

Learning and teaching programming and numerical methods with a system for automatic assessment

Gregor Jerše, Faculty of Computer and Information Science, University of Ljubljana,
Gregor.Jerse@fri.uni-lj.si

Matija Lokar, Faculty of Mathematics and Physics, University of Ljubljana, Matija.Lokar@fmf.uni-lj.si

Abstract

In the following draft version of the paper the web system for automatic assessment called Projekt Tomo is presented and it is shown how it can be used to teach programming and numerical methods. The paper itself is work in progress and we encourage the reader to write the authors for the current and updated version.

Introduction

In several courses at technical faculties courses called Introduction to numerical methods or similar have their primary goal to equip their students to be capable of application of their knowledge of numerical methods in the working process. There are several specific competences to be developed by the student:

- ability of employment of mathematical tools at practical problem solving,
- ability of identification of mathematical processes in professional environment,
- ability of result analysis,
- ability of presentation of results,
- ability of employment of mathematical tools in everyday environment,
- ability of mathematical modelling of technological processes,
- ability of employment of numerical methods at mathematical problem solving

The typical Course Description is as in [El-Araby_2012] “... *Numerical Analysis and Optimization methods to solve practical problems in computer science, business, engineering and science. Practical problem solving based on analyzing empirical, experimental or measured data where the precise mathematical model is approximated or not necessarily known. Limitations, trade-offs and margins of error are evaluated for various practical examples such as network traffic, engineering, science and business applications. MATLAB and/or C++ are used for computational problem solving. Suitable for computer science, mathematics, engineering, and business majors.*”

The emphasis in those courses is often on using computer to perform numerical methods like stated in [Coller_et_all_2009] “... *undergraduate mechanical engineering students must take a*

course in numerical methods. The goal of the course is to teach students the fundamentals of how to get a computer to perform calculations that are too difficult or too cumbersome to perform and check by hand. Major themes include discretization, iteration, sources of numerical error, and practical management of that error. These are the fundamentals that undergird most of the modern engineering and scientific computational tools that have become indispensable in contemporary practice.”

Therefore, the main emphasis in such courses is not on the mathematical rigorosity, but more on showing the students the practical usage of numerical mathematical methods. Beside basic knowledge on the mathematical numerical processes students are expected to learn about and learned to use computer tools suitable in various mathematical procedures.

In the curriculum in the above mentioned courses a lot of emphasis is put on using student's programming skills. The way students acquire those skills differ from faculty program to faculty program. Some intertwine the learning of programming language and numerical methods, the other expect the student to already possess basic programming skills. The later then enhance student's programming skills with adding programming language constructs specific to numerical methods (from using various libraries of procedures, additional data structures, programming language specific arithmetics... to using appropriate techniques (like recursion) students are not already familiar with).

Also several books use this “dual mode” (mathematics/programming) approach. For example [Otto_Denier_2009] *“This text provides an introduction to the numerical methods that are typically encountered (and used) in science and engineering undergraduate courses. The material is developed in tandem with MATLAB which allows rapid prototyping and testing of the methods. The package MATLAB (matrix laboratory) provides an environment in which students can learn to programme and explore the structure of the numerical methods.”*

So be it the students studying themselves using certain textbooks or being led by teachers, students have to be exposed to numerous problems and their attempts to solve them supervised . To support this teaching/learning approach we have developed a web service called Projekt Tomo.

Systems for automatic assessment of programming exercises

Programming is a skill that is best learned by practice and skillful use of numerical methods in solving problems also require a significant amount of practical examples. As said in [Langtangen_2012] “..., nobody can learn programming by just reading - one has to solve a large amount of exercises hands on. The book is therefore full of exercises of various types: modifications of existing examples, completely new problems, or debugging of given programs.” In order to achieve good programming skills, a learner has to solve many programming problems. Teaching programming is thus most efficient when the students perform the assigned tasks while the teacher points out the shortcomings and offers assistance with correcting the mistakes. As discussed before students should also know whether their solution is semantically correct, just syntactically correctness is not sufficient. Therefore good and swift feedback about the submissions is vital. Unfortunately, human grading of programming assignments is a tedious and error-prone task. As a result, students in such courses tend to be given fewer programming assignments than should be ideally given. Manual assessment of the large volumes of students' code is out of the question. However, since code can be fully understood by a computer, several different automatic assessment systems have been proposed (for a survey see [Ala-Mutka_2005] , [Caiza_DeAlamo_2013], [Douce_et_all_2005]). The obvious benefits of

automatic assessment are objectivity, consistency and speed of assessment, as well as constant availability [Ala-Mutka_Jarvinen_2004].

There are several possible approaches to Systems for automatic assessment of programming exercise (SAAPE), as discussed below.

Setting up a special server is one possible solution. Students send their programs to the server, which executes them and somehow checks the correctness and reports on the possible mistakes; all without any teacher intervention. Especially when using more time consuming methods and/or when using this approach in laboratory lessons setting where many students submit their solutions (almost) simultaneously, the server must be powerful, as it has to be able to run and check the programs of all the students in the class at the same time during the lesson itself. Authors can recall numerous teaching occasions when the servers became overloaded, forcing to switch back to ordinary teaching practice.

Using those servers have a significant drawback from the teacher's perspective as well, as they do not usually provide a teacher's insight into the students' submissions.

Another alternative is using the server for administration only, while the code itself runs on the student's computer. The CodeAcademy pages and Khan Academy are currently probably the best-known examples of such a service. This alternative eliminates security and technical issues, as well as provides even faster feedback to the students as they do not need to upload the files on the server. Despite all its advantages, the alternative has a drawback. It is most easily executed using JavaScript programs, which run via a web browser. Therefore, most services offer assistance with the learning of either JavaScript or some basic form of another programming language that can be simulated in JavaScript. But we needed the support also for numerical methods oriented languages (be it a special language like MATLAB or via the possibility to use numerically oriented libraries). Another possible issue is that the students cannot use and thus get used to the common development tools.

Just in order to be complete, as mentioned in [Douce_et_all_2005] other automated approaches that may be relevant have been developed, for example peer review [Gehring_2001] and intelligent training or tutoring systems [Sykes_Franek_2004].

It should be emphasized that a solution to support the learning process where the primary form of work are lab exercises with a group of students was needed, not an automated assessment tool, which requires relatively different design decisions. So in 2010 all the known solutions (to us) have failed to provide proper support to our education process.

Therefore, a web service called Projekt Tomo was developed. The first version of the system was created in 2010 by Matija Pretnar and Andrej Bauer [Pretnar_2014]. In 2015, a new and enhanced version of the system was developed by Matija Pretnar, Gregor Jerše, Sonja Jerše and Matija Lokar ([Lokar_Pretnar_2015], [Jerse_et_all_2015_A], [Jerse_et_all_2015_B]).

Project Tomo

The main goals we pursued when developing Tomo were local execution, ease of use and independence of a specific programming language and environment. Such service would enable its users to quickly start solving tasks while still using environments they are familiar with which means short learning curve. The independence of programming language is equally important

since Tomo will be used in various courses which use different programming languages, for instance Octave in Numerical Mathematics class.

The secondary goal was to help students use the service in the areas with poor internet coverage/quality (at home, during commuting, in dormitories). This is achieved by designing a system in such way that connection to the server is not a requirement during local testing. So once a problem description file is downloaded it can also be used when offline.

The approach with local execution is also budget friendly, since it does not require a purchase and maintenance of a strong computer which would execute the student code.

In order to provide swift and good feedback to the student our solution needed a flexible system for writing test programs. We would for instance like to cover a case where a solution must not use a certain methods build into the language, has to be recursive... This was achieved by allowing tests to be arbitrarily complex programs in a chosen programming language and the ease of use is achieved by using helper methods which make writing usual test cases very simple.

To summarize the main goals of our projects are

- local execution,
- Independence of programming environment,
- Independence of programming language and
- flexibility in writing test cases.

All the solutions known to us failed to provide these goals. Especially the independence of programming language was important to us since the service was about to be used by different courses. So the development of the web service called Projekt Tomo was started in 2010. The main goals of the project were meet and our service has the following capabilities.

Ease of use enables users low overhead and quick learn curve when working with our system. This enables them to focus on tasks and not spent a lot of time learning how to use yet another system. This goal the reason we chose to develop Tomo as a web service since every computer (and phone/tablet) has a web browser which enables easy access to the service.

Swift and good feedback enables users to progress faster since hints from test program enable them to hunt bugs in their code easier even when teacher is not around to help them.

Offline work is possible after the problem description file was downloaded to the users computer thanks to local execution. Users can therefore use the system also when internet connection is problematic or even non-existent.

Local execution allows users to keep the interaction with the service Tomo to the bare minimum while working on their solutions. All they have to do is execute a file, no file uploading or other interaction with the service is needed. It also spares us of setting up a server which would execute student code and grade it.

Independence of a programming environment goes hand in hand with ease of use since users do not have to get familiar with another programming environment. Existing tools users are

familiar with (Eclipse, IDLE, Visual Studio, Octave...) usually have better support, are more mature and are used in the industry..

Independence of programming language is achieved with service Tomo being a web application with a prescribed API for communication with the student computer when reporting students achievements. This API can be implemented in any programming language.

Easy tracking of students progress is given for the teacher with the use of minimalistic but easy-to-use web based interface. Teacher can easily track the progress of individual students as well of the entire group of students. All submitted solution are stored on the server which enables teachers additional insight into solutions. The student itself makes use of the same interface to track his/hers individual progress and determines which problem to tackle next.

Flexible tests enables of much richer possibilities for testing: checking whether recursion was used instead of loops, detecting changes in local variables, determining which methods were used in the solution... All features in a programming language are at the teacher disposal when administering tests.

Open source licence Though the service was primarily meant to be used internally, we imagined that it should prove useful in other teaching environments and committed ourselves to open sourcing the development on GitHub, allowing anyone to use and extend the service.

Using the service - students

When the student first visits the homepage of Projekt Tomo web service he is presented with a welcome page (see *Figure 1*) where basic information about the service can be found. To start using the service every user has to sign in. Users can login with their Google, Facebook or ARNES AAI usernames (popular in Slovenia). Other identity providers can be added easily. Once logged in the user is taken to the home page (*Figure 3*) and can start using the service. The typical workflow for student using the Projekt Tomo web service is shown on *Figure 2*.

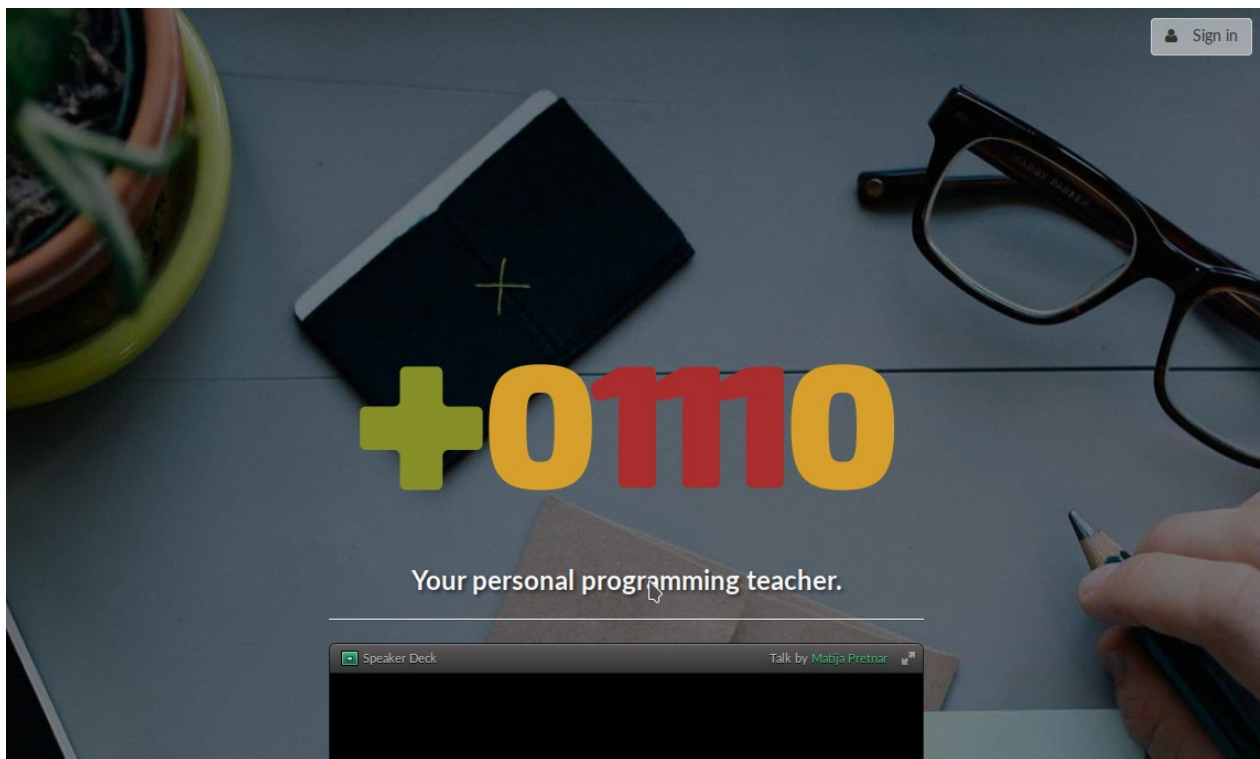


Figure 1. Welcome page.

Student's homepage (see *Figure 3*) consists of the list of their chosen courses in the content area and a list of all possible courses in the footer. For each chosen course a quick overview of accepted solutions is shown which enables a student to easily track their progress. The solutions are called accepted since correctness of the given solution is hard to test. We only test whether the solution passes our test cases and it is very possible that an accepted solution would fail on a different test.

The course page (see *Figure 4*) contains basic information about the course and its teachers on the right hand side and the list of problem sets on the left hand side. Each course is organized into problem sets, which contain one or more problems. The problems are independent and consist one or more parts. Teachers usually prepare the problem so that the parts are dependent

and their difficulty increases towards the end of the problem. On the problem set page (see *Figure 5*) a list of problems belonging to the given problem set is displayed.

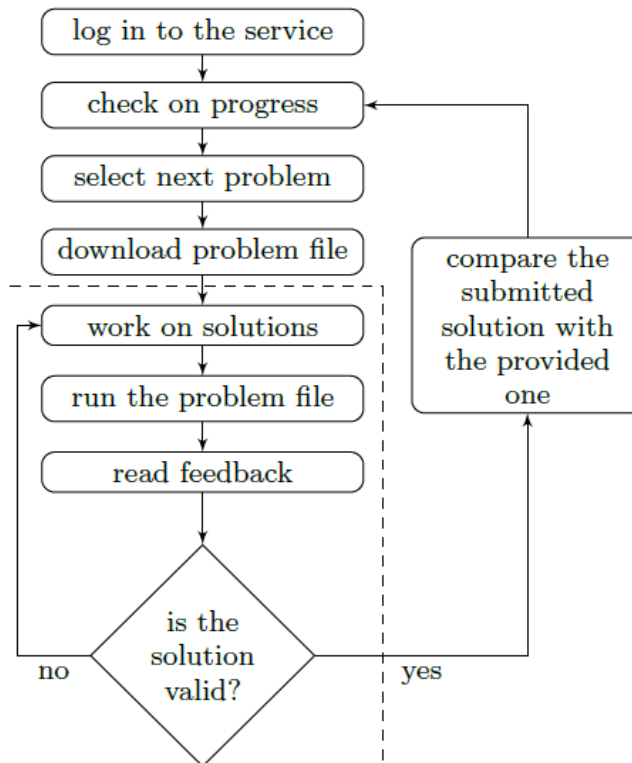


Figure 2. Workflow for the student using Projekt Tomo service. Adapted with permission from “A low overhead automated service for teaching programming” by Matija Pretnar and Matija Lokar, 2015, 15th Koli Calling Conference on Computing Education Research, November 19-22, 2015, Koli, Finland. Proceedings, (ACM International Conference Proceeding Series). New York: ACM, 2015, pg. 132-136.

To start working on a problem a student has to transfer a problem file from the server. The structure of the file depends on the problem type, currently we support python (.py) and octave (.m) file. In these files there are problem description with blanks below them where students fill in their solutions (see *Figure 6*).

The problem file can be opened in the students’ preferred editor and run with the chosen interpreter. The file contains all the validation and communication code so on each execution the solutions are validated against test cases and solutions are saved on the server if the connection is available. The validation code instantly informs student if their solutions are passing test cases (see *Figure 7*).

On the next download of the problem file the latest saved solutions are included so student can continue working from where he left even if he switches computers. Once a solution has been marked valid (or depending on the settings) a student can compare his solution with the official one (the one given by a teacher) on the web page of the Projekt

Tomo web service (see *Figure 8*). This is important for students since they can observe properly written code (of course the quality depends on the teacher) and compare it with their solution.

+0110

★

CADGME 2016

0% Simple problems

20% Workshop

0% Excercise 14

Drugi koraki v Python ☆

Matija Lokar, UROŠ VAUPOTIČ

@{Fakulteta za matematiko in fiziko}

Figure 3. Student's home page.

+0110 CADGME 2016 Testni Tomo (logout)

★

CADGME 2016

0% Excercise 14

20% Workshop

0% Simple problems

Gregor Jerse
Matija Lokar
Matija Lokar
Fakulteta za matematiko in fiziko
CADGME 2016 conference examples

Figure 4: Course page.

OMO CADGME 2016 Workshop Testni Tomo (logout)

Workshop

Problem set for ISSEP Workshop

Download this problem file

Body mass index

1. part

BMI or body mass index is a number that helps us to roughly estimate the nutritional status of human adults. If m is a person's weight in kilograms and h his height in meters, we calculate the BMI as follows:

$$ITM = \frac{m}{h^2}$$

A person's weight, measured in kilograms, is written in the variable `mass`. A person's height, measured in centimeters, is written in the variable `height`. Write a program that computes BMI and tells you what is your nutritional status. The computed BMI should be written to an accuracy of two decimal places, using function `two_decimal_places`:

```
def two_decimal_places(x):
    """ Function returns a string which contains a record of the decimal
        number, written to an accuracy of two decimal places. """
    return '{0:.2f}'.format(x)
```

Output should be as follows:

- ITM < 18.5: You are underweight.
- 18.5 ≤ ITM < 25: You have normal weight.
- 25 ≤ ITM < 30: You are overweight.
- 30 ≤ ITM: You are obese.

Body mass index: ●

Indiana Jones: ● ● ●

Equation: ●

Figure 5: Problem set page.

```

1 # =====
2 # Minmax 1
3 #
4 # Find the max and min values of a function  $sax^2 + bx + cs$  an a closed interval.
5 # =====@002828=
6 # 1. part
7 # Write a function minmax(a, b, c, d, e) that returns min value  $m$  and max
8 # value  $M$  of the function  $sax^2 + bx + cs$  an a closed interval  $[d, e]$ .
9 # Values must be returned as a tuple  $(m, M)$ .
10 #
11 # Example:
12 #
13 #     >>> minmax(1, 0, 1, -1, 1)
14 #     (1, 2)
15 #     >>> minmax(1, 0, 1, 1, 2)
16 #     (2, 5)
17 # =====
18
19 import numpy
20
21 def minmax(a, b, c, d, e):
22     def f(x):
23         return a*numpy.square(x) + b*x + c
24     values = [f(d), f(e)]
25     return (min(values), max(values))
26
--

```

Figure 6: Solution file.

```
Python 3.5.2 [Anaconda 4.1.1 (64-bit)] (default, Jul 5 2016, 11:41:13) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> runfile('C:/Users/gregor/Desktop/cadgme/minmax_1.py', wdir='C:/Users/gregor/Desktop/cadgme')
Saving solutions to the server... success.
Part 1 has no valid solution.
- For values (1, 0, 1, -1, 1) the min value is not 2. Did you consider value at the vertex of the parabola?
- Expression minmax(1, 0, 1, -1, 1) returns (2, 2) instead of (1, 2).
>>>
```

Figure 7: Feedback from the validation code.

The screenshot shows a web interface for a problem titled "Minmax 1" under the category "Simple problems". It is divided into two columns: "Your solution" and "Official solution".

Your solution:

```
import numpy

def minmax(a, b, c, d, e):
    def f(x):
        return a*numpy.square(x) + b*x + c
    points = [d, e]
    vertex_x = -b / (2*a)
    if d <= vertex_x and vertex_x <= e:
        points.append(vertex_x)
    values = [f(point) for point in points]
    return (min(values), max(values))
```

Official solution:

```
def minmax(a, b, c, d, e):
    # Store the x-coordinates of the candidates in the array.
    candidates_x = [d, e, -b/(2*a)]
    # Filter them out so that they lie on the interval [e, f]
    candidates_x = [c for c in candidates_x if c<=e and c>=d]
    # Compute function values at this points
    values = [a*x**2 + b*x + c for x in candidates_x]
    return (min(values), max(values))
```

Figure 7: Compare student's solution with the official one.

When an official solution is displayed is controlled by a teacher and can be one of the following: when a student's solution is accepted, always and never. The first option is the default one and the last one is very handy for students doing their lab exercises and the last one is used when Tomo is used for assessment..

Using the service - teachers

The teachers are presented with the same home page as the students, with some additional options, such as to add/edit problems and tracking student progress (see *Figure 8*).

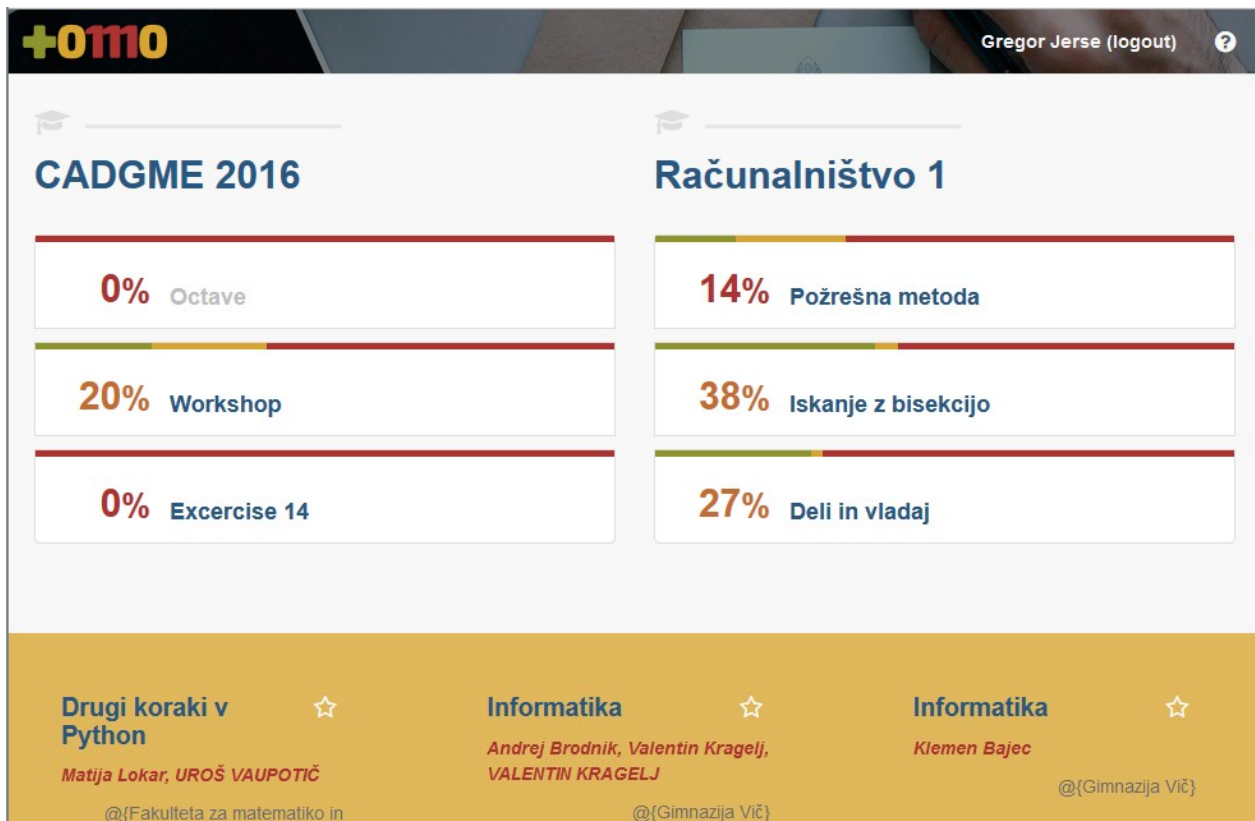


Figure 8: teacher's homepage.

As one can see the progress bars are used to quickly track the progress of the students enrolled into the courses teacher is teaching. The same progress bars are used in the course page (see *Figure 9*) where it enables the teacher to get additional insight into how well the students are progressing in various topics. This for example enables him to put more more effort into explaining topics students are weak in. On the right hand side of the course page the teacher can track progress of each individual student, which is shown with a pie chart next to the student's name.



Figure 9: Teacher course view.

In the problem set page (see Figure 10) the progress of students at individual problems can be seen by the use of pie charts on the right hand side of the page.

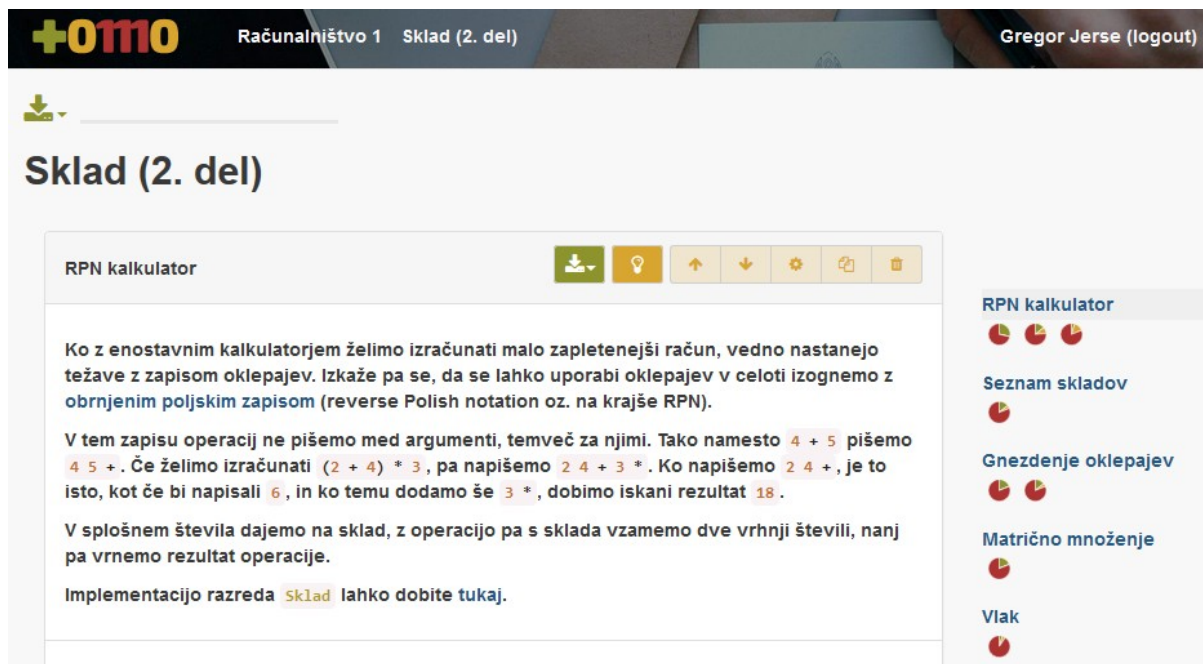
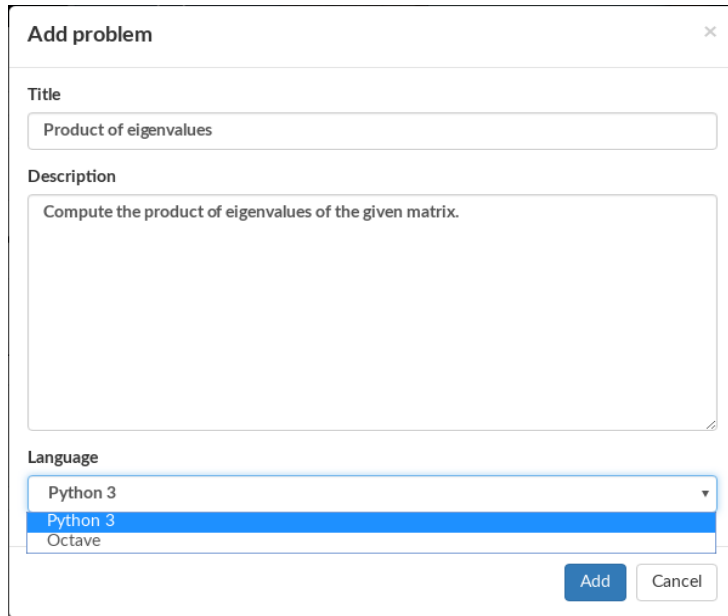


Figure 10: Teacher problem set view.

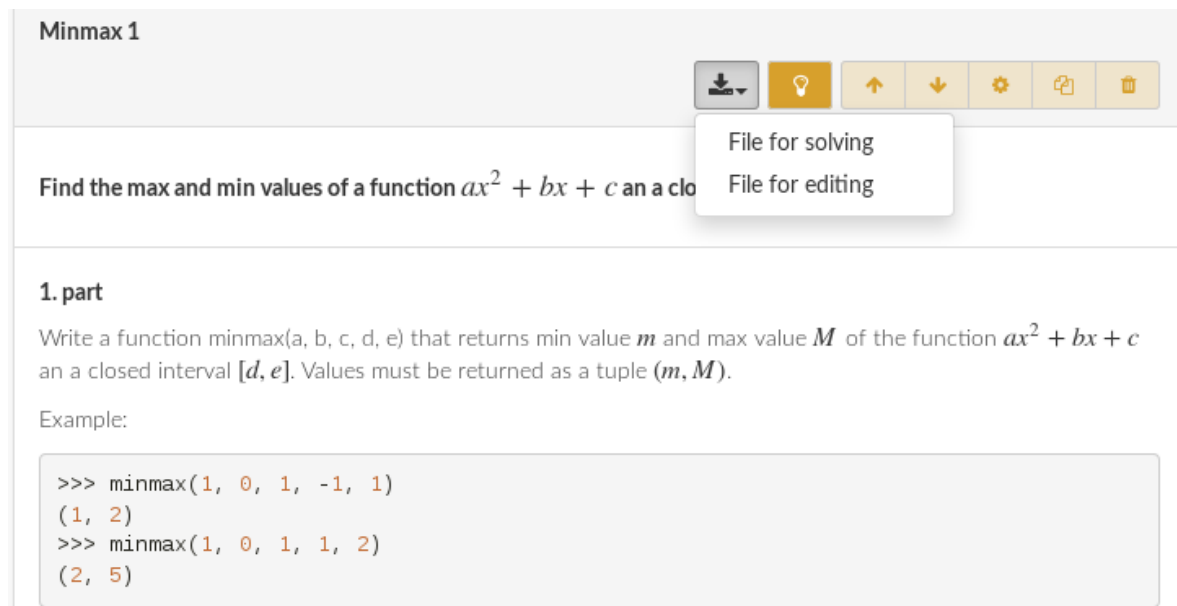
Let us describe how teachers add new tasks and how they write test cases. To start editing a problem it first has to exist on a web page. If the problem does not exist, the teacher can easily add it using web interface (see *Figure 11*). When adding a task teacher has to specify the programming language used in the given task. This can not be changed later on.



The screenshot shows a dialog box titled "Add problem" with a close button (X) in the top right corner. It contains three main sections: "Title" with a text input field containing "Product of eigenvalues"; "Description" with a larger text area containing "Compute the product of eigenvalues of the given matrix."; and "Language" with a dropdown menu showing "Python 3" as the selected option, with "Python 3" and "Octave" as visible options. At the bottom right, there are two buttons: "Add" (highlighted in blue) and "Cancel".

Figure 11: Add a problem.

When a problem exists editing is done in the same manner as a student solves a problem: teacher downloads a problem description file from the server, opens it in his favourite editor, edit the content of the file and executes it (see *Figure 12*).



The screenshot displays a problem page for "Minmax 1". At the top right, there is a toolbar with icons for download, lightbulb, up, down, settings, refresh, and delete. A dropdown menu is open over the download icon, offering "File for solving" and "File for editing". The problem text reads: "Find the max and min values of a function $ax^2 + bx + c$ on a closed interval $[d, e]$ ". Below this, it says "1. part" and "Write a function minmax(a, b, c, d, e) that returns min value m and max value M of the function $ax^2 + bx + c$ on a closed interval $[d, e]$. Values must be returned as a tuple (m, M) ". An "Example:" section shows a code editor with the following Python code and its output:

```
>>> minmax(1, 0, 1, -1, 1)
(1, 2)
>>> minmax(1, 0, 1, 1, 2)
(2, 5)
```

Figure 12: Download a file for editing or solving.

A problem description file starts with the global problem description followed by pieces which are called parts. Each part contains instructions for students, official solution and test cases as one can see in *Figure 13* and *Figure 14*.

Since the content of the file has to be a valid Octave/Python code the problem and part instructions are written as comments. The syntax of the instructions is Markdown format which allows teacher to add some simple formatting while the content remains readable in the plain text. In order to display mathematical equations LaTeX syntax can be used.

The solution and validation code for each part are written after the part instructions. They are separated by the expressions `Check.part()` in Python and `check_part()` in Octave. When the file is executed the content of the file is splitted into parts and test cases in each part are run against corresponding official solution.

If the official solution passes the test cases, the content of the file is uploaded to the server where it is parsed and changes are seen immediately on the Projekt Tomo web page. This is a failsafe which makes it impossible for the teacher to submit an official solution that does not pass the test cases. In the rest of this chapter the syntax of the file is explained in details, which we will demonstrate on a simple example of a task that computes the product of eigenvalues of the given matrix.

```
# =====
# Product of eigenvalues
#
# Compute the product of eigenvalues of the given matrix.
# =====@002693
# Write a method product_of_eigenvalues(A) which accepts a matrix A and
# returns the product of its eigenvalues.
#
# Example:
#
#     >>> product_of_eigenvalues([1 0, 0 1])
#     1
# =====
from numpy.linalg import eigvals
from numpy import prod

def product_of_eigenvalues(A):
    eigs = eigvals(A)
    return prod(eigs)

Check.part()
Check.equal('float(product_of_eigenvalues([[1, 0], [0, 1]]))', 1)
Check.equal('float(product_of_eigenvalues([[-1, 3], [2, 1]]))', -7)
Check.secret(product_of_eigenvalues([[2, 1], [0, 1]]))
```

Figure 13: Python with numpy problem example.

```

# =====
# Product of eigenvalues
#
# Compute the product of eigenvalues of the given matrix.
# =====@002696=
# Write a method product_of_eigenvalues(A) which accepts a matrix A and
# returns the product of its eigenvalues.
# #
# For example:
# #
#     octave> product_of_eigenvalues([1 0; 0 1])
#     ans = 1
# =====
function p = product_of_eigenvalues(A)
    p = det(A);
endfunction

check_part()
check.current_part;
check_equal('product_of_eigenvalues([1 0; 0 1])', 1);
check_equal('product_of_eigenvalues([-1 3; 2 1])', -7);
check_secret(product_of_eigenvalues([2 1; 0 1]));

```

Figure 14: Octave problem example.

Writing test cases

To make testing easier we have prepared some methods which enable teacher to quickly write most common tests. Their usage is also demonstrated in the examples in figures X and XX.

The most common test case equality testing, which is done by `Check.equal` in Python and `check_equal` in Octave. Both method accept two arguments: a string expression and `expected_result`. The string expression is evaluated and checked against `expected_result`. If they differ an error message with `expected` and the computed value from the expression is shown. The comparison between values is done in a smart way so that the teacher does not have to be concerned about the type of the objects being compared.

One potential drawback of using this method is that when a student opens a problem file these checks are included it (but hidden in the rest of the code). This makes it possible for a student to see the expected results for each test case and tweaks his solution to only pass test cases and thus being marked as valid on the server. This is where `Check.secret(expression)` in Python and `check_secret(expression)` in Octave comes into play. These tests are evaluated when a teacher executed the problem file and are stored on the server. When a student opens his file he can only see on which test cases his method is evaluated, but the expected result is hidden from him. Since the number of test cases can be quite big (using loops for instance a teacher can generate large number of test cases in a small amount of code) this does not help student a lot. The test cases are not limited to the use of our methods but can be any (arbitrarily complex valid problem. Furthermore the class/structure `Check/check` in Python/Octave is available to the test code. This structure provides insight into the student's test code the test program through the property `solution`. In *Figure 15* one can observe how a teacher can prohibit the student to use a

built-in method `det` in Octave. The test is done very simply by only testing whether the solution code contains 'det' keyword.

```
sol = check.current_part.solution;
if strfind(sol, "det")
    check_error("Your solution should not contain the string 'det'.")
endif
```

Figure 15: Octave test case.

Of course we can use all the tools available in the language to do the testing. In python we could determine whether the solution is using not allowed method using AST library (see *Figure 16*).

```
tree = ast.parse(Check.current_part['solution'])
for node in ast.walk(tree):
    if isinstance(node, ast.Call):
        name = node.func.id
        if name == 'det':
            Check.error('det function should not be used')
```

Figure 16: Python test case using AST.

Conclusion and future work

At Faculty of mathematics and physics we are using Projekt Tomo for couple of years in various programming courses with great results. We have collected feedback from the users of Projekt Tomo (students and teachers) and they were very satisfied with the system, especially with provided instant feedback and easy overview which enabled them to concentrate on the programming.

In the forthcoming year it is planned to be used at various other Faculties within University Of Ljubljana.

Since we have added support for Octave and numpy into Projekt Tomo we are planning to use it also at the Numerical Mathematics courses, where use of Linear Algebra and numerical tools is a must. After one year we plan to make an assessment based on feedback from teachers and students and further improve our service.

We also plan to add support for additional programming languages, possibly C# and R support coming in the near future, and add some code analysis tools. Having integrated support for MOSS would for instance come very handy when deciding which exams are candidates for plagiarism check. The Pylint report on the quality of the code and meeting python coding standards for instance would give student some information on the 'quality' of their source code to the student.

References

[Ala-Mutka_2005] K. Ala-Mutka. A survey of automated assessment approaches for programming assignments. *Comp. Sci. Ed.*, 15(2):83-102, 2005.

[Ala-Mutka_Jarvinen_2004] K. Ala-Mutka and H.-M. Jarvinen. Assessment process for programming assignments. In IEEE International Conference on Advanced Learning Technologies, pages 181-185, 2004.

[Caiza_DelAlamo_2013] J. Caiza and J. Del Alamo. Programming assignments automatic grading: Review of tools and implementations. In 7th International Technology, Education and Development Conference, pages 5691-5700, 2013.

[Coller_et_all_2009] B.D. Coller, M.J. Scott, Effectiveness of using a video game to teach a course in mechanical engineering, Computers & Education, Volume 53, Issue 3, November 2009, Pages 900-912, ISSN 0360-1315, <http://dx.doi.org/10.1016/j.compedu.2009.05.012>. (<http://www.sciencedirect.com/science/article/pii/S0360131509001201>)

[Douce_et_all_2005] C. Douce, D. Livingstone, and J. Orwell. Automatic test-based assessment of programming: A review. J. Educ. Resour. Comput., 5(3), 2005.

[El-Araby_2012] El-Araby, Numerical Analysis and Optimization CSC 380-01, retrieved from http://faculty.cua.edu/aly/my_home_page/data_files/CSC380_Spring2012_Syllabus.pdf

[Gehring_2001] GEHRINGER, E. F. 2001. Electronic peer review and peer grading in computer-science courses. In Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education. 139-143.

[Jerse_et_all_2015_A] JERŠE, Gregor, JERŠE, Sonja, LOKAR, Matija, PRETNAR, Matija. Preparing programming exercises with efficient automated validation tests. In: International Conference on Informatics in Schools - ISSEP 2015, September 28 - October 1, Ljubljana, Slovenia. JEKOVEC, Matevž (ur.). Proceedings. Ljubljana: Založba FRI, 2015, pg. 142-152, <https://issep15.fri.uni-lj.si/files/issep2015-proceedings.pdf>.

[Jerse_et_all_2015_B] JERŠE, Gregor, JERŠE, Sonja, LOKAR, Matija, PRETNAR, Matija. YASAAPE - Yet Another System for Automatic Assessment of Programming Exercises. In: International Conference on Informatics in Schools - ISSEP 2015, September 28 - October 1, Ljubljana, Slovenia. JEKOVEC, Matevž (ur.). Proceedings. Ljubljana: Založba FRI, 2015, pg. 47-48, <https://issep15.fri.uni-lj.si/files/issep2015-proceedings.pdf>.

[Langtangen_2012] Hans Petter Langtangen, A Primer on Scientific Programming with Python, Springer Publishing Company, Incorporated, 2012

[Lokar_Pretnar_2015] LOKAR, Matija, PRETNAR, Matija. A low overhead automated service for teaching programming. V: 15th Koli Calling Conference on Computing Education Research, November 19-22, 2015, Koli, Finland. Proceedings, (ACM International Conference Proceeding Series). New York: ACM, 2015, pg. 132-136. <http://dx.doi.org/10.1145/2828959.2828964>

[Otto_Denier_2009] Otto, S.R., & Denier, J.P. (2005). An Introduction to Programming and Numerical Methods in MATLAB. Springer-Verlag, ISBN: 1852339195.

[Pretnar_2014] PRETNAR, Matija. Spletna storitev za poučevanje programiranja. V: RAJKOVIČ, Vladislav (ur.), BERNIK, Mojca (ur.), RAJKOVIČ, Uroš (ur.). Vzgoja in izobraževanje v informacijski družbi - VIVID 2014 : zbornik referatov = Education in information society :

conference proceedings. Kranj: Fakulteta za organizacijske vede, 2014, pg. 194-201, <http://vivid.fov.uni-mb.si/sites/vivid.fov.uni-mb.si/files/IS2014-web.pdf>.

[Raadt_2005] M. de Raadt et al. Approaches to learning in computer programming students and their effect on success. In 28th HERDSA Annual Conference: Higher Education in a Changing World, 2005.

[Scott_Ghinea_2013] M. J. Scott and G. Ghinea. Educating programmers: A reflection on barriers to deliberate practice. In 2nd HEA Conference, 2013.

[Sykes_Franek_2004] SYKES, E. R. AND FRANEK, F. 2004. A prototype for an intelligent tutoring system for students learning to program in Java. Int. J. Comput.Appl.1, 35-44.