the `mathml("...")` command because Maple would immediately simplify the form anyway. With the Maple-commands **numer** and **denom** it is easy to calculate the separate answers for the answer fields.

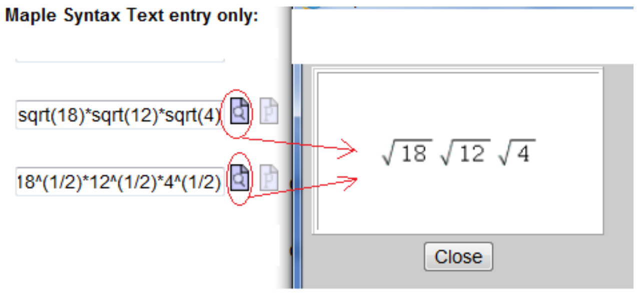
HINT: Instructions can also be included in the script of the *Algorithm*. You do that by placing a # at the front of the line.

HINT: Nice to see here is the prepared comment that can be called in the form of a variable to use in the *Feedback* section. Depending on the situation, the comments are different.

1.4.13 Square roots

It is very difficult to grade square root forms where simplification is requested.

You can go a long way with *StringTools* by checking the string of the student's response for whether or not certain characters or combinations thereof occur. See also section *Working with Stringtools (page 51)*.
But first we take a closer look at the student's input at the setting *Maple Syntax Text entry only*.



Maple Syntax Text entry only:

sqrt(18)*sqrt(12)*sqrt(4)

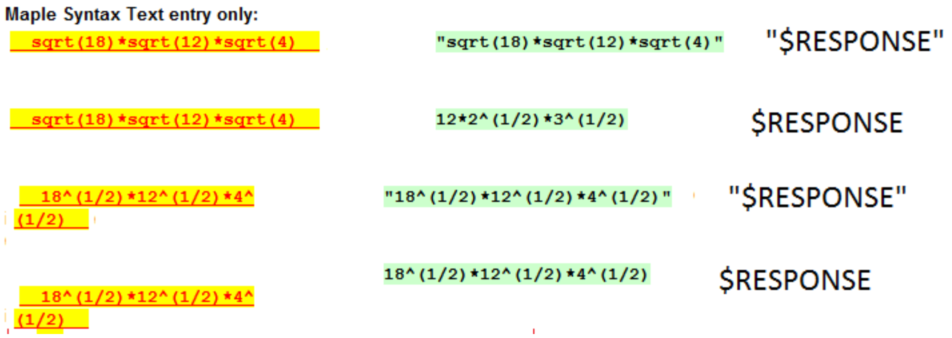
18^(1/2)*12^(1/2)*4^(1/2)

$\sqrt{18}\ \sqrt{12}\ \sqrt{4}$

Figure 1.95: How to simplify square roots

In the figure above we see that the *Preview* function in both cases (entered with sqrt or with ^(1/2)) yields the same: it does NOT simplify the size of the numbers under the root sign! (However, be careful with fractures.) Square roots are also not taken together and therefore not simplified in the *Preview* unless they are two of the same square roots. So you can quietly use the *Maple Syntax setting* with *Text entry only* for these types of questions.

With the trick explained in the section *The analysis of the Response (page 61)* we are now going to see what it yields if we examine the string of the answer and also the answer itself.



Maple Syntax Text entry only:

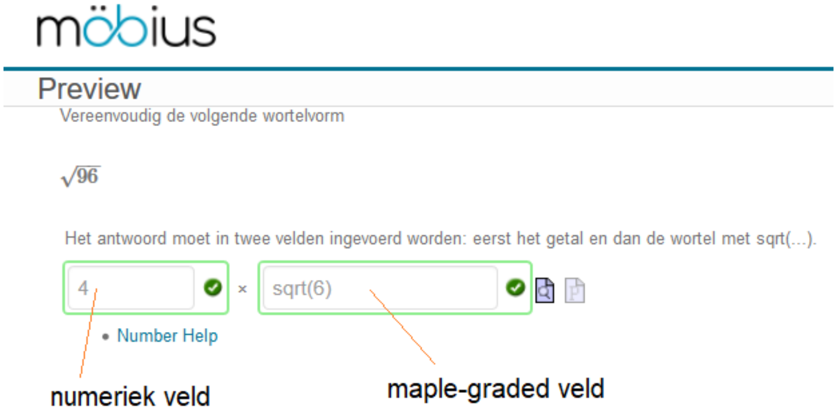
sqrt(18)*sqrt(12)*sqrt(4)	"sqrt(18)*sqrt(12)*sqrt(4)"	"$RESPONSE"
sqrt(18)*sqrt(12)*sqrt(4)	12*2^(1/2)*3^(1/2)	$RESPONSE
18^(1/2)*12^(1/2)*4^(1/2)	"18^(1/2)*12^(1/2)*4^(1/2)"	"$RESPONSE"
18^(1/2)*12^(1/2)*4^(1/2)	18^(1/2)*12^(1/2)*4^(1/2)	$RESPONSE

Figure 1.96: The response and the stringresponse in different ways of entering square roots

In the figure above you can see in the first line that the notation of the square roots, entered by the student with "sqrt", is exactly copied into the string.
In the second line in the student's response, the roots are converted to power (1/2) and at the same time simplified, but not taken together.
In the third line: enters the roots using ^(1/2), then the string of the input is taken over exactly as expected at this setting of *Text entry only*.
In the fourth line, you can see that strangely enough, the response is not simplified by making the numbers below the square root sign smaller. You should note that the student could use both notations.

In the following examples we will look at the ways in which the *Grading Code* can be arranged, always with the settings of *Text entry only*.
The last resort is to check that the student is typing exactly the answer you desire. There are always two possibilities when typing the square root, namely with sqrt and with a power with exponent (1/2).

Simplify the square root



Preview

Vereenvoudig de volgende wortelvorm

$\sqrt{96}$

Het antwoord moet in twee velden ingevoerd worden: eerst het getal en dan de wortel met sqrt(...).

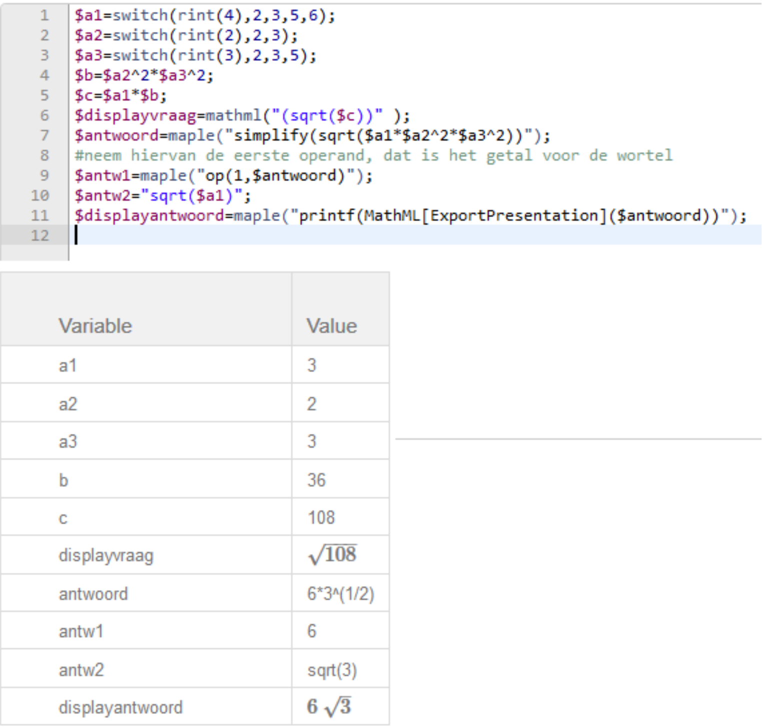
4 × sqrt(6)

- Number Help

Figure 1.97: Simplify Square Root

In the figure above you can see a question where the simplification of a square root is tested. You offer two fields: The first is a number and the second is a square root.
The *Preview* with the setting *Maple syntax Text entry only* shows two dimensional what the student has typed in.
In the *Feedback* you want the student to see sqrt(6).

In the following figure you can see in the corresponding *Algorithm* how the possible answers are prepared.



```
$a1=switch(rint(4),2,3,5,6);
$a2=switch(rint(2),2,3);
$a3=switch(rint(3),2,3,5);
$b=$a2^2*$a3^2;
$c=$a1*$b;
$displayvraag=mathml("(sqrt($c))" );
$antwoord=maple("simplify(sqrt($a1*$a2^2*$a3^2))");
#neem hiervan de eerste operand, dat is het getal voor de wortel
$antw1=maple("op(1,$antwoord)");
$antw2="sqrt($a1)";
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
```

Variable	Value
a1	3
a2	2
a3	3
b	36
c	108
displayvraag	$\sqrt{108}$
antwoord	6*3^(1/2)
antw1	6
antw2	sqrt(3)
displayantwoord	$6\ \sqrt{3}$

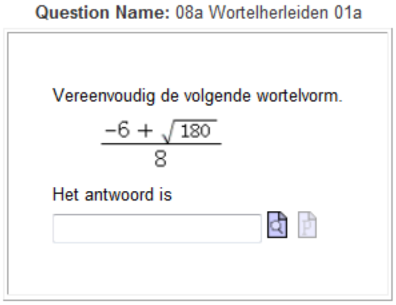
Figure 1.98: Symplify square roots in the Algorithm

In the figure above you can see the *Algorithm* where the answers are prepared and checked.

Note that with $antw2, you do not give a maple command to get the sqrt(3).
Note that the question $vraag is prepared using the mathml command, because with the MathML conversion with Maple, the automatic simplification would be unstopable here.
With this *Algorithm*, the feedback is also quickly prepared:
Decompose the number $c into prime factors $a1*$a2^2*$a3^2 and take the square root. The answer is $displayantwoord.
The correct answer for the first (numeric) answer field is $antw1.
For the correct answer of the second answer field (Maple graded) you fill in the section *Answer*: `printf("$antw2");`
For the *Grading Code* you give: `evalb($antw2=$RESPONSE);`
If the student gives the answer using 3^(1/2) it is also graded correctly.

Square roots with fractions

In this example, the way of entering for the student is a bit more flexible. Here again the setting to *Text entry only* and the command to simplify the shape. The point here is that at least the number under the square root sign must be made smaller for the rest it is not important.



Question Name: 08a Wortelherleiden 01a

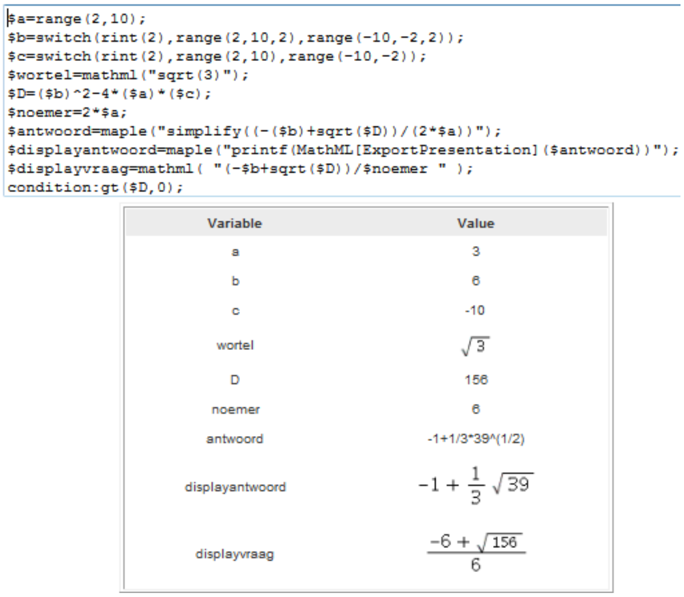
Vereenvoudig de volgende wortelvorm.

$$\frac{-6+\sqrt{180}}{8}$$

Het antwoord is

Figure 1.99: Simplify square root in the answer field

In the corresponding *Algorithm*, it is ensured that the number under the square root sign can always be made smaller (by $b an even number, this is always possible).



```
$a=range(2,10);
$b=switch(rint(2),range(2,10,2),range(-10,-2,2));
$c=switch(rint(2),range(2,10),range(-10,-2));
$wortel=mathml("sqrt(3)");
$D=($b)^2-4*($a)*($c);
$noemer=2*$a;
$antwoord=maple("simplify((-($b)+sqrt($D))/(2*$a))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$displayvraag=mathml( "(-$b+sqrt($D))/$noemer " );
condition:gt($D,0);
```

Variable	Value
a	3
b	6
c	-10
wortel	$\sqrt{3}$
D	156
noemer	6
antwoord	-1+1/3*39^(1/2)
displayantwoord	$-1+\frac{1}{3}\sqrt{39}$
displayvraag	$\frac{-6+\sqrt{156}}{6}$

Figure 1.100: Algorithm of the square roots

In the *Algorithm* above, $D is the form that will be placed under the square root sign. (Care has been taken to ensure that this $D greater than 0.)
Also pay attention to how the formula of the question is made with the mathml("...") command because with Maple the simplification cannot be stopped.

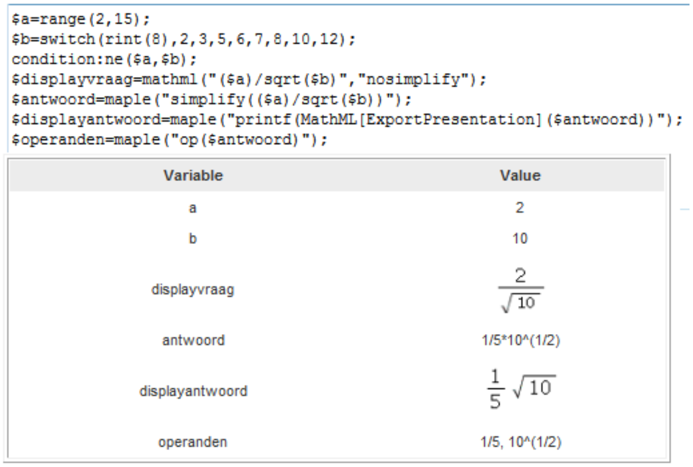
If you now always want to see the square root shape simplified, but otherwise do not have any requirements, you can, for example, program that in the example above the number $D should not appear.
In the *Grading Code* you program the following:

```
is(simplify(($antwoord)-($RESPONSE))=0) and
 evalb(StringTools[Search]("$D","$RESPONSE")=0);
```

The operands of an expression with square root

You can also check the operands of the correct answer and the student's answer separately.

In the *Algorithm* you can try out what the operands of an expression are.



```
$a=range(2,15);
$b=switch(rint(8),2,3,5,6,7,8,10,12);
condition:ne($a,$b);
$displayvraag=mathml("($a)/sqrt($b)","nosimplify");
$antwoord=maple("simplify(($a)/sqrt($b))");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$operanden=maple("op($antwoord)");
```

Variable	Value
a	2
b	10
displayvraag	$\frac{2}{\sqrt{10}}$
antwoord	1/5*10^(1/2)
displayantwoord	$\frac{1}{5}\sqrt{10}$
operanden	1/5, 10^(1/2)

Figure 1.101: The operands of an expression with sqare root

In this case you are somewhat limited and of course you have to make sure that the shape does indeed consist of two operands: in the sense that there is always a number or fraction in front of the square root followed by the square root. That is certainly the case if $a is not equal to $b.
If $a is equal to $b the answer only consists of a root and that is not the intention here, so you will need an extra condition in the *Algorithm*.
Mind that a single root consists of two operands (10^(1/2)).

The following lines show a number of Maple commands to try out in Maple itself.

```
> op(sqrt(10));
```

$$10, \frac{1}{2} \qquad (1.57)$$

```
> op(2*sqrt(10));
```

$$2, \sqrt{10} \qquad (1.58)$$

```
> op(2*sqrt(6)*7*sqrt(50));
```

$$70, \sqrt{6}, \sqrt{2} \qquad (1.59)$$

```
> op(3/sqrt(2));
```

$$\frac{3}{2}, \sqrt{2} \tag{1.60}$$

```
> op([2,2],3/sqrt(2));
```

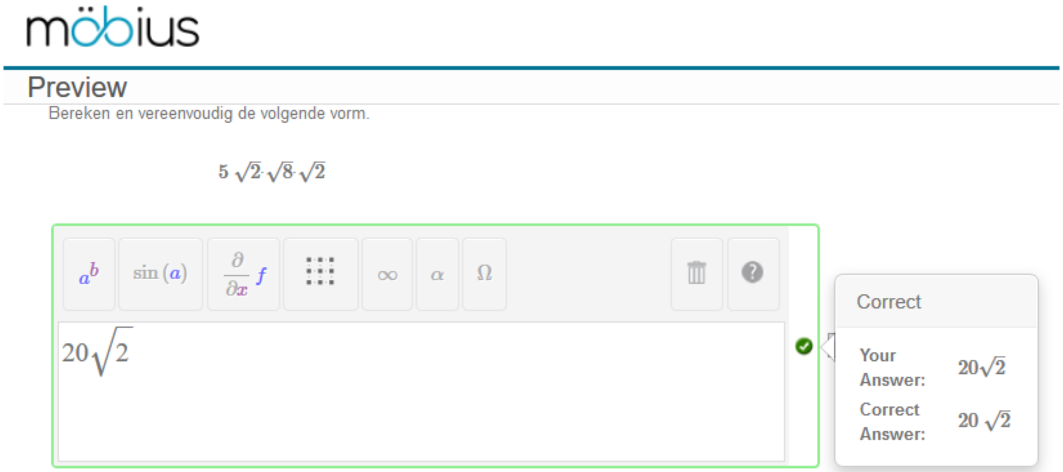
$$\frac{1}{2} \tag{1.61}$$

In the maple commands above it also becomes clear that the numbers are always automatically taken together, but the roots are not (unless there are two equal roots, which are taken together). In the last line, a list is asked for the second operand and from that the second operand is taken. So the square root sqrt(2) consists of two operands because it is in fact 2^(1/2). The first operand of these is 2 and the second operand is 1/2.

Simplify square roots by checking operands

If the student's response contains more roots, then these are all seen as operands. The roots that the student would type separately are never automatically taken together into one root in the $RESPONSE. Then make use of it to check the operands of the students response one by one.

Another example with testing the number of operands that an answer should consist of in a Maple graded question where you still decide whether to offer an editor or whether to offer a text field. In the figure below, the editor is offered for the student.



Preview

Bereken en vereenvoudig de volgende vorm.

$5\sqrt{2}\cdot\sqrt{8}\cdot\sqrt{2}$

$20\sqrt{2}$

Correct

Your Answer: $20\sqrt{2}$

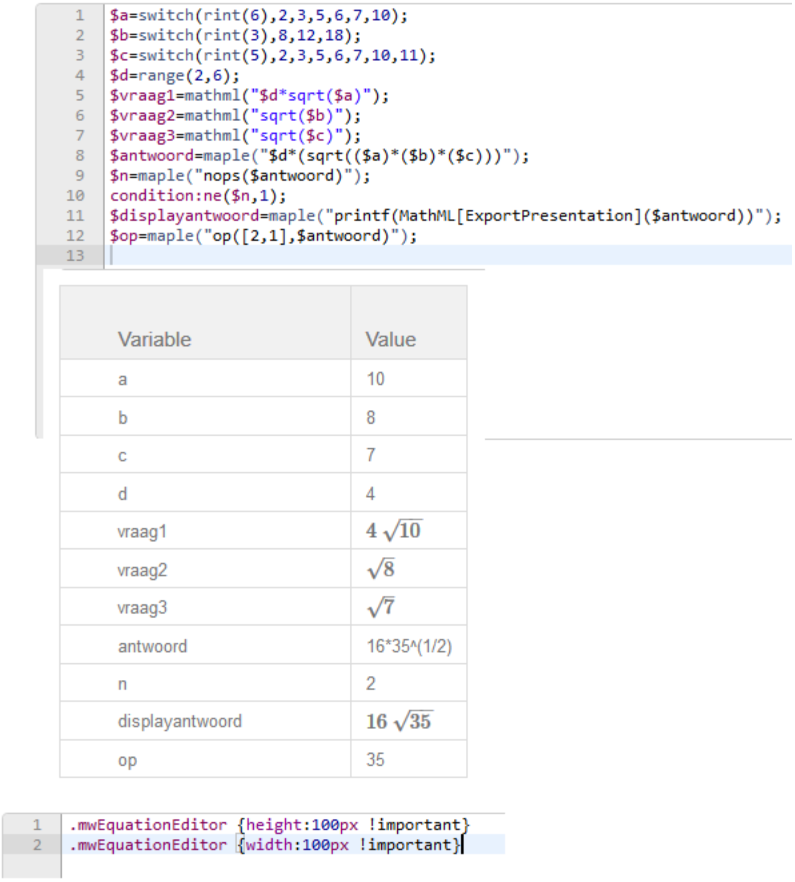
Correct Answer: $20\sqrt{2}$

Figure 1.102: Simplify Square Roots

With this question, it is difficult to ensure that the square roots do not cancel each other out and that at least a square root comes into the answer.
In the *Algorithm* you can program this as follows:

```
$a=switch(rint(6),2,3,5,6,7,10);
$b=switch(rint(3),8,12,18);
$c=switch(rint(5),2,3,5,6,7,10,11);
$d=range(2,6);
$vraag1=mathml("$d*sqrt($a)");
$vraag2=mathml("sqrt($b)");
$vraag3=mathml("sqrt($c)");
$antwoord=maple("$d*(sqrt(($a)*($b)*($c))");
$n=maple("nops($antwoord)");
```

```
#the number of the answers operands may not be equal to 1 because then the answer
 is a number.
condition:ne($n,1);
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$op=maple("op([2,1],$antwoord)");
```



```
$a=switch(rint(6),2,3,5,6,7,10);
$b=switch(rint(3),8,12,18);
$c=switch(rint(5),2,3,5,6,7,10,11);
$d=range(2,6);
$vraag1=mathml("$d*sqrt($a)");
$vraag2=mathml("sqrt($b)");
$vraag3=mathml("sqrt($c)");
$antwoord=maple("$d*(sqrt(($a)*($b)*($c)))");
$n=maple("nops($antwoord)");
condition:ne($n,1);
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$op=maple("op([2,1],$antwoord)");
```

Variable	Value
a	10
b	8
c	7
d	4
vraag1	$4\sqrt{10}$
vraag2	$\sqrt{8}$
vraag3	$\sqrt{7}$
antwoord	16*35^(1/2)
n	2
displayantwoord	$16\sqrt{35}$
op	35

```
.mwEquationEditor {height:100px !important}
.mwEquationEditor {width:100px !important}
```

Figure 1.103: Operands of the square root

In the *Algorithm* you can see that the parts of the question are now prepared with the command mathml("...") to prevent, for example, that the square root of 18 is not automatically simplified. Furthermore, it is programmed that the number of operands of the answer is in any case not equal to 1 so that you can be sure that the roots do not extinguish each other.
With the variable $op it is possible to check the *Grading Code* again whether the square root of $op in the string answer occurs in the form of "sqrt($op)".
With the setting *Symbol entry only* ALWAYS the sqare root is translated in sqrt.

```
evalb(op(1,$antwoord)=op(1,$RESPONSE)) and is(op(2,
$antwoord)=op(2,$RESPONSE)) and evalb(StringTools[Search]
("sqrt($op)","$RESPONSE")>0);
```

HINT: With the setting *Text entry only* is is not so wise, because the student also can type the square root with ^(1/2) unless you want to force students to work with sqrt.

HINT: To make the answer field of the editor a little smaller, enter a code in *Custom CSS* of the question:

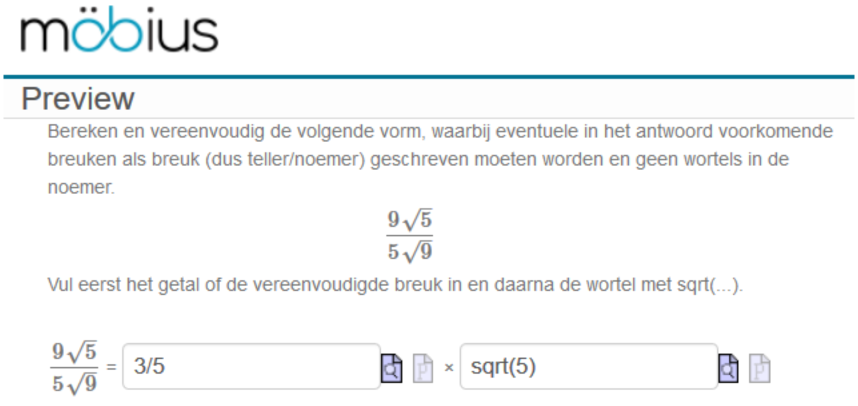
```
.mwEquationEditor {height:100px !important}
```

For the correct answer you put in the section *Answer*:

```
printf(MathML[ExportPresentation]($antwoord));
```

Square roots with StringTools

Another slightly more complicated simplification question with roots to program is the following:



Preview

Bereken en vereenvoudig de volgende vorm, waarbij eventuele in het antwoord voorkomende breuken als breuk (dus teller/noemer) geschreven moeten worden en geen wortels in de noemer.

$$\frac{9\sqrt{5}}{5\sqrt{9}}$$

Vul eerst het getal of de vereenvoudigde breuk in en daarna de wortel met sqrt(...).

$\frac{9\sqrt{5}}{5\sqrt{9}}$ = 3/5 × sqrt(5)

Figure 1.104: Simplify square roots with fractions

The student must first enter a simplified fraction and then a simplified square root in two separate answer fields

With the following *Algorithm* you prepare the two correct answers:

```
$a=range(2,10);
$b1=range(2,9);
$b=if(ne(($a),($b1)),($b1),($b1)+1);
$displayvraag=mathml("($b*sqrt($a))/($a*sqrt($b))" );
$antwoord=maple("($b/$a)*sqrt($a/$b)");
$displayantwoord=maple("printf(MathML[ExportPresentation]($antwoord))");
$n=maple("if type(op(2,$antwoord),`^`) then 0 else 1 end if");
condition:ne($n,1);
$antw1=maple("op(1,$antwoord)");
$stringantw1=maple("convert($antw1,string)");
$antw2=maple("op(2,$antwoord)");
```

To check if the second operand of the answer is a square root (i.e. of the type ^), test that with $n and make sure that $n is not equal to 1.
If you had sufficed just to demand the number of operands equal to 2, then it could also have been that the answer was a fraction, and that also consists of two operands: numerator and denominator.

In the first field, therefore, there must be a fraction that has been simplified. So you convert the first answer to a string and you compare the student's answer to this string with the following *Grading Code*:

```
stringrespons:=StringTools[Remove](IsSpace, "$RESPONSE");
evalb(StringTools[SubString](stringrespons, 1..-1)=$stringantw1);
```

Don't forget to enter in the *Custom Previewing Code* the following:

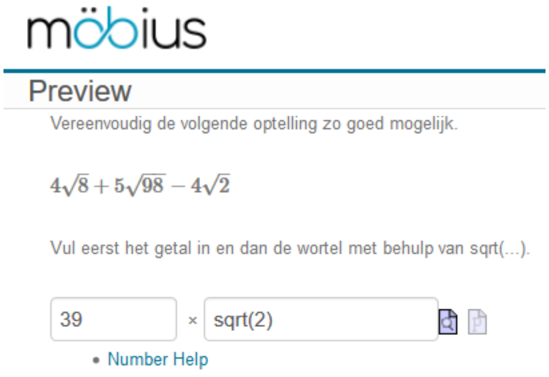
```
use InertForm:-NoSimpl in $RESPONSE: end: printf(InertForm:-
ToMathML(%));
```

In the second field should be a simplified root. This is easy to check with the *Grading Code*:

```
evalb(($antw2)-($RESPONSE)=0);
```

Square root with LaTeX

In the following example, the student gives a number in a numeric field and a square root in a maple graded field.



Preview

Vereenvoudig de volgende optelling zo goed mogelijk.

$4\sqrt{8} + 5\sqrt{98} - 4\sqrt{2}$

Vul eerst het getal in en dan de wortel met behulp van sqrt(...).

39 × sqrt(2)

- Number Help

Figure 1.105: Addition with square roots

With the following *Algorithm*:

```
$a=switch(rint(2),2,3);
$B=switch(rint(3),4,9,16);
$b=$a*$B;
$C=switch(rint(3),25,36,49);
$c=$a*$C;
$D=switch(rint(2),1,100,$a);
$d=$a*$D;
$p=range(2,5);
$q=range(2,5);
$r=range(2,5);
$pm1=switch(rint(2),"+","-");
$pm2=switch(rint(2),"+","-");
$vraagdisplay="\( $p\sqrt{$b} $pm1 $q \sqrt{$c} $pm2 $r \sqrt{$d}
 \)";
$antwoord=maple("simplify($p*sqrt($b) $pm1 $q*sqrt($c) $pm2
 $r*sqrt($d))");
$antw1=maple("op(1,$antwoord)");
$antw2=maple("op(2,$antwoord)");
```

This ensures that the answer always consists of two operands: an integer and a square root.

The first answer is for the numeric field and the second answer is for the maple graded field.
The $vraagdisplay question for the screen in the text is now created using LaTeX (see section about LaTeX *(page 25)*).

HINT: Finally, another possibility that can still be used in the case of working with square roots in a simplified form, is to deviate to a completely different question type namely the question type *Mathematical Formula* with special settings to grade simplified forms. See for more information in section *Formula without simplification (page 127)*.

1.4.14 Equations

When matching equations, it doesn't simply come down to subtracting the equations from each other and then demanding that zero come out.
If the student has to fill in an equation, he may simplify the equation or write it down differently.
The point is that he fills in an equation that is **equivalent** to the correct equation.

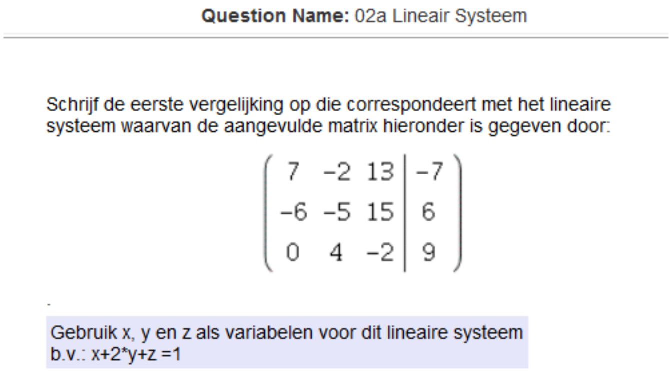
HINT: Use always the *Maple graded question type* with the setting *Maple syntax*.

HINT: Equations are more sensitive if you want to use decimal numbers:

The two numbers 4 and 4.0 are considered equal, but the equations x=4 and x=4.0 are not considered equal!!! In that case it is an idea to use the command `convert(vergelijking,rational)` to get rid of the decimal numbers.

Entering an equation

In the following example, the student has to fill in an equation in the variables x, y and z where it may also be that one or more of these variables is missing.



Question Name: 02a Lineair Systeem

Schrijf de eerste vergelijking op die correspondeert met het lineaire systeem waarvan de aangevulde matrix hieronder is gegeven door:

$$\left(\begin{array}{ccc|c} 7 & -2 & 13 & -7 \\ -6 & -5 & 15 & 6 \\ 0 & 4 & -2 & 9 \end{array}\right)$$

.

Gebruik x, y en z als variabelen voor dit lineaire systeem
b.v.: x+2*y+z =1

Figure 1.106: Linear system, equations

The correct answer is 7*x-2*y+13*z=-7.

IMPORTANT HINT: In order to check if **two equations are equivalent**: you will match the **solution of the students equation** with the **solution of the correct equation**. Then you are sure that these equations are equivalent.
It is easy if the solution of the equation consists of only one element.
If the solutions are the same, then the equations are equivalent, and that is what it is all about.
The programming of the *Grading Code* of the example above is the following:

```
evalb(solve($RESPONSE,x)=solve($antw,x)) or
 evalb(solve($RESPONSE,y)=solve($antw,y)) or
 evalb(solve($RESPONSE,z)=solve($antw,z)) and
 type($RESPONSE,equation);
```

In this case, expressions are matched. After all, if x is released from the equation with x and y and z, then you get an expression in the variables y and z.

HINT: In the above it is provided that perhaps the equation does not contain x or y or z. That is why three conditions have been given with `or` in between (so not `and`).

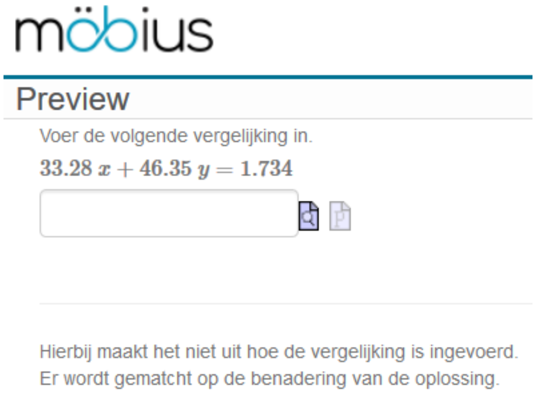
HINT: You may have seen that the *Grading Code* also has programmed: `and type($RESPONSE,equation);`
This was specially done because otherwise it is possible that, for example, the student reduces the equation to 0 and only fills in the left hand side. At Maple, the `solve` command not only works on equations but also on expressions on the assumption that they must then be set equal to 0.

HINT: It is wise to program in the *Custom Previewing Code* the following:

```
if evalb(StringTools[CountCharacterOccurrences]
("$RESPONSE","=")=0) then printf("Enter an equation (including =-
sign).") else printf(MathML[ExportPresentation]($RESPONSE)) end
 if;
```

Equations with decimal numbers

If you are working with decimals, round off the equation with some significant digits (with `evalf` for example). Then you can match the solutions of both equations.



Preview

Voer de volgende vergelijking in.

$33.28\,x + 46.35\,y = 1.734$

Hierbij maakt het niet uit hoe de vergelijking is ingevoerd.
Er wordt gematcht op de benadering van de oplossing.

Figure 1.107: Equations with decimal numbers

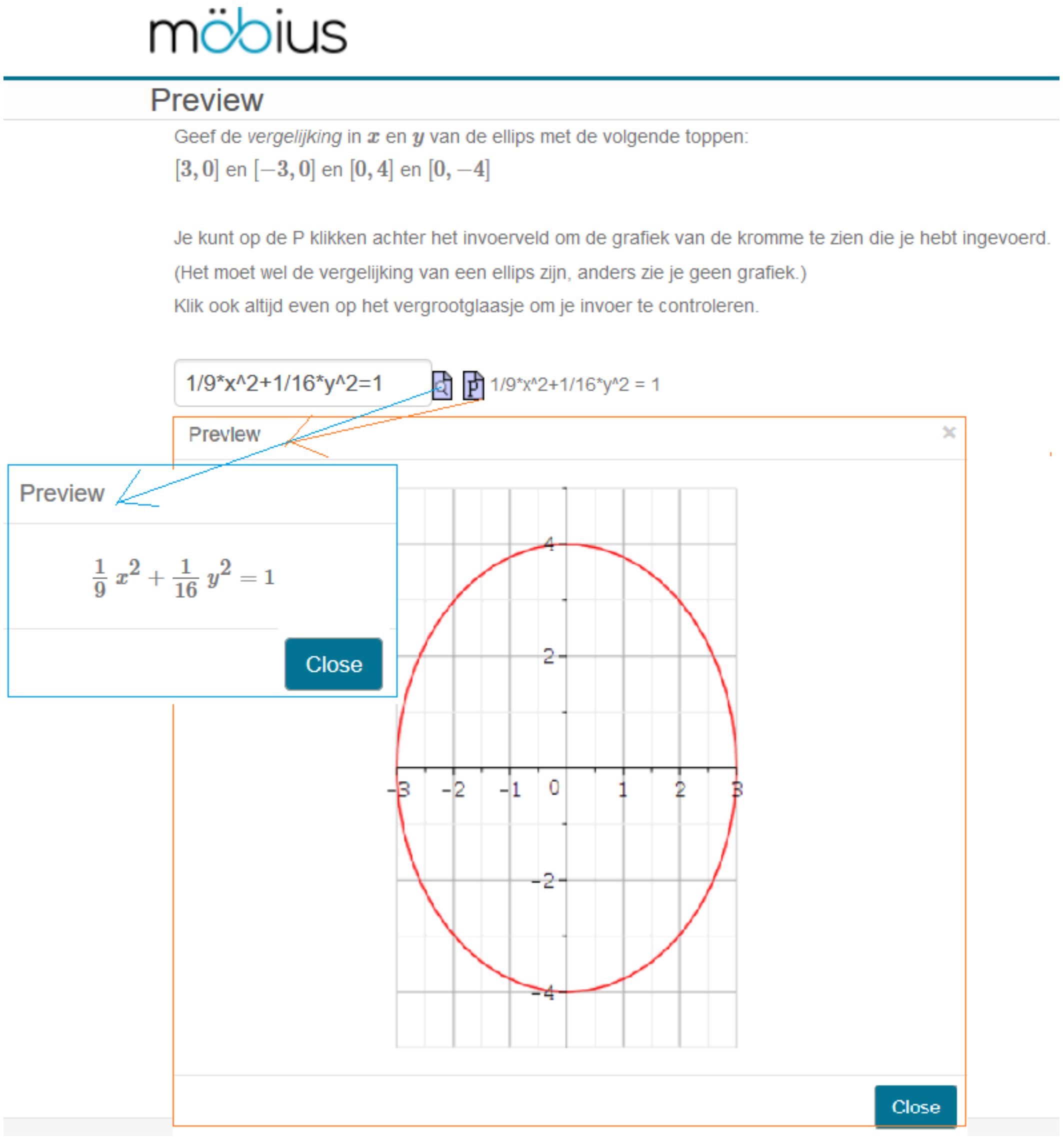
In case the equation has only one solution you can program in the *Grading Code* the following:

```
evalb(evalf[3](solve($verg,x))=evalf[3](solve($RESPONSE,x))) or
evalb(evalf[3](solve($verg,x))=evalf[3](solve($RESPONSE,x)))
and type($RESPONSE,equation);
```

In case the equation has more then one solution it is wise to work with sets as you will see in the next example with quadratic equations.

Quadratic equations

The following example shows that an equation of a distorted circle is requested.



Preview

Geef de *vergelijking* in x en y van de ellips met de volgende toppen:

$[3,0]$ en $[-3,0]$ en $[0,4]$ en $[0,-4]$

Je kunt op de P klikken achter het invoerveld om de grafiek van de kromme te zien die je hebt ingevoerd.
(Het moet wel de vergelijking van een ellips zijn, anders zie je geen grafiek.)
Klik ook altijd even op het vergrootglaasje om je invoer te controleren.

1/9*x^2+1/16*y^2=1

1/9*x^2+1/16*y^2 = 1

Preview

$$\frac{1}{9}\,x^2+\frac{1}{16}\,y^2=1$$

Figure 1.108: Equation of an ellipse

In this case, the student must complete a quadratic equation in x and y.
If now the solution of the equation that the student enters has to be matched with the solution of the correct equation, then there are more expressions that count as a solution. These exact solutions are listed in a row whose order is not important. The best thing is to make the solution a set and thus basically match two sets:

```
evalb({solve($antw,y)}={solve($RESPONSE,y)}) and
 type($RESPONSE,equation);
```

It means that the correct equation $antw the solution set is made with, for example, y as unknown. This can easily be done by putting curly brackets around it. Similarly, the solution set of the student's equation is also made with y as unknown.

HINT: Matching these solusion sets is only possible with exact solutions. See section *Sets (page 64)* for more hints for matching sets for the case you will work with decimal numbers with tolerance.

HINT: See also section *Simple equations (page 131)* to work with a quite different question type *Mathematical Formula*.

HINT: For the matching of **solutions** you will see in section *Ordered and unordered lists (page 93)* some interesting possibilities.

HINT: Another nice thing to switch the left hand side and the right hand side of an equation is the following:

```
$verg=maple("p=x+y");
```

```
$verg1=maple("(rhs = lhs)($verg)");
```

1.4.15 Systems of equations

When solving equations, you can use Maple again, which can effortlessly solve almost any system. There are two ways to generate the solution: with **sets** and with **lists**. Below are a few examples to see what Maple can do for you.
See also the manual *Handleiding Maple*17 section 2.6.7 and 2.6.8 about systems of equations.

```
> restart;verg1:=7*a-11*b=18;verg2:=3*a+4*b=2;
```

$$verg1 := 7\,a - 11\,b = 18$$

$$verg2 := 3\,a + 4\,b = 2 \tag{1.62}$$

```
> solve({verg1,verg2},{a,b});
```

$$\left\{a = \frac{94}{61}, b = -\frac{40}{61}\right\} \tag{1.63}$$

```
> solve([verg1,verg2],[a,b]);
```

$$\left[\left[a = \frac{94}{61}, b = -\frac{40}{61}\right]\right] \tag{1.64}$$

```
> verg3:=y=x^2+x-2;verg4:=y=4*x+2;
```

$$verg3 := y = x^2 + x - 2$$

$$verg4 := y = 4\,x + 2 \tag{1.65}$$

```
> solve({verg3,verg4},{x,y});
```

$$\{x = -1, y = -2\}, \{x = 4, y = 18\} \tag{1.66}$$

```
> solve([verg3,verg4],[x,y]);
```

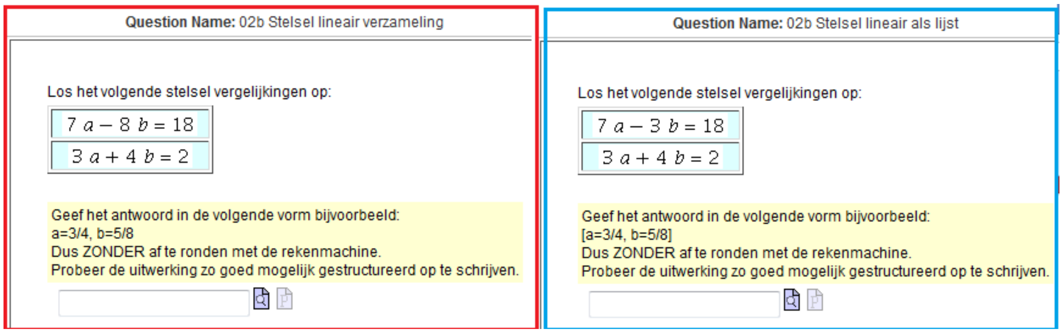
$$[[x = -1, y = -2], [x = 4, y = 18]] \tag{1.67}$$

In the above, you can see the difference between using sets and lists. Note that lists imply a fixed order! In the case of more than one solution, sets give you a **row of sets**, and lists in response, **a list consisting of lists**.

System of two linear equations. The solution is one number pair

The following example asks for the solution of a system of linear equations.

We approach this problem in two ways: with sets and with lists.



Question Name: 02b Stelsel lineair verzameling

Los het volgende stelsel vergelijkingen op:

$7\ a - 8\ b = 18$

$3\ a + 4\ b = 2$

Geef het antwoord in de volgende vorm bijvoorbeeld:
a=3/4, b=5/8
Dus ZONDER af te ronden met de rekenmachine.
Probeer de uitwerking zo goed mogelijk gestructureerd op te schrijven.

Question Name: 02b Stelsel lineair als lijst

Los het volgende stelsel vergelijkingen op:

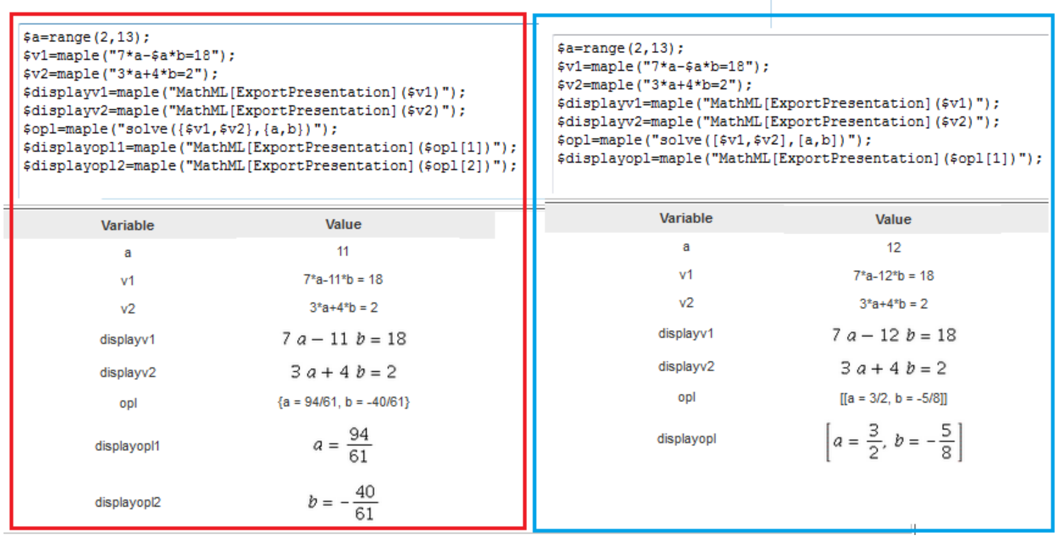
$7\ a - 3\ b = 18$

$3\ a + 4\ b = 2$

Geef het antwoord in de volgende vorm bijvoorbeeld:
[a=3/4, b=5/8]
Dus ZONDER af te ronden met de rekenmachine.
Probeer de uitwerking zo goed mogelijk gestructureerd op te schrijven.

Figure 1.109: System of equations

The corresponding *Algorithms* can be seen below:



```
$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve({$v1,$v2},{a,b})");
$displayopl1=maple("MathML[ExportPresentation]($opl[1])");
$displayopl2=maple("MathML[ExportPresentation]($opl[2])");
```

Variable	Value
a	11
v1	7*a-11*b = 18
v2	3*a+4*b = 2
displayv1	$7\ a - 11\ b = 18$
displayv2	$3\ a + 4\ b = 2$
opl	{a = 94/61, b = -40/61}
displayopl1	$a = \frac{94}{61}$
displayopl2	$b = -\frac{40}{61}$

```
$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve([$v1,$v2],[a,b])");
$displayopl=maple("MathML[ExportPresentation]($opl[1])");
```

Variable	Value
a	12
v1	7*a-12*b = 18
v2	3*a+4*b = 2
displayv1	$7\ a - 12\ b = 18$
displayv2	$3\ a + 4\ b = 2$
opl	[[a = 3/2, b = -5/8]]
displayopl	$\left[a = \frac{3}{2},\ b = -\frac{5}{8}\right]$

Figure 1.110: System of equations with sets and with lists

For the set

The corresponding *Grading Code*: `evalb(($opl)={$RESPONSE});`
In the *Answer* section you give the correct answer as a row: `$opl[1],$opl[2];`

You see in the *Algorithm* the variable $opl is a set and you will match it in the *Grading Code* with the students set by putting curly brackets round it. The advantage is that the student does not have to place brackets or curly brackets and the order of typing is also not important for the student. Just as well, the student could have swapped the variables a and b.

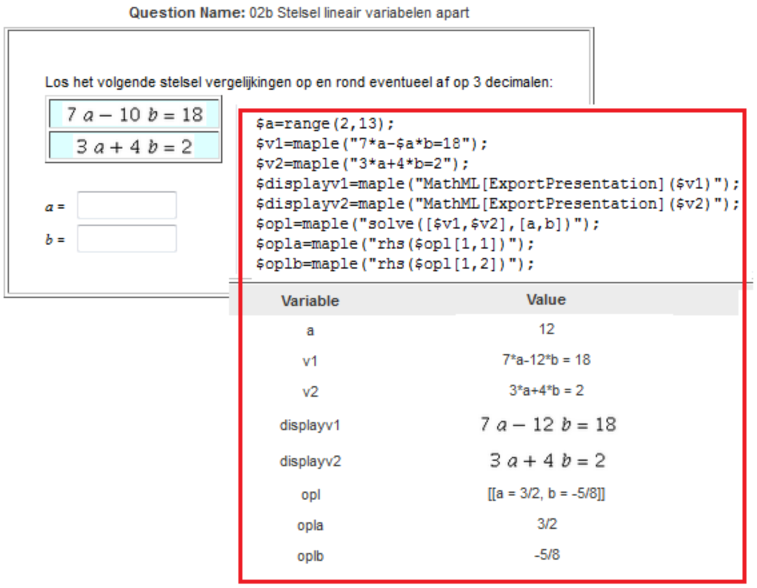
HINT: Note that in the *Answer* section, do not present the set of curly brackets. The student might think when he sees the correct answer that he should have entered curly brackets. If necessary, work with operands of the set. (See section *(page 66)*.)
For the list:
The corresponding *Grading Code*: `evalb(($opl[1])=$RESPONSE);`
In the *Answer* section you give the correct answer as: `$opl[1];`

The correct answer is the first operand of the list $opl: [a=3/2,b=-5/8].
The disadvantage of a list is that the order of this list is fixed. The student must therefore first handle a and then b and place square brackets.

HINT: Make sure that you have some unity in the way you enter the answer. Otherwise, the student always has to use different ways of input and that can be confusing.

Questioning the individual variables

Optionally, you can also question the individual variables, so that there are more possibilities for partial grading:



Question Name: 02b Stelsel lineair variabelen apart

Los het volgende stelsel vergelijkingen op en rond eventueel af op 3 decimalen:

$7\ a - 10\ b = 18$

$3\ a + 4\ b = 2$

a =

b =

```
$a=range(2,13);
$v1=maple("7*a-$a*b=18");
$v2=maple("3*a+4*b=2");
$displayv1=maple("MathML[ExportPresentation]($v1)");
$displayv2=maple("MathML[ExportPresentation]($v2)");
$opl=maple("solve([$v1,$v2],[a,b])");
$opla=maple("rhs($opl[1,1])");
$oplb=maple("rhs($opl[1,2])");
```

Variable	Value
a	12
v1	7*a-12*b = 18
v2	3*a+4*b = 2
displayv1	$7\ a - 12\ b = 18$
displayv2	$3\ a + 4\ b = 2$
opl	[[a = 3/2, b = -5/8]]
opla	3/2
oplb	-5/8

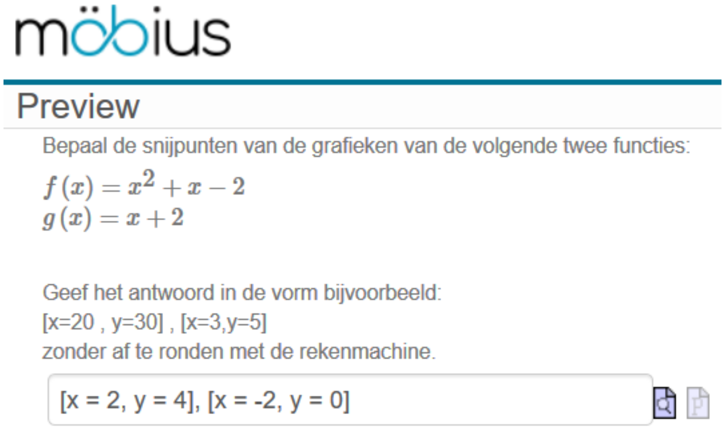
Figure 1.111: Sets of equations with individual variables

Then use lists again in the `solve` command, because you can be sure that the order of the variables is fixed. For example, by calling `rhs($opl[1,2])` you get the right member of the second element of the first list of the list. After all, the solution is generated as a list containing lists (here only one).

HINT: In this last example, you can create numeric answer fields and possibly allow rounding and tolerance if you want. You can also choose a *Formula* answer field and force the exact answer.

System of nonlinear equations

In the following example, there are more solutions of a system of equations.
It asks for the coordinates of the intersections of two graphs.



Preview

Bepaal de snijpunten van de grafieken van de volgende twee functies:

$f(x) = x^2 + x - 2$

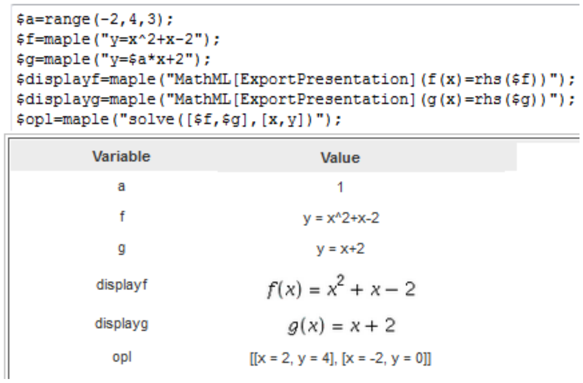
$g(x) = x + 2$

Geef het antwoord in de vorm bijvoorbeeld:
[x=20 , y=30] , [x=3,y=5]
zonder af te ronden met de rekenmachine.

[x = 2, y = 4], [x = -2, y = 0]

Figure 1.112: System of equations with more solutions

The corresponding *Algorithm* is the following:



```
$a=range(-2,4,3);
$f=maple("y=x^2+x-2");
$g=maple("y=$a*x+2");
$displayf=maple("MathML[ExportPresentation](f(x)=rhs($f))");
$displayg=maple("MathML[ExportPresentation](g(x)=rhs($g))");
$opl=maple("solve([$f,$g],[x,y])");
```

Variable	Value
a	1
f	y = x^2+x-2
g	y = x+2
displayf	$f(x) = x^2 + x - 2$
displayg	$g(x) = x + 2$
opl	[[x = 2, y = 4], [x = -2, y = 0]]

Figure 1.113: System of equations with more solutions Algorithm

Here you can see that the solution $opl is given in the form of a list. It is a list consisting of two lists.
Within maple, you can get the separate lists with $opl[1] and $opl[2].

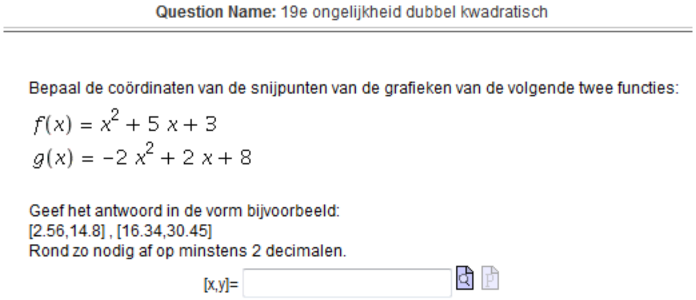
With the *Grading Code* of this answer field you work with a set containing two lists, so that the order of the lists does not matter:
`evalb({$opl[1],$opl[2]}={$RESPONSE});`

For the correct answer in the *Answer* section: `$opl[1],$opl[2];`

Care has been taken to make it work out nicely so that the student is not tempted to apply numerical approaches.

Numerical solution of a system of nonlinear equations

If you plan to allow a numerical approach, the following way of doing things is an option.



Question Name: 19e ongelijkheid dubbel kwadratisch

Bepaal de coördinaten van de snijpunten van de grafieken van de volgende twee functies:

$f(x) = x^2 + 5\,x + 3$

$g(x) = -2\,x^2 + 2\,x + 8$

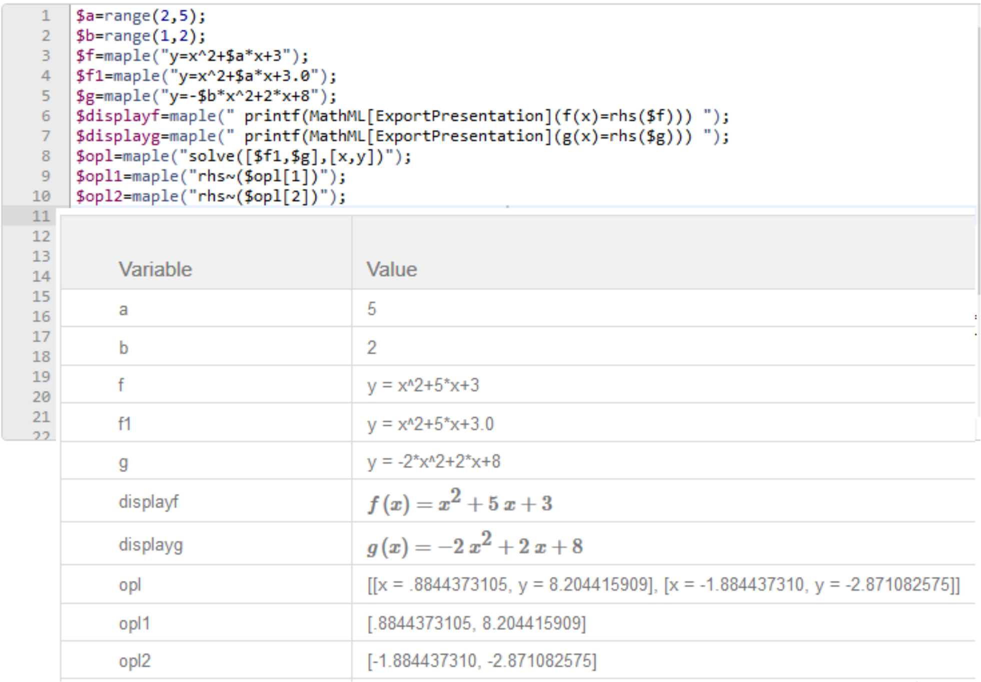
Geef het antwoord in de vorm bijvoorbeeld:
[2.56,14.8] , [16.34,30.45]
Rond zo nodig af op minstens 2 decimalen.

[x,y]=

Figure 1.114: System of equations with more solutions numeric

In this question, the coordinates of the intersections of the two graphs are no longer numbers that come out nicely. It is obvious that a numerical approach must now be applied. You can of course ask for the x-values and create two separate numeric answer fields for them, but it can also be done at once as done here.
Because this issue is part of a larger whole, in this case it is not bad if the student uses tools such as graphing calculator or computer algebra for this calculation. (In a follow-up question, the intervals in which one function is larger than the other is asked and the answer to this first question is necessary.)

The *Algorithm* is as follows:



```
$a=range(2,5);
$b=range(1,2);
$f=maple("y=x^2+$a*x+3");
$f1=maple("y=x^2+$a*x+3.0");
$g=maple("y=-$b*x^2+2*x+8");
$displayf=maple(" printf(MathML[ExportPresentation](f(x)=rhs($f))) ");
$displayg=maple(" printf(MathML[ExportPresentation](g(x)=rhs($g))) ");
$opl=maple("solve([$f1,$g],[x,y])");
$opl1=maple("rhs~($opl[1])");
$opl2=maple("rhs~($opl[2])");
```

Variable	Value
a	5
b	2
f	y = x^2+5*x+3
f1	y = x^2+5*x+3.0
g	y = -2*x^2+2*x+8
displayf	$f(x) = x^2 + 5\,x + 3$
displayg	$g(x) = -2\,x^2 + 2\,x + 8$
opl	[[x = .8844373105, y = 8.204415909], [x = -1.884437310, y = -2.871082575]]
opl1	[.8844373105, 8.204415909]
opl2	[-1.884437310, -2.871082575]

Figure 1.115: Algorithm of a question about calculating intersections of graphs

In the *Algorithm* above, you can see that in addition to the variable $f, a variable $f1 is also defined. This is done deliberately, because if a decimal number occurs somewhere when solving the system of equations, Maple immediately switches to numerical calculation and you also get the solution in the form of decimal numbers, whatever you want here.

Because the assignment asks for the coordinates of the intersections, prepare them with the variables $opl1 and $opl2.
From the lists, the right hand side of the equations is always taken and the order is maintained.
In the *Grading Code*, the set of the two lists will be matched with the set of the two lists that the student fills in. However, you cannot compare lists of decimal numbers, so multiply the elements of those lists by 100 and then round as follows:

```
lijst:=[$RESPONSE]: evalb({map(round,100*$opl1),map(round,100*
$opl2)}={map(round,100*lijst[1]),map(round,100*lijst[2])});
```

In this *Grading Code* you compare sets with each other. So the order of the lists (the coordinates (x,y)) that the student fills in is not important.

HINT: It does not matter if the student would fill in more than 2 decimal places, because the student's answer is still multiplied by 100 and then rounded off. Less than 2 decimal places will not be calculated correctly, unless the second decimal place would be a 0.

HINT: In the *Answer* section you can give for example the following:

```
evalf[3]($opl1),evalf[3]($opl2);
```

The student then at least has the correct answer at hand in case he had answered this question incorrectly.

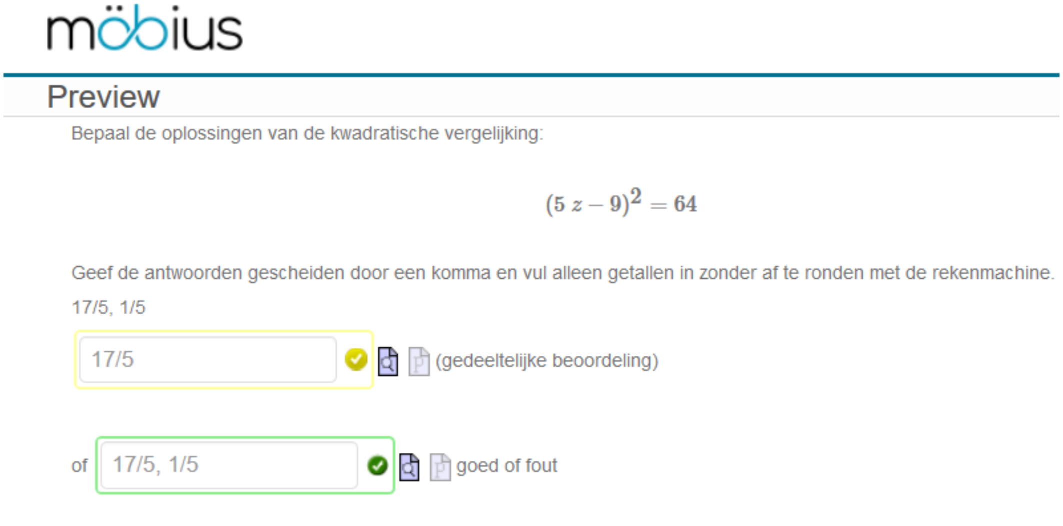
1.4.16 Ordered and unordered lists

For solving equations with more than one solution, you can grade an ordered or unordered list with the *Maple graded question type* and then you can match the solution of the equation with the correct solution possibly with matching sets or, if necessary, with separate answer fields. But beware! If you work with sets, it can only be done with exact answers, and if there are decimal numbers in the equation, then it does not work with sets. Then in the *Grading Code* use the multiplication and rounding trick mentioned in section *Sets (page 64)*.

HINT: For more information about ordered and unordered lists in a different question type see section *(page 129)*.

Unordered list with a set

The following example is about solving a quadratic equation exactly.



Preview

Bepaal de oplossingen van de kwadratische vergelijking:

$$(5\,z-9)^2=64$$

Geef de antwoorden gescheiden door een komma en vul alleen getallen in zonder af te ronden met de rekenmachine.

17/5, 1/5

17/5 ✓ (gedeeltelijke beoordeling)

of 17/5, 1/5 ✓ goed of fout

Figure 1.116: Solving a quadratic equation exactly with partial grading

The *Algorithm* of this question is as follows.

```
$a=range(1,9);
$b=$a^2;
$c=switch(rint(2),range(-9,-1),range(1,9));
$d=range(2,6);
$vraag=maple("($d*z-($c))^2=$b");
$displayvraag=maple("printf(MathML[ExportPresentation]($vraag))");
$opl=maple("solve($vraag,z)");
$Antw1=maple("[solve($vraag,z)][1]");
$Antw2=maple("[solve($vraag,z)][2]");
$antw2=maple("max([$Antw1,$Antw2])");
$antw1=maple("min([$Antw1,$Antw2])");
```

The *Grading Code* for one answer field is as follows:

```
aantal:=nops({$RESPONSE} intersect {$opl}): if StringTools[Search]
("solve","$RESPONSE")>0 then 0 else evalf(aantal/2) end;
```

In this code, the student's answer is converted to a set by placing curly brackets around it. The number of elements of the intersect of the student's set and the correct set is counted. As a result, this number divided by 2 then gives a number between 0 and 1 which corresponds to the rating for this question if, for example, there were two answers.
But then, of course, there should not be the Maple command "solve" in the student's answer. This way you can also provide partial grading.

HINT: Of course, you must be assured that the correct answer definitely consists of two elements, but this can be provided for when choosing the variables in the *Algorithm*.

HINT: For the first answer field, do not opt for the setting *Formula*, because then the student will get brackets in the *Preview* to see his answer and that is not what you want. However, if you use the setting of *Maple syntax with text entry only*, the student can use the Maple command "solve" in his answer. In that case, you still have to program extra with *StringTools*, which "solve" should not appear in the answer, as above. If you don't want to program that, you can opt for *the Symbol entry only* setting and the following *Grading Code* will suffice:

```
aantal:=nops({$RESPONSE} intersect {$opl}): evalf(aantal/2) ;
```

HINT: In this case, there are two answers for the solution of the equation. These answers can also be graded separately in two different answer fields of, for example, the question type *Numeric*. That way you can also easily allow decimal numbers with any tolerance. In the *Algorithm*, the two answers are prepared separately here ($antw1 and $antw2). The largest with the Maple command `max` and the smallest with the Maple command `min`.

HINT: When it comes to *decimal numbers*, first convert the sets to integer sets, otherwise matching won't work. For example, multiply the elements of the set by 100 and then round them off:

```
aantal:=nops(map(x->round(100*x),{$RESPONSE}) intersect map(x-
>round(100*x),{$opl})):
if StringTools[Search]("solve","RESPONSE")>0 then 0 else
 evalf(aantal/2) end;
```

Or if you don't opt for partial grading:

```
evalb(map(x->round(100*x),{$antw1a,$antw2a}) = map(x->round(100*x),{$RESPONSE}));
```

See also section *Sets (page 64)*.

HINT: For example a partial grading with different weights for 3 answers you have the following *Grading Code*:

```
if evalb(StringTools[Search]("solve","$RESPONSE")=0)  then

`if`(member($antw1,{$RESPONSE}),0.3,0)

+

`if`(member($antw2,{$RESPONSE}),0.3,0)

+

 `if`(member($antw3,{$RESPONSE}),0.4,0)

else 0 end if;
```

Ordered list with vector

Matching ordered lists is only possible if all elements are exactly the same. A list is always ordered if there are square brackets around it.
In the following example, the A and B lists are not exactly the same because the first element is a decimal number and the second is an integer.
So if you want to **match lists**, make sure that all elements are rational **or switch to vectors**.

```
> A:=[3.0,5.23,1/3];B:=[3,5.23,1/3];
```

$$A := \left[3.0, 5.23, \frac{1}{3}\right]$$

$$B := \left[3, 5.23, \frac{1}{3}\right] \qquad (1.68)$$

```
> evalb(A=B);
```

$$\mathit{false} \qquad (1.69)$$

```
> convert~(A,rational);
```

$$\left[3, \frac{523}{100}, \frac{1}{3}\right] \tag{1.70}$$

Here, all elements of the list A have been converted to rational numbers.

```
> evalb(convert~(A,rational)=convert~(B,rational));
```

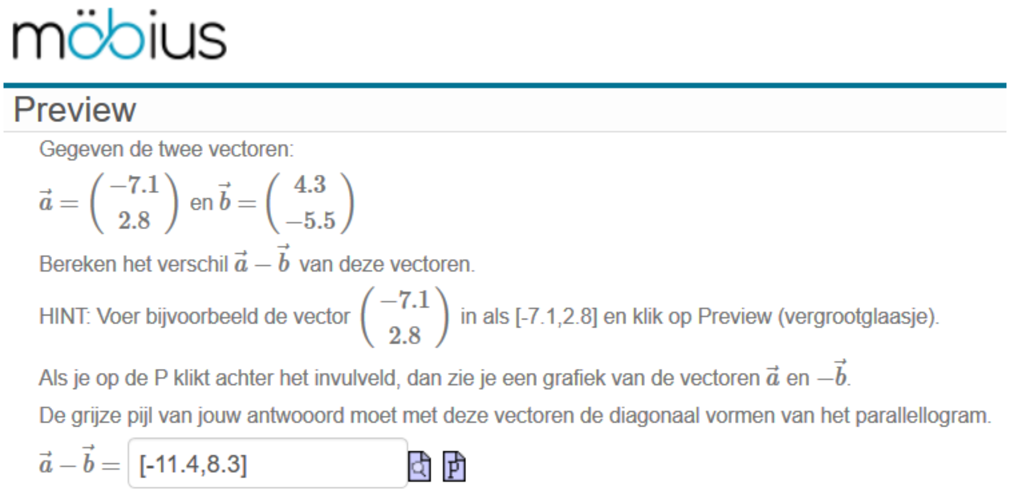
$$true \tag{1.71}$$

If you use vectors in stead of lists in the *Grading Code*, it is more robust for decimal numbers and integers.

```
> LinearAlgebra[Equal](Vector(A),Vector(B));
```

$$true \tag{1.72}$$

In the following example, the student must type an ordered list in the answer field. This ordered list is basically a vector. It is therefore useful to treat them as such.



Preview

Gegeven de twee vectoren:

$\vec{a} = \begin{pmatrix} -7.1 \\ 2.8 \end{pmatrix}$ en $\vec{b} = \begin{pmatrix} 4.3 \\ -5.5 \end{pmatrix}$

Bereken het verschil $\vec{a} - \vec{b}$ van deze vectoren.

HINT: Voer bijvoorbeeld de vector $\begin{pmatrix} -7.1 \\ 2.8 \end{pmatrix}$ in als [-7.1,2.8] en klik op Preview (vergrootglaasje).

Als je op de P klikt achter het invulveld, dan zie je een grafiek van de vectoren $\vec{a}$ en $-\vec{b}$.

De grijze pijl van jouw antwoord moet met deze vectoren de diagonaal vormen van het parallellogram.

$\vec{a} - \vec{b} =$ [-11.4,8.3]

Figure 1.117: Vector as an ordered list

The *Algorithm* of this question is as follows:

```
$a1=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$a2=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$b1=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$b2=switch(rint(2),decimal(1,rand(-10,-1)),decimal(1,rand(1,10)));
$detabs=maple("abs($a1*($b2)-($b1)*($a2))");
condition:gt($detabs,3);
$A=maple("Vector([$a1,$a2])");
$Adisplay=maple("printf(MathML[ExportPresentation]($A))");
$B=maple("Vector([$b1,$b2])");
$Bdisplay=maple("printf(MathML[ExportPresentation]($B))");
$AminusB=maple("map(evalf[2],($A-($B)))");
$AminusBdisplay=maple("printf(MathML[ExportPresentation]($AminusB))");
$p=plotmaple("plots[display]({plottools[arrow]([0,0],
$A,0.08,0.3,0.2,color=red),plottools[arrow]([0,0],-
($B),0.08,0.3,0.2,color=blue),plottools[arrow]([0,0],
$AminusB,0.07,0.2,0.1,color=gray),plottools[line]
(convert($AminusB,list),convert($A,list),linestyle=dash),plottools[line]
(convert($AminusB,list),convert(-
($B),list),linestyle=dash)},scaling=constrained)");
```

For the *Grading Code*, in this case, you turn the lists into a vector and you match the vectors with each other. You also immediately ensure that only one list is typed by the student by checking the string of the answer for number of square brackets.

```
LinearAlgebra[Equal]($AminusB,Vector($RESPONSE)) and
 evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","[")<2);
```

If there really is rounding to two decimal places, you can move on to rounding correct answer and student response and match these two vectors.

```
LinearAlgebra[Equal]
(map(round,100*($AminusB)),map(round,100*(Vector($RESPONSE)))) and
 evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","[")<2);
```

Also pay some attention to the feedback by giving the correct answer in the *Answer* section and optionally a note using the *Custom Previewing Code* if you opt for *Text entry only*.

HINT See for more information in the section about *Vectors (page 118)*.

1.4.17 Inequalities

Testing inequalities requires some knowledge of the Maple computer algebra system because you will outsource a lot to it anyway.
First, Maple has an obnoxious habit of always presenting everything with the < sign. Although x>-2.82 is considered the same as -2.82<x, Maple will always present the last form with the < sign, so that there is ambiguity about the left and right sides of the inequality. Possibly the problem must be adjusted in terms of presentation because Maple will first simplify the inequality as much as possible before creating the MathML code. The MathML coding probably won't always deliver the inequality you want to offer. You can use MathML and then also offer the left hand side and the right hand side of the inequality separately. Or work with LaTeX.
The inequality can be solved using Maple, but it has to be done in a special way. Numerical solving is not possible anyway, but then the command evalf can still be given after solve.
The `solve` command must always be accompanied by the unknown in the form of a set, so `solve(inequality,{x})`. You will then receive the answers in the form of inequalities: one time a row of sets and the other time a single set. Something to watch out for if you want to match sets in the *Grading Code*.

```
> restart;ong1:=4*x+5>-4;
```

$$ong1 := 0 < 4\,x + 9 \tag{1.73}$$

```
> lhs(ong1);rhs(ong1);
```

$$0$$

$$4\,x + 9 \tag{1.74}$$

```
> solve(ong1,{x});
```

$$\left\{-\frac{9}{4} < x\right\} \tag{1.75}$$

```
> ong2:=x^2>8;ong3:=x^2<8;
```

$$ong2 := 8 < x^2$$

$$ong3 := x^2 < 8 \tag{1.76}$$

```
> solve(ong2,{x});
```

$$\{x < -2\sqrt{2}\}, \{2\sqrt{2} < x\} \tag{1.77}$$

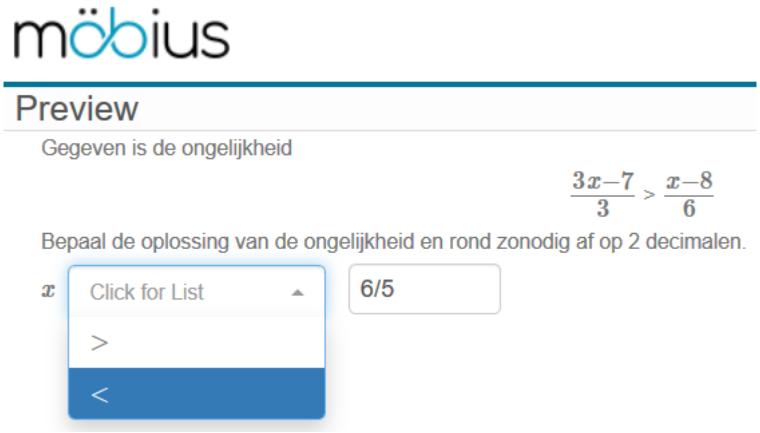
```
> evalf[3](solve(ong3,{x}));
```

$$\{-2.82 < x, x < 2.82\} \tag{1.78}$$

There are several possibilities to check the solution of an inequality for correctness.
Below we give several examples to vary in approach.

Linear inequality

The student has a separate field for the < or > sign and a numerical field for entering the number (with a margin if necessary). This has the advantage that rounding off can be handled properly.



Preview

Gegeven is de ongelijkheid

$$\frac{3x-7}{3} > \frac{x-8}{6}$$

Bepaal de oplossing van de ongelijkheid en rond zonodig af op 2 decimalen.

x Click for List ▴ | 6/5

>

<

Figure 1.118: Linear inequation

For example, the student gets a weighting of 1 for the sign and 2 for the number he has to fill in. The *Algorithm* is as follows:

```
$a=range(1,7);
$b=switch(rint(4),2,4,5,6);
$ongleft=mathml("(3*x-$a)/3");
$ongright=mathml("(x-8)/$b");
$index=rint(2);
$teken=switch($index,">","<");
$tekenfout=switch($index,"<",">");
$rechtslinks=switch($index,"lhs","rhs");
$ong1=maple("(3*x-$a)/3 $teken (x-8)/$b");
$opl1=maple("solve($ong1,{x})");
$Getal=maple("$rechtslinks($opl1[1])");
$opl=maple("solve((3*x-$a)/3 = (x-8)/$b,x)");
```

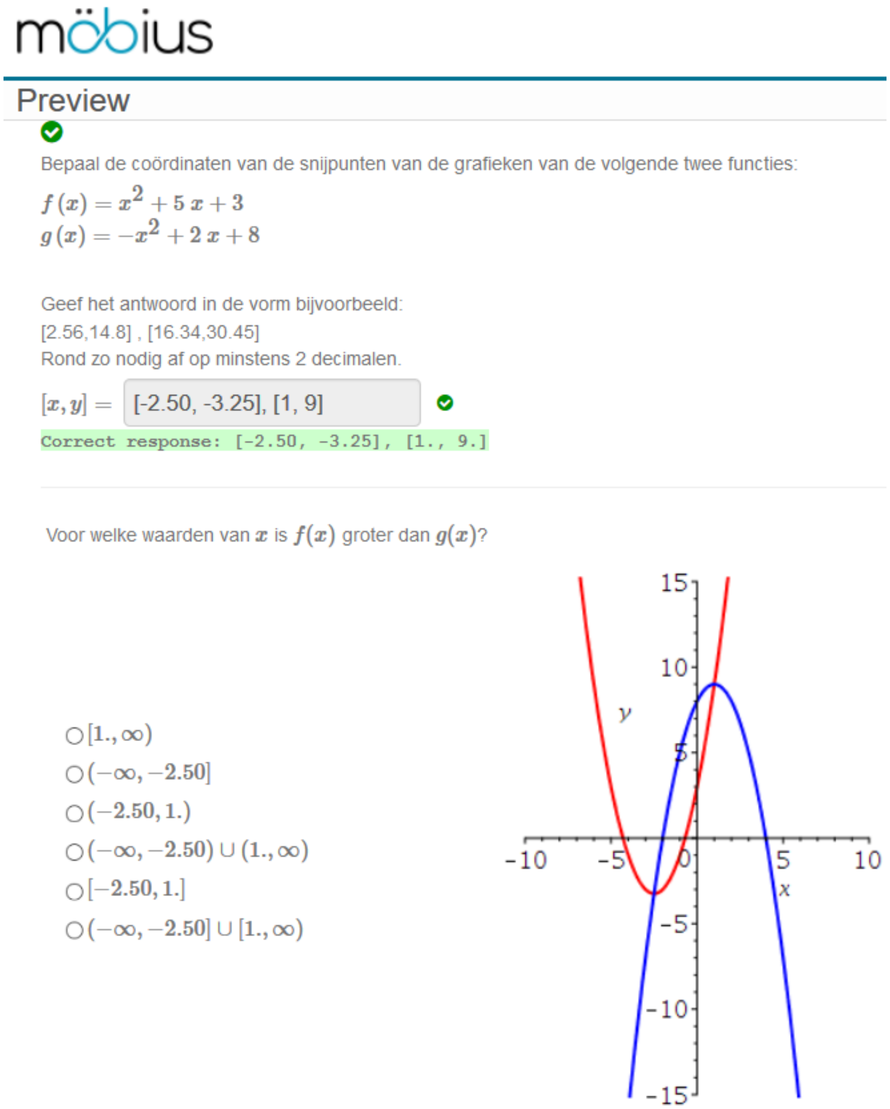
When defining the left and right sides of the inequality, it is taken into account that no reversal of the sign occurs during the inequality solution.
Defining the left and right hand sides as MathML code implies that the inequality can be presented in unsimplified form in the text of the question with $ongleft $teken $ongright.
Maple displays the solution as {number < x} or as {x < number} because Maple always uses the less than sign, hence the definition of $Getal in the *Algorithm*.
Although that can also be solved here in another way by not solving the inequality, but the equation with:
`$number=maple("solve((3*x-$a)/3 = (x-8)/$ b,x)");` (There is only one answer to this linear equation.)

The variable $teken can be used not only when presenting the problem, but also when defining the correct answer. The wrong answer for the character in the answer field of the *List* question type is defined as $tekenfout.

When defining the *Numerical* answer field, a match is made with the variable $Getal and opted for *Accept arithmetic* so that the student can also fill in the exact answer if he does not round. The tolerance is set to 0.01 in case the student rounds to two decimal places.

Inequality with intervals

The following example uses intervals to represent the solution of an inequality. Multiple choice can then be used, adaptively preceded by a calculation of the intersections of two graphs. Then the student can translate the solution of the inequality into the correct interval of the variable.
To create open answer fields at intervals, refer to *Creating Items Manual Part C*.



Preview

Bepaal de coördinaten van de snijpunten van de grafieken van de volgende twee functies:

$f(x) = x^2 + 5\,x + 3$
$g(x) = -x^2 + 2\,x + 8$

Geef het antwoord in de vorm bijvoorbeeld:
[2.56,14.8] , [16.34,30.45]
Rond zo nodig af op minstens 2 decimalen.

$[x, y] =$ [-2.50, -3.25], [1, 9]

```
Correct response: [-2.50, -3.25], [1., 9.]
```

Voor welke waarden van x is $f(x)$ groter dan $g(x)$?

○ $[1., \infty)$
○ $(-\infty, -2.50]$
○ $(-2.50, 1.)$
○ $(-\infty, -2.50) \cup (1., \infty)$
○ $[-2.50, 1.]$
○ $(-\infty, -2.50] \cup [1., \infty)$

Figure 1.119: Inequation with intervals

The first part of the question asks about the intersections of two graphs. That comes down to solving a system of equations. See also in section *Systems of equations (page 89)*.
The second part of the question is in the form of multiple choice. The student will in any case be shown the answer to the first question, if that was not correct, and in this adaptive way the student can still show that he can translate the solution of the inequality into interval notation.
In this case, the alternatives to the multiple choice question in the *Algorithm* are prepared in the form of the variables $test1 to $test6, the latter of which is therefore the correct choice.

```
$a=range(2,5);
$b=range(1,2);
$f=maple("y=x^2+$a*x+3");
$f1=maple("y=x^2+$a*x+3.0");
$g=maple("y=-$b*x^2+2*x+8");
$displayf=maple(" printf(MathML[ExportPresentation](f(x)=rhs($f)))
 ");
$displayg=maple(" printf(MathML[ExportPresentation](g(x)=rhs($g)))
 ");
$opl=maple("solve([$f1,$g],[x,y])");
$opl1=maple("rhs~($opl[1])");
$opl2=maple("rhs~($opl[2])");
$oplossing=maple("solve(rhs($f1)=rhs($g),x)");
$oplmin=maple("evalf[3](min([$oplossing]))");
$oplmax=maple("evalf[3](max([$oplossing]))");
$displayopl1=maple(" printf(MathML[ExportPresentation]($opl[1]))
 ");
$displayopl2=maple(" printf(MathML[ExportPresentation]($opl[2]))
 ");
$plot1=plotmaple("plot([rhs($f),rhs($g)],x=-10..10,y=-15..15,color=[red,blue],thickn
 width=250'");
$plot=plotmaple("plot([rhs($f),rhs($g)],x=-10..10,y=-15..15,color=[red,blue],thickne
 width=250'");
$optie1=maple("printf(MathML[ExportPresentation](x>$oplmin))");
$optie2=maple("printf(MathML[ExportPresentation](x<$oplmin))");
$optie3=maple("printf(MathML[ExportPresentation](x>$oplmax))");
$optie4=maple("printf(MathML[ExportPresentation](x<$oplmax))");
$test1="\([$oplmin,$oplmax]\)";
$test2="\((-\infty,$oplmin]\)";
$test3="\([$oplmax,\infty)\)";
$test4="\(($oplmin,$oplmax)\)";
$test5="\((-\infty,$oplmin] \cup [$oplmax,\infty)\)";
$test6="\((-\infty,$oplmin) \cup($oplmax,\infty)\)";
```

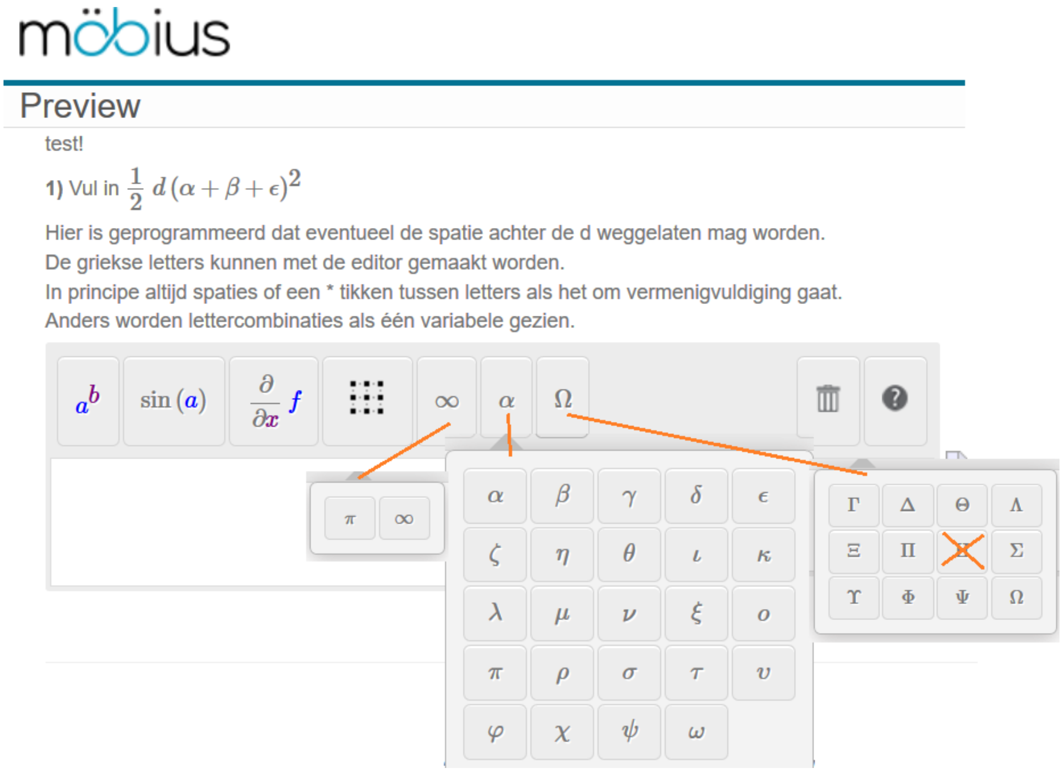
The variables $test 1 to 6 can be called when creating the multiple-choce question, however this LaTeX code (text) can also be entered directly in the alternatives of the multiple-choice answer field.
HINT: See for the dynamic graphs in section *Dynamic graphs (page 134)*.
HINT: To create open answer fields at intervals, refer to *Creating Items Manual Part C*.

1.4.18 Symbols en Greek characters

With the *Maple graded question type* it is also possible to test formulas with Greek characters and some other symbols.
In that case, for the *Maple syntax* setting, take *Symbol entry only* so that the student is offered an editor.
It is also possible without an editor in a *Text entry only* setting, but then the student must receive some more instruction on how to enter the symbols and Greek characters.
HINT: Different *Grading Codes* need to be taken into account for the different settings.

Greek characters

All lowercase characters are in a pallet under the button α. In the pallets with the ∞-sign and the Greek captial Ω there are some other characters.



möbius

Preview

test!

1) Vul in $\frac{1}{2}\, d\,(\alpha+\beta+\epsilon)^2$

Hier is geprogrammeerd dat eventueel de spatie achter de d weggelaten mag worden.
De griekse letters kunnen met de editor gemaakt worden.
In principe altijd spaties of een * tikken tussen letters als het om vermenigvuldiging gaat.
Anders worden lettercombinaties als één variabele gezien.

Figure 1.120: Entering Greek characters

HINT: Also note the bin button on the right, which allows the student to start over if the formula does not work.

Always use the standard Greek characters as defined in Maple. That is, the lowercase characters with alpha, beta, gamma, delta etc..
The capital characters with Alpha, Beta, Gamma etc..

Some problematic characters and symbols include the following. So keep this in mind the prepared variables in the *Algorithm* you can use in the *Grading Code* and in the feedback and everywhere in the question.

- The symbol ∞ is known as infinity in maple.
- The character π is known in maple by Pi. In the pallettes is π always meant as Pi (3.14....).
- The character ϵ is known in maple by epsilon. In the pallet this character is translated to varepsilon (ε).
- The character ζ is known in maple by zeta. In the pallet this character is tranlated to Zeta.
- The capital Π is known in maple by PI. In the pallettes it is the same.
- The capital Γ is known in maple by Gamma needs a special treatment. In the pallettes it is translated by GAMMA.
- The capital Y is known in maple by Upsilon. In the palettes it is the capital Y from the keyboard.
  The grading is not possible only with the capital Y from the keyboard.
- The symbol next to the capital PI does not work.

The correct answer and also the question of the above example are prepared in the *Algorithm* as shown below.

```
$antw1=maple("d(alpha+beta+epsilon)^2/2");
$antw1a=maple("d*(alpha+beta+epsilon)^2/2");
$displayantw1=maple("printf(MathML[ExportPresentation]($antw1))");
```

Two correct answers have been programmed for this situation due to whether or not there is an intended space after the d.
The *Grading Code* is as follows:

```
evalb(($antw1)-($RESPONSE) = 0) or evalb(($antw1a)-($RESPONSE) =
 0);
```

It is actually best to choose the *Maple syntax Symbol entry only* setting, with which the student is then presented with an Editor and can click on the Greek palette. You will see that this *Grading Code* is not sufficient in that case, because the editor translates the character ε to varepsilon.
To make the *Grading Code* also work properly in case you opt for the setting *Symbol entry only*, take the following coding:

```
evalb(subs(varepsilon=epsilon,($antw1)-($RESPONSE)) = 0)
or
evalb(subs(varepsilon=epsilon,($antw1a)-($RESPONSE)) = 0);
```

HINT: Optionally, the settings with this last *Grading Code* can also be set to *Text entry only*, but then the student must know how to type the Greek characters such as alpha and theta and Pi and the like. However, in these kinds of situations, offering an Editor is less risky.

HINT: Please note that if a ζ or a π is used in the formula, these characters are translated with a capital: Zeta and Pi. Maple assigns certain functions or values to these characters. So working with substitution here too.

HINT: You can prepare the Greek capitals in the formulas with, for example, Delta and Theta and Sigma.

HINT: If you want to grade Δx, this is possible if you
prepare the correct answer as Delta*x. It looks like

Δx looks like Δx. The

The combination Deltax is not seen as a Greek character followed by an x. This is also something

combination Deltax is not seen as a Greek character followed by an x. This is also something the students should know and train.

1.4.19 Differentials and differential equations

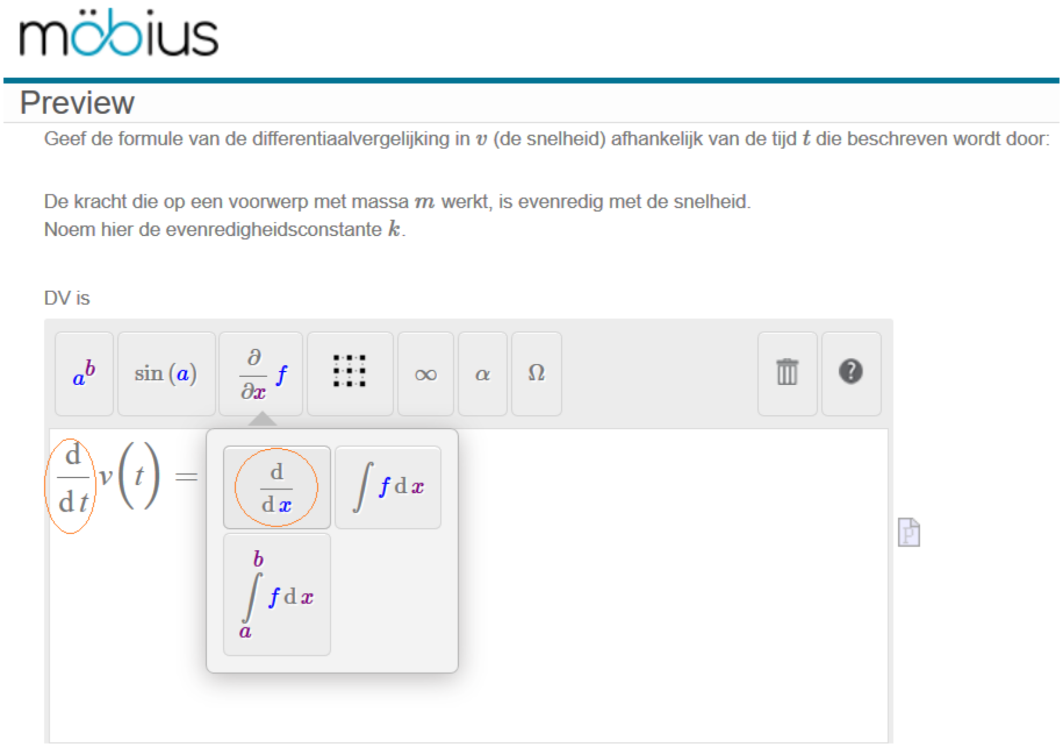
For the grading of differentials and integrals, it is often useful and highly recommended to make the Editor available with *Symbol entry only* in the settings with *Maple syntax*.
Knowledge of some syntax for both the student and the item-makers is required.
HINT: If you will grade the derivative of a function or the integral where the student himself has to calculate it, keep in mind that the student can also use the Maple commands "diff" and "int" and that the answer is therefore graded correctly. This applies not only to the *Text entry only* settings, but also to the use of the Editor (*Symbol entry only*). If the student takes a differential or an integral from the palette in the editor, the response is the **value** of the differential or the **value** of the integral. With *StringTools*, the string of the student's response can be checked for use of these commands with, for example, the *Grading Code*:

```
evalb(StringTools[Search]("diff","$RESPONSE")=0);
```

(This applies to both the settings for *Text mode* and *Symbol mode*.)

Entering a differential equation

If the student needs to enter a differential equation, make the Editor available with the settings *Maple syntax with Symbol entry only*. The student then has access to the operator $\frac{\mathrm{d}}{\mathrm{d}x}$.



Preview

Geef de formule van de differentiaalvergelijking in v (de snelheid) afhankelijk van de tijd t die beschreven wordt door:

De kracht die op een voorwerp met massa m werkt, is evenredig met de snelheid.
Noem hier de evenredigheidsconstante k.

DV is

Figure 1.121: Entering a differential equation

Het *Algorithm* van deze vraag is als volgt:

```
$answ1=maple("diff(v(t),t)=k/m*v(t)");
$answ2=maple("m*diff(v(t),t)=k*v(t)");
$displayansw1=maple("printf(MathML[ExportPresentation]($answ1))");
$displayansw2=maple("printf(MathML[ExportPresentation]($answ2))");
```

The *Grading Code* can be as follows:

```
evalb(dsolve($RESPONSE,v(t),explicit)=dsolve($answ1,v(t),explicit));
```

This means that the explicit solution of the DE the student enters ($RESPONSE) is matched with the explicit solution of the correct answer ($answ1).

This is a fairly simple question and the differential equation can be noted in more ways, but that is not important, because in the *Grading Code* its solutions are matched anyway.
The last lines in the *Algorithm* are for the purposes of the *Feedback* section of the question.

HINT: With the Editor for the student, the differential equation can be graded excellently. The function must then definitely be noted with its independent variables. So here's that v(t). Only with v will not succeed.

With the Editor, it makes no sense to grade expressions such as $\frac{d}{dx} y$. After all, this expression has the meaning 0 because y is seen as independent of x. Similarly diff(v,t).

HINT: See for more information section *Symbol Mode (page 4)* for working with the Editor for which students still need to receive some instruction.

HINT: The solution of a DE in Maple always is an equation by iteself where you can see the integration constant is called _C1.

So if you want the right hand side with the constant equals C, you can get it with the commands `subs` and `rhs`.

```
> dv:=diff(y(x),x)+1/2*y(x)=0;
  dsolve(dv,y(x));
```

$$dv := \frac{\mathrm{d}}{\mathrm{d}x}\, y(x) + \frac{y(x)}{2} = 0$$

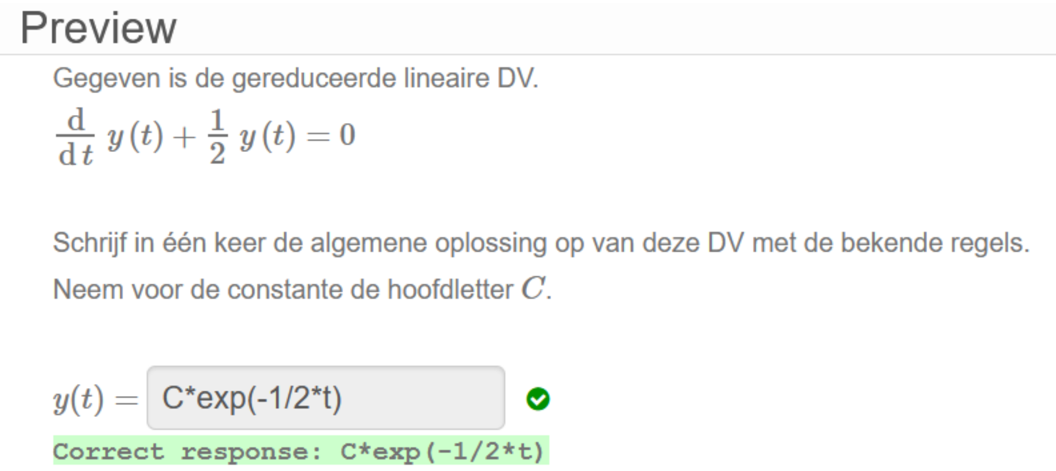
$$y(x) = \_C1\, \mathrm{e}^{-\frac{x}{2}} \tag{1.79}$$

```
> subs(_C1=C,rhs(dsolve(dv,y(x))));
```

$$C\, \mathrm{e}^{-\frac{x}{2}} \tag{1.80}$$

Grading the solution of a differential equation

For the *Grading Code* of the solution of a differential equation, you can fill in the student's solution in the given DV and demand that it be correct as in the following example.



Preview

Gegeven is de gereduceerde lineaire DV.

$$\frac{\mathrm{d}}{\mathrm{d}\,t}\, y\,(t) + \frac{1}{2}\, y\,(t) = 0$$

Schrijf in één keer de algemene oplossing op van deze DV met de bekende regels.

Neem voor de constante de hoofdletter C.

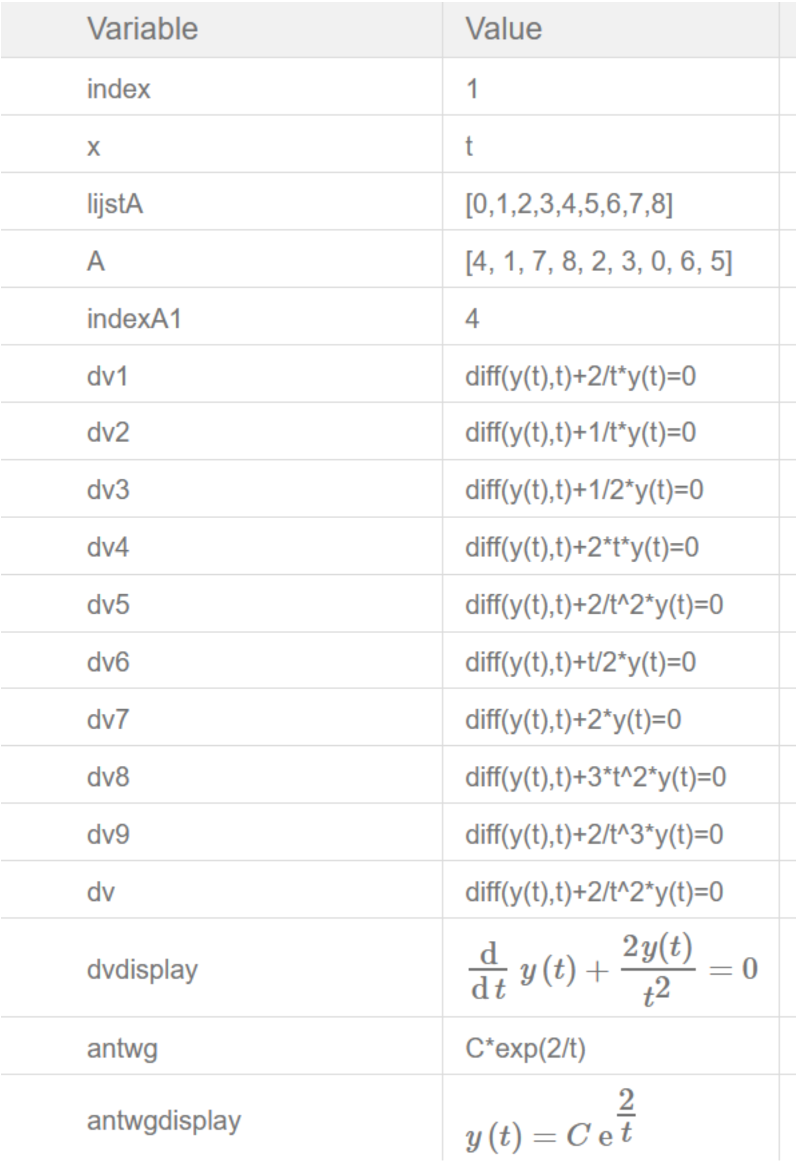
$y(t) =$ C*exp(-1/2*t) ✓

`Correct response: C*exp(-1/2*t)`

Figure 1.122: The grading of the solution of a DE

The *Algorithme* is as follows with randomization of several differential equations and randomisation with the independent variable (x or t):

```
$index=rint(2);
$x=switch($index,"x","t");
$lijstA="[0,1,2,3,4,5,6,7,8]";
$A=maple("StringTools[Randomize]():combinat[randperm]($lijstA)");
$indexA1=switch(0,$A);
$dv1="diff(y($x),$x)+2/$x*y($x)=0";
$dv2="diff(y($x),$x)+1/$x*y($x)=0";
$dv3="diff(y($x),$x)+1/2*y($x)=0";
$dv4="diff(y($x),$x)+2*$x*y($x)=0";
$dv5="diff(y($x),$x)+2/$x^2*y($x)=0";
$dv6="diff(y($x),$x)+$x/2*y($x)=0";
$dv7="diff(y($x),$x)+2*y($x)=0";
$dv8="diff(y($x),$x)+3*$x^2*y($x)=0";
$dv9="diff(y($x),$x)+2/$x^3*y($x)=0";
$dv=switch($indexA1,"$dv1","$dv2","$dv3","$dv4","$dv5","$dv6","$dv7","$dv8","$dv9");
$dvdisplay=maple("printf(MathML[ExportPresentation]($dv))");
$antwg=maple("subs(_C1=C,rhs(dsolve($dv,y($x))))");
$antwgdisplay=maple("printf(MathML[ExportPresentation](y($x)=$antwg))");
```



Variable	Value
index	1
x	t
lijstA	[0,1,2,3,4,5,6,7,8]
A	[4, 1, 7, 8, 2, 3, 0, 6, 5]
indexA1	4
dv1	diff(y(t),t)+2/t*y(t)=0
dv2	diff(y(t),t)+1/t*y(t)=0
dv3	diff(y(t),t)+1/2*y(t)=0
dv4	diff(y(t),t)+2*t*y(t)=0
dv5	diff(y(t),t)+2/t^2*y(t)=0
dv6	diff(y(t),t)+t/2*y(t)=0
dv7	diff(y(t),t)+2*y(t)=0
dv8	diff(y(t),t)+3*t^2*y(t)=0
dv9	diff(y(t),t)+2/t^3*y(t)=0
dv	diff(y(t),t)+2/t^2*y(t)=0
dvdisplay	$\frac{\mathrm{d}}{\mathrm{d}\,t}\, y\left(t\right) + \frac{2y(t)}{t^2} = 0$
antwg	C*exp(2/t)
antwgdisplay	$y\left(t\right) = C\,\mathrm{e}^{\frac{2}{t}}$

Figure 1.123: The solution of the DE

For the *Grading Code* you can program the following:

```
evalb(simplify(eval(subs(y($x)=$RESPONSE,$dv))))
and
evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","C")=1);
```

There must also be a C in the general answer. This is checked with the help of *StringTools*.

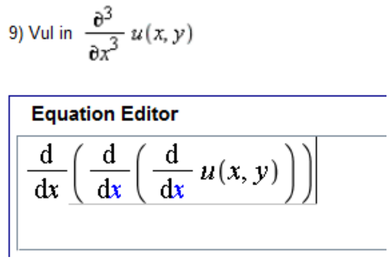
Higher derivatives

For the function u depending on x and y, the third derivative to x can be noted if the correct answer is prepared in the *Algorithm* as follows:

```
$antw1="diff(u(x,y),x,x,x)";
```

This formula can be graded in the setting with the *Editor* (*Symbol entry only*) but also with the setting *Text entry only* where the student is able to use the *Preview* button. Here too, it is useful if the student is somewhat trained in handling the Editor or is aware of the syntax.

In the Editor situation:



9) Vul in $\frac{\partial^3}{\partial x^3} u(x,y)$

$\frac{d}{dx}\left(\frac{d}{dx}\left(\frac{d}{dx} u(x,y)\right)\right)$

Figure 1.124: Higher derivatives with Editor

For the *Grading Code* use simply: `evalb(($antw1)=($RESPONSE));`

For the correct answer in the *Answer* section fill in:

```
printf(MathML[ExportPresentation]
(diff(diff(diff(u(x,y),x),x),x)));
```

In the Text entry only situation

For the *Grading Code* use simply: `evalb(($antw1)=($RESPONSE));`

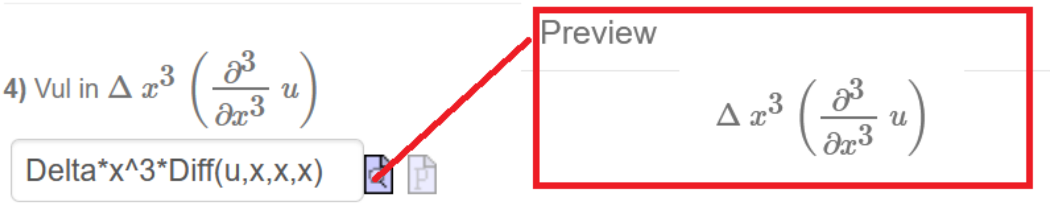
For the correct answer in the *Answer* section fill in: `printf("$antw1");`

HINT: If you put in the *Answer* section the maple statement $antw1, then you will get in the feedback something like diff(u(x,y),x$3) and with the dollar sign is not what you want. That is why you are working with `printf` and not using maple by defining `$antw1="diff(u(x,y),x,x,x)";`

Symbolic notation for higher derivatives

For the symbolic notation, it is better to offer a text field with the setting *Text entry only* and to instruct the student well that this is the symbolic notation. After all, the third derivative of u (a constant) to x is of course 0 so that the grading will not really be conclusive if you do not use the symbolic notation with the *inert command* Diff.

In the *Algorithm* the correct answer you define (without maple) as the string: `$antw4="Delta*x^3*Diff(u,x,x,x)";`



4) Vul in $\Delta\, x^3 \left(\frac{\partial^3}{\partial x^3}\, u\right)$

Delta*x^3*Diff(u,x,x,x)

$\Delta\, x^3 \left(\frac{\partial^3}{\partial x^3}\, u\right)$

Figure 1.125: Higher derivatives with Text

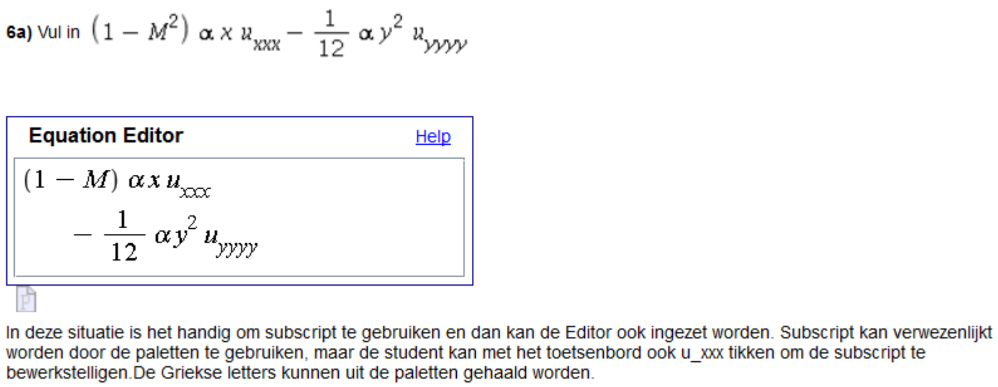
The *Grading Code* simply is: `is(($antw4)=($RESPONSE) );`
The correct answer in the *Answer* section is: `printf("$antw4");`

HINT: An alternative is to work with subscript for the derivative. Then the Editor can be used again: y_x .

The student can also work with underscore to achieve subscript, so for example u_xxx. The Editor will then automatically be subscripted. However, the formula of the answer in Maple notation should then be prepared in the *Algorithm* as follows:

```
$antw6=maple("(1-M^2)*Delta*x*u[xxx]-Delta*y^2/12*u[yyyy]");
```

So in Maple, subscript is with square brackets.



6a) Vul in $\left(1-M^{2}\right)\alpha\, x\, u_{xxx}-\frac{1}{12}\,\alpha\, y^{2}\, u_{yyyy}$

$(1-M)\,\alpha\, x\, u_{xxx}$
$-\frac{1}{12}\,\alpha\, y^{2}\, u_{yyyy}$

In deze situatie is het handig om subscript te gebruiken en dan kan de Editor ook ingezet worden. Subscript kan verwezenlijkt worden door de paletten te gebruiken, maar de student kan met het toetsenbord ook u_xxx tikken om de subscript te bewerkstelligen.De Griekse letters kunnen uit de paletten gehaald worden.

Figure 1.126: Subscript in the Editor

The *Grading Code* does not match the solutions of the DV, but the formula can be matched in the normal way as follows:

```
evalb(($antw6)=($RESPONSE) );
```

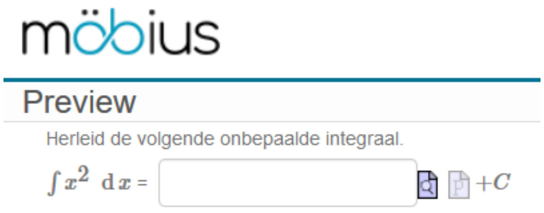
In the *Answer* section you prepare the correct answer as follows:

```
printf(MathML[ExportPresentation]($antw6));
```

1.4.20 Integrals

1.4.20.1 Indefinite integration

Indefinite integration is the computation of antiderivatives of functions.
You must put the integration constant behind it already so that the student does not have to do that.



Herleid de volgende onbepaalde integraal.

$\int x^{2}\ \mathrm{d}x =$ $+C$

Figure 1.127: Computation of antiderivatives

In the *Algorithm* you have programmed the following:

```
$integraal=maple("Int(x^2,x)");
$intdisplay=maple("printf(MathML[ExportPresentation]
($integraal))");
$antw=maple("value($integraal)");
```

HINT: Note that you phrase the question with the capitalized inert command (`Int`). With the maple command `value`, the value of this indefinite integral can be generated (the integration constant is set to 0 by maple).
The formula to be typed by the student (in the setting *Text entry only*) can be easily matched by differentiating the student's answer with the *Grading Code*:

```
evalb((diff($antw,x)-diff($RESPONSE,x))=0)
and evalb(StringTools[Search]("int","$RESPONSE")=0);
```

With this *Grading Code* you make sure that the student does not use the maple command.

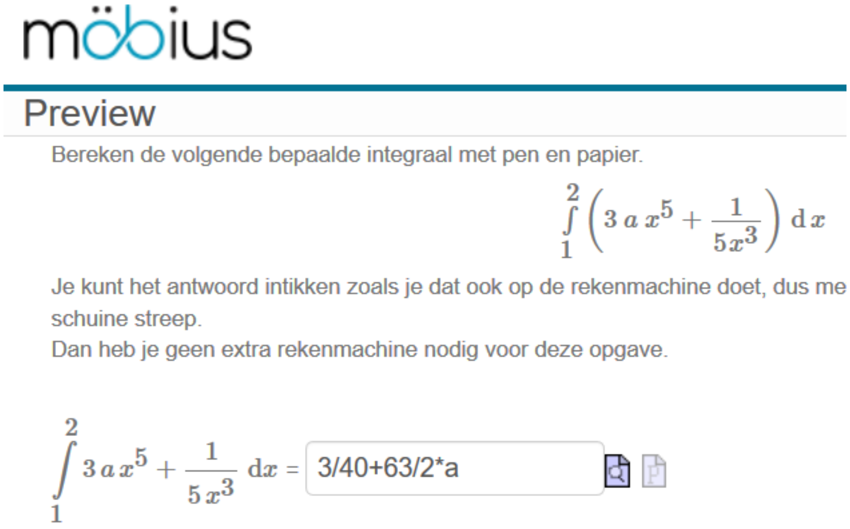
If you opt for the Editor, with the setting *Symbol entry only*, then this last restriction `and evalb(StringTools[Search]("int","$RESPONSE")=0)` is very important, because withhout this restriction the student can take the integral from the palettes of the Editor to get a full grading.
Because an arbitrary integration constant C still has to be given with indefinite integration, we match the derivative of the correct answer with the derivative of the student's answer. The integration constant then doesn't matter and if the student still adds a constant, it doesn't matter.

1.4.20.2 Definite integration and different notations

If the student needs to calculate the outcome of the definite integral, you can simply match the correct answer to the student's answer with the *Grading Code*:

```
evalb(simplify($antwoord-$RESPONSE)=0);
```

The following is about how the integral is neatly displayed on the screen.



Preview

Bereken de volgende bepaalde integraal met pen en papier.

$$\int_{1}^{2} \left(3\,a\,x^{5}+\frac{1}{5x^{3}}\right)\mathrm{d}x$$

Je kunt het antwoord intikken zoals je dat ook op de rekenmachine doet, dus me schuine streep.
Dan heb je geen extra rekenmachine nodig voor deze opgave.

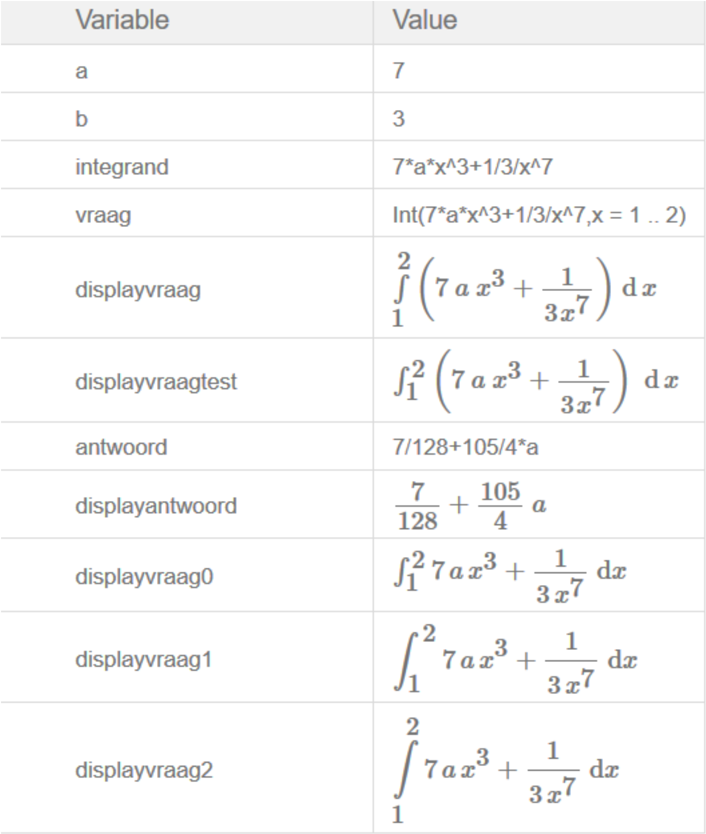
$\int_{1}^{2} 3\,a\,x^{5}+\frac{1}{5\,x^{3}}\,\mathrm{d}x =$ 3/40+63/2*a

Figure 1.128: De bepaalde integraal berekenen

In the *Algorithm* you can program the following and below in the figure the result:

```
$a=range(1,9);
$b=range(1,5);
$integrand=maple("a*$a*x^($b)+1/($b*x^($a))");
$vraag=maple("Int($integrand,x=1..2)");
$displayvraag=maple(" printf(MathML[ExportPresentation]($vraag))
 ");
$displayvraagtest=maple("printf(MathML[ExportPresentation]
($vraag))");
$antwoord=maple("simplify(value($vraag))");
$displayantwoord=maple("printf(MathML[ExportPresentation]
($antwoord))");
$displayvraag0="\( \int_{1}^{2} $a\,a\,x^{$b}+\frac{1}{$b\,x^{$a}}
 \ \mathrm{d}x\) ";
```

```
$displayvraag1="\( \displaystyle{\int_{1}^{2} $a\,a\,x^{$b}+
\frac{1}{$b\,x^{$a}} \ \mathrm{d}x}\) ";
$displayvraag2="\( \displaystyle{\int\limits_{1}^{2} $a\,a
\,x^{$b}+\frac{1}{$b\,x^{$a}} \ \mathrm{d}x}\) ";
```

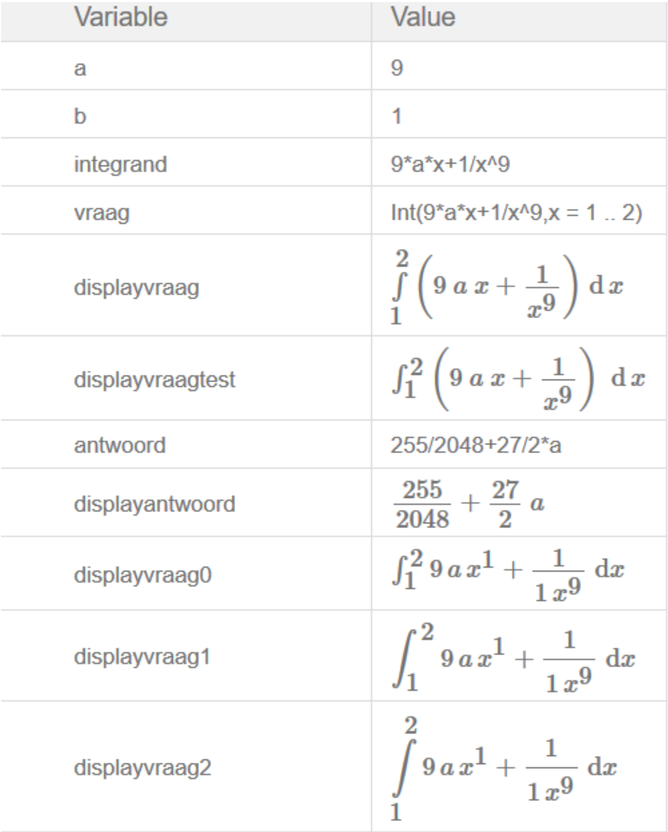


Variable	Value
a	7
b	3
integrand	7*a*x^3+1/3/x^7
vraag	Int(7*a*x^3+1/3/x^7,x = 1 .. 2)
displayvraag	$\int\limits_{1}^{2} \left(7\,a\,x^{3}+\frac{1}{3x^{7}}\right) \mathrm{d}\,x$
displayvraagtest	$\int_{1}^{2} \left(7\,a\,x^{3}+\frac{1}{3x^{7}}\right) \mathrm{d}\,x$
antwoord	7/128+105/4*a
displayantwoord	$\frac{7}{128}+\frac{105}{4}\,a$
displayvraag0	$\int_{1}^{2} 7\,a\,x^{3}+\frac{1}{3\,x^{7}}\ \mathrm{d}x$
displayvraag1	$\displaystyle\int_{1}^{2} 7\,a\,x^{3}+\frac{1}{3\,x^{7}}\ \mathrm{d}x$
displayvraag2	$\displaystyle\int\limits_{1}^{2} 7\,a\,x^{3}+\frac{1}{3\,x^{7}}\ \mathrm{d}x$

Figure 1.129: Verschillende schrijfwijzen van de bepaalde integraal

The integral as it appears on the screen in the question text can be generated in various ways.
The easiest way is to do that with maple. Note the spaces next to the quotes in the *Algorithm* for the variable $displayquestion.
If you place the quotes tightly against the command, the integral sign is smaller and the boundaries are not nicely above and below the integral sign.
So pay attention to the spaces in the quotes.
A disadvantage of generating the mathml code by maple is that there are brackets around the integrand and this is not always necessary or desirable.
Furthermore, the disadvantage is also that there is a space between d and x where it should not be in fact because the d is an operator.

HINT: It is nicer to get the definite integral on the screen with the help of LaTeX. You can then choose whether or not to include brackets around the integrand.
You can also make sure that there is no space between the d and the x. Furthermore, the size of the integral sign can still be controlled and the boundaries can be nicely above and below it. A number of possibilities have been suggested with the variables $displayquestion0 to 2.
A disadvantage of LaTeX is that if the variables have the value 1, you will also see a 1 in the formula. It is better not to include the 1 as a variable in the range for the randomization of the integrand. Even with negative variables it becomes difficult to work with LaTeX because then you may see +- in succession.



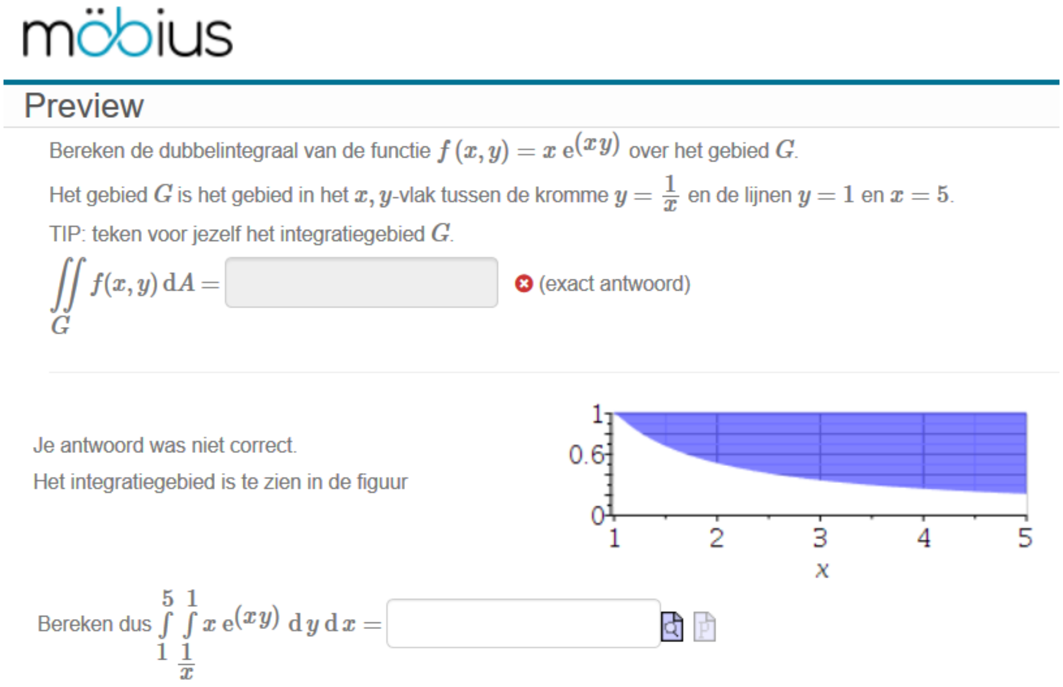
Variable	Value
a	9
b	1
integrand	9*a*x+1/x^9
vraag	Int(9*a*x+1/x^9,x = 1 .. 2)
displayvraag	$\int\limits_{1}^{2}\left(9\,a\,x+\frac{1}{x^{9}}\right)\mathrm{d}\,x$
displayvraagtest	$\int_{1}^{2}\left(9\,a\,x+\frac{1}{x^{9}}\right)\;\mathrm{d}\,x$
antwoord	255/2048+27/2*a
displayantwoord	$\frac{255}{2048}+\frac{27}{2}\,a$
displayvraag0	$\int_{1}^{2} 9\,a\,x^{1}+\frac{1}{1\,x^{9}}\,\mathrm{d}x$
displayvraag1	$\displaystyle\int_{1}^{2} 9\,a\,x^{1}+\frac{1}{1\,x^{9}}\,\mathrm{d}x$
displayvraag2	$\displaystyle\int\limits_{1}^{2} 9\,a\,x^{1}+\frac{1}{1\,x^{9}}\,\mathrm{d}x$

Figure 1.130: LaTeX en de integrand met variabele 1

HINT: The numerical question type can also be used for these kinds of questions if the answer consists of a number without variables. In that case, a tolerance can also be added if you want to accept rounding.

1.4.20.3 Multiple integrals

In this example, the same procedure is used as in the example of the simple definite integral. If a single number comes out without variables, you can switch to the numeric question type and opt for the exact answer or even accept rounding.



Preview

Bereken de dubbelintegraal van de functie $f(x,y) = x\,\mathrm{e}^{(xy)}$ over het gebied G.

Het gebied G is het gebied in het x, y-vlak tussen de kromme $y = \frac{1}{x}$ en de lijnen $y = 1$ en $x = 5$.

TIP: teken voor jezelf het integratiegebied G.

$\iint\limits_{G} f(x,y)\,\mathrm{d}A =$ ☐ ✖ (exact antwoord)

Je antwoord was niet correct.

Het integratiegebied is te zien in de figuur

Bereken dus $\int_{1}^{5}\int_{\frac{1}{x}}^{1} x\,\mathrm{e}^{(xy)}\,\mathrm{d}y\,\mathrm{d}x =$ ☐

Figure 1.131: Dubbelintegraal berekenen

The first integral is prepared in the *Algorithm* using LaTeX:

```
$formule="\( \displaystyle{\iint\limits_{G}f(x,y)\, \mathrm{d}A}\) ";
```

The correct answer is also prepared in the *Algorithm* with variable function and variable limits:

```
$Int=maple("Int(Int($f,y=$l1..$u1),x=$l2..$u2)");
$antw=maple("value($Int)");
```

For the *Grading Code* you have the following simple code:

```
evalb(($antw)-($RESPONSE)=0)
and evalb(StringTools[Search]("int","$RESPONSE")=0);
```

Or perhaps with simplification:

```
evalb(simplify(($antw)-($RESPONSE))=0)
and evalb(StringTools[Search]("int","$RESPONSE")=0);
```

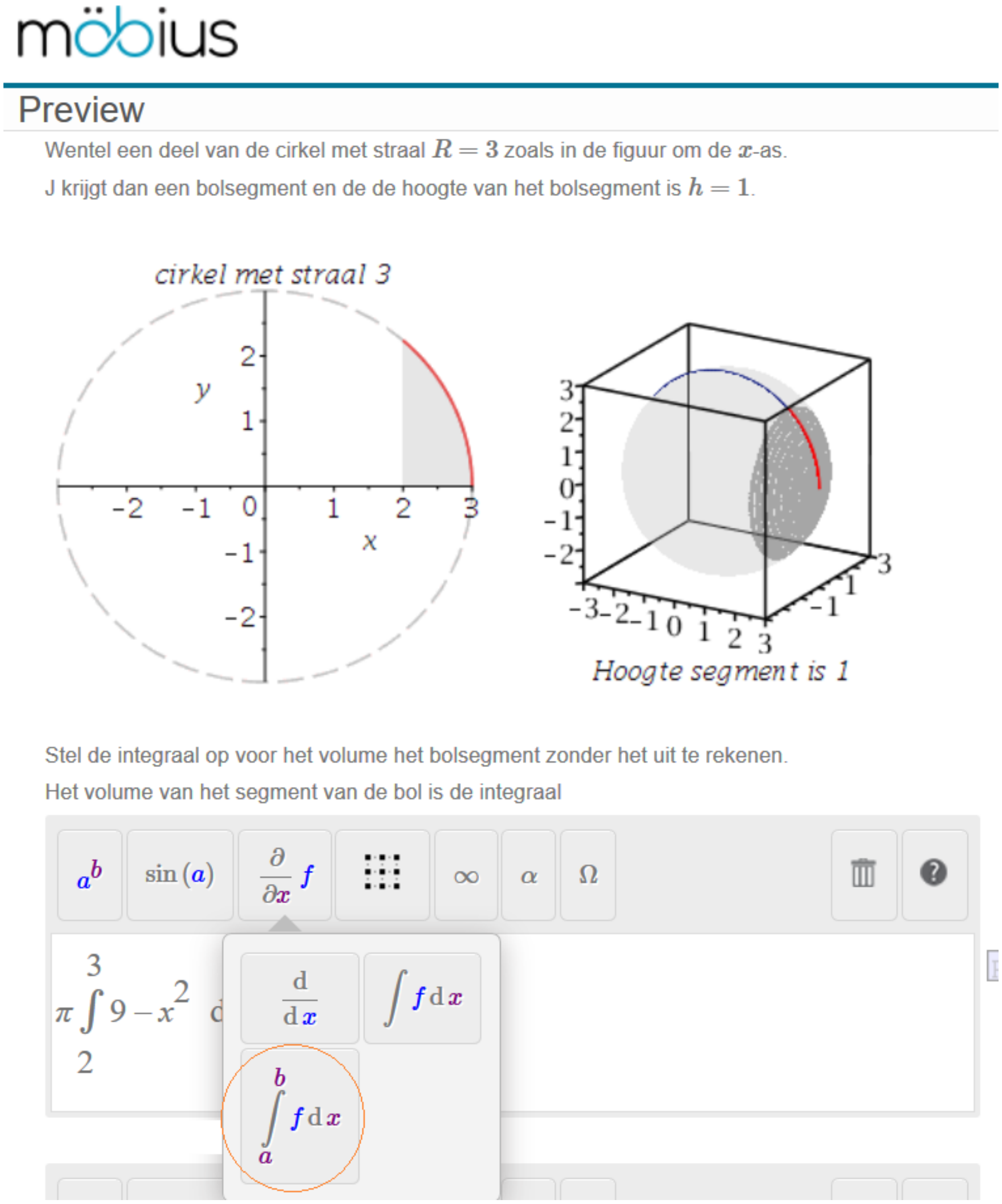
HINT: For the stepwise feed back you can prepare the steps by simply change the inner integral from `Int` to `int` with a lower case i.

```
$Inthalf=maple("Int(int($f,y=$l1..$u1),x=$l2..$u2)");
$inthalfdisplay=maple(" printf(MathML[ExportPresentation]
($Inthalf)) ");
```

1.4.20.4 The student enters the integral

If you want the student to write the integral and also type it in as an integral, you can match the **value** of the integral entered with that of the correct integral.

Working with the editor entering integrals



Wentel een deel van de cirkel met straal $R = 3$ zoals in de figuur om de x-as.

J krijgt dan een bolsegment en de de hoogte van het bolsegment is $h = 1$.

Stel de integraal op voor het volume het bolsegment zonder het uit te rekenen.

Het volume van het segment van de bol is de integraal

Figure 1.132: Student enters an integral

First prepare the *Algortihm* as follows:

```
$kleur=switch(rint(8),"red","yellow","khaki","gray","green","magenta","cyan","turquoise");
$R=range(3,5);
$r=range(1,2);
$h=$R-$r;
$f=maple("sqrt($R^2-x^2)");
$fdisplay="\(y=\sqrt{$R^2-x^2}\)";
$volume=maple("simplify(Student[Calculus1][VolumeOfRevolution]($f,x=$r..
$R,output=integral))");
$volumedisplay=maple(" printf(MathML[ExportPresentation]($volume)) ");
$antw=maple("value($volume)");
$antw1=maple("subs(Pi=pi,$antw)");
$antwdisplay=maple("printf(MathML[ExportPresentation]($antw))");
```

```
$vb=maple("printf(MathML[ExportPresentation](Int(f,x=a..b)) )");
$plot=plotmaple("plots[display]([plots[implicitplot](x^2+y^2=$R^2,x=-$R..$R,y=-
$R..$R,color=grey,linestyle=dash,title=`cirkel met straal $R`),
plot(sqrt($R^2-x^2),x=$r..$R,color=red),
plot(sqrt($R^2-x^2),x=$r..$R,color=$kleur,filled=true,transparency=0.6)]),
plotoptions='height=250, width=250'");
$plotbolkapje=plotmaple("plots[display]([Student[Calculus1]
[VolumeOfRevolution](sqrt($R^2-x^2),x=-$R..
$R,output=plot,volumeoptions=[color=grey,transparency=0.9],caption=`Hoogte segment
 is $h`,axes=boxed),
Student[Calculus1][VolumeOfRevolution](sqrt($R^2-
x^2),x=$r..$R,output=plot,volumeoptions=[color=
$kleur,transparency=0.5],functionoptions=[thickness=3,color=red],title=`bol met
 straal $R`,caption=``,axes=boxed)],orientation=[-60,70]),
plotoptions='height=250, width=250'");
$cirkeldisplay=maple("printf(MathML[ExportPresentation](x^2+y^2=R^2) )");
$ydisplay=maple("printf(MathML[ExportPresentation](y=sqrt(R^2-x^2)) )");
$dVdisplay="\(\pi\,y^2\,\text{d}x\)";
```

If the value of the correct integral in the *Algorithm* is programmed as $antw (the **value** of the integral), you can use the following *Grading Code*:

```
evalb(simplify($antw-(value($RESPONSE)))=0) and
 evalb(StringTools[Search]("nt","$RESPONSE")>0);
```

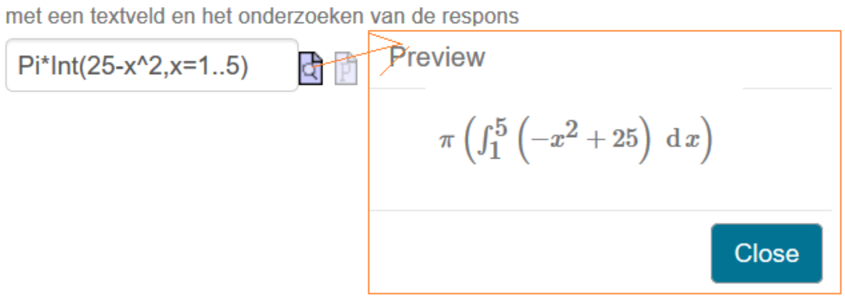
In this way the student can also use a different formulation of the integral that has the same **value** and you demand that an integral has been entered.
With the setting *Symbol entry only*, the students integral is always translated in the sring with the maple command "int".
(The character π is translated by the editor always in Pi.)

Working with a text field entering an integral

Basicly the same *Grading Code* you can use for this situation.

```
evalb(simplify(subs(pi=Pi,$antw-(value($RESPONSE))))=0) and
 evalb(StringTools[Search]("nt","$RESPONSE")>0);
```

The fact that "int" is not searched for in the string of the response here, but for "nt" has to do with the fact that this same *Grading Code* can also be used in the *Text entry only* situation as well in the *Symbol entry only* situation with the editor. The student can type int or int there. Both will be approved with the same *Grading Code*.
But with the additional option with `subs(pi=Pi)` the character π is always meant to be Pi.



met een textveld en het onderzoeken van de respons

Pi*Int(25-x^2,x=1..5)

Preview

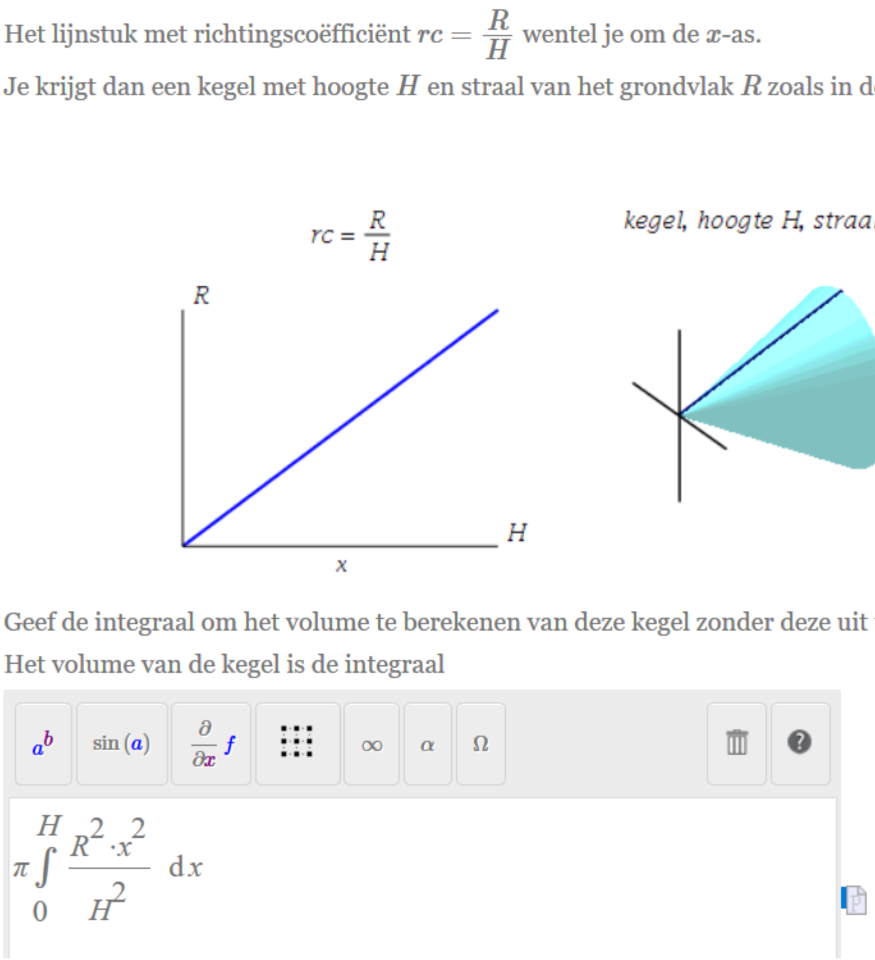
$\pi\left(\int_1^5 \left(-x^2+25\right)\,\mathrm{d}x\right)$

Figure 1.133: Entering an integral in text mode

Entering an integral with parameters

In case the integrand and/or the boundaries contain parameters, it is not always said that you can match the entered integral of the student with the correct integral by calculating the **value** of the integral.

The following example shows you how to act.



Het lijnstuk met richtingscoëfficiënt $rc = \frac{R}{H}$ wentel je om de x-as.

Je krijgt dan een kegel met hoogte H en straal van het grondvlak R zoals in de figuur.

Geef de integraal om het volume te berekenen van deze kegel zonder deze uit te rekenen.

Het volume van de kegel is de integraal

$$\pi \int_0^H \frac{R^2 \cdot x^2}{H^2}\, dx$$

Figure 1.134: Inhoud kegel met integreren

The *Algorithm* is as follows:

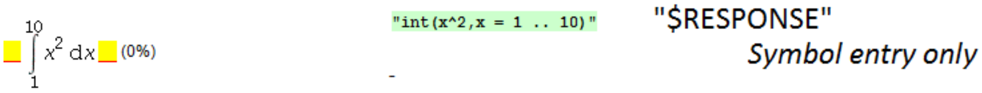
```
$kleur=switch(rint(8),"red","yellow","khaki","gray","green","magenta","cyan","turquoise");
$R=range(2,5);
$H=$R+range(1,5);
$f=maple("$R/$H*x");
$F=maple("R/H*x");
$volume=maple("Student[Calculus1][VolumeOfRevolution]
($F,x=0..H,output=integral)");
$volumedisplay=maple(" printf(MathML[ExportPresentation]($volume)) ");
$antw=maple("value($volume)");
$antw1=maple("subs(Pi=pi,$antw)");
$antwdisplay=maple("printf(MathML[ExportPresentation]($antw))");
$vb=maple("printf(MathML[ExportPresentation](Int(f,x=a..b)) )");
$plot=plotmaple("plots[display]({plot($f,x=0..
$H,color=blue,thickness=2,tickmarks=[0,0],title=typeset(rc=R/
H),scaling=constrained),plots[textplot]([[$H,0,H],[0,
$R,R]],align={right,above})}),plotoptions='height=250, width=250'");
$plotkegel=plotmaple("Student[Calculus1][VolumeOfRevolution]
($f,x=0..$H,output=plot,tickmarks=[0,0,0],volumeoptions=[color=
$kleur,transparency=0.5],functionoptions=[thickness=3],title=`kegel, hoogte H,
 straal R`,caption = ``),plotoptions='height=250, width=250'");
$dVdisplay="\(\pi\,y^2\,\text{d}x\)";
```

The *Grading Code* is as follows: The program needs to know that the parameters are positive in order to calculate the integral and do the simplification to get 0. (You can also use something similar for roots and logarithms, for example).

```
assume(R>0): assume(H>0): evalb(simplify($antw-
(value($RESPONSE)))=0) and evalb(StringTools[Search]
("nt","$RESPONSE")>0);
```

The analysis of the students response

In the following figures you see the student's input on the left, then the response in the middle column and the symbol or text setting in the right column and whether the response or the string of the student's response is being looked at in different situations



$\int_{1}^{10} x^2\, dx$ (0%)

"int(x^2,x = 1 .. 10)"

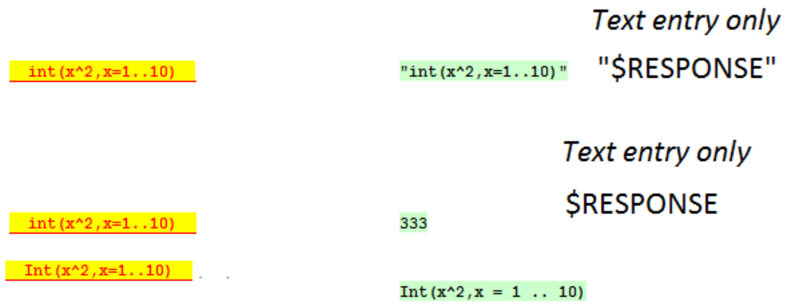
"$RESPONSE"

Symbol entry only

Figure 1.135: The analysis of the students respons with an integral in Symbol mode

However, the $RESPONSE is the **value** of the integral that maple computes immediately and matches it in the *Grading Code*.

It is the same in case of *Symbol mode* or *Text entry only*. The student types in the integral (with int) and that can also be checked using *StringTools* and furthermore the **value** of the integral is used again to match in the *Grading Code*.



Text entry only

int(x^2,x=1..10) "int(x^2,x=1..10)" "$RESPONSE"

Text entry only

$RESPONSE

int(x^2,x=1..10) 333

Int(x^2,x=1..10) Int(x^2,x = 1 .. 10)

Figure 1.136: The analysis of the students respons with an integral in Text mode

In this figure you can see that entering an integral in the answer field with *Text entry only* in the string of the answer translates literally to what the student enters. In the second field, the pure response ($RESPONSE) is directly translated to the value of the integral calculated by maple. In the third answer field, the student types "Int" with a capital character. In the latter case, the response is no longer the value of the integral but the integral itself. In that case you can also test the integrand separately and separately the variable with the boundaries. In fact, the *Grading Code* always matches with the $RESPONSE, although there are plenty of possibilities to include something in the *Grading Code* that can be traced back to the string of the answer.

HINT: In both cases the students input "int" or "Int" in the text field will be seen in de *Preview* as the integral itself and not as the value.

HINT: If you require your students to capitalize "Int" in their input into the text field, you can do two things in the *Grading Code*:

```
evalb(simplify($antw-value($RESPONSE))=0) and
 evalb(StringTools[Search]("Int","$RESPONSE")>0);
```

But you can grade the integrand and variable and bounds separately with operands if the variable $antw in the *Algorithm* is prepared as the correct integral with the following *Grading Code*:

```
is((op(1,$antw))=(op(1,$RESPONSE)) )
and is( (op(2,$antw))=(op(2,$RESPONSE)) )
and evalb(StringTools[Search]("Int","$RESPONSE")>0);
```

HINT: You can prepare a few things in the *Algorithm*.
Below you can see what Maple does with the inert integral when its operands are requested.

```
> op(1,Int(x^2,x=0..10));
```

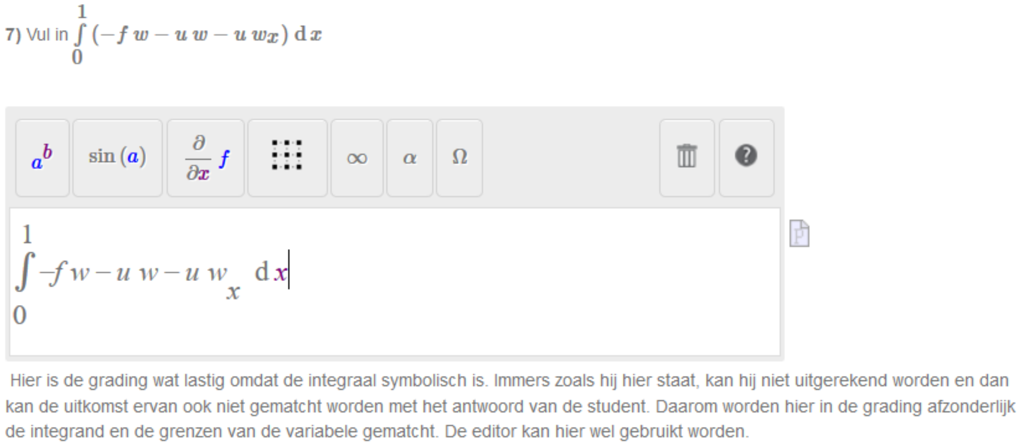
$$x^2 \tag{1.81}$$

```
> op(2,Int(x^2,x=0..10));
```

$$x = 0\,..10 \tag{1.82}$$

Symbolic integrals

In the following example, the student must also enter an integral. However, this integral is symbolic and in that case it is difficult to match its **value** with that of the student if the value of the integral cannot be calculated. So here you can work with operands.



7) Vul in $\int_0^1 (-f\,w - u\,w - u\,w_x)\,dx$

$\int_0^1 -f\,w - u\,w - u\,w_x \;\mathrm{d}x$

Hier is de grading wat lastig omdat de integraal symbolisch is. Immers zoals hij hier staat, kan hij niet uitgerekend worden en dan kan de uitkomst ervan ook niet gematcht worden met het antwoord van de student. Daarom worden hier in de grading afzonderlijk de integrand en de grenzen van de variabele gematcht. De editor kan hier wel gebruikt worden.

Figure 1.137: Invoeren van een integraal symbolisch

You program the correct answer in the *Algorithm* with

```
$antw7=maple("Int(-w[x]*u-w*u-w*f,x=0..1)");
$displayantw7=maple(" printf(MathML[ExportPresentation]($antw7))
 ");
```

Operands of an integral

Here you can test with maple whether this symbolic integral has a value or not.
You see an example where it is not possible to calculate the value of the integral.
The x in the subscript will throw a spanner in the works and the integral will be returned with no effect.

```
> int(-w*u-w*u-w*f,x=0..1);
```

$$-w\,f - 2\,w\,u \tag{1.83}$$

```
> int(-w[x]*u-w*u-w*f,x=0..1);
```

$$\int_0^1 \left(-w f - w u - w_x u\right) \mathrm{d}x \tag{1.84}$$

```
> op(int(-w[x]*u-w*u-w*f,x=0..1));
```

$$-w f - w u - w_x u, x = 0 ..1 \tag{1.85}$$

When querying the operands of an integral, use the inert command Int, so that the integrand and the variable with the bounds are neatly displayed. However, here it would not have mattered if the value of the integral cannot be calculated with int (because the integral is then simply returned). So there are two possibilities for matching integrals. You compare the **value** of the integrals if they can be calculated or you compare the **operands** of the integrals.

In the example (*Figure 1.137 (page 116)*) the grading is a bit tricky because the integral is symbolic. After all, as it stands here, it cannot be calculated and then its value cannot be matched with the value of the student's answer. Therefore, here in the *Grading Code*, the integrand and the bounds of the variable are matched separately.

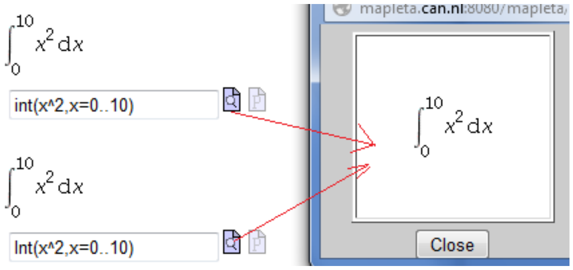
There is no problem to offer the Editor here with the *Symbol entry only* setting, but also if the student knows the Maple syntax of integrals and subscripts (with square brackets), you can set *Text entry only* with the same *Grading Code*:

```
is( (op(1,$antw7))=(op(1,$RESPONSE)) )
and is( (op(2,$antw7))=(op(2,$RESPONSE)) )
and evalb(StringTools[Search]("nt","$RESPONSE")>0);
```

In the latter case, it may still be necessary to check whether "Int" or "int" occurs in the student's answer string.
This means that you match the first operand (the integrand) of the student's answer with that of the correct answer and likewise the second operand: the integration variable with the correct bounds.

HINT: This way of doing things only works if the integral cannot be calculated. After all, if you have set the settings to *Symbol entry only*, the integral from the palette is translated to the non-inert command "int". So if the integral could be calculated, then it is not possible to work with the operands of the integral, such as the integrand and the variable with the limits.

HINT: If you require your students to explicitly enter the integral and want to check whether both the integrand and the variable with any limits are checked separately, you have no choice but to set the settings to *Text entry only*. The student must then definitely use the inert command "Int" to type in the integral. The *Preview* then gives no difference between "Int" and "int" but the student's response is translated to the value of the integral in the case of "int" and in the case of "Int" to the pure integral.



$\int_0^{10} x^2 \, dx$

int(x^2,x=0..10)

$\int_0^{10} x^2 \, dx$

Int(x^2,x=0..10)

Figure 1.138: Difference with input Int and int and no difference in the Preview

1.4.21 Matrices and Vectors

1.4.21.1 Introduction

With Matrices and Vectors the *Grading Code* is not obvious. You need to know a little more about this. Simple equating is not possible here.
But otherwise with the settings of *Maple syntax* it is wise to opt for *Symbol entry only* to let the students enter the Matrix or Vector with the help of an Editor.
If **two matrices or two vectors are equal** to each other, then **all corresponding elements are equal** to each other. This is checked by Maple with the following *Grading Code*:

```
LinearAlgebra[Equal]($RESPONSE,$antw);
```

Here $RESPONSE is always by definition the response of the student and $antw is the answer prepared in the *Algorithm*. This *Grading Code* can be used not only for Matrices but also for Vectors.

HINT: You can also require that some individual elements of a matrix (or vector) have certain values with the following *Grading Code*:

```
is ($RESPONSE[2,1]=0) and is($RESPONSE[3,1]=0);
```

Here it is checked whether the element of the second row and the first column and the element of the third row and the first column of the response are equal to 0.

HINT: If you are using decimal numbers (two decimals for example), the following *Grading Code* may be useful to use:

```
LinearAlgebra[Equal]
(map(round,100*($RESPONSE)),map(round,100*($antw)));
```

This ensures that all elements of the matrix or vector are first multiplied by 100 and then rounded. It is rather close that with Equal from the LinearAlgebra package the Matrices and Vectors also exactly match with regard to the elements.

HINT: When testing Matrices and Vectors, students should be properly instructed especially in choosing the correct dimensions of the Matrix. In the field of the Equation Editor, there is a menu in which the student chooses, for example, a square 2×2 matrix. A matrix will then appear in which the numbers must be adjusted. See also the figure below.

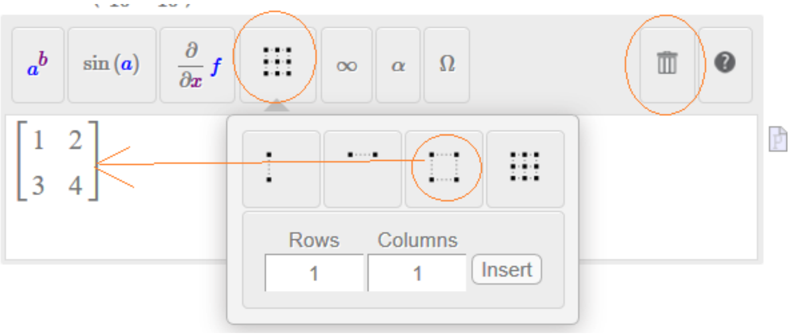

a^b
sin (a)
$\frac{\partial}{\partial x} f$
∞
α
Ω
$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
Rows
Columns
1
1
Insert

Figure 1.139: Entering a matrix in the answer field with editor

HINT: If students want to improve a lot in the formula entered, have them clean everything with the bin button in the menu and start over.
When matching matrices and vectors, it is difficult to apply tolerances. If possible, have the student type in the exact answer or indicate clearly how the rounding should be and prepare the correct answer in that way in the *Algorithm*.

HINT: Note that a column vector or a row vector is not considered as a Matrix by the system. The correct answer must therefore be prepared as a column vector (in the *Algorithm*) to match. So note that when using the Editor that you offer in the settings *Maple syntax with Symbol entry only* that if the student opts for a matrix of 1 column, the system will always see it as Vector! so when matching, the prepared answer must also be defined as Vector! Likewise for a row vector.

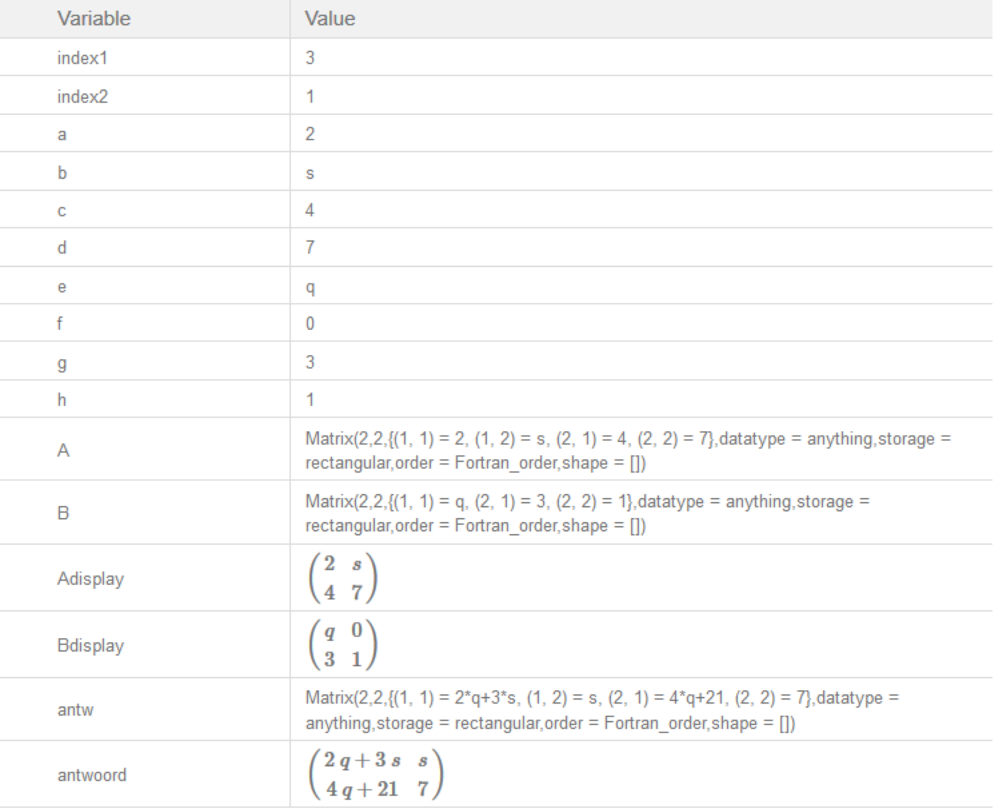
HINT: Because you will in fact always use the Editor for questions with matrices and vectors, it is useful to always program the correct answer (for *Answer*) with mathml, so that the student always sees the two-dimensional presentation in the feedback for the correct answer:

```
printf(MathML[ExportPresentation]($antw));
```

1.4.21.2 Defining a matrix and a vector

In the *Algorithm* you can already define and calculate a matrix and a vector.

```
$index1=range(0,3);
$index2=range(0,3);
$a=range(0,9);
$b=switch($index1,p,q,r,s);
$c=range(0,9);
$d=range(0,9);
$e=switch($index2,p,q,r,s);
$f=range(0,9);
$g=range(0,9);
$h=range(0,9);
$A=maple("Matrix([[$a,$b],[$c,$d]])");
$B=maple("Matrix([[$e,$f],[$g,$h]])");
$Adisplay=maple("printf(MathML:-ExportPresentation($A))");
$Bdisplay=maple("printf(MathML:-ExportPresentation($B))");
$antw=maple("$A . $B");
$antwoord = maple("printf(MathML:-ExportPresentation($antw))");
```



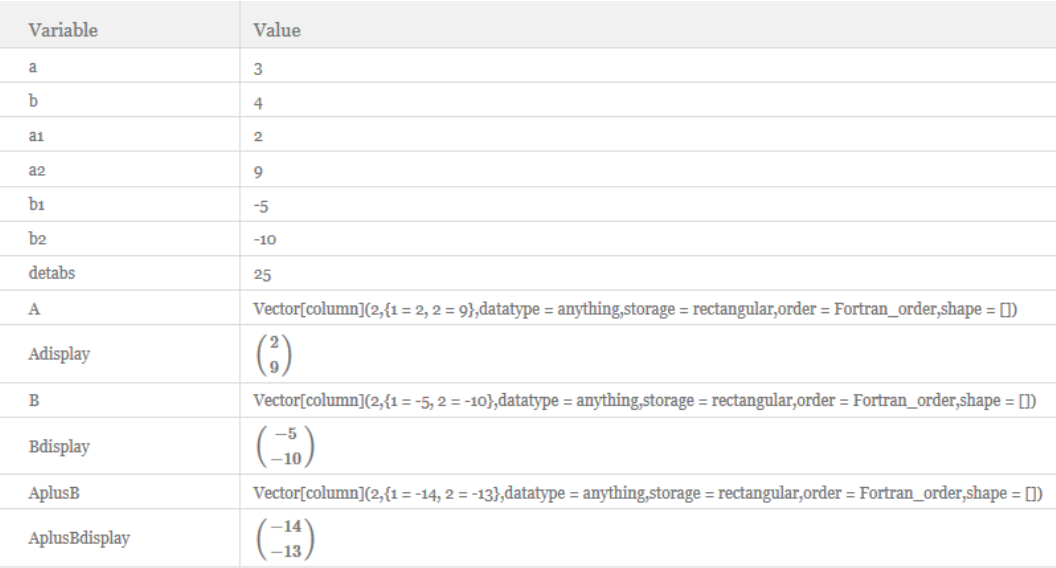
Variable	Value
index1	3
index2	1
a	2
b	s
c	4
d	7
e	q
f	0
g	3
h	1
A	Matrix(2,2,{(1, 1) = 2, (1, 2) = s, (2, 1) = 4, (2, 2) = 7},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
B	Matrix(2,2,{(1, 1) = q, (2, 1) = 3, (2, 2) = 1},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Adisplay	$\begin{pmatrix} 2 & s \\ 4 & 7 \end{pmatrix}$
Bdisplay	$\begin{pmatrix} q & 0 \\ 3 & 1 \end{pmatrix}$
antw	Matrix(2,2,{(1, 1) = 2*q+3*s, (1, 2) = s, (2, 1) = 4*q+21, (2, 2) = 7},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
antwoord	$\begin{pmatrix} 2q+3s & s \\ 4q+21 & 7 \end{pmatrix}$

Figure 1.140: Thee matrix in the Algorithm

In the above *Algorithm* you can see how the matrix A is stored. This allows you to simply calculate within maple with the well-known commands for matrices. For screen readability, write out the matrix using mathml code you generate with Maple.

A vector in the *Algorithm* is also easy to define.

```
$a=range(2,5);
$b=range(2,5);
$a1=range(-10,10);
$a2=range(-10,10);
$b1=range(-10,10);
$b2=range(-10,10);
$detabs=maple("abs($a1*($b2)-($b1)*($a2))");
condition:gt($detabs,0);
$A=maple("Vector([$a1,$a2])");
$Adisplay=maple("printf(MathML[ExportPresentation]($A))");
$B=maple("<$b1,$b2>");
$Bdisplay=maple("printf(MathML[ExportPresentation]($B))");
$AplusB=maple("$a*$A+$b*$B");
$AplusBdisplay=maple("printf(MathML[ExportPresentation]
($AplusB))");
```



Variable	Value
a	3
b	4
a1	2
a2	9
b1	-5
b2	-10
detabs	25
A	Vector[column](2,{1 = 2, 2 = 9},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Adisplay	$\begin{pmatrix} 2 \\ 9 \end{pmatrix}$
B	Vector[column](2,{1 = -5, 2 = -10},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
Bdisplay	$\begin{pmatrix} -5 \\ -10 \end{pmatrix}$
AplusB	Vector[column](2,{1 = -14, 2 = -13},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
AplusBdisplay	$\begin{pmatrix} -14 \\ -13 \end{pmatrix}$

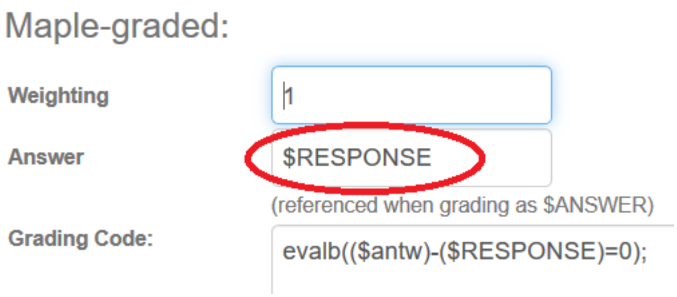
Figure 1.141: The definition of a vector in the Algorithm

In the above *Algorithm* you can see two different ways to define a (column) vector: $A and $B. It comes down to exactly the same thing and the $B way is less typing.

1.4.21.3 Analysis of the input

In the *Grading Code* you will on the one hand use the student's $RESPONSE to match with the correct answer, on the other hand it is important to be able to use the string of the student's answer to see exactly what the student has typed in. With *StringTools* you can then further program how you want the student's input by looking at the string of the student's response: "$RESPONSE".

In the figure below you can see in some examples of input in the Editor how this is translated to a string.
You do this by entering $REPSONSE (or "$RESPONSE") in the *Answer* section of the *Maple graded* answer field if you want feedback on what you've entered.



Maple-graded:

Weighting: 1

Answer: $RESPONSE

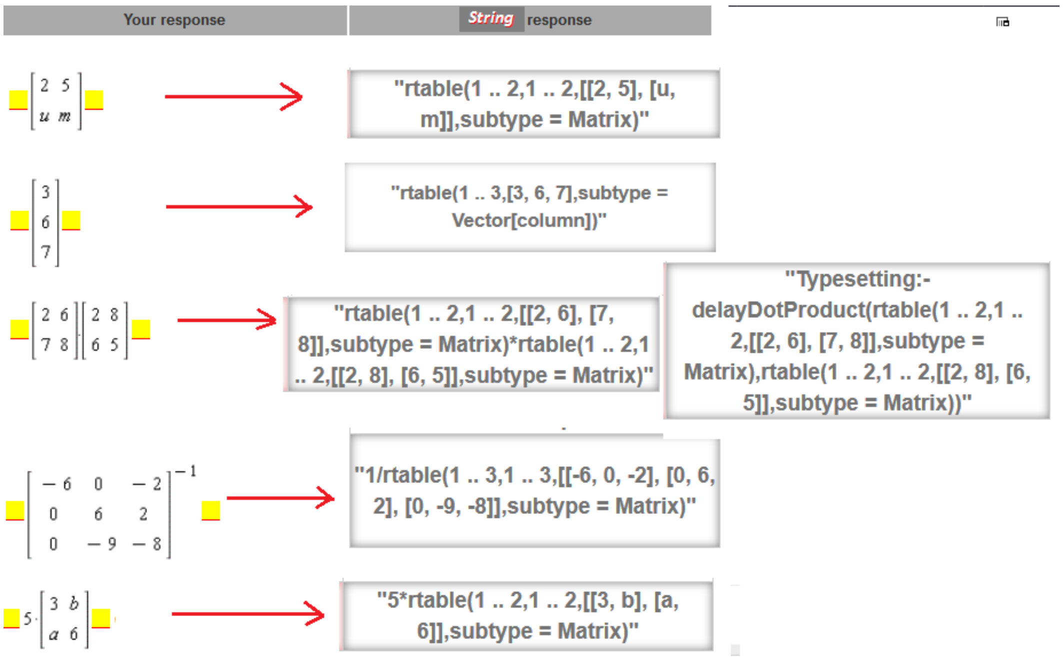
(referenced when grading as $ANSWER)

Grading Code:

```
evalb(($antw)-($RESPONSE)=0);
```

Figure 1.142: Analysing the students response

In the *Preview* of the question you fill in something and click on *Grade* after which you see the translation of what you have entered.



Your response	String response
$\begin{bmatrix}2&5\\u&m\end{bmatrix}$	"rtable(1 .. 2,1 .. 2,[[2, 5], [u, m]],subtype = Matrix)"
$\begin{bmatrix}3\\6\\7\end{bmatrix}$	"rtable(1 .. 3,[3, 6, 7],subtype = Vector[column])"
$\begin{bmatrix}2&6\\7&8\end{bmatrix}\begin{bmatrix}2&8\\6&5\end{bmatrix}$	"rtable(1 .. 2,1 .. 2,[[2, 6], [7, 8]],subtype = Matrix)*rtable(1 .. 2,1 .. 2,[[2, 8], [6, 5]],subtype = Matrix)" "Typesetting:-delayDotProduct(rtable(1 .. 2,1 .. 2,[[2, 6], [7, 8]],subtype = Matrix),rtable(1 .. 2,1 .. 2,[[2, 8], [6, 5]],subtype = Matrix))"
$\begin{bmatrix}-6&0&-2\\0&6&2\\0&-9&-8\end{bmatrix}^{-1}$	"1/rtable(1 .. 3,1 .. 3,[[-6, 0, -2], [0, 6, 2], [0, -9, -8]],subtype = Matrix)"
$5\cdot\begin{bmatrix}3&b\\a&6\end{bmatrix}$	"5*rtable(1 .. 2,1 .. 2,[[3, b], [a, 6]],subtype = Matrix)"

Figure 1.143: How the input is translated to a string

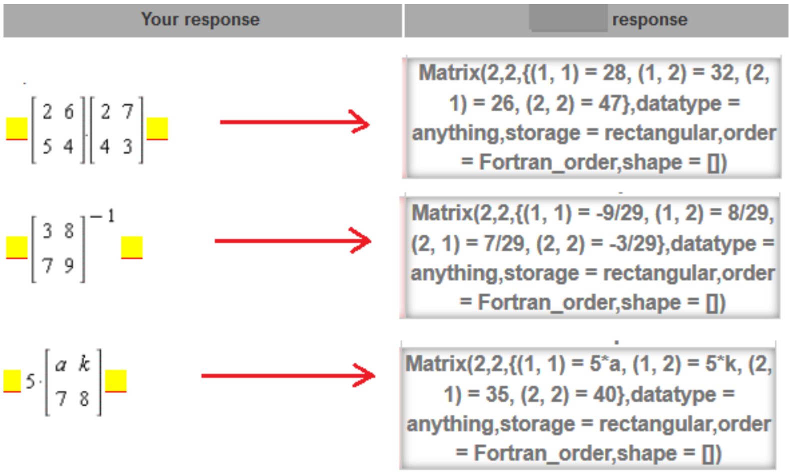
The figure above shows how the input is translated to a string.

- How is a 2×2-matrix translated into a string.
- A matrix with one column entered is translated to a Vector[column].
- If the student enters two matrices with an asterix between them, it is **not** considered a matrix multiplication. The translation to the string is simply "rtable() * rtable()" . It will not be graded correctly.
  But with a (regular) dot in between (from the keyboard), these two matrices are regarded as the dot product (DotProduct) of the two matrices and is correct.
  In that case you can, for example, check with *StringTools* whether DotProduct occurs in the string of the response.
  For example, if you absolutely want the student to multiply the two matrices himself, only one matrix may be entered.
  For example, you check with the *Grading Code*:

```
LinearAlgebra:-Equal( simplify( $RESPONSE ), simplify($antw))
and evalb(0=StringTools[Search]("DotProduct","$RESPONSE"));
```

- At the inverse, the student has typed the matrix to the power of -1. This would be graded correct if you don't take your measures with *StringTools* with the following *Grading Code*:
```
LinearAlgebra:-Equal( simplify( $RESPONSE ), simplify($antw))
and evalb(0=StringTools[Search]("/rtable","$RESPONSE"));
```
- If the student has to get the factor 5 before the matrix, you can use this multiplication to check whether there is an asterix in the string of the response. The student types an asterix or a space or nothing. The system always translates that with an asterix in Symbol mode.
```
LinearAlgebra:-Equal( simplify( $RESPONSE ), simplify($antw))
and evalb(0=StringTools[Search]("*","$RESPONSE"));
```

The figure below shows that the $RESPONSE (so not the string, but the value of the students answer) has sometimes already been calculated by the system. In the Editor the student cannot type commands, however the student can use (matrix) multiplication (with the dot), inverse (with(^(-1)), addition and subtraction and the like and that is sometimes not your intention.



Your response	response
$\begin{bmatrix} 2 & 6 \\ 5 & 4 \end{bmatrix} \cdot \begin{bmatrix} 2 & 7 \\ 4 & 3 \end{bmatrix}$	Matrix(2,2,{(1, 1) = 28, (1, 2) = 32, (2, 1) = 26, (2, 2) = 47},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
$\begin{bmatrix} 3 & 8 \\ 7 & 9 \end{bmatrix}^{-1}$	Matrix(2,2,{(1, 1) = -9/29, (1, 2) = 8/29, (2, 1) = 7/29, (2, 2) = -3/29},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
$5 \cdot \begin{bmatrix} a & k \\ 7 & 8 \end{bmatrix}$	Matrix(2,2,{(1, 1) = 5*a, (1, 2) = 5*k, (2, 1) = 35, (2, 2) = 40},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])

Figure 1.144: The response of a matrix input

The figure above shows the following:

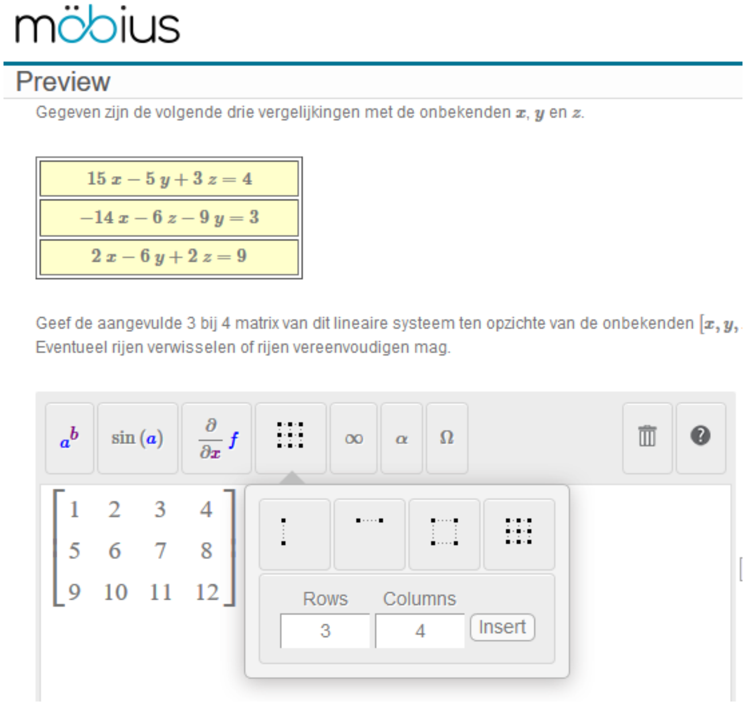
- The student enters a matrix multiplication (using the regular dot between the two matrices, so not with an asterix). The response ($RESPONSE) is then translated into the final matrix that yields this multiplication. This then compares the correct answer with, for example, the *Grading Code*:
```
LinearAlgebra[Equal](($RESPONSE ),($antw));
```
- It can also be seen that the student can enter the matrix raised to the -1 power to get a correct grading when asked about the inverse. This will happen too if the student enters the following: $\dfrac{1}{\begin{bmatrix} 3 & 8 \\ 7 & 9 \end{bmatrix}}$. The response has already been calculated by the system. You can of course do extra programming here with StringTools if the student also applies this trick and demand that the string does not contain, for example, "/rtable....." with the *Grading Code*:
```
LinearAlgebra[Equal](simplify($RESPONSE),$antw) and
 evalb(StringTools[Search]("/rtable","$RESPONSE")=0);
```

- The scalar multiplication entered by the student using an asterix (the student sees a middot in the Editor) is translated by the system into the resulting matrix where all elements with this scalar are multiplied. The same goes for addition, subtraction and multiplication of matrices.

If you are aware of this sort of things, you can take it into account in the *Grading Code*.

1.4.21.4 Example

In the following example, a 3 × 4 matrix must be entered by the student.



Preview

Gegeven zijn de volgende drie vergelijkingen met de onbekenden x, y en z.

$15\,x - 5\,y + 3\,z = 4$

$-14\,x - 6\,z - 9\,y = 3$

$2\,x - 6\,y + 2\,z = 9$

Geef de aangevulde 3 bij 4 matrix van dit lineaire systeem ten opzichte van de onbekenden $[x, y,$
Eventueel rijen verwisselen of rijen vereenvoudigen mag.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

Rows 3 Columns 4

Figure 1.145: Matrix invoeren

The corresponding*Algorithm* is:

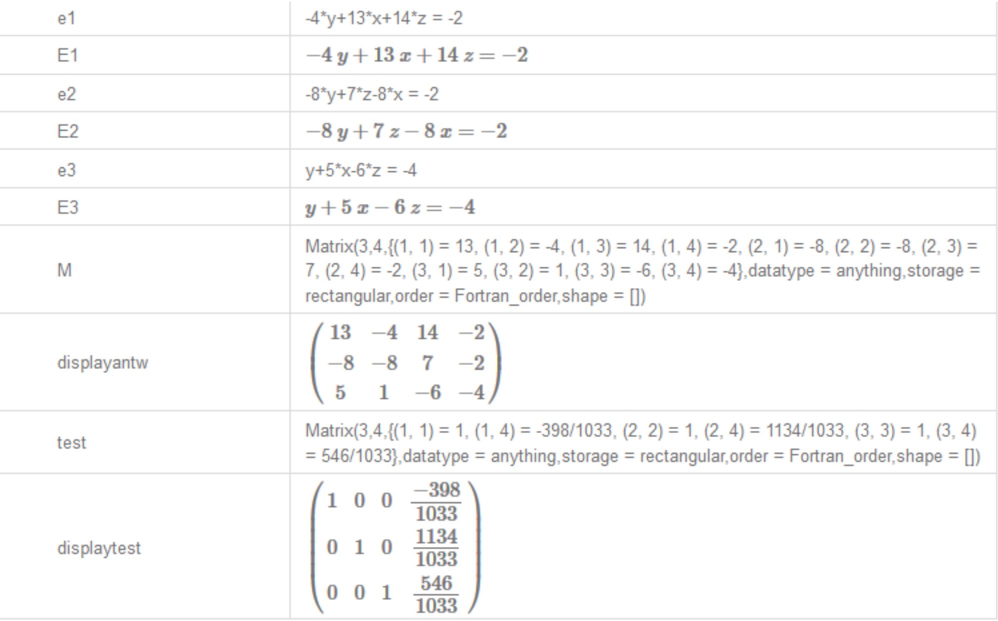
```
$a=range(-15,15);
$b=range(-15,15);
$c=switch(rint(2),range(-15,-1),range(1,15));
$d=range(-15,15);
$e=switch(rint(2),range(-15,-1),range(1,15));
$f=range(-15,15);
$g=switch(rint(2),range(-15,-1),range(1,15));
$h=range(-15,15);
$i=range(-15,15);
$j=range(-15,15);
$k=range(-15,15);
$l=range(-15,15);
$index=rint(3);
$lijstA=maple("[1,2]");
$A=maple("StringTools[Randomize]():combinat[randperm]($lijstA)");
$ad1=maple("$A[1]");
```

```
$ad2=maple("$A[2]");
$x=switch($index,"x","y","z");
$y=switch($index+$ad1,"x","y","z","x","y","z","x");
$z=switch($index+$ad2,"x","y","z","x","y","z","x");
$e1=maple("($a)*$x+($b)*$y+($c)*$z=$j");
$E1=maple("printf(MathML[ExportPresentation]($e1))");
$e2=maple("($d)*$x+($e)*$z+($f)*$y=$k");
$E2=maple("printf(MathML[ExportPresentation]($e2))");
$e3=maple("($g)*$x+($h)*$y+($i)*$z=$l");
$E3=maple("printf(MathML[ExportPresentation]($e3))");
$M=maple("LinearAlgebra[GenerateMatrix]([$e1,$e2,$e3],
[x,y,z],augmented=true)");
$displayantw=maple("printf(MathML[ExportPresentation]($M))");
$test=maple("LinearAlgebra[ReducedRowEchelonForm]($M)");
$displaytest=maple("printf(MathML[ExportPresentation]($test))");
```

In the above *Algorithm*, a number of variables are first prepared for the coefficients of the linear equations.
Furthermore, the variables of the equations are quite mixed up using $index and $ad1 and $ad2.
See the manual *Handleiding Randomiseren* about the Möbius command `switch`.
The equations are coded with MathML for the purpose of presentation in the question.
The matrix $M is generated by Maple. The figure below shows how this is stored in code in the *Algorithm*.
So this is not something to communicate to the student. That's why you prepare the correct answer at *Answer* with the following code:

```
printf(MathML:-ExportPresentation($M));
```

The variables $displayanswer and $displaytest are for checking with the variable $test being used in the *Grading Code*.



e1	-4*y+13*x+14*z = -2
E1	$-4\,y+13\,x+14\,z=-2$
e2	-8*y+7*z-8*x = -2
E2	$-8\,y+7\,z-8\,x=-2$
e3	y+5*x-6*z = -4
E3	$y+5\,x-6\,z=-4$
M	Matrix(3,4,{(1, 1) = 13, (1, 2) = -4, (1, 3) = 14, (1, 4) = -2, (2, 1) = -8, (2, 2) = -8, (2, 3) = 7, (2, 4) = -2, (3, 1) = 5, (3, 2) = 1, (3, 3) = -6, (3, 4) = -4},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
displayantw	$\begin{pmatrix} 13 & -4 & 14 & -2 \\ -8 & -8 & 7 & -2 \\ 5 & 1 & -6 & -4 \end{pmatrix}$
test	Matrix(3,4,{(1, 1) = 1, (1, 4) = -398/1033, (2, 2) = 1, (2, 4) = 1134/1033, (3, 3) = 1, (3, 4) = 546/1033},datatype = anything,storage = rectangular,order = Fortran_order,shape = [])
displaytest	$\begin{pmatrix} 1 & 0 & 0 & \frac{-398}{1033} \\ 0 & 1 & 0 & \frac{1134}{1033} \\ 0 & 0 & 1 & \frac{546}{1033} \end{pmatrix}$

Figure 1.146: Echelon matrix

In the *Grading Code* you can program as follows:

```
LinearAlgebra[Equal]($test,LinearAlgebra[ReducedRowEchelonForm]
($RESPONSE));
```

This reduced form of the augmented matrix is unambiguous and independent of the calculation method and the order of the equations used by the student.
In the *Algorithm*, the variable $test, the reduced matrix, is in fact already prepared.

HINT: If you want to allow rounding when matching the Matrices and Vectors, try something like this in the *Grading Code*:

```
LinearAlgebra[Equal](LinearAlgebra[ReducedRowEchelonForm]
(map(round,100*($RESPONSE)),LinearAlgebra[ReducedRowEchelonForm](map(round,100*
$antw)));
```

This will cast the student's rounded answer into the Echelon form and then match it with the rounded answer from the correct answer model and also cast it into the echelon form.
It is thus checked whether the augmented coefficient matrix covers the system of the given equations, where $M is the correct answer. It does not matter in which order the equations are.

1.4.21.5 Random matrix

To begin with, program a random Matrix. This can also be done by defining all coefficients one by one as a range, but the following method for the *Algorithm* is just as easy.

```
$n=3;
$A=maple("randomize():LinearAlgebra[RandomMatrix]($n,
$n,density=0.75,generator=rand(-9..10)) ");
$displayA=maple("printf(MathML[ExportPresentation]($A))");
$B=maple("randomize():LinearAlgebra[RandomVector](3,generator=rand(-9..9)) ");
```

1.4.21.6 Vectors

Vectors you match with each other in the same way as with matrices. Make sure that the correct answer is prepared in the *Algoritm* as a vector.
The *Grading Code* needed for that:

```
LinearAlgebra[Equal]($RESPONSE,$antw);
```

- To test a **direction vector**, it is not useful to match elements of the vector of the student $RESPONSE and of the correct answer $antw.
  However, a direction vector can be longer or shorter and must therefore be calculated correctly.
  If it concerns a two-dimensional direction vector, you can for example take the following in the *Grading Code*:
  ```
  evalb(LinearAlgebra[Determinant](Matrix([$RESPONSE,$antw]))=0)
  and evalb(signum($antw[1])=signum($RESPONSE[1]));
  ```
  In the above *Grading Code* you look at the 2×2 matrix formed by the two vectors to be matched. If the system is dependent, then the determinant of the matrix is 0.
  If it is also important in which direction, as with gradient and the like, you can check the sign of the first number that must match with both vectors and you do that with signum. (Or other elements if they don't have to be 0.)
- A direction vector in higher dimensions is more difficult to match. You then make another matrix of the two vectors and you demand that the rank equals 1. That's all:
  ```
  evalb(LinearAlgebra[Rank](Matrix([$RESPONSE,$n]))=1);
  ```
  Possibly the direction with the signum of one of the elements.

1.5 Question type Mathematical Formula

1.5.1 Introduction

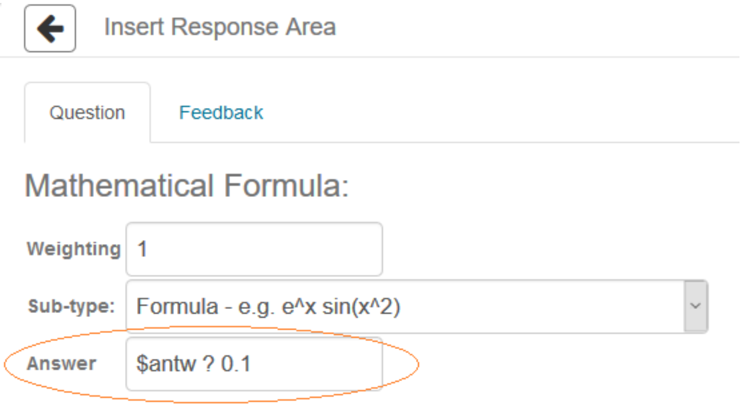
The *Mathematical Formula* question type offers a number of possibilities to do additional things with formulas.

Maple graded question type requires a *Grading Code*. For the *Mathematical Formula* question type, the correct answer is sufficient.

For ordinary formulas, this question type is often insufficient.

HINT: With this question type it is possible to grade **numbers and formulas** in different situations with possibly a tolerance. Then give in the correct A*nswer* a question mark in front of the tolerance as you can see in the figure below.

- A possibility for a formula is (-17.78? 0.01)+(0.56? 0.01)*x. You then have the option to apply a tolerance to each of the numeric parts of the formula. (Do work with brackets for each part). The disadvantage is that you can't use formulas with character combinations, and students are allowed to omit the asterix for multiplication.
- To grade a number where the correct answer $antw in the *Algorithm* is prepared, type in the settings for the correct A*nswer*: $antw? 0.1. So it means a tolerance of 0.1 in the answer.
- The nice thing is that for the tolerance with the question mark, an algorithmic variable can also be used!



Mathematical Formula:

Weighting 1

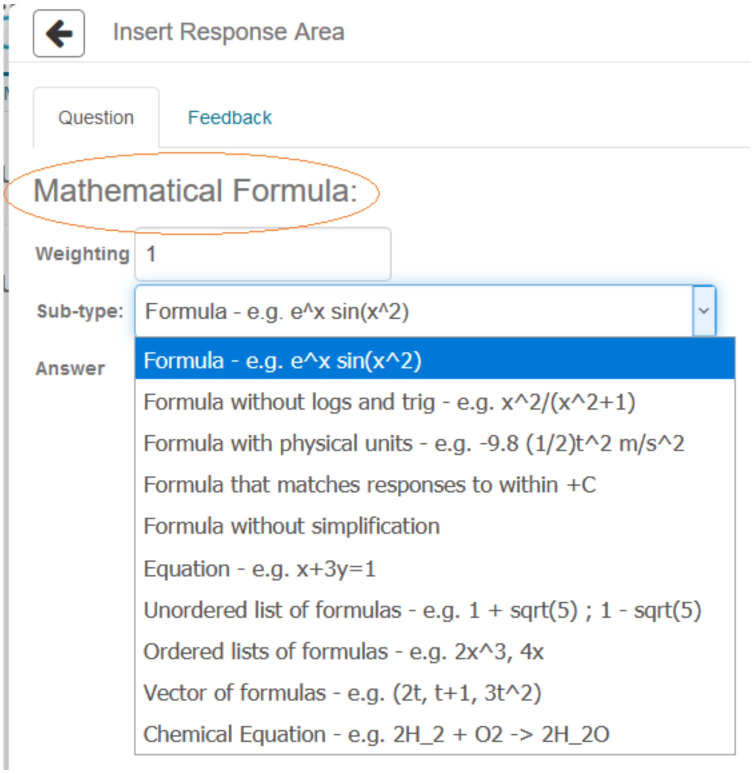
Sub-type: Formula - e.g. e^x sin(x^2)

Answer $antw ? 0.1

Figure 1.147: Tolerance in numbers of a formula

HINT: Please note that a **thousand separators** are definitely not accepted in this question type. If you still want to use it, resort to the question type *Numeric*.

Below you can see which **Sub-types** are possible for the question within this question type.


Insert Response Area
Question
Feedback
Answer
Formula - e.g. e^x sin(x^2)
Formula without logs and trig - e.g. x^2/(x^2+1)
Formula with physical units - e.g. -9.8 (1/2)t^2 m/s^2
Formula that matches responses to within +C
Formula without simplification
Equation - e.g. x+3y=1
Unordered list of formulas - e.g. 1 + sqrt(5) ; 1 - sqrt(5)
Ordered lists of formulas - e.g. 2x^3, 4x
Vector of formulas - e.g. (2t, t+1, 3t^2)
Chemical Equation - e.g. 2H_2 + O2 -> 2H_2O

Mathematical Formula:

Weighting 1

Sub-type: Formula - e.g. e^x sin(x^2)

Figure 1.148: Subtypes of the question type Mathematical Formula

The possibilities are discussed in the following sections.

1.5.2 Formula - e.g. e^x sin(x^2)

The setting with *Formula* in the *Mathematical Formula* question type is supposedly "user-friendly" but NOTE: character combinations are always seen as multiplications. For example, the character combination ab is always translated as a*b and p2 is understood as p*2.
You fill in the correct answer and there is no *Grading Code* involved. Maple is not enabled here with grading or with the correct answer.
With the number π, it does not matter whether the student enters Pi or pi. Both times it is taken as the intended constant and therefore not seen as p*i.Calculations sin(pi/2) or sin(Pi/2) or directly the answer 1 are all graded correctly.

At some settings (here at *Formula*) the student himself can choose the style in which the answer must be filled in with the Σ button. That is, choosing between text input or using an Editor.

However, the *Preview* function on text input is very limited and does not suffice in all respects (for exponential functions it is downright bad). However, there is no automatic simplification as with the *Preview* of the *Maple graded* question type with *Maple syntax Text entry only* setting and that can sometimes be an advantage. The advice is **not to use this question type with this setting**.

1.5.3 Formula without Simplification

With this question type *Mathematical Formula* is the possibility to choose sub-type *Formula without simplification*. This means that the answer should not be given as a calculation, but really in the form in which the correct answer is also given. With this setting, for example, the calculation is not correct, but a fraction with the same value is.

For example, if the answer is Pi/2, then the *Formula without simplification* setting does not grade 5*Pi/10 correctly, 1/2*Pi is not graded correctly.

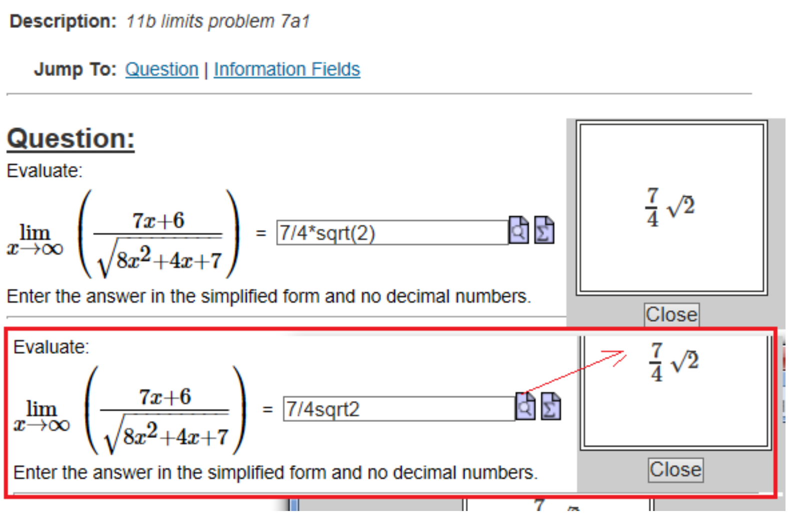
For example, if the answer is 1/2*Pi, the *Formula without simplification* setting grades 5*Pi/10 correctly, but Pi/2 is not.

At the setting with *Formula*, this would all be calculated correctly. It is advisable to try out this kind of things.

For example, in the setting with *Formula* you can calculate correctly: Pi/2 = pi/2 = 5*pi/10 = 5pi/10 etc. Uppercase and lowercase character pi or Pi does not matter.

HINT: In the setting with *Formula*, a tolerance can always be specified using the question mark, but of course not in the *Formula without simplification* setting!

We prefer not to use this question type to grade formulas, but in special cases it can come in handy. In the following example, the limit is graded with square roots. It is the intention that the possible root form that must be given as an answer is simplified as well as possible and that is why we use this question type *Formula without simplification*.



Description: *11b limits problem 7a1*

Jump To: Question | Information Fields

Question:

Evaluate:

$\lim_{x\to\infty}\left(\frac{7x+6}{\sqrt{8x^2+4x+7}}\right)$ = 7/4*sqrt(2)

Enter the answer in the simplified form and no decimal numbers.

$\frac{7}{4}\sqrt{2}$

Evaluate:

$\lim_{x\to\infty}\left(\frac{7x+6}{\sqrt{8x^2+4x+7}}\right)$ = 7/4sqrt2

Enter the answer in the simplified form and no decimal numbers.

$\frac{7}{4}\sqrt{2}$

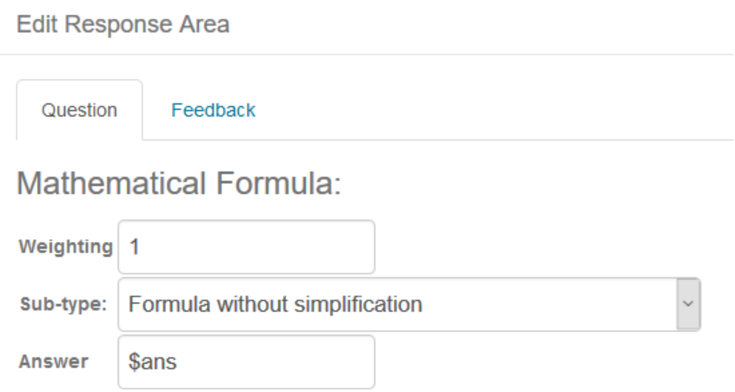
Figure 1.149: Limits with square roots

In the figure above you can see why this question type for formulas is in fact very unsuitable. The student can even manage to enter: 7/4sqrt2. But this is still translated to the right formula according to the *Preview*. And in both situations, it is graded to be correct. However, these kinds of things are not recommendable!

Apart from that, you must therefore explicitly communicate when asking the question that the answer must be given in a simplified form. You prepare this simplified form in the *Algorithm*. With square roots this is a bit difficult, but you have to make sure that the answer is prepared in the simplified form with sqrt (and not with ^(1/2)).
For example, in the following way (without maple because it makes it a power again with exponent 1/2).

```
$ans="$d/$a1*sqrt($a1)";
```

Then you create the answer field with question type *Mathematical Formula and subtype Formula without simplification* as follows.



Edit Response Area

Mathematical Formula:

Weighting: 1

Sub-type: Formula without simplification

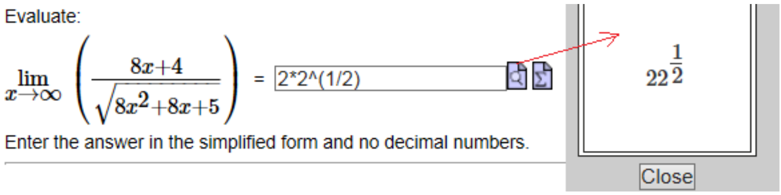
Answer: $ans

Figure 1.150: Question type Formula with square roots

This question type allows students to decide for themselves whether they want to work with the Editor or not. The student must enter the square root shape using sqrt and will then see a square root sign in the *Preview* such as *Figure 1.149 (page 128)* shows. The input 7/sqrt(8) will be graded as incorrect, because the setting *Formula without simplification* and the answer is prepared as a simplified form.

In the Editor the student enters the following: $\frac{7}{4}\sqrt{2}$ or $\frac{7}{4}\cdot 2^{\frac{1}{2}}$. It will be graded correctly. But the input $\frac{7}{\sqrt{8}}$ or $\frac{7}{8^{\frac{1}{2}}}$ or something else wll not be graded correctly.

All this only on condition that the correct answer is prepared in simplified form with sqrt (with quotes). If you don't, the student should definitely enter the answer in the form ^(1/2) and then the student could get something in the Preview that puts him completely on the wrong track.



Evaluate:

$\lim_{x\to\infty}\left(\frac{8x+4}{\sqrt{8x^2+8x+5}}\right)$ = 2*2^(1/2)

$22^{\frac{1}{2}}$

Enter the answer in the simplified form and no decimal numbers.

Figure 1.151: Preview with square root

HINT: You may grade the answer in, for example, two parts with a separate answer field for the number and a separate answer field for the square root.
See also in the section on square roots and *Grading Code* for the *Maple graded question type* (*(page 77)*)

1.5.4 Formula that matches responses to within +C

The *sub-type Formula that matches responses to within +C* means that the formula is graded as correct with a constant. So x+3 and x+6 could both be graded as correct with this setting.

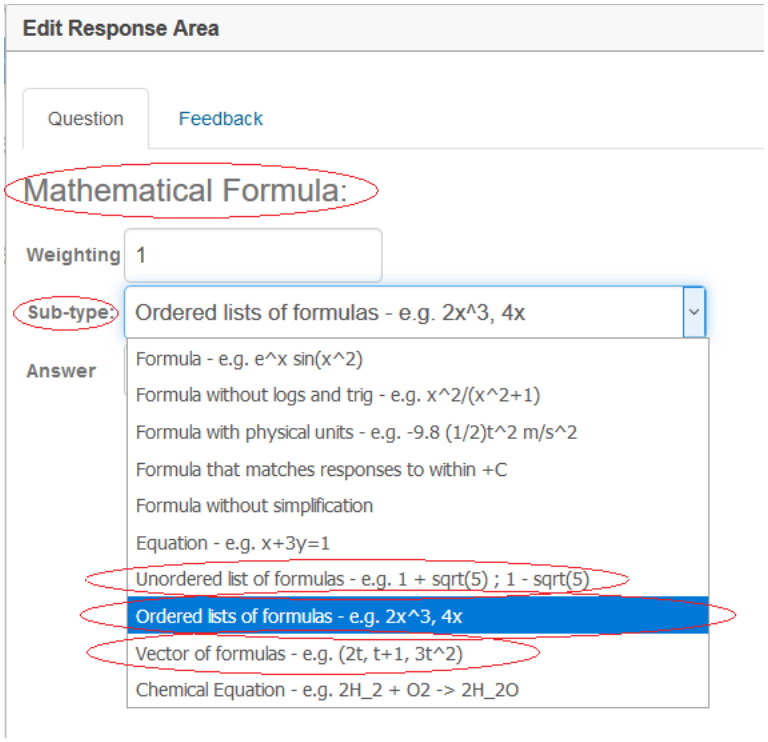
1.5.5 Ordered and unordered lists and vectors

With this question type *Mathematical Formula* it is possible to grade **ordered lists with numbers or formulas** with the *Sub-type* setting *Ordered list of formulas*. You simply give the row of

ordered objects with the correct answer (*Answer*). Whether or not brackets (square or not) around it does not matter.
Exactly the same works the *subtype Vector of formulas*, but also the *subtype Unordered list of formulas*. These three Sub-types are basically all the same with a few things of interest:

- Order is important: use commas as a delimiter.
- Order not important: Use semicolons as a delimiter.
- Brackets (square brackets or ordinary brackets) around it or not does not matter. (Communicate this with the student.)

HINT: The student cannot choose between Text and Editor at these settings.



Edit Response Area

Question | Feedback

Mathematical Formula:

Weighting: 1

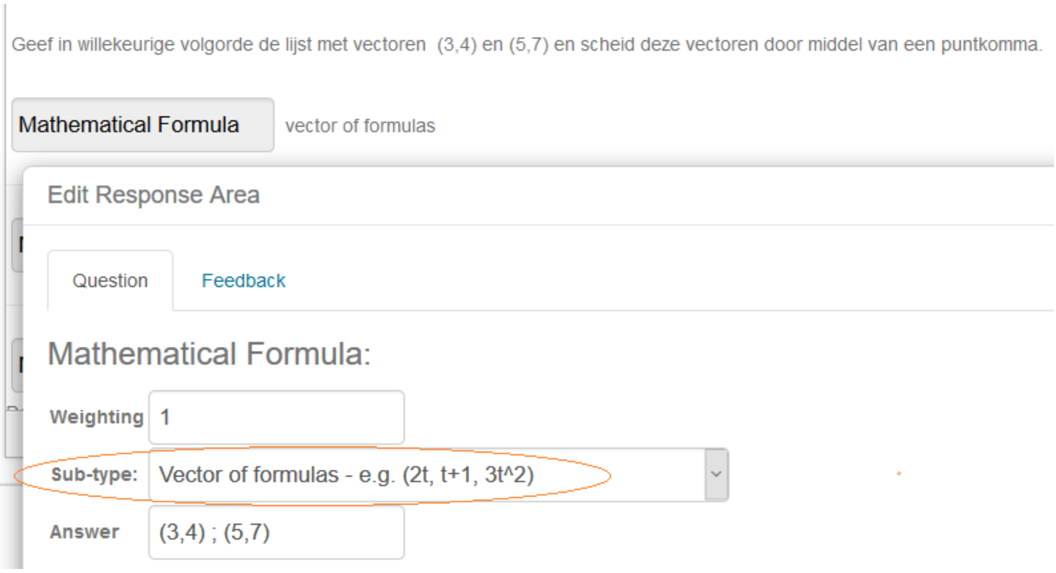
Sub-type: Ordered lists of formulas - e.g. 2x^3, 4x

Answer

Formula - e.g. e^x sin(x^2)
Formula without logs and trig - e.g. x^2/(x^2+1)
Formula with physical units - e.g. -9.8 (1/2)t^2 m/s^2
Formula that matches responses to within +C
Formula without simplification
Equation - e.g. x+3y=1
Unordered list of formulas - e.g. 1 + sqrt(5) ; 1 - sqrt(5)
Ordered lists of formulas - e.g. 2x^3, 4x
Vector of formulas - e.g. (2t, t+1, 3t^2)
Chemical Equation - e.g. 2H_2 + O2 -> 2H_2O

Figure 1.152: Unordered and ordered lists question type Mathematical Formula

HINT: Then there is also the possibility to grade a sequence of vectors (a row of numbers or formulas with brackets around them and separated by commas).



Geef in willekeurige volgorde de lijst met vectoren (3,4) en (5,7) en scheid deze vectoren door middel van een puntkomma.

Mathematical Formula vector of formulas

Edit Response Area

Mathematical Formula:

Weighting: 1

Sub-type: Vector of formulas - e.g. (2t, t+1, 3t^2)

Answer: (3,4) ; (5,7)

Figure 1.153: Lists and vectors

HINT: If parts of the ordered or unordered list are not correct, the whole answer is wrong. There is no possibility of partial grading in this question type. This is possible with the *Maple graded question type* by working with sets (*(page 64)*)

HINT:Ordered lists or vectors with formulas can also be entered, but please note that the formulas should not be too advanced because character combinations are seen as multiplications and the like.

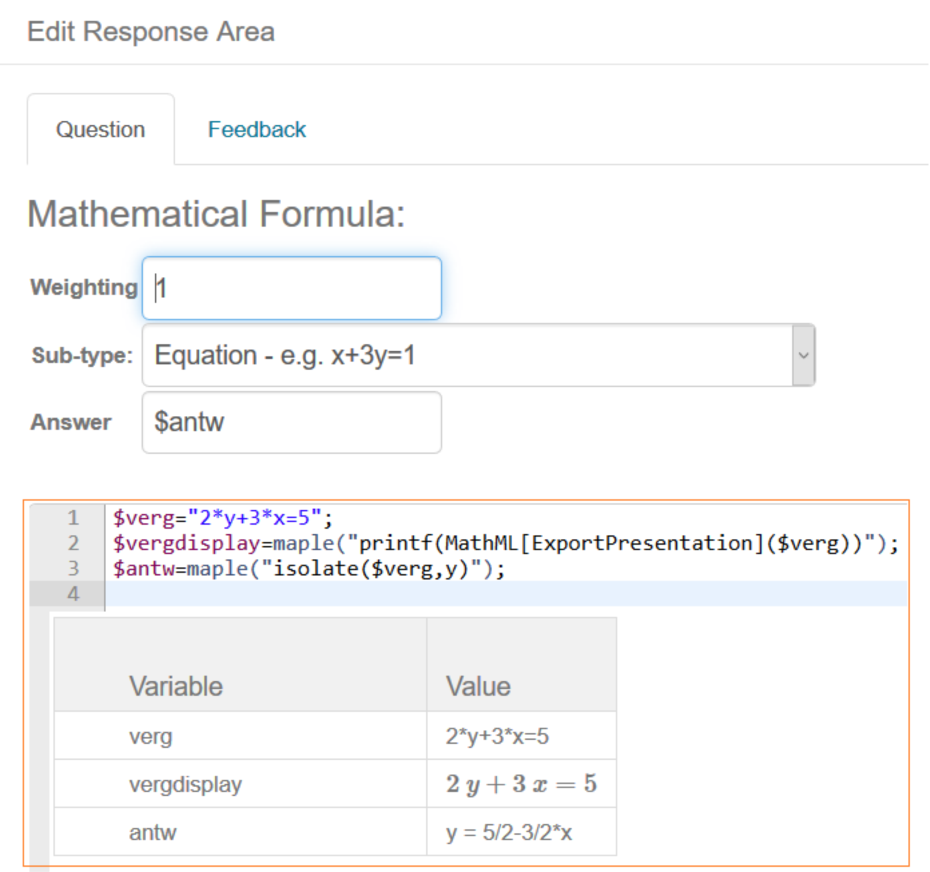
HINT: You can also add tolerance to the answer with the question mark such as (3.1? 0.1,5.7? 0.1).

1.5.6 Simple equations

To use the *Equation sub-type* of the *Mathematical Formula* question type, some additional information is needed.
Use this question type only for simple equations. The student can type the equation in any form. The disadvantage of this is that the student does not have to type in an asterix for multiplication (but it is allowed). So the syntax is not strict. There is no *Grading Code* possible so you have to make do with this.
If you enter the correct answer at *Answer*, it should definitely be in **explicit form**! It is somewhat misleading when it says Equation e.g. x+3y=1. Formulating the correct answer in this way will not lead to the desired results. But if you give the correct answer in **explicit form**, then the student can make anything out of it if it is an **equivalent equation**.

So for example, for the correct answer, enter: x=(5-2*y)/3 or y=5/2-3/2*x, then the student can type in any equation that is equivalent to this and that will then be graded as correct.
This only works if one of the variables can indeed be written in explicit form, so not in the equation$x^2 + y^2 = 9$ or something like. Then resort to the *Maple graded question type* (section *(page 34)*).

The disadvantage of this question type is that the student does not have to type an asterix for multiplication (but it is allowed) and that therefore character combinations cannot be graded. A Σ button is available for the student to switch between Editor and text field. It is the same low-level editor as in the *Formula* setting of the *Maple graded* question type.



Mathematical Formula:

Weighting: 1

Sub-type: Equation - e.g. x+3y=1

Answer: $antw

```
$verg="2*y+3*x=5";
$vergdisplay=maple("printf(MathML[ExportPresentation]($verg))");
$antw=maple("isolate($verg,y)");

```

Variable	Value
verg	2*y+3*x=5
vergdisplay	$2\,y+3\,x=5$
antw	y = 5/2-3/2*x

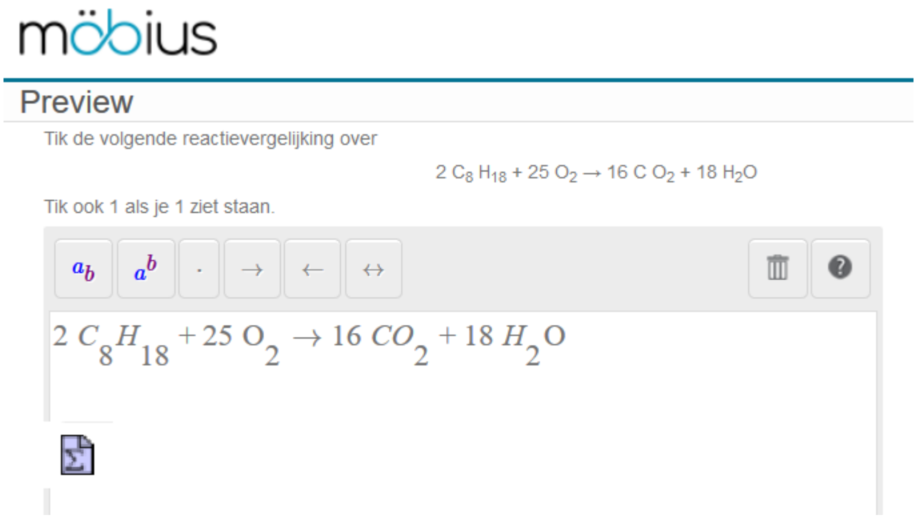
Figure 1.154: Equations with Mathematical Formula

In the figure above, the *Answer* code must contain the equation in **explicit form**. In the *Algorthm* this answer is prepared in the right form by means of the maple command `isolate`, so it could also be in the form x=(5-2*y)/3. The student may then fill in any **equivalent equation** to get the full grade.

1.5.7 Chemical formulas

With the subtype *Chemical Equation e.g.* 2H_2+O_2->2H_2O, the correct answer can be entered. At this setting, the student himself can choose text input or using the Editor. The student turns on the editor using the ∑ button below the answer field if the field is visible as a text field.

In the editor, a menu with buttons is offered for sub- and superscript and arrows.
The arrow can also be made with the keyboard with -> and superscript and subscript with ^ and underscore_.
The Editor is not that extensive, but for chemistry formulas it offers good tools with the arrows and the subscripts.



Preview

Tik de volgende reactievergelijking over

$2\ C_8\ H_{18} + 25\ O_2 \rightarrow 16\ C\ O_2 + 18\ H_2O$

Tik ook 1 als je 1 ziet staan.

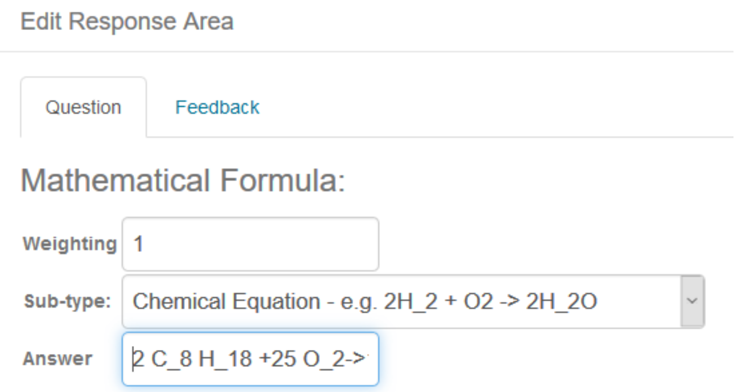
$2\ C_8 H_{18} + 25\ O_2 \rightarrow 16\ CO_2 + 18\ H_2 O$

Figure 1.155: Chemische formules

Here too, it is important that the student practices typing formulas in this editor. Create a few questions to enter formulas.

Type in the correct answer as follows:

```
2 C_8 H_18 +25 O_2->16 C O_2+18 H_2 O
```



Edit Response Area

Mathematical Formula:

Weighting: 1

Sub-type: Chemical Equation - e.g. 2H_2 + O2 -> 2H_2O

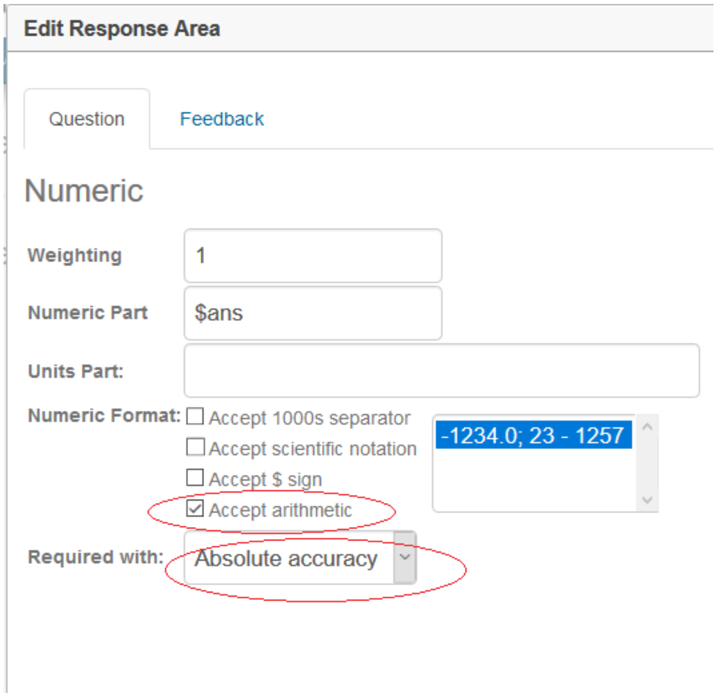
Answer: 2 C_8 H_18 +25 O_2->

Figure 1.156: Correct anwer of a chemical formula

1.6 Question type Numeric

See *Handleiding Möbius Items Maken Deel A*.

HINT: The only setting for the *Numeric question type* that may be of interest to formulas is the *Accept arithmetic* setting.



Numeric

Weighting: 1

Numeric Part: $ans

Units Part:

Numeric Format: □ Accept 1000s separator □ Accept scientific notation □ Accept $ sign ☑ Accept arithmetic

-1234.0; 23 - 1257

Required with: Absolute accuracy

Figure 1.157: title of the figure

The student can then enter the calculation with brackets ^, * and the like without using the calculator.

In that case you can put *accuracy* on *Absolute*.

HINT: Because the grading is outside Maple, here the meaning of logarithm with base 10, and ln is the logarithm with base e.
See also the section about *Exponents, logarithm and Pi* (*(page 55)*).

HINT: The number e can used in the prepared correct answer in this *Numeric question type*, but it cannot be entered as an answer even though *Accept arithmetic* is checked.

1.7 Dynamic graphs

This section you can find in *Handleiding Möbius Items Maken Deel C*

1.8 Maple commands

Information about all the commands below can be found in the book

Handleiding Maple 16, Metha Kamminga
ISBN: 9789039526750
Uitgeverij Academic Service

1.8.1 Algemene commando's

- `abs` The absolute value of a number.
- `allvalues` Sometimes the solution of an equation is given in code. To get all the solutions, you use allvalues.
- `assign` Convert an equation in an assignment

- `coeff(expr,x,n)` To get the coefficient from x to the power n of an expression.
- `coeffs(expr)` Gives all the coefficients of an expression in more variables. From that you can create a set and then see if all the coefficients in the set match. See also sets.
- `ceil(getal)` Round up (see also floor).
- `collect` To take terms together.
- `combinat[permute]([0,1,2])` Displays all permutations of the specified list.
- `combinat[randperm]([0,1,2,3])` Gives a random permutation of the elements of the list.
- `combinat[randcomb]([0,1,2,3],2)` Gives a random combination of two elements from the list.
- `convert(formule,string)` Convert a formula to a string. More conversions are possible!
- `degree(f,x)` Gives the degree of the polynomial in x.
- `denom(breuk)` Gives the denominator of the fraction.
- `Digits:=5` Means that Maple has to calculate with 5 significant digits.
- `diff(f,x)` Differentiates the function f with respect to x.
- `discrim(kwadratischevorm,x)` Calculates the discriminant of the quadratic form in x.
- `dsolve(dv,onbekende functie)` Solves a differential equation to the unknown function.
- `dsolve({dv,randvoorwaarde}, onbekende functie)`Solves a differential equation to the unknown function with boundaries.
- `eval(F,x=1)` Evaluates F for the value of x equal to 1.
- `evalb` So evaule boolean true or false for example.
- `evalb(StringTools[CountCharacterOccurrences]("$RESPONSE","a")=0)` This checks whether the number of character a in the response is equal to 0.
- `evalb(0=StringTools[Search]("factor","$RESPONSE"))` This checks whether there is also the character combination "factor" in the string of the student's response. If the combination does not occur, it provides 0 and if the combination does occur, the command Search provides a number that indicates where in the string the character combination begins.

-
  ```
  stringresponse:=StringTools[Remove](" ","$RESPONSE"):
  evalb(SubString(stringresponse, 1..-1)=$stringantwoord);
  ```
  or
  ```
  with(StringTools); stringresponse:=Remove(IsSpace, "$RESPONSE");
  evalb(stringresponse=$stringantw);
  ```
  This command ensures that the spaces are first removed from the student's response and then the result is compared with the correct answer as a string. The spaces can also be deleted with `DeleteSpace(string)`. And `SubString(.., 1..-1)` means the first character up to and including the last character.
- `evalc` Evaluates a complex number to the form a+i b assuming that all variables in the formula are real.
- `evalf` Evaluates a number to a decimal number (with floating point).
- `evalf[3]` Evaluates a number to a decimal number (with floating point) with significance 3.
- `factor` Factorizes an expression.

- `frac(breuk)` Gives the fractional part of a number
- `Float(round((6 * $var1/($var1 + 4))*10^1), -1)` Returns the result of the calculation with only one decimal place.
- `MapleTA:-Builtin:-decimal(3,5*1.7^4)` Returns the result of the calculation with three decimal places.
- `floor(getal)` Gives the greatest integer less than or equal to a number.
- `fsolve(vergelijking, onbekende, range)` Solves the equation numerically within a certain range.
- `igcd(getal1,getal2,…)` Gives the greatest common divisor of integers.
- `ilcm(getal1,getal2,....)` Gives the least common multiple of integers.
- `if $a>$b then 1 else 0 end if` Gives the least common multiple of integers.
- `if $a<0 then 0 elif $a=0 then 1 else 2 end if` This allows indices to be created for switch.
- `ifactor(125)` Displays the factors of the number (integer)
- `Int(f,x)` or `int(f,x)` Integrates the function f with respect to x. The first with capital I gives the integral, the second gives the evaluated integral.
- `intersect` For sets to get the intersect
- `isolate(eqation,expr)` returns an equation equivalent to its input. It isolates a subexpression to the left hand side of an equation.
- `isprime(a)` Asks if it is a prime number (boolean), `nextprime(a)` gives the next prime, `prevprime(a)` gives the previous prime.
- `ithprime(n)` Gives the n-nd prime number (the first is 2).
- `lcm(getal1,getal2,…)` Gives the least common multiple of a sequence of integers.
- `Limit (f,x=0)` of `limit(f,x=0)` Specifies the limit for x approaches to 0.
- `LinearAlgebra[Equal] (M1,M2)` To compare matrices (or vectors) true or false.
- `LinearAlgebra[Determinant]($matrix)` To calculate the determinant of a matrix.
- `LinearAlgebra[GenerateMatrix]([$e1,$e2,$e3], [x,y,z],augmented=true)` Calculates the augmented matrix of the three equations $e1 and $e2 and $e3 in the variables x, y and z. (3 rows and 4 columns).
- `LinearAlgebra[Rank]($matrix)` To calculate the Rank of a matrix.
- `LinearAlgebra[ReducedRowEchelonForm]($matrix)` To calculate the reduced matrix.
- `LinearAlgebra[Transpose]($A)` Gives the transformed of a matrix A.
- `evalf(ScientificConstants[Constant](R));`
  Gives the value of the universal gas constant. You can request the unit with:
- `ScientificConstants[GetUnit](ScientificConstants[Constant](R));`
- `map(round,A)` Rounds all elements of A (matrix, vector, or set or list).
  This action can also be achieved with the help of the elementwise postfix (~) so with `round~(A)` .
- `map(x->100*x,A)` Multiplies all elements of A by 100, useful for a set where one. scalar multiplication does not work. Thus, each function can be used to treat all elements.
  This action can also be achieved with the help of the elementwise postfix (~) so with `100*~A`.
- `Matrix([[a,b],[c,d]])` Gives a 2×2 matrix.

- `max([a,b,c])` Gives the maximum of a list of numbers.
- `min([a,b,c])` Gives the minimum of a list of numbers.
- `minus` To substract two sets. See also intersect and union.
- `nops` Specifies the number of operands of an object (set, list, expression).
- `numer(breuk)` Gives the numer of the fraction.
- `numtheory[factorset](G)`Specifies the set of factors that make up the integer G.
- `op(...)` Gives the operands of an expression.
- `op(1..4,expressie)` Gives the operands 1 to 4 of an expression.
- `op([1,3],expressie)` First takes the first operand of an expression and then gives the third operand of it. It should be seen as the nesting of operands of an expression. This is the same as op(3,op(1,expression)) or the third operand of the first operand of an expression.
- `plot` To plot a function.
- `printf(MathML[ExportPresentation]($antwoord))` Gives the MathML code of a formula.
- `printf("tekst")` To communicate text where the quotes are omitted by means of printf.
- `randomize():LinearAlgebra[RandomVector](6,generator=rand(1..5))` To make a random vector.
- `randomize():LinearAlgebra[RandomMatrix]($n, $n,density=0.75,generator=rand(-9..10))` To make a random matrix.
- `remove(has,[seq(seq(i*k^2,i=2..10),k=2..10)],[seq(k^2,k=2..31)])` Of all the numbers in the list i*k^2 where i = 2 to 10 and k = 2 to 10, "has" checks whether there are elements of the list of numbers k^2 where k runs from 2 to 31 and then these elements are removed.
- `max(remove(has,[$g1,$g2,$g3,$g4,$g5,$g6],$grootste))` If first $grootste is defined as the largest of the relevant list of numbers, then this largest can be removed and then the next largest can be chosen.
- `parse ("...")` Turns a string into a normal formula without quotes. (stripping quotes).
- `rhs(verg) lhs(verg)` Displays the right hand side or the left hand side of an equation, respectively.
- `round` Rounds a number to an integer.
- `seq` To make a sequence
- `series(f,x=0,n)`To make the series of powers of the function *f* near $x = 0$, optional to which degree.
- `simplify(vorm)` Simplifying an expressiong.
- `solve(verg, x)` or `solve(verg, {x})` To solve an equaion with respect to *x*.
- `sort` Can sort an expression in order of descending powers or in lists by order of size (only for rational numbers, with square roots does not work).
- `sqrt` square root
- `StringTools[CountCharacterOccurrences]("$RESPONSE","a")`
- `StringTools[Remove]("a","$RESPONSE")`
- `StringTools[Search]("factor","$RESPONSE")`Gives a number where the first character combination in the string is found.

- `StringTools[SearchAll]("a","$RESPONSE")` Displays a row of numbers where all the characters you are looking for in the string are found.
- `subs(x=3,f)` Substitute $x = 3$ in the expression f.
- `Sum/sum` Summation
- `type($RESPONSE,expanded)` Returns true when all the brackets are gone.
- `union` The union of two or more sets (see also intersect and minus).
- `value(..)` Gives the value of something, for example of an integral.
- `Vector([1,2,3])` Returns a column vector.
- `[seq(seq(i*k^2,i=2..10),k=2..10)]` Displays a list of values sequentially "i*k^2" where i runs from 2 to 10 and k runs from 2 to 10.
- `Float(round(10*69.4567), -1)` To get a number with 1 decimal.
- `Float(10*round(24.7752809),-1)` Gives 25.0

```
> zip(op,[2,1],[x^2+3,x^2+3]);
```

$$\left[3, x^2\right] \tag{1.86}$$

Means you get a list of the second operand of the first element and the first operand of the second element.

```
> zip((x,y)->x+y,[1,2,3,7],[4,4,3,1]);
```

$$[5, 6, 6, 8] \tag{1.87}$$

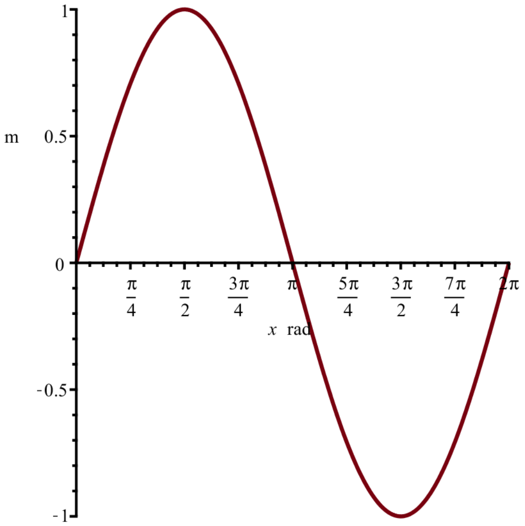
The lists are the same length. According to the prescription of the function, all elements are added up.

```
> zip((x,y)->x+y,[1,2,3,7],[4,4],0);
```

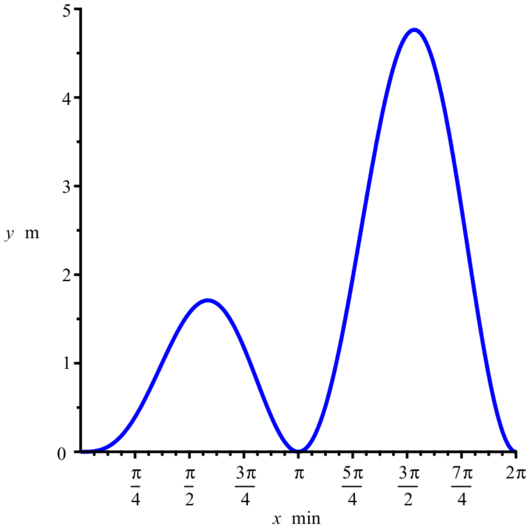
$$[5, 6, 3, 7] \tag{1.88}$$

Here the lists are not the same length. The meaning is that the elements must be added up and if the second list does not have enough elements, add 0.

```
> plot(sin(x)*Unit('m'), x = 0 .. 2*Pi*Unit('rad'));
```



```
> plot(sin(x)^2*x, x = 0 .. 2*Pi,y=0..5, useunits = [Unit('min'),
   Unit('m')], color = blue);
```



Index