International Journal of Engineering Education Vol. 29, No. 2, pp. 403–414, 2013
Printed in Great Britain
0949-149X/91 $3.00+0.00

Gender-Aware Course Reform in Scientific Computing*

ELISABETH LARSSON, STEFAN PÅLSSON, JARMO RANTAKOKKO, LINA VON SYDOW and MICHAEL THUNÉ
Department of Information Technology, Uppsala University, Box 337, SE-75105 Uppsala, Sweden. E-mail: lina.von.sydow@it.uu.se

A holistic approach was used to thoroughly redesign courses in Scientific Computing at Uppsala University. The objectives were two-fold: to improve the learning outcome for students in general and to make the courses more appealing to women students in particular. The redesigned courses include a combination of learning activities motivated by previous research on preferences of women students regarding learning environments and educational approaches. Moreover, particular care was taken to design the courses according to the principles of constructive alignment. This was achieved by structuring the course content into thematic modules, where each module was organized into different learning activities with several cycles of action, observation and reflection. Indications that the resulting course design fulfils the objectives stated above are, for example: that the students get a clearer and more coherent view of the subject; that they consider the courses to be well-structured with well-connected activities; and that the student–student and student–teacher interactions are increased. The new course structure is clearly appreciated, especially by female students, and considered to be important for the learning process. The conclusions are supported by qualitative as well as quantitative evidence.

Keywords: scientific computing; gender; holistic; constructive alignment; action research

1. Introduction

1.1 Background and motivation

In the Swedish university system, the M.Sc. degree programmes in Engineering are among the most prestigious study programmes. There are employers, students and faculty who consider them to be elite programmes, and many of their graduates obtain leading positions in Swedish industry. Thus, it is particularly troublesome from a gender equality perspective that women are under-represented among the students in these programmes. Among Swedish university students in general, women are instead slightly over-represented.

In the course reform project described here we consider education in our particular subject area, Scientific Computing. Most of the Scientific Computing courses at Swedish universities are taught within the five-year Engineering programmes, and there is a progression from mandatory courses at a basic level to eligible courses at an advanced level. The Scientific Computing courses inherit the under-representation of women from the programmes involved, but for the eligible courses there is at least a potential to increase the numbers.

Scientific Computing is about using computers and mathematical models to simulate phenomena related to nature or technical artefacts. This is an interdisciplinary area, in the intersection between Mathematics, Computer Science, and various fields of Science and Engineering where computer simulations are used for research and development. Educators in Scientific Computing struggle with the task of creating possibilities for students to 'see' this fascinating interdisciplinary connection between different areas. Before the redesign of the courses, the majority of our students demonstrated an adequate learning outcome concerning particular algorithms, methods for analysis, and even solutions of realistic application problems. Despite this, learning was unsuccessful, because course evaluation questionnaires revealed that most students experienced these different kinds of knowledge as fragments that did not align with each other into a coherent 'whole'. They did not perceive Scientific Computing as a coherent discipline.

The main objective for our course reform was to improve the retention of female students in Scientific Computing. More specifically, we wanted female students to become sufficiently interested in Scientific Computing to continue with optional advanced courses after having taken the mandatory introductory courses. The secondary objective was to improve the learning outcomes for the student body as a whole to mirror our understanding of the subject as coherent and relevant better.

Based on the following argument we assumed that our two objectives were correlated in such a way that they prompted the same kind of course redesign. Research on women's preferences in education has shown that women students value learning experiences that contribute to sense making [1, 2]. By 'sense making', we mean the process of making sense of what you learn, for example by coming to see relations between various aspects of the subject that you are studying and by seeing an overall purpose in the subject matter in terms of

* Accepted 7 December 2012.

applicability, social relevance and/or other features. Consequently, we assumed that the fragmented learning outcome for our students in general was at the same time a key factor behind the relative failure to attract women students for continued studies in Scientific Computing. The success of the redesign is an indication of the validity of this assumption.

We used an action research approach to redesigning the courses in Scientific Computing at Uppsala University with the aim of supporting sense making. We consider action research to be an iterative research approach, where actions are planned with regard to certain objectives, for example to improve learning in some respects. The actions are then carried out and observations are made that provide material for reflection. The conclusions from the observations and reflections may provide suggestions for further or revised actions that are then planned, carried out, etc. Each iteration consisting of planning, action, observation and reflection is said to be an action research cycle (see, e.g., [3]).

The redesign did not consist in introducing very different educational activities. Rather, we took a holistic approach (cf. [4]) where typical educational activities in Scientific Computing were realigned and modified in a way that was informed by previous results from pedagogical research. The changes were made with two main purposes:

1. to align the activities with each other and with the course goals;
2. to increase student–student and student–teacher interaction.

As a structuring mechanism for the alignment, we organized the activities to support students' cycles of action, observation, reflection and revised understanding. These seemingly minor modifications did in fact imply (i) a thorough restructuring of the internal relations between the learning activities and (ii) significantly more active engagement of the students.

To monitor the effects of the course redesign process continuously we have used the course evaluations that are collected at the end of each course, implying that we have a rich collection of data both from before the project started and during the project. The qualitative effects on the observed learning outcome were very encouraging, indicating that the redesign has successfully fulfilled its objective concerning sense making. Quantitative evidence also supports that the new course structure is clearly appreciated by female students.

1.2 Related work that informed the redesign

Results from previous research strongly indicate that women students value sense making. In their overview of gender issues in postsecondary computing education, McGrath Cohoon and Aspray [2] mention several aspects of sense making in an educational context, such as the importance of real-life applications and student centred pedagogy.

In our course redesign project, we were particularly inspired by a research report from Albinsson *et al.* [1]. In this report a summary of different projects to increase the number of women students in M.Sc. Engineering programmes in Computer Science in Sweden is presented. A number of suggestions for changes and wishes from the female students are given where some can be described as:

- Increased possibility of:
  - seeing how different parts of their knowledge connect;
  - understanding the usefulness of their knowledge in the study environment and future professional life;
  - seeing relations between subject area knowledge and reality.
- Technical discussions should have a social perspective.
- More project and group work.
- More oral presentations and writing of reports.
- Better contact and communication with teachers and fellow students.

The items in the upper part of this list are clearly about sense making. The additional items about group work, reports, and communication also have a relation to sense making. They concern forms of education with a large degree of student activity, as well as student-student and student-teacher interaction. When appropriately aligned with course goals and forms of examination such activities have the potential to contribute to sense making [5].

In a seminal paper based on empirical observations and phenomenographic analysis, Marton and Säljö [6] made a distinction between a deep and a surface approach to learning. This was further explored in subsequent studies [7]. Students taking a surface approach focus on memorising and reproducing pieces of information, whereas a deep approach is characterised by a search for meaning, in that the student actively tries to understand the relation between various aspects of the subject matter at hand. The learning approach is *not* a personality attribute of the student. An individual student can be observed to use a deep approach to learning in one context and a surface approach in another context. The challenge for educators is to design learning environments and learning activities that trigger a deep approach to learning (see, e.g., [8]). This has been a key ambition for our course redesign, due to the close relation between a deep approach and sense making.

1.3 Traditional education in Scientific Computing

In order to explain the changes we have made, there is need for a very brief introduction for readers who are unfamiliar with Scientific Computing. Scientific Computing is concerned with the simulation of phenomena in areas of application such as physics, biology, finance and technology. The simulations are based on mathematical models of the phenomena of interest. A typical everyday example is a weather forecast, where future weather is simulated on the basis of a mathematical description of the weather physics. The simulation consists of solving the equations in the mathematical model and presenting the results in an understandable form, for example as a weather map. The equations are 'unsolvable' in the sense that they cannot be solved by hand using paper and pen. The complexity of the problem precludes such a solution. The computations to solve the problem rather require the use of so-called 'numerical methods', i.e., computer-based algorithms to solve mathematical problems. In real-life applications, such as weather forecasting, huge amounts of computational operations are involved. It is thus necessary to write an efficient computer program to carry out the simulation within a reasonable time. The solutions achieved in these numerical computations are always approximations and one part of the subject is to interpret and validate the numerical results.

In summary, to understand what Scientific Computing is, one has to be able to grasp the whole picture and see the relation between different aspects:

- application areas (i.e., areas where various phenomena are studied via computer simulations);
- numerical methods;
- computer science aspects, such as programming techniques.

The challenge for the teacher in a beginners' course is to organize learning activities that provide possibilities for the students to discern these aspects and to see the connection between them. Only then can the students make sense from the material.

The traditional way of teaching Scientific Computing does not achieve this. A partial explanation is that the traditional Scientific Computing beginners' course is largely a fragmented presentation of various aspects of the subject. Teachers may lecture about theory without providing sufficient context to help the students to see the purpose of the theory or demonstrate how to solve small model problems by hand, without providing the students with a rationale for why these problems are relevant.

Another part of the explanation is that the traditional Scientific Computing beginners' course lacks constructive alignment. Briefly, constructive alignment means that curriculum objectives should be clearly stated, that assessment should be made with regard to those objectives and that learning activities should be designed to provide the students with training towards achieving learning outcomes according to the objectives. In the traditional Scientific Computing course there is too much emphasis on abstractly formulated, small model problems to solve by hand. This does not align with the overall goal to make students aware of real-life applications where computers are required in order to perform the computations. A traditional course in Scientific Computing consists of:

- lectures,
- problem solving classes,
- computer labs and assignments,
- final exam.

In all these activities, except for computer labs and assignments, the emphasis is completely on theory and small-size model problems. The lab and project assignments are intended to provide the 'big picture' of the subject, but students fail to see the connection between these assignments and the material covered in lectures and problem solving classes. We claim that this lack of alignment is why traditional education in Scientific Computing fails to help a majority of students attain a coherent view of the subject. In fact, if Scientific Computing consisted of solving small problems by hand, then the subject would truly lack purpose. Sadly, this is the impression that many students have acquired after a traditional first course in the subject.

The course activities listed above are typically organized in the order in which they are listed, i.e., theory comes prior to practical work and applications. This might seem perfectly natural from a teacher point of view, first explain the theory and then illustrate the theory on the computer screen. However, if a student fails to understand the theory, the computer lab based on the theory will not make any sense whatsoever. It illustrates and gives answers to questions never raised by the student. We claim that this is one of the factors leading to students' fragmented understanding in traditional introductory courses in Scientific Computing.

1.4 The reformed courses in context

The courses that we set out to redesign were Scientific Computing I and Scientific Computing II at Uppsala University. During the project period the course content was redistributed and a third course, Scientific Computing III was introduced as part of a syllabus change*. Thus, when we discuss the different courses below, we refer to the changed course structure. At least one, and in most cases two, of these courses are included in the curriculum for the study programmes at the Faculty of Science and Technology. A typical course instance involves around 100 students, coming from different study programmes. Lectures are given for the whole group. For problem solving sessions and labs, students meet in subgroups of about 25 students per group. Each academic year several instances of each of the two courses are given, involving a total of about 1100 students per year.

Each of our two courses corresponds to 5 credit points. This is a measure of the 'size' of the course and also reflects the expected workload for the students. One year of full-time studies corresponds to 60 credit points.

In most of the study programmes, Scientific Computing I appears in the first-year curriculum. The course requires only very elementary university mathematics and no previous knowledge of computer programming is assumed. The course syllabus encompasses an introduction to Matlab programming, an introduction to numerical methods, and an introduction to numerical analysis. The kinds of mathematical models addressed are linear systems of equations, integrals and non-linear equations. Scientific Computing II is typically in the second-year curriculum and addresses numerical solution of ordinary differential equations, curve-fitting (interpolation and least-squares approximation), and Monte Carlo methods.* The combined contents of Scientific Computing I and II are typical for introductory courses in Scientific Computing at universities worldwide, often under alternative names, such as Numerical Methods. This is reflected in a large supply of textbooks for this kind of course (see, for example, [9, 10]).

2. Methodology

2.1 Holistic and systemic approach

The course redesign presented here takes a holistic approach. Thota [4] argues eloquently for such an approach and also demonstrates how she successfully used it in an introductory object-oriented programming course. In advocating the holistic approach, she cites Ramsden [11]: 'We cannot reduce a relational view to one which concentrates on the parts of the process. When it comes to applying our understanding to improving the process, a relational view has distinct advantages'.

In our interpretation, the holistic approach to teaching is analogous to the holistic or deep approach to learning. As educators we should take a relational view on the various elements of the teaching and learning process and design the learning environment and the learning activities in a way that takes these relations into account. By taking a holistic approach, our course redesign addresses a complex interplay between various 'parts of the process'. To mention only one example of such an interplay: Our previous experiences with realistic application problems was that students tended to take a surface approach when working with these problems, since they felt stressed by the difficulty of the problems and interpreted the situation as if they were not allowed to ask the teacher for help. By *both* introducing challenging mini-projects *and* encouraging students to interact with the teachers in working on these problems, we noted that students seemed more relaxed and were more prone to adopt a deep approach to the mini-projects. This is one example of many in our project, pointing to the relevance of a holistic approach to course design.

A systemic approach is a kind of holistic approach, where the whole learning environment is regarded as a system, including not only the students, teachers, learning resources and learning activities, but also such issues as institutional policies and support. In a recent paper, Barker, Cohoon and Thompson [12] propose a systemic change model of undergraduate education. Their particular focus is on undergraduate computing and the objective is to accomplish gender parity. As expressed by Barker, Cohoon and Thompson: 'Rather than view women as needing to be modified or repaired to fit the system, this model advocates changing the system to fit the needs of all students.' This is precisely in the spirit of the course reform project described here. The model proposed by Barker and McGrath Cohoon was not published when this project started, but in retrospect our project can be seen as a case study for three of the components in their model:

1. retention through pedagogy,
2. retention through student-student and student-faculty interaction,
3. institutional policies and support.

As for retention through pedagogy, Barker and McGrath Cohoon, distinguish two aspects: (i) Col-

* A Faculty decision to streamline the size of the courses meant that the syllabus was changed so that both courses became significantly smaller in terms of credit points and with less content. An additional course, Scientific Computing III, was introduced to cover the numerical solution of partial differential equations. At the beginning of the project period this topic was addressed in Scientific Computing II. It can be observed that the work made in the initial stages of our project gave the courses a clear modular structure that turned out to be beneficial for meeting the challenge of restructuring the courses to meet the new Faculty requirements on course size.

laborative Learning and (ii) Meaningful and Relevant Assignments. As explained above, both of these are key elements in our course redesign. Moreover, our focus on collaborative learning and frequent student-teacher feedback implies increased student-student and student-faculty interaction. In addition, the activities we use are in line with well-established best practices recommended in pedagogical literature, e.g., Felder and Silverman [13] and McKeachie [14].

Finally, with regard to institutional policies and support, Barker and McGrath Cohoon claim that change in higher education is more likely to occur when there is visible support from upper administrators and leaders. These prerequisites were clearly at hand in our project: gender-aware education is strongly supported by policy documents at Uppsala University; the project had a reference group consisting of educational leaders on central university and faculty level; one of the authors (MT) was Counsellor to the Vice Chancellor on gender equality issues during a five-year period that included the years when the project was running; and one of the authors (SP) is Director of Studies at the Division of Scientific Computing since more than a decade.

2.2 The course redesign in brief

As described above, in order to meet the two-fold goal of improving the retention of female students and to improve the learning outcome for students in general, the objective of the course redesign was to support students' sense making better. To this end, we used constructive alignment as a key approach. As a means to achieve such alignment, we designed learning activities to engage students in a learning process inspired by the action research model. These will be elaborated on below.

For the courses in Scientific Computing in our project, the curriculum objectives, both before and after the reform, can be summarized into four main objectives. After having taken the course, the student should be able to: apply the numerical algorithms presented in the course; use and explain the concepts introduced in the course; apply various methods of analysis to assess properties of the numerical algorithms; make arguments about the advantages and disadvantages of using a certain numerical algorithm to address a given problem. Even prior to the course redesign, learning activities were designed to provide training towards these objectives. As mentioned in Section 1, the learning outcomes were satisfactory, in that a majority of the students managed to demonstrate learning outcomes according to the objectives. However, from the point of view of sense making, there is an additional objective, not explicitly stated in the curriculum, namely that the students should be able to see the relations between various aspects of the subject and to see an overall purpose of the subject matter in terms of applicability, social relevance, etc. Prior to the course reform, this objective was not met, as discussed in Section 1. The constructive alignment in the course reform project was made with the purpose of achieving alignment with regard to the objective related to sense making.

To achieve such alignment, we made a number of seemingly minor modifications that we hoped would have a large effect taken together. In the reformed courses Scientific Computing I and II, the course matter is organized into thematic modules, where each module corresponds to 1–1.5 credit points. The 'theme' for such a module is typically a certain type of mathematical model. The activities in the module are intended to provide the student with: examples of real-life applications that can be modelled by this type of model; examples of numerical methods that can be used to perform simulations based on such models; experience of how to implement such numerical methods in computer programs; insight into important properties of such numerical methods and how to analyse and argue about those properties; and, finally, experience in carrying out simulations of some 'real' phenomenon modelled by this type of model. As an example, students in Scientific Computing II carry out simulations of the circadian rhythm in a cell, using models from a research paper in systems biology [15]. In one course module the circadian rhythm is modelled by a system of nine differential equations that the students solve numerically to simulate the rhythm. In another course module a stochastic model is used and simulations are carried out with a Monte Carlo method.

Each thematic module includes the following learning activities, in chronological order:

1. computer lab,
2. lecture,
3. workout,
4. problem solving,
5. mini-project.

These activities serve different purposes and are aligned with the goal that students should be able to make sense of the subject matter and thus get a relevant overall understanding of what Scientific Computing is. All of these activities except the lectures are student centred. For a more detailed description, see [16].

All of these activities are aligned with the four main curriculum objectives discussed above, in that they are designed to provide training towards them. However, in order to achieve alignment with the additional objective of sense making, i.e., to help students to see the relations between the different aspects of the subject matter as represented by the four main objectives, we further aligned the different learning activities with each other by making them support cycles of *action*, *observation*, *reflection*, *tentative understanding*, followed by *new action*, *observation*, *reflection*, *revised understanding* for the students. In other words, the course is designed to engage the students in a learning process inspired by the action research model [3]. Together, the learning activities in each module constitute two action research cycles for the students. The computer lab includes action, observation and reflection. This reflection is continued during the lecture, which is based on the lab in order to support that reflection better. The lecture should lead to a tentative understanding that is tested and revised or reinforced in the workout. Then the problem solving and mini-project activities provide new action, observation, reflection and revised or reinforced understanding.

3. Results and discussion

3.1 Reflections and observations

Not only did we design the courses to support a learning process inspired by the action research model. We also used an action research approach in planning and carrying out the course reform project. The project was carried out iteratively, with cycles of *planning*, *action*, *observation*, *reflection*, followed by *revised planning*, *action* [3]. In this way, we redesigned the two introductory courses Scientific Computing I and II at Uppsala University. The iterative process involved fifteen course instances with a total of ca 1500 students over a two-year period (11 instances of Scientific Computing I and 4 instances of Scientific Computing II). For us, each course instance constituted one action research cycle. Here, we will briefly summarize the reflections and observations made during these cycles. The discussion will be centred on the different learning activities. The evidence reported in a subsequent section further supports the reflections and observations made here.

3.1.1 Computer labs

Computer labs are standard ingredients in Scientific Computing courses. The novel aspect of our course design is that the computer labs come *before* the lectures and are intended to generate questions rather than answers. During the lab sessions, students work in groups of 2 or 3 students. We observed that discussions with fellow students about the lab assignments support students' reflection and tend to trigger a deep learning approach.

With their focus on group work, the computer lab sessions are designed to be *collaborative learning* experiences (cf., e.g., [17, 18]). This is one of the aspects of the systemic change model recommended by Barker, Cohoon and Thompson [12]. They point at several studies to support the claim that collaborative learning is beneficial for retention (e.g., [19, 20]).

A teacher is present during the lab sessions to give advice and feedback when needed. We observed that this opportunity for the students to interact individually or in small groups with the teacher helped to establish a more 'personal' although still professional relationship between students and teacher. This was instrumental in making students feel comfortable and confident in the learning situation, which in turn made them less stressed, and consequently more prone to adopt a deep approach to learning. This agrees with the statement by Entwistle [21] that in teaching 'it is generally explanation, enthusiasm and empathy which are most likely to evoke a deep approach'. The importance of a high degree of student-faculty interaction is also emphasised by Barker, Cohoon and Thompson [12] in their systemic change model. In our course design, student-teacher interaction in smaller groups is a key to a good classroom climate. There is plenty of evidence that a classroom climate that reduces students' anxiety enhances the learning outcome (see, e.g., [22]).

3.1.2 Lectures

In the traditional course design, each new topic is introduced in a *lecture* before it is practised in a lab session. Prior to the course reform project we observed that students had difficulty benefiting from the lectures, since they had no previous experience to which to relate. In our redesigned courses, the computer lab comes before the lecture and the role of the lab is to give students some experience of the kinds of computations and applications on which that the particular course module is focused. This provides a context to relate to in the subsequent lectures, where the teacher can explicitly refer to the lab, so that new theoretical elements can be motivated by observations that students have made during the lab session. We have observed that this close interaction between lab and lecture supports students' reflections and leads to a more coherent understanding of the subject matter. From the point of view of constructive alignment it is also very important that each computer lab contains at least one example of a realistic case where the computations would not be feasible to carry out by hand, but really require the use of a computer program.

3.1.3 Workout

The lectures are followed by a *workout* session. The name 'workout' was deliberately chosen to give associations to gym workout sessions. This should give the signal that the students are expected to be active. Moreover, all students know that a workout in the gym is hard work but makes you feel in better shape afterwards. We wanted the workout sessions in the 'scientific computing gym' to have the same effect.

Our workout sessions are another example of collaborative learning experiences with a high degree of student-student and student-teacher interaction. Students work in groups of 2 or 3, solving problems designed to give a basic understanding of numerical methods, algorithms and related theory. Discussions with fellow students about the problems help to support understanding. In addition, a teacher is present during the workout session, to provide further explanations when needed. Also, when a student group has completed a workout exercise, the teacher assesses their solution, in dialogue with the students. In this way, the workout session provides an additional opportunity for individual student-teacher interaction and feedback, and we observed the same benefits here as discussed above.

3.1.4 Problem solving and mini-projects

The next link in the chain of learning activities during a thematic module is the *problem solving session*. During the problem solving session, the teacher is available to give advice when needed. The goal is that when leaving the session the student groups should understand the problem and have a reasonable solution outline or algorithm formulated on paper. The work that remains for a complete solution of the problem is to be carried out as a *mini-project*.

The students get a realistic application problem to work on, in groups of 2 or 3. The problems are context-rich. As reported by Benckert [23], it has been demonstrated that Physics students working on context-rich problems in groups of 3 found this to be an engaging learning experience and that female students were particularly appreciative of this kind of learning activity. This agrees well with our observations and also relates to a component of the systemic change model in [12], namely *Meaningful and Relevant Assignments*. Understanding the relevance of the subject matter to their personal experiences, interests and career plans enhances the students' learning [13].

3.1.5 Guest lectures

As an additional component in relating the courses to real-life, one or two *guest lecturers* from industry or applied science were invited to present how they use scientific computing methods in their profession and how issues discussed in class have a practical impact. Students especially appreciated seeing that the methods they had learned about in the course are actually used outside the students' educational context.

3.1.6 Learning process

As mentioned above, the learning activities were designed and aligned to support a learning process inspired by the action research model. We observed that students' understanding evolved gradually during each thematic module. Via tentative questions during the lab sessions and theory-supported reflection in lectures and workouts, a preliminary understanding emerged. This was subsequently combined with the students' additional experiences during the problem solving session and mini-project, where we could note in discussions with students that they attained a more complete, often revised understanding. This supports the claim by Kember and Gow [3] that action research can be a viable instrument in a learning process.

3.1.7 Assessment

It is crucial for constructive alignment that the *assessment* of students' progress is aligned with the curriculum objectives. Following Biggs' recommendation, we distinguish between formative and summative assessment [5]. We have observed that formative assessment, where the students are explicitly encouraged to interact with the teacher for feedback while working on the assessment task, makes the students relax, and they become more prone to adopting an in-depth approach to learning.

The summative assessment is in the form of a written exam. We have introduced grading criteria inspired by the SOLO taxonomy [24]. For the top grade, we require students to demonstrate a relational understanding, i.e., that they are able to see and use relations between various aspects of the subject matter at hand. This is a central aspect of making sense of the subject. In order to assess this kind of understanding, we have introduced new kinds of exam problems that relate to real-life situations where the methods studied in the introductory course become meaningful. To test the students' ability to see the 'big picture' we have also used essay questions where students were required to write short essays. By introducing exam questions that relate to applications and that require a relational understanding of Scientific Computing issues, we have aligned the exams with the curriculum objectives. Our impression is that this has been of importance for the students' study approach in the course. Students realise that they are not likely to reach the understanding required for a high grade if they adopt a superficial approach and this insight make them more prone to taking an in-depth approach.

3.2 Evidence

In this section we present evidence supporting our reflections and observations reported in previous sections. We focus on how students experience the change, i.e. on student appeal, rather than student performance. To provide evidence supporting our reflections and observations, as reported above, we have used course evaluations that are conducted at the end of each course. These are part of the regular monitoring system at Uppsala University and have been used for several years. Thus, we have a large collection of students' comments from course instances prior to the course reform. The evaluation form is developed at departmental level and questions that are course-specific can be supplemented. The evaluations are always anonymous and incorporate quantitative as well as open-ended qualitative questions. Below, we discuss the quantitative and qualitative responses as two sources of input. The qualitative ones have been used to collect students' comments on their experiences of the redesigned courses. These were compared with corresponding comments made by students who took the courses before the redesign. There is a striking difference between these two collections of comments, indicating that our redesign has met its objectives. We also present statistics that show the outcome for some of the qualitative questions.

It should be noted that the group of teachers in these courses at Uppsala University include both women and men. For the results of the redesign project, we have not seen any difference depending on whether a particular course instance was taught by women, men, or both men and women (all these combinations occurred during the project, with courses taught by both men and women as the most common alternative).

We will now present these results in more detail, beginning with the qualitative ones.

3.2.1 Qualitative evidence

As mentioned above, the regular form of course evaluation has been used for several years, and as a result, we have a large collection of students' comments from course instances prior to the course reform. When these are compared with the course evaluation comments made by students who had taken the redesigned course, it is obvious that the students' view of the courses has changed. After the redesign:

- students got a clearer view of Scientific Computing as a topic;
- students considered the courses to be well-structured and with well-connected activities;
- students felt relaxed and encouraged to interact with the teacher;
- female students clearly appreciated the changes, while the men have a more neutral attitude.

It can also be noted that:

- in some study programmes the student attitudes and study culture significantly affects the way they look upon the course design.

Each item is described in more detail below. The first three items are clearly related to sense making, alignment, and the hypothesis that female students put a greater value on sense making. *Students got a clearer view of scientific computing as a topic*. Prior to the course redesign, comments made by students in the course evaluations often indicated a lack of understanding of what the course and the subject was all about. These comments were there even if the teacher and the course in general got high grades in the evaluation. In fact, this was one of the observations that triggered our course reform project. The quotes below are all taken from course evaluations in a traditionally taught instance of Scientific Computing I, spring term 2007, and can serve as typical examples (all quotes are translations from the Swedish original):

> What was the course all about??? Clearer explanation of where and how the course can be of use later. Was it a course about learning Matlab and how demanding it is for a computer program to do calculations? Or was it about something completely different?

> Difficult course to grasp.

> It was hard to get what we were doing, not until the end of the course did one begin to get what it was about.

Teachers in the course noted the same underlying problem and became frustrated. The students usually understood details in the course, but not the underlying motivation and overall meaning.

With the new course design, *this type of comment has disappeared completely*. Instead, as shown in the next section, many of the comments express how well the different activities are connected, how well they support the learning, and that the students have learned a lot. This indicates that the course has become well aligned and that the course contents make sense to the students.

Well-structured courses with well-connected activities. A key aim for the project was to change the course activities to make them well aligned and supportive for students' sense making and their overall understanding of the subject matter. As described in previous sections the courses are divided into modules and each module contains a computer lab, lectures, a workout, a problem solving session and a mini-project, in chronological order.

From our evaluation data for the redesigned course, it is clear that the different course activities are considered by students to be well-structured and well-connected. Some comments from course evaluations in the autumn term 2007 and the spring term 2008 exemplify this:

> Above all it's the interplay between different activities that has been good. To begin with a computer lab, then lecture, workout and end with problem solving gave understanding and well needed repetition. When one block was finished you grasped the content well. (Scientific Computing I, spring term 2008)

> To use the knowledge acquired in workout sessions, computer labs, and mini-projects was very good. It leads to consolidated knowledge and reduces the difficulties to prepare for the final exam, since you have used most of the elements of subject matter included in the course. (Scientific Computing I, spring term 2008)

> The course set-up has been super. To schedule the computer labs at the beginning of the course modules, so that they do not require any prior knowledge and results to present was really good. It felt stimulating and totally right. The workouts were very good as well. I particularly appreciate the group work and the presentation of material (i.e., no written report to hand in, but discussions with the instructor during the workout session). (Scientific Computing II, autumn term 2007)

> It was good to have seen and tested the course content (theory) in the computer labs before the lecture, that way you could relate to what was said. (Scientific computing I, spring term 2008)

It is notable that the different course activities are seen as a whole to a larger extent than before the course reform. It is not one activity that supports the learning, but rather the whole structure of activities. Again, this indicates a well-aligned course.

Some students also commented explicitly on the relaxed atmosphere that encourages students to interact with the teacher. An example:

> The teaching has been well structured and the problems are carefully explained step by step, and it has been possible for everybody to ask without feeling stupid, if you did not understand. (Scientific Computing I, spring term 2008)

The significance of student attitudes, study culture and study programmes. The quotes and results presented so far have all been positive to the changes. There are also students who are negative or less positive. An interesting observation is that the number of less positive students is strongly correlated to the study programme. In most programmes there are only a few negative students, whereas we experienced a polarized opinion in one particular study programme, where approximately half of the student group were negative to the changes. The following are examples of negative comments from students in that study programme:

> Turn back to the old structure. I believe that would make both teachers and students feeling better. It is indeed proven that you learn by watching somebody else solving problems.

> More methods, and less understanding in the final exam. There are too few credit points to motivate the time needed for understanding. Understanding is really a good thing and it is good in the long run, but if one should go for understanding all courses you wouldn't manage to cope with the entire study program.

> Return to traditional lessons where the teacher solves problems. Old-fashioned but efficient.

Most comments have to do with efficiency. These students feel that it is more efficient to let the teacher present problems and solutions for the students. They do not personally see 'understanding' as a goal for their participation in the course, because they feel that there is not enough time for understanding. It can be observed that the success of our course redesign in all the other study programmes provides a convincing counter argument to this attitude.

The course reform from a teacher's perspective. In addition to the project members, several other teachers have been involved in the various instances of the redesigned courses. The teacher experiences can be summarized as follows:

- better contact between teachers and students and more teacher-student feedback and interaction, than prior to the course reform;
- much more student activity and discussions between students;
- a general feeling that the students understand what Scientific Computing is all about;
- the students learn successively during the course;
- teaching is much more fun now than prior to the course reform.

A common fear when working with educational reforms is that they will result in more work and a higher cost. This is not true in this case and a cost-neutral reform was in fact one condition for the project. Some parts of this reform imply time saving, or rather reallocation of teacher resources. For instance, presenting solutions on the blackboard has more or less been replaced with individual discussions with student groups in workout and problem solving sessions. Time-consuming marking of assignments has been reduced and, to a large extent, been replaced with discussions and oral feedback.

3.2.2 Quantitative evidence

Female students appreciate the changes. The answers to the questionnaires were anonymous, but can be

The course is divided into modules with computer lab, workout, problem solving in each module. How has this structure worked?

	#	Graph (%)	%	Σ
1 = **Bad**	A		0	0
	B		9	2
2	A		0	0
	B		0	0
3	A		13	2
	B		32	7
4	A		40	6
	B		32	7
5 = **Very good**	A		47	7
	B		27	6
No opinion	A		0	0
	B		0	0

Fig. 1. This graph summarizes answers to a question about students' experiences of the course. The answers are divided into two categories: answers by women (A) and men (B), respectively. Response rate: 70% (out of 53 students).

The computer labs come before the lectures. Has it worked well?

	#	Graph (%)	%	Σ
1 **= No, not at all**	A		0	0
	B		0	0
2	A		0	0
	B		9	2
3	A		27	4
	B		45	10
4	A		27	4
	B		18	4
5 **= Yes, definitely**	A		47	7
	B		23	5
No opinion	A		0	0
	B		5	1

Fig. 2. This graph summarizes answers to a question about students' experiences of having computer labs before the corresponding lectures. The answers are divided into two categories: answers by women (A) and men (B), respectively. Response rate: 70% (out of 53 students).

divided into male and female categories. When studying these, an interesting observation can be made, namely that female students clearly appreciate the new course structure and consider it to be important in the learning process. The male students show a more neutral attitude. The same trend can be noticed in all course instances where we have used the new course structure.

Two figures taken from a course evaluation in Scientific Computing I, spring term 2008 can serve as examples. The figures show students' opinions on the course structure in general (Fig. 1), and the construction 'computer lab before lecture' (Fig. 2). In both figures, Category A and B denote female and male students, respectively. On the first question, 87% of the women answer 'Good' or 'Very good', and on the second question, 74% of the women give positive answers. The corresponding numbers for men are 59% and 41%, respectively.

The fact that a large majority of the women answer positively to these and similar questions indicates that our redesign has met its goal to make the two courses Scientific Computing I and II more appealing to women. At the same time, the qualitative evidence above indicates that the redesign has met the objective that students in general, women *and* men, have made better sense of the course and the subject area, as indicated in the previous subsections. In other words, even though the men do not express their appreciation of the new course structure as clearly as the women, their learning outcome has nevertheless benefited from the redesign.

4. Conclusions

The course reform project described in this paper had a two-year time frame. The objectives were twofold: to improve the learning outcome for students in general and to make the courses more appealing to female students in particular. In view of the empirical data reported in the previous section, we conclude that the redesigned courses meet both of these objectives. Consequently, we continue to use this course design in Scientific Computing I and II, and we are also applying the same ideas in some of the optional, advanced courses in Scientific Computing at Uppsala University.

Constructive alignment was achieved through activities that were consciously designed to address the various curriculum objectives. In addition, we changed the assessment to better reflect these objectives. Not only were the course activities aligned with the curriculum objectives, they were also designed to fit together into action research cycles for the students. The project results show the viability of this approach. This claim is supported by our observations and by student comments cited above, where students express how they experienced the activities to link together precisely in the way that we intended when we designed these action research cycles.

In summary, the course reform project reported here met its goals. It has led to a significantly better learning outcome for all students in our introductory Scientific Computing courses. In particular, it has led to female students appreciating the subject much more than before the course redesign. It is yet too early to see if the latter leads to more female students taking optional advanced courses in Scientific Computing in later years of the M.Sc. Engineering programmes. However, it is safe to claim that the prerequisites for this to happen have improved as a result of the project.

Acknowledgements—The project was supported by The Swedish National Authority for Networking and Cooperation in Higher Education (NSHU). A pre-study was funded by the Faculty of Science and Technology at Uppsala University.

We thank Mats Daniels from the Uppsala Computing Education Research Group for comments on a draft of this paper. Moreover, thanks are due to the members of the project reference group for valuable input during the project. The reference group members were: Maja Elmgren, Leader of the Council for Educational Development in the Faculty of Science and Technology; Åsa Kassman, Chair of the Board of Engineering Education; and Mia Lindegren, Head of Uppsala Learning Lab, all at Uppsala University. Last but not least, we are grateful to the two students who acted as reference persons in the project and provided very useful feedback: Görel Andersson and Ronak Naderi.

References

1. S. Albinsson, H. Andersson , D. Dervisevic and H. Sjöstedt, Genus och *programmering (in Swedish), Linköping University, Department of Computer Science, 2002.*
2. J. McGrath Cohoon and W. Aspray, Women's participation in postsecondary computing education, *Women and Information Technology*, MIT Press, Cambridge, 2006.
3. D. Kember and L. Gow, Action research as a form of staff development in higher education, *Higher Education,* 1992, pp. 297–310.
4. N. J. Thota, Developing a holistic approach to learning and teaching introductory object-oriented programming, Ph.D. Thesis, University of Saint Joseph, Macau, 2010.
5. J. B. Biggs, *Teaching for Quality Learning at University*, 2nd edn, Open University Press, 2003.
6. F. Marton and R. Säljö, On qualitative differences in learning—I, Outcome and process*, British Journal of Educational Psychology,* **46**, 1976, pp. 4–11.
7. N. J. Entwistle and P. Ramsden, *Understanding Student Learning*, Croom Helm, London, 1983.
8. P. Ramsden, The context of learning in academic departments, *The Experience of Learning,* 3rd (Internet) edn, Edinburgh, University of Edinburgh, Centre for Teaching, Learning and Assessment, 2005.
9. S. Chapra, *Applied Numerical Methods with Matlab for Engineers and Scientists*, 2nd edn, McGraw-Hill, International Edition, 2008.
10. M. T. Heath, *Scientific Computing—An Introductory Survey*, 2nd edn, McGraw-Hill, International Edition, 2002.
11. P. Ramsden, Improving teaching and learning in higher education: The case for a relational perspective, *Studies in Higher Education,* 1987, pp. 12:275–286.
12. L. C. J. T. L. Barker, Work in progress—A practical model for achieving gender parity in undergraduate computing: Change the system, not the student, *Frontiers in Education conference (FIE)*, 2010.
13. R. Felder and L. Silverman, Learning and teaching styles in engineering education, *Engineering Education,* 1988, pp. 78:674–681.
14. W. J. McKeachie, *Teaching Tips: Strategies, Research, and Theory for College and University Teachers*, 10th edn, Houghton Mifflin, Boston,1999.
15. J. Vilar, H. Y. Kueh, N. Barkai and S. Leibler, Mechanisms of noise-resistance in genetic oscillator, *PNAS,* 2002, pp. 99:5988–5992.
16. E. Larsson, S. Pålsson, J. Rantakokko, L. von Sydow and M. Thuné, *Gender-Aware Course Reform in Scientific Computing*, Swedish Council for the Renewal of Higher Education, 2010.
17. P. G. George, Using cooperative learning in the college classroom, *Thought & Action*, National Education Association, 1999, pp. 1999:33–38.
18. D. Teague and P. Roe, Collaborative learning—towards a solution for novice programmers, *Proc. Tenth Australasian Computing Education Conference*, Wollongong, New South Wales, 2008.
19. A. Astin, *What Matters in College? Four Critical Years Revisited*, Jossey-Bass, San Fransisco, 1993.
20. E. Pascarella and P. Terenzini, *How College Affects Students: Findings and Insights from Twenty Years of Research*, Jossey-Bass, San Fransisco, 1991.
21. N. J. Entwistle, Promoting deep learning through teaching and assessment: conceptual frameworks and educational contexts, *TLRP Conference*, Leicester, 2000.
22. C. Pierce, Importance of classroom climate for at-risk learners, *The Journal of Educational Research,* 1994, pp. 88:37–42.
23. S. Benckert, *Conversation and Context in Physics Education*, Swedish Council for the Renewal of Higher Education, 1997.
24. J. B. Biggs and K. F. Collis, *Evaluating the Quality of Learning: The SOLO Taxonomy*, Academic Press, New York, 1982.

Elisabeth Larsson is a senior lecturer at the department of Information Technology at Uppsala University. She has been teaching courses in Scientific Computing, first as a teaching assistant and then as a lecturer, since 1995. She was also a member of the gender equality committee at the faculty of science and technology 2003–2008, and chaired the same committee 2008–2011. Currently she is involved in an international EU-funded project aimed at strengthening the role and increasing the numbers of women taking part in research activities and decision making at European universities.

Stefan Pålsson is a lecturer at the department of Information Technology at Uppsala University and, since 1997, has been the Director of studies in Scientific Computing. As such, he has been managing pedagogical development and been active in numerous pedagogical projects at the department. His academic teaching experience began in the early 1990s, and over this period he has taught a range of courses. Currently he is also a member of the Council for Educational Development at the faculty of Science and Technology (TUR) at Uppsala University.

Jarmo Rantakokko has been a senior lecturer at the department of Information Technology at Uppsala University since 2000. He has been teaching and developing courses in Scientific and Parallel computing. For developing the pedagogy in these courses he has received a grant from the Swedish Council for the Renewal of Higher Education. Currently he is the program coordinator for the Master program in Computational Science.

Lina von Sydow is a senior lecturer at the department of Information Technology at Uppsala University. She has been teaching courses in Scientific Computing since 1989, first as teaching assistant and then as lecturer. She has been both a member of and the chair of the gender equality group at the department. For her pedagogical work she has received grants from the Faculty of Science and Technology and she was also the main grant holder for the project described in this paper, which was funded by The Swedish National Authority for Networking and Cooperation in Higher Education (NSHU). She has also served as Director of graduate studies at the department and Dean of education at the faculty, as well as a member of the Council for Educational Development at the faculty (TUR).

Michael Thuné is professor of Scientific Computing at Uppsala University. From 2003 to 2008 he was Counsellor to the Vice Chancellor of Uppsala University on gender equality issues. He was Head of Education at the Department of Information Technology 2009–2012 and is currently Head of Department in the same department, which has around 1000 full-time equivalent students per year. Thuné has been active in educational development for more than 30 years and received the Uppsala University Educational Award in 1996. Since 2001, he has been a member of the Uppsala Computing Education Research Group.