A MATLAB code for automatic generation of calculated questions in MOODLE.

Olivier LOUISNARD

September 2025

Contents

1 Introduction

1.1 Can it help you ?

You are concerned by this code if you use MOODLE to design quizz for your students, in particular if you want to generate the so-called "calculated questions".

If you don't know MOODLE, you probably don't need this code.

1.2 Calculated questions

Calculated questions is one of the most elaborated question types that can be used within MOODLE quizz. The student must perform come computations in function of input data and enter the correct numerical result within some tolerance. A typical and naive calculated question would be :

```
Compute the sum of a = (some value) and b = (some value) within 5 %
error.
```

Result:

In general, you want the data indicated above by "(some value)" to change from one student to another (within some defined bounds), so that cheating becomes difficult. In this case you cannot code directly the correct numerical result in your question, since it will vary from student to student. You will have therefore to code the operation(s) that must be performed on input data to compute the correct result.

This is exactly the motivation of "calculated questions", whose basic steps are :

- phrase the question involving the varying data,
- define the formula for computing the correct result.
- define N sets of input data, each data varying in the range you specify,

In the above case, you would have to enter the following information in various MOODLE dialog boxes (disregarding various fine tuning options):

Question text: Compute the sum of a = {a} and b = {b}.

Answer : {a} + {b}

Tolerance : 0.05 Type : Relative

Elements to add :

Variable (a)	Minimum	Maximum
	5.1	41.3
Number of decimals	1	
Distribution	Uniform	

Variable (b)	Minimum	Maximum
	10.2	37.8
Number of decimals	1	
Distribution	Uniform	

1.3 Motivations

This is definitely a great tool, and I've used it intensively. But if you have already experimented it, you will probably have noted the following limitations:

- Syntax for formula is painful.
- The formula must cast in a single line.
- You do not have access to elaborated functions.
- You can ask the student only one numerical result.

- Dialog boxes are verbose, often confusing, and most tuning options are generally not needed.
- You can forget one input data in the question.
- The tolerance for the answer is not displayed to the student by default.

The first four drawbacks listed above are the most annoying in my opinion and motivated me to find another system extending the same concept. The ideal tool for me would include the following features:

- Write the question in near-WYSIWYG mode.
- Code the formula yielding the correct answer in a standard easy-to-use computing language.

This is exactly what OLMOODLE does. The question is designed under Excel, because most of my collegues are familiar with it. The correct formula must then be coded in a prebuilt MATLAB script. The tool then automatically builds a LaTeX file containing as many CLOZE questions as you required. Compilation of the latter yields an XML file, that can be directly imported into MOODLE.

1.4 Do I need to know LaTeX?

Short answer: no. In absolute, you just need to know how to compile the generated LaTeX file (see Sec. 2), and anyway, the code will remind you how to do that. So:

You don't need to know the LaTeX syntax but *you need to have an installed LaTeX on your computer.*

If you don't know anything about this, look at the TeXlive page. From there you'll manage to install LaTeX easily on whichever platform.

Of course, if you have some basic knowledge of LaTeX, good! You'll be able to use greek and math symbols in your questions. All you have to know is that any math formula should be enclosed by "`$`". To write "Symbol α represents the first greek letter", typeset "`Symbol $\alpha$ represents the first greek letter`". As simple as that...

Look at examples, it will soon become clear.

1.5 Do I need to know Matlab?

Yes. But barely.

MATLAB was initially designed (in the 80's) for people that want to compute but are not very fond of programmation. Its syntax is thus as simple as possible. That's why you will probably

have no difficulties in programming the correct answers with MATLAB. If you experienced the ugly MOODLE syntax for calculated questions (see Sec. 1.2), you will realize how using MATLAB is simple...

2 Usage. Workflow.

The process of creating a calculated question using this package is depicted in Fig. 1. It follows a few steps:

1. Create a void directory somewhere, and there create your Excel file, say "`myquestion.xls`".
2. In MATLAB, go to that directory, and launch "`olmoodle_main`".
3. An input filename will be asked. Answer "myquestion.xls" (with extension). In the following calls, if any, the latter will be proposed by default.
4. On the first run, if no errors are detected, you'll be informed that a `mycode.m` snippet program has been created, and it will be opened in MATLAB editor. The script is paused.
5. Fill the snippet code to program the correct answers to the questions. The comments in the pre-built snippet will guide you. Don't forget to use `.*`, `./` and `.^` in multiplications, divisions and exponentiations, respectively.
6. Once you are done, click ENTER in the Command Window. The main script will build the output LaTeX file "`myquestion.tex`".
7. In a terminal, or in your favorite LaTeX system, type "`xelatex myquestion`".
8. A file "`myquestion-moodle.xml`" has been created. Import it into moodle. You're done !

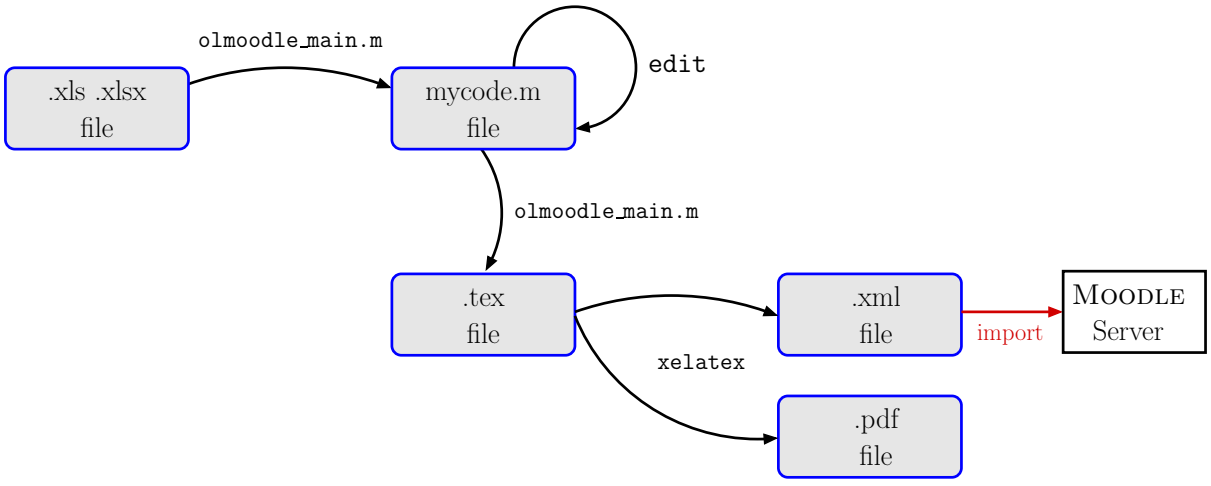

Figure 1: Workflow of OLMoodle

3 Syntax of Excel input file

3.1 Overview

In a numerical question, you will have mainly:

- **Pure text:** an informative text with no numerical data inside.
- **Fixed input data:** a numerical data that you provide *identical to all students*. Think for example of gravity in a mechanics exercise, atmospheric pressure, or a physics constant (light speed, Boltzmann constant, ideal gas constant ...) or whatever input data you don't want to change from student to student.
- **Variable input data:** a randomly generated numerical data that you want to be *different from one student to the other*. In this case you will need to define a minimum and a maximum.
- **Calculated input data:** an data depending on the other input data (fixed or variable), but that *you provide pre-calculated to the student*. This happens when some intermediate calculation is necessary for the student to compute his/her answers, but you don't want to annoy him/her with this painful step.

  For example, you have a variable input data representing temperature, and the students needs the (temperature-dependent) viscosity of the fluid to perform his/her calculation. But the formula is awful and/or this is not the skill you want to evaluate in your student, so that you compute yourself the viscosity and provide the result directly to him/her.
- **Answer:** a numerical data that your student must compute in function of input data, which will be rated.

Each among the latter will be defined on a single Excel line, and the corresponding output will be written in the output LaTeX file, in the same order as they appear in the Excel file. You can therefore mix text, data definition, questions, text again and so on... The order of appearance will be the same as in the Excel file. Of course, logically, you must give all input data to the student

You will also find :

- The number of different data sets you want to generate.
- The tolerance allowed in the results.
- The question title.
- The moodle category where you want to import the resulting questions.

Comments are also allowed. The Excel cell formatting is free, and the format of cells will not be considered by the code. This allows you to format your cells as you want, especially to use colors for better readability.

Each data among the above lists is defined on one line, the first column containing a code constituted by a letter or character string, defining what kind of data you are going to define. If the first column is void, all other cells of the line will be considered as comments.

3.2 Lines involving numerics

F: Fixed input data

A	B	C	D	E	F	G	H	I
F	Text	LaTeX symbol	Matlab variable	Value			Unit	Format

- Text: the text to be displayed
- LaTeX symbol: the displayed symbol for this quantity.
- Matlab variable: the value of the input data will be available in this variable in your matlab snippet.
- Value: the value you want to define for this quantity.
- Unit: the unit you want to display after the value. See section 3.4.
- Format: the way you want to format the displayed numerical value. See section 3.5.

Example :

A	B	C	D	E	F	G	H	I
F	Pression atmosphérique	p_{\text{atm}}	patm	101300			Pa	F0

will display :

Atmospheric pressure $p_{\text{atm}} = 101300.\,\text{Pa}$

V: Variable input data

A	B	C	D	E	F	G	H	I
V	Text	LaTeX symbol	Matlab variable	Min Value	Max value	Precision	Unit	Format

- Text: the text to be displayed
- LaTeX symbol: the displayed symbol for this quantity.
- Matlab variable: the value of the input data will be available in this variable in your matlab snippet.

- Min value: the minimal value that this variable can reach within all sets.
- Max value: the maximal value that this variable can reach within all sets.
- Precision: a signed integer p defining the last non-systematically zero digit, from right to left. For any $n < -p$, the 10^n digit will be zero.

  For example, if you specified min and max values of 100 000 and 200 000, respectively, if you specify 0 here, a value of 134 739 can be randomly generated. If you specify -2, values for digits 1 and 0 will be zero, and you will get values like 134 700 or 198 300.

  Another example: you specified min and max values of 0.2 and 0.4, respectively. If you specify 1 here, you can only get 0.2, 0.3 or 0.4. If you specify 3, you will get all values with zero fourth decimal and following, typically 0.237, 0.389 etc.

  This avoids to provide data with unexpectedly large number of significant digits to the student (remember that he will have to do some calculations. . .).
- Unit: the unit you want to display after the value. See section 3.4.
- Format: the way you want to format the displayed numerical value. See section 3.5.

Example :

A	B	C	D	E	F	G	H	I
V	Vitesse	U_\infty	Uinf	2	10	1	m.s-1	F1

will display :

Velocity $U_\infty = 9.0\,\mathrm{m\,s^{-1}}$

Note that 8.7 is in the range $[2-10]$, it is defined with 1 decimal (so that value 8.71 for example cannot appear) and (consistently), is displayed with one decimal digit. Note also how a LaTeX syntax was used in third column.

C: Calculated input data

A	B	C	D	E	F	G	H	I
C	Text	LaTeX symbol	MATLAB variable				Unit	Format

- Text: the text to be displayed
- LaTeX symbol: the displayed symbol for this quantity.
- MATLAB variable: you must assign this variable in your matlab snippet.
- Unit: the unit you want to display after the value. See section 3.4.
- Format: the way you want to format the displayed numerical value. See section 3.5.

Example :

A	B	C	D	E	F	G	H	I
C	Diffusion coefficient	D_{AV}	DAV				m2.s-1	E2

will display :

Diffusion coefficient $D_{AV} = 2.52\ 10^{-5}\ \mathrm{m^2\ s^{-1}}$

Q: Result

A	B	C	D	E	F	G	H	I
Q	Text	LaTeX symbol	MATLAB variable	Points			Unit	Format

- Text: the text to be displayed
- LaTeX symbol: the displayed symbol for this quantity.
- MATLAB variable: you must assign this variable in your matlab snippet.
- Points: integer. Defines the number of points assigned to this question. Default is 1.
  This must be understood as follows: the total number of points assigned to the
- Unit: the unit you want to display after the value. See section 3.4.
- Format: the way you want to format the numerical values for displaying the range of acceptable answers. See section 3.5.
  In addition to the codes described there, you can also enter:
  - "N" to disable the display of the range.
  - Any user-defined string that will be displayed as is (any LaTeX sentence is allowed).

Example :

A	B	C	D	E	F	G	H	I
Q	Incident vapor concentration	C_\infty	Cvinf	2			mol.m-3	F2

will display :

Incident vapor concentration : $C_{V,\infty}$ = [ ] $\mathrm{mol\ m^{-3}}$ (0.44 → 0.94)

There is a variant code `Q*` that will skip the display of the LaTeX symbol.

Example :

In the above example, this would yield :

Incident vapor concentration : [ ] $\mathrm{mol\,m^{-3}}$ $(0.67 \rightarrow 1.17)$

3.3 Other lines

T: Text

A	B	C	D	E	F	G	H	I
T	Text							

- Text: the text to be displayed. LaTeX is allowed. Don't bother about linebreaks, Moodle will manage that by himself, depending on the window size.

Example :

A	B	C	D	E	F	G	H	I
T	A spherical fruit of diameter D, whose surface is wet, is dried by an airflow at temperature T_∞, relative humidity ψ, and velocity U_∞.							

will display :

A spherical fruit of diameter D, whose surface is wet, is dried by an airflow at temperature T_∞, relative humidity ψ, and velocity U_∞.

Z: Tolerance

One line of tolerance in your Excel file is mandatory. This is because the student needs to know within which error bounds his/her result will be rated. This is especially important if you expect the student to find his/her result with a diagram (in thermodynamics for example).

The tolerance defined is meant relative and applies to all results expected from the student.

The tolerance line is structured similarly as fixed inputs or calculated inputs variables, except that:

- no LaTeX and Matlab names are required,

- unit can only be blank or `percent`.

A	B	C	D	E	F	G	H	I
Z	Text			Value			Unit	Formatting

- Text: the text to be displayed. A typical message should be informative for the student, for example:

  `Your answer is considered correct within a relative error of`

- Value: the value of relative tolerance accepted in the student answers.
- Unit: leave the cell either blank or typeset `percent`. For example 0.05 and 5 % have the same meaning.
- Format: the way you want to format the displayed numerical value. See section 3.5.
  - If set in `percent`, format `F0` is probably what you need.
  - If set unitless, `F2` or `F3` may be adequate. Using `E1` is also possible but less readable.

N: Number of sets

One, and only one N line is required. It defines the number of different data sets that you want to define. If you have 40 students, the rationale is probably to define 40 sets. . . .

Think that for very large numbers of set (typically hundreds), the compilation of the LaTeX file will take some time (a few minutes), and so does the import task in moodle. Thus, don't define large sets if you don't need to.

A	B	C	D	E	F	G	H	I
N	Number of sets							

Of course the number of sets must be a positive integer.

H: Header

The title of the question. The student will see this header in the list of questions in a quizz. If N datasets are generated, all N generated questions will have the same title.

A	B	C	D	E	F	G	H	I
H	Title							

There is a variant code `H*` which acts only on the PDF file generated by the LaTeX compilation. If used, the generated PDF file will not exhibit the correct answers[1]. There is no consequence on the generated XML file.

[1] This corresponds to the option "handout" in the moodle LaTeX class

M: Moodle category

In Moodle, when dealing with a large collection of questions (chichever their type), you can arrange your questions in categories, sub-categories etc, as you do with your files in folders. Thus the syntax to design a category is similar to the one used to design folders (Windows user, beware, in normal word, the separator is a slash). There is a root category corresponding to your course, and all relative paths refer starting from the latter.

A	B	C	D	E	F	G	H	I
M	Category							

Category should look like `subpath1/subpath2/subpath3/etc`. Each subpath can contain blank spaces (it works but I hate that...). If the destination category does not exist, it will created and the generated questions will be created inside. Otherwise, the questions will be added to existing ones in the destination category.

Example :

A	B	C	D	E	F	G	H	I
M	MASS TRANSFER/NUMERICAL							

will create a subcategory MASS TRANSFER from your the root location of your course, a subcategory NUMERICAL inside the latter, and deposit the generated questions inside. I used uppercase in the present example, but this is not required.

3.4 Units

Any physcial quantity requires a unit. Units in the present tool are just an agreement between you and students. You must specify the units of both the input data and of the results you expect from the student. There is no conversion, no check of whether the unit you specify exists or not.

For the lines involving numerical quantities (I, V, C, or Q, see above), a unit string is required and must be composed as:

`xxn.xx-n.xxn`

where `xx` is the unit symbol, possibly with prefix (milli, kilo etc), and `n` is an integer representing exponent (the minus sign allows the standard notation for "per something"). If the integer is not present, 1 is assumed.

The point separator *must be present* but can be replaced by virtually anything (including one or several blank spaces), except an alphanumeric character or a minus symbol. It can even be a string as far as it does not contain forbidden characters.

Example :

- `m1.s-1` or just `m.s-1` will display $\mathrm{m\,s^{-1}}$
- `m1 s-1` or `m1=s-1` or `m1$-1` or `m=/#s-1` will also display $\mathrm{m\,s^{-1}}$ (OK the last ones are not very readable...)
- `kg1.mm-2.ns-1` or just `kg.mm-2.ns-1` will display $\mathrm{kg\,mm^{-2}\,ns^{-1}}$
- `kgm-2s-1` is forbidden because there is no separator.

There are special cases:

- degree Celsius should be noted `degC` and degree Fahrenheit `degF`. Lucky you, it is case insensitive.
- percentages should be noted `percent`.
- if you really need some fractional exponents, use `SI`. This is sometimes required by some weird physical quantities such as thermal effusivities.
- micro is the only prefix that should be displayed with greek letter. Use "`u`" to define it, for example if you typeset `uF.m-1`, it will be displayed as $\mathrm{\mu F\,m^{-1}}$.
- if your favorite weird unit or prefix is missing, contact me.

Important: this code will not check that your unit symbol and/or prefix indeed exists. As far as the above rules are respected, they will be formatted as is in the output Latex string. If you decide for example that Lo is an official unit (the "louisnard"), the following will work:

`kLo^2.m-3`

which means "squared kilolouisnard per cubic meter"...:-)

3.5 Formatting numerics

There are two methos for displaying numerics in OLMOODLE:

Ep: Floating point

The numerical data will be displayed as:

$$M.\underbrace{mm\ldots m}_{p\text{ digits}}\ 10^{\pm n}$$

where M is nonzero, there are p figures after decimal point, and n is an integer.

This is the most versatile and secure way to display numerics, because you control the number of significant figures, whatever the value. Default is `E2`.

Fp: Fixed point

The numerical data will be displayed as:

$$D\,\text{—}\,D\underbrace{0\,\text{—}\,0}_{-p \text{ digits}}. \quad \text{if } p \leqslant 0$$
$$D\,\text{—}\,D.\underbrace{d\,\text{—}\,d}_{p \text{ digits}} \quad \text{if } p > 0$$

Example :

Assume the number is 1524.628 37

Depending on the formatting code, you will get:

`F-3`	2000.
`F-2`	1500.
`F-1`	1520.
`F0`	1525.
`F1`	1524.6
`F2`	1524.63
`F3`	1524.628
`F4`	1524.6284

The fixed point formatting is less secure, but more readable. Use it if you are sure of the order of magnitude of your data. It is especially useful to display the suggested range for a result (see code Q in Sec. 3.2).

4 Writing the Matlab code snippet

After launching `olmoodle_main` in your working directory, a MATLAB code snippet will be created from the data found in the Excel file and open in the editor. Instructions will appear as comments on what you should do here.

Let's take the folloging basic example: we want to design an exercise to compute the sum, product and difference of two numbers. The Excel source file is:

	A	B	C	D	E	F	G	H	I
1	M	NUMERICAL/BASIC							
2	H	Basic operations							
3		Number of sets							
4	N	10							
5									
6		Text	LaTeX_Name	MATLAB_Name	Min	Max	Precision	Unit	Formatting
7	T	You are are going to perform operations between two numbers							
8	V	First number	a	a	10	100	0		F0
9	V	Second number	b	b	10	100	0		F0
10	T								
11	T	Compute the quantities below							
12					Points				
13	Q*	Sum $ a + b $	S	sum	1				F0
14	Q*	Product $ a \times b $	P	prod	1				F0
15	Q*	Difference $ a - b $	D	difference	1				F0
16									
17					Tolerance in %				
18	Z	Your result will be accepted within an relative tolerance of			1			percent	F1

Look at column D entitled `MATLAB_Name`. These are the names of the MATLAB symbols you are going to manipulate.

Initially, the snippet code `mycode.m` in the editor will look like (some extended comments in the real file have been skipped here):

```
% ========================================================================
% Fixed input data (identical for all students)
% These are the data corresponding to 'F' code in your Excel file.
% ========================================================================

% ========================================================================
% Variable input data (different from one student to the other)
% These are the data corresponding to 'V' code in your Excel file.
% ========================================================================
% a : First number
% b : Second number

% ========================================================================
% Calculated input data
% These are the data corresponding to code 'C' in your Excel file.
% ========================================================================
% ========================================================================
% Answers asked to student
% These are the data corresponding to code 'Q' in your Excel file.
% ========================================================================
% ----------------------------------------
% Sum $ a + b $
% ----------------------------------------
sum = zzz ;

% ----------------------------------------
% Product $ a \times b $
% ----------------------------------------
prod = zzz ;

% ----------------------------------------
% Difference $ a - b $
% ----------------------------------------
difference = zzz ;
```

You can see that some material was retaken in comments from the Excel file: the input variable

names, their definition. This explains clearly what you can manipulate.

Now, you must assign the answer variables and replace "zzz" by something more convenient. In the present case, you have to typeset:

```
% ------------------------------------------------------------------------
% Sum $ a + b $
% ------------------------------------------------------------------------
sum = a + b ;

% ------------------------------------------------------------------------
% Product $ a \times b $
% ------------------------------------------------------------------------
prod = a .* b ;

% ------------------------------------------------------------------------
% Difference $ a - b $
% ------------------------------------------------------------------------
difference = a - b ;
```

Nothing really difficult in the present example, but note however that we had to use operator `.*` instead of `*`. The same holds for division (`./`) and exponentiation (`.^`). You are largely informed of that in the `mycode.m` comments.

The reason for that is that the output variables are vectors, each component corresponding to a single dataset, and MATLAB uses the special operator `.*` to multiply arrays term by term (because the classical `*`, when used on arrays, is understood as matrix multiplication).

5 F.A.Q.

Q: Why bother me with LaTeX?

A: MOODLE wants to be fed with XML, which is very obscure for me and many people. Building directly an XML file could have been a choice, but programming and debugging would have been painful.

Luckily, clever guys[2] have developed what most of us were dreaming of: a LaTeX class to produce MOODLE questions (package moodle). With that at hand, it was much more easy for me to generate LaTeX code by a matlab script.

Q: Why not have developed this in Python ?

A: Because I'm scared of snakes. . . :-)

The true reason is that I manage much more easily MATLAB than Python. Note that if you are familiar with Python syntax, you will have no difficulty in programming your snippet code in MATLAB, since both languages are very close, at least for calculations.

[2] Anders Hendrickson developed the class, and Matthieu Guerquin now maintains it.

The only drawback is that MATLAB is a commercial code whereas Python is free, so that maybe you don't want to (or just can't) pay a MATLAB license just to use the present code ...

Q: I've change things in my Excel file but the output remains the same. Why?

A: Did you save your Excel file after last change? ;-)

Q: I am asked the following question upon execution:

```
I found an existing mycode.m file...  Should I erase it ?  Y/[N] :
```

but I don't really know what to answer. I'm afraid to erase something important...

A: At first run, a `mycode.m` was created with dummy program inside (see Sec. 4), and you should have normally already edited this file, possibly with precious coding inside.

If you have some doubts, edit it before answering, check that it is OK, save, and afterwards answer "N" (or press ENTER key).