

A Study on Variation Technique in Courses on Scientific Computing

Dara Maghdid, Piotr Szybek, Claus Führer

Center for Mathematical Sciences, Lund University, Lund, Sweden

Email address:

dara@maths.lth.se (D. Maghdid), piotr@maths.lth.se (P. Szybek), claus@maths.lth.se (C. Führer)

To cite this article:

Dara Maghdid, Piotr Szybek, Claus Führer. A Study on Variation Technique in Courses on Scientific Computing. *Science Journal of Education*. Vol. 3, No. 3, 2015, pp. 60-67. doi: 10.11648/j.sjedu.20150303.13

Abstract: The background of this study is a project aiming at assessing the quality of teaching and learning in scientific computing in different cultural settings. This, we hope will lead us to constructing standards, which can provide outcomes of comparable quality in scientific computing in different countries and societies. Specifically we want to gain insight which quality benchmarks are suitable for the project. The tool we use in teaching is a set of variation techniques. The presented pilot study aims at the examination of the role variation theory for the quality of elementary courses in scientific computing. Earlier studies by others confirmed that variation theory offers a comprehensive set of variables characterizing teaching, well described and easy to follow and measure and which can result in improving teaching. The main data for this investigation was collected via interviewing students.

Keywords: Variation Theory, Programming, Teaching Culture, Learning Quality, Scientific Computing, Python

1. Introduction

We present a study on the use of variation technique as a pedagogical tool to improve the teaching quality in mathematical courses with strong programming components. This study is part of a bigger research project on the impact of different socio-cultural settings on the teaching outcome and how students respond on various levels of variation techniques depending on their socio-cultural background.

As a benchmark course we choose the course *Computational Mathematics with Python*, an introductory course in scientific computing at Lund University.

The presented pilot study will guide a future implementation of the same course at Soran University, Kurdistan/Iraq to investigate two different socio-cultural education environments.

The pilot study is inspired by the results of phenomenographic research and aims at showing the role of variation theory in assessing teaching and learning.

A series of studies by Marton and Morris [9] gave the motivation to base the study on variation technique. These studies confirm that using variation theory in an object of learning is a promising approach for improving the quality of teaching and learning. Runesson ensured that to learn a topic in a certain way requires that the student experiences patterns

of variation, [13]. Experiencing patterns of variation is required to improve the space of teaching and feedback in the area of computing subjects, [14].

2. Principles of Variation Technique

2.1. Theory of Variation

We have taken the critical aspects into consideration when we implemented different variation techniques. The definition of the patterns of variation given by Marton et al., [11], has been used to understand the difficulties in teaching scientific computing. Following [14] we investigate what kinds of variation are suitable for teaching various subjects in scientific computing.

Teachers have free hand in teaching, which results in qualitatively different outcomes. Similarly, it is natural that students learn different topics or even the same topic in different ways [9]. Phenomenographic researchers claim that students' learning outcomes depend on their way of seeing concepts. Thus, it is a varying ways of seeing concepts which should be the object of research, [3, 14, 11, 9, 10, 8, 13].

2.2. Object of Learning

A set consisting of a teacher, a student group, a topic, learning materials, a space for teaching, discussion and

feedback forms an *object of learning*. The teacher's plan to portion and structure the content is called the *intended object of learning*. Creating a space (classroom environment) for teaching and feedback is then called the *enacted object of learning*. Finally, the lived object of learning is what students have experienced in lectures and exercises [9]. Marton and Morris state that the object of learning does not exist in curriculum or any other static documents. In contrast, it has to be found, discovered and developed in every new teaching situation. The main reason is due to differences in the enacted object of learning.

Let us denote by OL1 and OL2 two different objects of learning with the same topic:

- OL1=function1(teacher1, studentgroup1)
- OL2=function2(teacher2, studentgroup2)

The difference between OL1 and OL2 is the difference between the enacted OL1 and OL2. The question is how teachers can reduce this difference to achieve a good quality of teaching in every teaching situation and every possible setting.

Enacted objects of learning emerge in the course of a lesson, in the interaction between the teacher and the students. On the other hand the lived object of learning is what students take with them after the lesson.

To better understand and examine objects of learning one has to consider the conditions of experiencing them (internal horizons, [8]). These are structure and meaning. *Structure* refers to the relationship between parts, and between the parts and the whole. *Meaning* is given by relationships between different aspects.

Based on the available studies on variation in teaching, e.g. [9, 3, 14] one can assume that this theory may constitute a

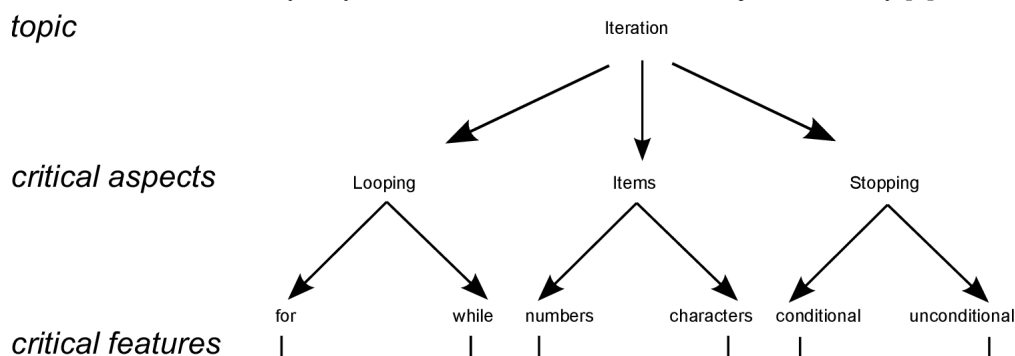


Fig. 1. Critical aspects and critical features.

For example in scientific computing programming iterations is a frequent task. It requires an initialization step, the construction of loops and the definition of a termination criterion as well as complete exception handling. In that way initialization, loops, termination criteria and exception handling are *critical aspects* of *iterations*. Their understanding is essential for a student to be able to make iterative computations. To describe an example of an *iteration* one has to know that a loop can be written by *for* or *while*. The constructs *for* and *while* are values of the dimension of variation on the critical aspect *loop*, and are called critical features Fig. 1.

fruitful approach to improve the quality of teaching and learning.

2.3. Variation Theory and Different Ways of Experiencing

Variation theory explains "how the same thing or the same situation can be seen, experienced or understood in a limited number of qualitatively different ways", [13].

Marton and Morris state, [9]:

- There is no learning without discernment, and there is no discernment without variation.
- Variation is the necessary condition for learning in a certain way.
- If we need to measure the quality of the space of learning we have to focus on the following questions:
 1. How is variation used by the teacher?
 2. How do teachers structure the content to maximize variations?

The range of what can be learned is defined by the space of learning and the range of variations presented this space [14].

2.3.1. Critical Features and Critical Aspects

In order to understand an object in a certain way, some aspects must be discerned that are critical for this way of seeing [13, p. 72].

Critical aspect is a space of dimension of variation (topics that are conceptually difficult to understand), whereas critical feature is the value of the dimension of variation. These terms can be used interchangeably, but they are not equivalent [8]. When we focus on a critical feature of an object we see it in a particular way. But when we focus on another critical feature, we see the same object differently [8].

2.3.2. Patterns of Variation

In [11] those patterns of variations are defined, which are significant in every variation process:

- Contrast: offer contrasting alternatives.
- Generalization: point out the underlying principle behind concrete examples.
- Separation: vary only those aspects you want to focus on.
- Fusion: focus on several aspects at the same time.

Marton and Morris pointed out that the necessary condition for learning in a certain way requires a recurring experience of patterns of variation. This experience may reveal extra

conditions for learning computing subjects [9].

2.4. Application of Variation Theory in Computing Subjects

Frequently, it can be observed in computing oriented introductory courses that novice programmers have difficulties to program and that the students' background becomes more and more diverse, [14]. Students have difficulties in grasping key concepts and key programming techniques. A phenomenographic study may reveal these difficulties, [14]. In [14] it is claimed that variation theory is a promising approach for improving teaching and learning of computing subjects. This paper points out that there is a need for a novel pedagogical approach in teaching computer science.

2.5. Kinds of Variation

According to [14] programming in computing subjects has at least two dimensions: The first one focuses on producing concepts and the other focuses on problems which can be treated by these concepts. This motivates two different kinds of applying variation technique in this context:

- Different concepts used for solving one problem.
- One concept used for solving different problems.

In this study we attempt to test the first one.

3. Two Different Teaching Cultures

We mention here two kinds of teaching cultures the first is *problem focused teaching* -- a technique often used at Asian universities. Here, teachers start by demonstrating a problem and then demand individual solutions (to find different concepts) as a first step, then continue by group works, so that different groups present their solutions and afterwards compare this group solutions. The teacher then comments on strengths and weaknesses of the different approaches. This amounts to the creation of a *space of learning*. The teacher finally summarizes the most powerful ideas that have come up during the lesson. Thus students are facing a single problem to which they are expected to find different solutions. So, the problem (the content) is kept constant, while the methods vary. This lets students learn methods [9, p.26].

The second technique is *method focused teaching* -- a technique often found at US American or European universities. Teachers introduce a method and then students are expected to apply it to a large number of problems. They are thus facing many problems which are to be solved in the same way. Here the method of solution is kept constant and the problem parameters¹ are varied. Here students learn which types of problems fit the given method of solution [9, p.26].

4. Use and Exchange Value in Mathematics Education

Educational research distinguishes students' motivations

inspired by either a course's *exchange value* or by its *use value*.

Williams explains that exchange value relates to the possibility that a grade can be used to "purchase" entry to further educational steps, [16, p. 59]. While use value relates to the competence and understanding required to use and apply the course content in practice. Williams derived these values first from Vygotsky's activity theory, [15], and from Bourdieu's sociology of education, [4].

Williams found out that today's students experience the exchange value of a mathematics courses as more important than its use value.

We claim that variation theory can increase the students' perception of the use value of a course in mathematics and even more it gives the use value precedence over its more abstract exchange value.

Beach investigated students' preferences for different types of classroom teaching, [1]. He describes how students prefer a boundary-setting pedagogy, favoring reproduction of knowledge in formulaic forms to a pedagogy leaving room for their own creativity. His findings corroborate the conclusions drawn by Williams.

5. Design of an Empirical Study

5.1. Selection of a Benchmark Course

This study attempts to examine via interview questions how variation theory can change the teaching quality in a course in scientific computing. For this end patterns of variation for critical aspects are considered.

The long-term interest in this question is to find possibilities to improve the quality and to develop teaching material of the course "Computational Mathematics by Python". This course was chosen because it was one of the main introductory courses in the area of computing subjects. Python is an object oriented programming language which is taught with strong relation to mathematics and scientific computing, [6]. The course is an introductory course and attending students came from mathematics, physics and electrical engineering programs at Lund University. The majority of the participating students were in the second term of the year, while some few were from later stages or even graduates. The course was also taught in Soran (Kurdistan/Iraq) to find out differences and similarities in a separate comparative study for investigating variation theory in different socio-cultural environments.

The course takes an entire study period with eight weeks (seven study weeks and one final week for project work). The total of 42 lecture were distributed on 14 hours theory and 28 hours for guided computational practice. Teaching staff were one professor for the theoretical hours and one student tutor per ten students for laboratory hours. Finally the course got one "observer" to initiate and select the experiments with different variations. To pass this course, students had to program three bi-weekly home assignments in groups of maximum two students. Furthermore co-programming was trained by the final project work which was done in groups by

¹In [9] referred as "content".

ca seven students. There were 30 students registered in the beginning of the course. Those 24 students who completed the course were interviewed. The students who quited the course could not be reached for interview purposes.

5.2. Material Collection

During the course in Lund we collected the student's different approaches to bi-weekly lab-exercises and used them to inspire plenary discussions of variations in the solutions and to jointly develop a prototype solution. This exemplified the teaching principles: contrast, generalization, separation and fusion. Four techniques have been used for collecting information for our study.

5.2.1. Participatory Observation

The teaching material was designed by observing the way students coped with given problems. The teacher presented the problems. A qualitative analysis of the students' solutions gathered by an observer gave insight into the different reception alternatives. This enabled us to find potential variation patterns.

The entire class was then confronted with so collected solution variations and asked to comment and validate different approaches.

5.2.2. Interview

The interview was a semi-structured interview as described in [5]. We refrained from open interviews since we wanted to ensure that all relevant aspects of the course in question were covered. At the same time we wanted enough flexibility for follow-up questions. Another reason for selecting this kind of interview is an intended future multiple-case study for which we need to ensure cross-case comparability [5].

We generated 45 research questions. All questions were relevant for the students we interviewed.

In order to attract the students' interest, we showed them the significant relationships between our topics. In addition, we had a list of eleven predetermined questions in order to investigate different aspects of the course, [3]. The list of questions covered the way variation theory was applied and some other less focused issues. Questions were followed exactly allowing occasional subquestions.

In total 24 students were interviewed - half of them individually, the rest in three focus groups. In these groups the students got the opportunity to discuss each other's opinion to refine answers, [5]. The groups had to arrive to a common view, and it is this that is represented in the answer we summarize below.

The average interviewing time per individual student was 20 minutes and 30 minutes per group. After an opening question concerning the students' background the interview was focused on the way of applying variation theory and on teaching culture. The interview process took place in a separate room in Mathematics Center at Lund University. The interview language was either Swedish or English depending on the students' native language. Students were free to talk and encouraged to voice their personal impressions. The

exemplifications of answers are marked S_i for student number i and $G_{i,j}$ for student number j in group i .

The interview started by introducing the objectives of this research. It was concluded by an explanation of the future use of the data, [5]. The questions about the use of variation techniques were preceded by an explanation of variation theory.

5.2.3. Questionnaire

The two teaching assistants, (TA_1 , TA_2), working in the course were given a questionnaire focusing on their impressions of the application of variation technique in laboratory work and homework presentation. Some general questions about the way of teaching and teaching culture were included.

5.3. Trustworthiness (Validity and Reliability)

5.3.1. Reliability (Dependability)

Reliability of a qualitative study is governed by other factors than of a quantitative study. The central criterion is here whether questions and categories generate a consistent picture of learning outcomes and the educational background of these outcomes in teaching. This background is constituted by the application of variation theory. Bryman [5] speaks about the so-called security zone generated by a well-founded theory, which in this case is variation theory.

5.3.2. Validity

A central validity criterion is whether data collection, analysis of data and conclusion are integrated. This has been achieved in the present study.

Content-related Validity

Data collection was based on a good knowledge and experience in scientific computing and related pedagogy. The person who ran the interviews has a sound background in scientific computing and is also familiar with the course content and university pedagogy.

Methodological validity

All material was collected according to the study as explained in Sec. 5.2. Questions are based on variation theory and some general teaching and learning issues. Data collection, outcomes, analysis of data and conclusions had always been related to each other.

Internal and external validity

Experts and teachers in mathematics and mathematics education have been consulted. Internal seminars were held where the relationship between scientific computing and its pedagogical understanding was discussed.

6. Outcomes

The collected data were analyzed with the objective to make a statement about the value in using variation theory extensively in a scientific computing course. After the outcomes were coded a collection of themes was obtained.

As mentioned, the methodology used was of a semi-structured interview with fixed questions. The results concern six themes:

- special aspects of the answers,
- students' programming procedures,
- students' perception of critical features and critical aspects,
- students' perceptions of variation (patterns of variation),
- students' perception of different teaching cultures,
- students' attitude to programming with respect to their individual background.

Additionally, we present the

- results of the questionnaire given to the teaching assistants, TA1 and TA2.

6.1. Special Aspects of the Answers

Seven students expressed delight (“it was fun...”). Five of them said this furthered their understanding and even pointed out what it was they understood better. Two students said it was useful but were more vague about how it was useful. Six answers focused rather that variation theory helped to find the best solution. Here, one answer stands out:

“Are there so many ways to write the same program? It helped a lot to know there is not just one way. We often tend to look at which one is more effective.”

6.2. Students' Programming Procedures

We have asked students a couple of questions about their work plan and how they structured their work before they start to write in groups a specific program. The aim of these questions was to understand if they apply alternative ways by using critical aspects and critical features or if they were just going ahead directly without determining any aspect.

This part of the answers gives a picture of two clusters of strategies. One cluster consists of the more planning and designing type of students (13 answers). They stress prior *understanding* as important for the steps to write a particular program, i.e. the *planning phase*. Here, four answers concern the importance of understanding the task of a particular homework, four students stress that a priori understanding of the typical problem solving steps is essential, other stress the importance of a deeper understanding the *mathematical background*. Two answers describe the details (still stressing understanding) and one says that searching in Google is necessary.

The second cluster is production oriented rather than focused on a priori planning (ten answers). Here a computer program emerges directly while writing. Seven students mention the importance to look at how the steps are interrelated, “just grope” was another answer and “look at what the result should be!” a third one. One student is just rushing “head on”.

One group of students (five answers) says that they plan

sometimes, and sometimes not; it may depend on whether the program is long or short.

Some students approach the program via its parts, trying to control constant and variable objects (S_3 and S_{10}), some again say that working with the entire program is better (S_9 and S_{12}). One student tells us that “in our group we split up the program [everybody does a part of it]” (S_6).

6.3. Students' Perception of Critical Features and Critical Aspects

The perception of what is critical for learning to program is very varying and rather vague. Students named 14 different components of the programming procedure, cf. Table 1:

Table 1. Critical aspects.

Procedure component	Number of answers indicating this component as critical
Dictionary	15
Profiling	9
Generators	9
Tests	8
Class	6
Function	4

6.4. Students' Perceptions of Variation (Patterns of Variation)

The patterns of variation which students were asked about were *contrast*, *separation* and *fusion*. Here students did see contrast: *For loop* is contrasted with: *generator* (2 students), *list comprehension* (1 student), *while* (1 student), anything else (1 student). *List* is contrasted with: *array* (3 students), *dictionary* (1 student). *Function* is contrasted with *class* by 3 students. Three students do not mention any specifics, while stressing that contrast is important.

We already mentioned that some students divide programs into parts. Here we look at it from the perspective of variation theory (separation), cf. Table 2.

Table 2. Answers related to separation.

Control of parts	“[...] you divide the program into different functions and you change just one of [them] rather than changing the whole program” (S_3) “If we change something in one subclass we may change some other subclasses to control the program” (S_{10})
Holistic approach	“I try to avoid it” (S_9) “I usually change everything” (S_{12})

The way how students understood fusion is presented in Table 3 by summarizing the students' answers about controlling several aspects simultaneously.

Table 3. Students' answers on controlling several aspects simultaneously (fusion).

Ways of expressing fusion	Examples
Identifying difficulties	“You must always control changes in parts of the program at the same time.” (S_5)
Progressing step by step	“I [...] test a little part at a time [...]. I want my functions separated from one another so changes in one of them don't affect the other.” (S_2)

Ways of expressing fusion	Examples
Visualizing	"I used to print a lot [...]" (S5) "I print as strings to see what happens" (S11)
Expressing a function in another function	"[...] call a function by another function [...] during this process you realize that some parts are correct and some not." (S12)
Using classes	"[...] through one class you can control all [...] functions [...]" (S6) "More exercises make you check dimensions increasingly. So from lines to functions and so on to classes, in this way you can control all dimensions." (G1,2)
No understanding	"Never thought of such controls." (G3,1)

6.5. Students' Perception of Different Teaching Cultures

Seven answers point out the problem oriented teaching culture as superior; two answers express a belief that the problem oriented way is better for beginners but they were unspecific with respect to the general case. One answer is about *drawbacks of the problem oriented teaching culture* and claims that it creates more *stress*. Of the answers preferring the method oriented teaching culture two claim that the material becomes technically easier, two that it furthers understanding, two that it is more fun, and one that it avoids memorization problems.

6.6. Students' Attitude to Programming with Respect to Their Individual Background

The time when the student came first in their life in contact with programming varies and has an impact on their attitude

towards programming: Of the 16 students who said programming and Python were fun 13 did not encounter programming before the university. Nine students learned to program in high school. It was among them that the four students could be found, who did not say what their feelings were toward programming.

6.7. Results of the Questionnaire Given to the Teaching Assistants

Teaching assistants are a valuable resource in course evaluation as they are students in a teaching role, so they are on both sides of the teaching process.

The two teaching assistants in the course were given interview questions related to the teaching culture and the application of variation theory. The answers are summarized in Table 4 and Table 5.

Table 4. Application of variation techniques according to teaching assistants.

Role of previous knowledge of patterns of variation	"[teaching assistants has discussed variation technique] because students have different approaches than me. ...those who need more help think in radically different ways." (TA1) "[tries to adapt to student's ability and interest] some of the students understood his/her way of solving the problem and were interested in... other ways... other students had difficulties in solving the problem." (TA2)
When did students experience variation technique?	"[all the time during the course] but we weren't aware that there is a name for [it]" (TA1) "during exercises they got help from us to think of different ways." (TA2) "in the lectures... the teacher was keen to compare different ways. Often starting with the worst way to motivate the use of a different way as easier" (TA2)

Table 5. Patterns of variation applied according to teaching assistants.

Examples of patterns of variation	
Contrast	"while loop iterates as well [for loops iterate too], but it is not necessary to stop after x iterations" (TA1)"Comparing lists with arrays. Different types of loop mechanisms." (TA2)
Generalization	"for loops, loops over a sequence of something, it doesn't need to be 1, 2,3,etc; the important aspect is iteration itself." (TA1) "many different functions students have seen/written highlight what is common for one function." (TA2)
Separation	"I can't give any good example, possibly inheritance and subclasses" (TA2)
Fusion	"The class concept in object-oriented language is a good example of this, where several critical aspects need to be addressed simultaneously, so that full understanding can be gained." (TA2)

Both approaches, the *problem focused* and the *method focused* approach, were used according to the teaching assistants.

TA₁ says that the method focused approach was applied primarily and only occasionally some few problem focused

aspects were used as a teaching concept. The problem focused approach is more difficult to apply at basic level courses, while the method focused approach seems more appropriate for beginners. This teaching assistant summarizes his experience by emphasizing that students frequently have own

ideas how to attack a problem, but they lack experience to structure the approach in such a way, that it can be expressed in a programming language and becomes programmable. So the method needs to be focused rather than the problem.

TA₂ is not that firm about the preponderance of the method focused approach. Rather this teaching assistant proposes to mix both ways of teaching.

7. Discussion

The introduction of some variation techniques in a Python based course on scientific computing was well received by the students. A majority of the participants characterized it as being “fun” and useful. This is rather encouraging, considering Dennis Beach's results, [1], who describes how students prefer a boundary-setting pedagogy, favoring reproduction (“mimesis”) of knowledge in formulaic forms to a pedagogy leaving room for their own creativity. This also runs counter the findings cited by Williams, [16], that the use value of education comes second after its exchange value. This, again, is described by Beach, [1]. Thus a teacher sees as commonplace that students work hard to get the best jobs - rather than to accrue knowledge useful for their personal development, and students tell Beach “we need good grades to get good jobs” (rather than saying “we have to understand the stuff they teach us, so we get smarter”).

The result of learning is, in any case, knowledge relevant for a future job; especially so in the case of students studying our course, who are to become engineers. But to attain such knowledge means to transform one's way of life, to attain a new identity: in this case, that of an engineer. In the case of students of statistics, Petocz and Reid write about the importance of acquiring personal qualities directly connected to being a statistician. What is essential is thus not just learning the rules of a practice but identifying oneself with it, in our case with the use of programming. It seems plausible to see understanding of subject matter (and not just knowing how to answer tests) as the first step toward this end [12].

How these students value understanding can be seen in what they say about programming procedures: 13 of them claim to strive for understanding prior to planning the procedure.

It is also interesting that students of engineering (five out of 13), can appreciate a teaching method that shows them ways of discovering new possibilities rather than the right answers. Whether this is more than just a statistical fluctuation cannot be said on this stage of our research, but the very existence of such an attitude might be relevant.

Beach, [1], Beach and Player-Koro, [2], and Williams, [16], are trying to capture the essence of the teaching culture in Western educational systems. The main aspect they point to is that getting grades is the overruling educational objective which in consequence leads to the supremacy of rote learning. “Playing the grades game” (Beach) makes the student give up trying to understand (since teachers do not demand it and tests can be passed without it). There is a hint of that in the answers of the teaching assistants. When they claim the method

focused teaching culture should be applied to beginners, it might be because they are accustomed to a rigid way of presenting the subject. This is merely hypothetical and should be more investigated.

Besides, as Kwan and Ng, [7], point out, repetition can be a “road to understanding” - it is, in fact, in the East Asian teaching culture. Here they discern “meaningful memorization” as opposed to “mechanical rote learning”.

Students mentioned some critical aspects (features), which were class, function, generator, dictionary, test, profiling, input-output, Debugging and file management [9].

During our prototype presentations we explained to the students the principles of the pattern of variation, so that most of students could apply the definition to find examples when we interviewed them.

Students confirmed that *for* is a *contrast* of *while* which are values of dimension of variation for a critical aspect *loop*. They were able to relate classical for-loops to other concepts of iterations like *generator* and *list comprehension*. Furthermore they could see the contrast between some other programming aspects for example *list* with *array* and *dictionary* and also between *function* and *class*.

Some of the students confirmed that learning by contrasting alternative concepts is important, but they could not mention any example. One of students considered that the course was a chain of contrast aspects. Loops use different forms of iterable objects in Python.

The generalization from classical lists of numbers (cf. Python's *range* command) to other objects introduces an additional dimension of variation for the critical aspect *loop-items*, cf. the *generalization principle* [8].

Dividing a program into different functional units was the students' favorite example for a *separation* principle. After we have different functions for one program we can vary one of the functions that focused on, while other functions remain unchanged (here the critical feature is a function and the other subfunctions *function1*, *function2*, etc. are values of the dimension of variation).

There were some students who avoided these principles and attempted to delete everything when the output is wrong.

Students mentioned that it is difficult to control all parts of a program simultaneously and understood that the concept of *fusion* can remedy this difficulty. In computer science in particular within object oriented programming a typical example for the *fusion* principle is the *class* concept, which allows to pack, control and access several methods (functions) in a single object. In scientific computing an example for such an object might be quadrature with different algorithms like the trapezoidal rule as its methods. By putting these functions inside one class all methods can be controlled simultaneously.

Students meant that it is practice to focus first on lines and then on the methods which contain these lines and finally all methods can be controlled by on class.

Students confirmed also that “print” command is also an instrument which can control all outputs simultaneously. Calling one function by another function is also an example of the *fusion* principle. There were other students who had never

thought about such simultaneous controls. Most students do not seem to have understood the variation techniques. When asked about *separation* and *fusion* they described procedures and not aspects. Separation in Marton's and Booth's variation theory is the separation of aspects of an object, so that a new possibility to see the object is opened [10]. Most students, however, separated parts of the program and saw them as aspects. The students had better luck with contrast, this being probably easier to grasp.

The object of learning is scientific computing which in the answers of the students is described by three dimensions:

- effectiveness,
- a manifold of approaches,
- affective (fun).

These dimensions show what positions students take. For some students variation techniques in teaching scientific computing are useful because they make it easier to find the best solutions to problems. Other students say that it is important to see that there are many possible approaches. Some of these latter students even say it is "fun" to discover these possible approaches. So, we can see that there are three things which can appear important to students:

- to find the best way to solve problems and doing it swiftly,
- to see that there are many possible approaches,
- that solving a problem can be fun because there are so many different possible approaches.

8. Conclusions

The use of variation techniques seems promising in teaching scientific computing. We have not conducted a formal evaluation yet, but the method is well received by the students. What stands out is their interest in different possibilities to design a solution, not limited to finding the best solution.

For our ongoing investigation on how variation theory is best implemented in similar courses in different socio-cultural environments the presented study proved to be a good and promising platform.

After having implemented the course identically at Soran University/Kurdistan we will conduct the same interviews to find out which changes have to be made in the teaching concept when changing the cultural environment.

We have also come to the conclusion that students and even teaching assistants were not familiar with variation theory and the concept of patterns of variation, but when we showed this technique via presentations they could in the interviews directly identify the process and could describe some examples.

Principles of variation theory are certainly used in many courses and in particular variation theory is implicitly implied in nearly all courses in scientific computing. But the presented study shows that it can be of advantage for the quality of teaching if variation theory is introduced to all students and teachers involved in the course.

We even believe that a more active application of variation

theory can bridge over differences in the acceptance of a course in different socio-cultural environment. This point will be the topic of a future study.

References

- [1] Dennis Beach: The paradoxes of student learning preferences. *Ethnography and Education*, 3(2): 145--159, 2008.
- [2] Dennis Beach and Catarina Player-Koro: Authoritative knowledge in initial teacher education: studying the role of subject textbooks through two ethnographic studies of mathematics teacher education. *Journal of Education for Teaching*, 38(2): 115--125, 2012.
- [3] Shirley Booth: Learning to program: A phenomenographic perspective. PhD thesis, University of Gothenburg, 1992.
- [4] Pierre Bourdieu: The forms of capital. *Readings in economic sociology*, 4: 280, 2008.
- [5] Alan Bryman: Social research methods. Oxford university press, 2012.
- [6] Claus Führer, Jan-Erik Solem, and Olivier Verdier: Computing with Python. Pearson, 2013.
- [7] Kwan, T., Ng, F.P. & Chik, P. "Repetition and Variation". In Marton, F. & Morris, P. (Eds.) What Matters? Discovering critical conditions of classroom learning. Chapter 3, pp.39-58, November 2002. ACTA: Universitatis Gothoburgensis.
- [8] Mun Ling Lo: Variation theory and the improvement of teaching and learning. Göteborg: Acta Universitatis Gothoburgensis, 2012.
- [9] Ference Marton and Paul Morris: What matters? Discovering critical differences in classroom learning, Göteborg: Acta Universitatis Gothoburgensis, 2001.
- [10] Ference Marton and Shirley Booth: Learning and awareness. The educational psychology series, Routledge, 1997.
- [11] Ference Marton, Amy B.M. Tsui, Pakye P.M. Chik, Po Yuk Ko, and Mun Ling Lo: Classroom discourse and the space of learning. Lawrence Erlbaum, 2004.
- [12] Peter Petocz and Anna Reid: On becoming a statistician - a qualitative view. *International Statistical Review*, 78(2):271--286, 2010.
- [13] Ulla Runesson: Beyond discourse and interaction. Variation: a critical aspect for teaching and learning mathematics. *Cambridge journal of education*, 35(1): 69--87, 2005.
- [14] Jarkko Suhonen, Janet Davies, Errol Thompson and Kinshuk: Applications of variation theory in computing education. Proceedings of the Seventh Baltic Sea Conference on Computing Education Research-Volume 88, 217--220. Australian Computer Society, Inc., 2007.
- [15] Lev S. Vygotsky: The collected works of L.S. Vygotsky: Volume 1: Problems of general psychology, including the volume Thinking and Speech. Springer, 1987.
- [16] Julian Williams: Use and exchange value in mathematics education: contemporary CHAT meets Bourdieu's sociology. *Educational Studies in Mathematics*, 80 (1-2): 57--72, 2012.