31

Learning to bridge classroom and computer laboratory activities in mathematics education

Vincenzo Auricchio, Giuliana Dettori, Simonetta Greco and Enrica Lemut
I. M. A., C.N.R.
Via De Marini 6, 16149 Genova, Italy

Abstract
When introducing computer laboratory activities in mathematics education, some training for in-service teachers appears essential, to help them revise school programmes and to address issues related to teaching habits, didactical planning and computer laboratory organisation. In order to make computers an effective support to mathematics teaching, we think that class and computer laboratory uses should be given equal cognitive importance. Training courses should be hands-on, provide models rather than recipes, and include the formation of working groups connected to the research world.

Keywords
Teacher education, classroom practice, curriculum development, innovation, mathematics, software

1 INTRODUCTION

Computer laboratory activities in secondary schools are more and more diffused across subjects in many countries, and in the past few years most schools have been equipped with personal computers and software packages. Though experiences with different topics are mentioned in the literature (Plomp and Pelgrum, 1991; Veen, 1993), computer laboratories are mostly used in connection with mathematics, due to its natural conceptual relationship to informatics. The introduction of computer laboratory activities in mathematics curricula has been proposed for different purposes in the official programmes of different countries. One approach consists of focusing mainly on raising computer abilities; mathematics is only a privileged application field (Selwood and Jenkinson, 1995). The opposite position considers the use of computer laboratories as a means to improve teaching and learning of mathematics, hence shifting the focus from informatics to mathematics.

In Italy, a National Informatics Plan (PNI, 1991) was proposed at the end of the last decade with the aim of introducing concepts, languages and methods of computer science in non-specialised secondary schools; a secondary aim was to apply them in mathematics teaching. This plan included a training phase for mathematics teachers, which did not result in really effective outcomes, basically for two reasons: the courses tried to fulfil both the above mentioned aims and did not sufficiently deepen either of them; the interaction between teachers and trainers was limited to the training period and no further assistance was provided during actual school work.

Some training for teachers appears essential (Burke, 1994) not only because some of them are still not familiar with computers, but also because it is necessary to modify mathematics programmes in some ways. This task is difficult because there is no previous teaching tradition to rely upon. A suitable revision program entails deciding whether to focus the activities on informatics introduction, or upon mathematics support. We think that the basic computer introduction should not be the responsibility solely of the mathematics teachers, but should instead use computer laboratory hours available to improve teaching of their discipline.

This paper is focused on mathematics rather than informatics, and on how to retrain mathematics teachers (which we will mention as teachers from now on) to fruitfully make use of computer laboratories. We argue that a suitable integration of class and computer laboratory activities is necessary to make it a valid support in mathematics education. This requires that teachers not only revise school programmes and teaching habits but also extend to the computer laboratory the didactic contract established in the class.

In this paper, we start from an analysis of computer laboratory characterisation; then we analyse problems faced by teachers when introducing computers in mathematics education; next we consider different relationships between class and computer laboratory; finally we discuss training of in-service teachers.

2 COMPUTER LABORATORY CHARACTERISATION

Computer laboratory activities in mathematics education can exploit, either for introducing or for deepening concepts, the characteristics of software of different kinds, such as educational software, commercial software or programming languages. For instance, in the case of algebra, spreadsheets can be used for introducing basic concepts such as variables and functions, symbolic computation systems can help in defining and applying algebraic manipulation rules, and graphical representation systems can provide an easy graphical resolution of equations. Using computers affects the learning environment, since it raises in students new mental images that teachers need to take into account (Noss and Hoyles, 1992), and provides consistent feedback (Tall, 1991), hence changing the traditional "pedagogical triangle" (Rosvik, 1994) of the relations between students, teachers and knowledge into a "pedagogical tetrahedron". However, computers cannot substitute for teachers, whose leading role remains essential.

There are different opinions about the role actually played by computers: on the one hand, some teachers consider it simply as a tool, like a text book or a compass, hence failing to exploit its full potential; on the other hand, others consider it like a real subject of interaction, without emphasising its limits from a cognitive point of view, hence not stressing what it can be expected to perform and what must always be done by the user. Computers should be considered rather as very particular tools with characteristics different from any other tool used in school: its potentialities stem from its capabilities of interaction, its limits from the fact that its interaction can follow only fixed schemes depending on the running program, without any personal possibility of changing the thread of discussion.

Using a computer requires abilities firstly of how to learn to interact with an operating system and with the syntax and the logic of different programs. Hence, every time a new software program is used, it is necessary to spend some time getting used to it. This delay, however, has the advantage of forcing the user to pay more attention to the mathematical elements that are being learned; for instance, using a symbolic computation system can help students to become aware of the difference between semantic and syntactic rules.

Computers allow fast computations, visualisations of two- and three-dimensional objects, simulations of casual events, and this offers several advantages. It is possible to:

- discuss the same mathematical concept from different points of view, which usually leads to a deeper understanding and a better mastery of concepts (Duval, 1994; Schwarz and Dreyfus, 1995); for instance, geometrical transformations can be completely described by means of equations, but also using a graphical representation clarifies the concept better;
- change the presentational order of some concepts, based on their relative degree of difficulty (Tall, 1991); for instance, the resolution of equations can be introduced graphically before completing the study of algebraic manipulation rules;
- focus on resolution strategies rather than on performing calculations, which can improve problem solving activities (Fey, 1989);
- emphasise the meaning of mathematics concepts (Rothery, 1995) before being able to formally handle them; for instance, derivatives computed by some software can be used to find minima and maxima of a function, based on their meaning, before learning to compute them.

Computers lead students to a more active role, not only because they are required "to do things by themselves", but also because they feel motivated to ask more questions, as is witnessed also in the literature (Offir et al, 1994). This deeper involvement can also give the occasion for some cognitive considerations:

- exploring conjectures can lead to distinguishing conjectures from demonstrations.
- solving problems by trial-and-error can emphasise the power of formal methods.
- discovering rules by analysing the output of some software can make it easier to remember them and to understand their meaning.

3 PROBLEMS TEACHERS FACE IN THE COMPUTER LABORATORY

When including a computer laboratory activity in a mathematical course, teachers face problems of several kinds during preparation, development, and delivery. Problems faced are made more difficult by the fact that a teaching tradition in this field is not yet available. Before starting this activity for the first time, some *preliminary problems* need to be solved. In particular, teachers should:

- face possible personal idiosyncrasies against using computers.
- face the concern of seeing their role weakened (Chaptal, 1994).
- overcome the fear of using a tool which is possibly newer to them than to their students (Sendova, 1994).
- accept that they will need to change their teaching habits (Marshall, 1994).
- understand that using computers aims at a more effective teaching, not to make their work less tiring.
- reach a sufficient knowledge of the hardware and some software that will be used.
- learn that computer arithmetic is somehow different from the traditional form, since numbers that can be represented by a computer are a discrete and limited set, unlike real numbers; this does not result in some applications, but is evident in others.

Concerning the organisation of courses, *didactical planning problems* arise that should be analysed by the teacher before starting a course, or in the class preparation phase during the school year. They include:

- establishing whether and how their didactic "contract" should be modified;
- considering which concepts can be better mastered by the students if introduced by using computers;
- evaluating which software is most suitable for topics to be taught; since more than one software program related to the same topic can exist, and there may be no didactical guides or experience reports to rely on, so performing this choice can be puzzling;
- understanding when students are able to use a given tool so that they can really take advantage of it.

Teaching problems arise during the course of computer laboratory sessions themselves; they can be tackled a priori only in a general way; how exactly these problems will show up cannot be completely predicted, since they arise from the interaction with the students:

- time should be more accurately controlled than in traditional teaching (Jones, 1995), because computer laboratories are usually accessed for a limited number of hours, which need to be used to their best effect, and because it is easier for students to work at different paces in computer laboratories than it is in traditional teaching;
- often there are fewer computers than students in the computer laboratory, hence it is necessary to organise working groups; this entails deciding how many students can fruitfully work together, and if groups should be formed always by the same members and, if not, how often and how they should be changed, if and how individual work should alternate with group work;
- it is more difficult to control the work of all students, because of possible different technical problems at the same time;
- helping students to learn is more complex, since the teacher needs to understand not only obstacles related to mathematics, but also those related to the logic operation of computers;

- evaluating student's work is more difficult (Passey and Ridgway, 1994), because most software currently in use does not keep track of all performed steps, while in traditional tests the students themselves can be required to record the steps of their work and reasoning.

4 POSSIBLE RELATIONSHIPS BETWEEN CLASS AND COMPUTER LABORATORY

How to give an answer to the above issues is strictly dependent on the mutual relationship between class and computer laboratory organisation, which can take different forms.

In some cases, class and computer laboratory are conceived by the teacher as separate modules; in both of them students are taught mathematics, but activities performed are not explicitly connected. Though both environments can be fruitful, some opportunities of cognitive growth are missed. For instance, if equations are solved in the class by formal methods, and in the computer laboratory graphical software is used to study function intersections (but the teacher does not call the student's attention to the fact that an equation can be interpreted as an intersection of two functions so that a graphical approach can be used to solve equations), then a possibility of gaining a general view and a better mastery of the involved concepts is missed.

Sometimes the teacher tries to connect class and computer laboratory activities, but gives greater emphasis to one of the two environments. If the computer laboratory is used only to support conjectures or to apply theorems already demonstrated in the class, then this activity is conceived as a service to class work; though meaningful exercises can be carried out, students get the impression that new concepts and theorems can be introduced only by a theoretical approach. Vice versa, if new concepts and theorems are introduced always by heuristic work in the computer laboratory, and only later is it completed from a theoretical point of view, students can get the message that all mathematical knowledge is obtained by generalising particular cases, and they can be led to underestimate the importance and the need of a theoretical approach to mathematics concepts.

The advantages of both approaches can be achieved by considering class and computer laboratory as environments with equal importance, which is possible, since their potentialities are not in conflict. In both of them teachers can carry out directive activities aiming to build a shared knowledge, students can solve problems autonomously, and ideas can be compared by common discussions. Thus, the teacher should integrate class and computer laboratory activities from a cognitive and a cultural point of view, by considering each as a starting point for the other one, according to the current need, hence producing a continuous and balanced learning experience.

5 TRAINING IN-SERVICE TEACHERS

Training experiences have been carried out in many countries, but often teachers have not been satisfied with the outcomes achieved (Handler, 1993). In particular, they complain that at the end of the training period there are no support personnel to contact in case of need, and that it is difficult to apply what they have learned in everyday teaching. This last issue seems to arise, at least in Italy, because of the structure of courses offered, which lasted a short period and concerned technical aspects of computers and programming languages, and marginally application software, without any self-contained and well organised didactical material. Some mathematical topics were treated as examples of software use, rather than as the subject of the teaching and learning process. Based on this experience and on the teaching problems mentioned in Section 3, we can identify issues related to training in-service teachers.

The three groups of problems are equally important, since each of them, if not suitably addressed, can lead teachers to take decisions not to use computer laboratories or to use them ineffectively. The preliminary issues, except for the acquisition of technical knowledge that must be covered during the training courses, are essentially psychological or cultural, and can be overcome by addressing the remaining groups of problems during the training, provided that the teachers become aware of their existence. Indeed, understanding potentialities and limits of the tools that they are going to use, and getting aware of possible teaching issues and suitable approaches to them, can raise the self confidence that is essential to positively accept this new experience.

Concerning the other two groups of problems, it is important to give models, rather than recipes, in order to make explicit the philosophy underpinning a proposal, not only because teaching problems can arise in different forms, but also because the technology available changes very rapidly so that teachers need to be able to take autonomous decisions on various issues. Hence, training courses should offer criteria, to analyse the potentialities and the applicability of packages rather than be limited in showing their use. Moreover, since comparison and discussion are fundamental for taking balanced decisions, teacher training, after an initial intensive period, should be organised in the form of working groups, strongly linked to research in mathematics education, in order that experiences, problems and suggested solutions can be shared, in particular that concerning their actual work in school. Both intensive courses and working groups should be hands-on, that is, they should give teachers direct experience of using a computer laboratory on the topic they are going to treat.

Though it is not possible to exhaustively address the above-mentioned problems in this paper, we want to outline solutions that seem reasonable, arising from our experiences.

The choice of topics to tackle in the computer laboratory depends both on an analysis of difficulties, usually met by students, and on the software available for that purpose. When more than one software program on a given topic is available, simplicity, robustness and richness should be guiding criteria to make a choice. Simplicity is important, since software which is too sophisticated would require that both teachers and students use long periods of time to learn to use it, and probably also distract the user from concentrating on the proposed mathematical concepts. Simplicity of use includes a user-friendly interface, but does not mean that a software program should solve problems without requiring cognitive effort or much work from the students.

Robustness means that an error in the input data will not cause the student to input all data again or even to restart the program; this feature is important since students are usually not very skilled when typing data.

Richness of features does not mean choosing the package with more functions, but looking at which functions are useful for teaching, since many complex operations are not required in school. In this respect, it is not necessary to periodically get the most recent version of the applied software, but rather to evaluate whether the newly added features are really an improvement from a didactical point of view. Moreover, sometimes it seems convenient to use different software for different purposes, even within the same topic, so as not to limit the choice to didactical software. For instance, in the case of algebra, Derive is a package which can be used effectively, but a more suitable introduction to equations can be made by using a spreadsheet (Dettori et al, 1995; Auricchio and Greco, 1995), while using some software that allows the integration of text and formulas in the same working sheet helps students to comment on the work done, hence reflecting on it more than they usually would. Another reason for using more than one software program is that one can be more suitable for making conjectures, another for developing applications. However, the number of different packages should be as limited as possible, in order to avoid wasting effort with technical rather than mathematical notions.

The choice of a suitable software program depends on didactical aims, which are related to a student's current abilities and knowledge. Using a software program to do faster what is already mastered allows them to focus on concepts that are to be learned. For instance, if a teacher wants his students to learn graphical representations of function, he should not use a program that draws graphs automatically, but one that requires all parameters involved to be chosen. Later, for introducing the graphical method to solve equations, a graphical software program is adequate, because students need to focus on the solution method.

If a student needs individual support, it is appropriate not to force him to use a different software program, but rather the same software in a different way; for instance, the teacher can ask him to verbalise his reasoning before translating it into software commands, or not to use some commands that automatically produce operations he does not yet master. Hence, the didactical contract established in the class should be confirmed also in the computer laboratory, extending it to meet the characteristics of this learning environment.

Let us now consider problems arising during the course of computer laboratory sessions. The first problem to solve is its physical organisation, that is, how to match students and computers, especially if there are more people than machines. Working in groups has the advantage of compelling students to share and compare their thoughts, provided that groups are small enough to give space to everyone; in general, groups of two seem the most productive. On the other hand, individual work leads each student to face problems by himself, and this gives a measure of individual abilities and limits. Working together influences the level of attention: it can help students to concentrate more on their work, but it can also be an occasion for distraction. Moreover, working with a class mate is not always immediately engaging, but it is a skill that needs to be acquired and refined. From the point of view of teaching, it may be easier to control group work than individual work, since the number of resulting strategies to be analysed is smaller, but it can happen that interesting approaches or means will not emerge. We believe that it would be advisable to alternate individual and group work, both in the class and in the computer laboratory, possibly with students taking turns to work individually, if the number of computers is not sufficient; this alternation can also increase the student's self confidence. Exploring conjectures and solving very complex problems seems to be performed better in groups, while verifying concept acquisition and solving standard problems seems more appropriate for individual work. The initial group setting can be decided by the teacher or by the students; what matters is that the teacher checks that all groups work effectively, possibly changing the composition of some of them according to need, but not so often as to prevent the group members developing an understanding of each other.

Group work can make the evaluation of the performances of every student more difficult, which in the computer laboratory is difficult as well for individual work, since most software does not keep track of single steps performed. Evaluation can be improved by asking the students to mark on paper the performed operations, as well as recording the obtained results. Individual evaluation can be supported by personal observation and by experiences of individual work. While class and computer laboratory are equally important from an educational point of view, they require different cognitive abilities. A global evaluation of the students should give the right balance to skills developed in each environment.

Since occasions for discussing collateral problems are more frequent in the computer laboratory than in the class, and it is easier to loose the thread of the lesson, we suggest that teachers prepare a detailed scheme of the content to be developed in every lesson, which refers also to maintaining the student's work at a similar pace. Only questions asked by few students, or groups of students, should be discussed individually, while issues shared by most students should be the objects of common discussion, in order both to save time and to contribute to the building of a shared base of knowledge.

6 REFERENCES

Auricchio, V. and Greco, S. (1995) *Un itinerario didattico per l'introduzione all'algebra supportato da software commerciali.* Tech. Rep. n.16/95, IMA CNR, Genova, Italy.

Burke, W. (1994) The search for a coherent approach to the development of Information Technology in the school curriculum, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Chaptal, A. (1994) Policy for change, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Dettori, G., Garuti, R. and Lemut, E. (1995) From arithmetic to algebraic thinking by using a spreadsheet, in R. Sutherland (ed.) *Algebraic processes and structures.* Kluwer.

Duval, R. (1994) Représentation sémiotique et coordination des registres. Paper presented at the *46th Congress of the CIEAEM*, Toulouse.

Fey, J.T. (1989) School algebra for the year 2000, in *Research issues in the learning and teaching of algebra*, *Vol. 4, NCTM*. L. Erlbaum.

Handler, M.G. (1993) Preparing new teachers to use computer technology: perceptions and suggestions for teacher educators. *Computers and Education,* **20**, 2, 147-156.

Jones, D. (1995) The introduction of an Information Technology project: personal computing in the classroom, in J.D. Tinsley and T.J. van Weert (eds.) *Proceedings of the WCCE 95*, pp. 327-336.

Marshall, G. (1994) Cautionary verses: prospects and problems in achieving the aims of the computer revolution, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Noss, R. and Hoyles, C. (1992) Looking back and looking forward, in C. Hoyles and R. Noss (eds.) *Learning math with Logo*. MIT Press.

Offir, B., Katz, Y.J. and Passig, D. (1994) The utilisation of "open" courseware for developing an enquiry approach in the study of non-scientific subjects, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Passey, D. and Ridgway, J. (1994) The development of informatics capability in elementary schools: what to do, and how to make it work, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Plomp, T. and Pelgrum, W.J. (1991) Introduction of computers in education: state of the art in eight countries. *Computers and Education*, **17**, 3, 249-258.

PNI (1991) La verifica del Piano Nazionale per l'Informatica nelle scuole secondarie superiori. *Annali della Pubblica Istruzione.*

Rosvik, S. (1994) A look back at the conference, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Rothery, A. (1995) Computer algebra and the structure of the mathematics curriculum, in L. Burton and B. Jaworski (eds.) *Technology in mathematics teaching*, 329-343.

Schwarz, B. and Dreyfus, T. (1995) New actions upon old objects: a new ontological perspective on functions. *Educational Studies in Mathematics*, **29**, 259-291.

Selwood, I. and Jenkinson, D. (1995) The delivery of Information Technology capability in secondary schools in England and Wales, in J.D. Tinsley and T.J. van Weert (eds.) *Proceedings of the WCCE 95*, pp. 565-575.

Sendova, E. (1994) A computer in the classroom: a catalyst for a creative partnership, in J. Wright and D. Benzie (eds.) *Exploring a New Partnership: Children, Teachers and Technology, IFIP Transactions* **A-58**, 49-56. North-Holland, Amsterdam.

Tall, D. and Thomas, M. (1991) Encouraging versatile thinking in algebra using the computer. *Educational Studies in Mathematics*, **22**, 2, 125-147.

Veen, W. (1993) How teachers use computers in instructional practice: four case studies in a Dutch secondary school. *Computers and Education*, **21**, 1/2, 1-8.

7 BIOGRAPHY

Giuliana Dettori and **Enrica Lemut** are mathematicians, working as researchers at the Institute for Applied Mathematics of the National Research Council of Italy. Their main research interest is in mathematics and technology education.

Vincenzo Auricchio and **Simonetta Greco** are mathematicians, collaborators of the Institute for Applied Mathematics of the National Research Council of Italy. They are interested in mathematics and technology education.