LINEAR INDEPENDENCE FOR CORK

R.E. Harte

In Summer 1978, Question 4 of the Second Arts Matrix Theory paper at U.C.C. begins as follows:

> "Explain what it means for a finite subset of a real vector space to be linearly independent. If $\{x_1, x_2, x_3\}$ is linearly independent and t is real, show that
>
> A $\{x_1 - tx_2, x_2 - tx_3, x_3 - tx_1\}$ linearly independent $\iff t \neq 1$".

The embarrassing fact is that the solution is wrong, but in a rather subtle way. In fact provided the three vectors x_1, x_2 and x_3 are distinct from one another, the statement (A) is correct, as can be and was verified by Second Year Arts students. Suppose however

B $\quad x_1 \neq x_2 \neq x_3$

still assuming that the set $\{x_1, x_2, x_3\} = \{x_1, x_2\}$ is linearly independent, then:

C $\{x_1 - tx_2, x_2 - tx_3, x_3 - tx_1\}$ linearly independent $\iff t = 0$.

Finally if $x_1 = x_2 = x_3 \neq 0$ then the statement (A) is again valid.

MORAL: linear independence should be defined for sequences, not sets.

Mathematics Department,
University College,
Cork.

COMPUTER SCIENCE AND THE MATHEMATICS CURRICULUM

Anthony Karel Seda

> "As soon as an Analytical Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will then arise - By what course of calculation can these results be arrived at by the machine in the shortest time?"
>
> Charles Babbage, 1864.

§1. Introduction

In any discussion of computer science and its relationship with mathematics, from an educational viewpoint, certain obvious questions come to the fore:

(1) What is the role of mathematics in computer science education?

(2) What is the role of computer science in mathematics education?

(3) What is, or has been, the response of mathematicians to computer science in relation to the mathematics curriculum?

There are two viewpoints, at least, from which these questions can be contemplated. One is that of the computer scientist engaged in teaching/research in a third level institution peering over the ramparts at the mathematicians. The other, which is ours, is that of the mathematician similarly engaged in teaching/research and similarly peering at the computer scientists. Having thus declared my vantage point, and for reasons of space, I wish to concentrate here on Question 1, and only to touch on Questions 2 and 3. Specifically, I wish to bring to the attention of readers of the *Newsletter* the discussion contained in the articles [4] and [5] of Professor Anthony Ralston. Ralston's conclusion is,

> "It is time to consider (i.e., try)
> an alternative to the standard
> undergraduate mathematics curriculum
> which would give discrete analysis an
> equivalent role to that now played
> by calculus in the first two years
> of the undergraduate curriculum".

In §3 I have listed the topics which Ralston proposes in order to achieve his aim. Actually, [4] is a detailed version (83 pages) of [5], and [5] will suffice to support the main thread of the argument here.

In the quotation above, Babbage is of course talking about algorithms, and algorithms in the words of Knuth [3], are "... really the central core of the subject (computer science), the common denominator which underlies and unifies the different branches". Indeed, Knuth has, just prior to writing this, chosen to describe computer science as "the study of algorithms". Now, as confirmed by Knuth, the study of algorithms is very mathematical and it is worth stating this fact in order to dispose of the short, negative reply to Question 1 which just might be proposed from the other vantage point! Further confirmation of this fact, i.e. of the mathematical nature of computer science, can be gained by consulting the list of topics in Section 68 of the 1980 Subject Classification of Mathematical Reviews, or by actually reading some recent reviews in this section; see also [1].

2. Some History and Some Educational Philosophy

Whilst our main discussion centres on Question 1, it will not be out of place to devote a few words to Questions 2 and 3.

One might wonder why it is today that there is a division

between computer scientists and mathematicians, and that there is not more sympathy shown by each for the other's subject. After all, computer science grew out of mathematics and in its early days, some twenty five-thirty years ago, it was necessarily closely bound to mathematics. However, today, digital computers vastly predominate over analogue computers and digital computers are essentially discrete. What, though, is being taught in most mathematics departments? I suspect that it is largely either continuous mathematics, such as analysis, or relatively abstract mathematics, to the great exclusion of discrete mathematics. Certainly this is true in U.C.C., but may be less so in non-university departments. Indeed, Ralston [4] argues that in American universities the present-day structure of the mathematics curriculum (mainly calculus/linear algebra - at least in the first two years) has come about for reasons more to do with history and inertia (human) than with a judicious choice of topics to meet the educational requirements of those students other than majors in physical science and engineering.

As far as Question 3 is concerned, there are at least three discernible responses:

(a) Ignore the problem - maybe it will go away.

(b) Continue teaching traditional material but illuminate it with examples/projects worked on the computer.

(c) Meet the problem head-on and design/update courses to more nearly meet the needs of those students studying computer science.

Response (a) needs no comment; (b) is outside the scope and limits of this note but surely has a lot of merit, see [2] and its references for some experiments, and also elsewhere in this *Newsletter*; (c) is the main topic of this discussion, see §3.

Before leaving this section, there is another aspect worth noting. Mathematics courses are widely held to be educational, irrespective of their content, for purposes of training the mind. Can the same be said of computer science? This touches on Question 2, because the solution Ralston has in mind for (c) is best framed in terms of a mathematical sciences degree programme and, naturally, the educational value of such a programme, over and above its content, has to be considered. To quote G.E. Forsythe, see [3], "The most valuable acquisitions in a scientific or technical education are the general-purpose mental tools which remain serviceable for a lifetime. I rate natural language and mathematics as the most important of these tools, and computer science as a third". Some of Knuth's own views on this can also be found in [3].

3. Ralston's Proposals for the Mathematics Curriculum

I want, now, to list the topics which Ralston believes could form a suitable basis for the discrete component in a better balanced curriculum for mathematics students, computer science students and others. The headings below are taken from [4] and [5] and the topics from [4].

i) Algorithms and their Analysis. Topics: the notion of an algorithm; notation for expressing algorithms; basic analysis of algorithms.

ii) Introductory Mathematical Logic. Topics: the notion of mathematical proof; the propositional calculus; Boolean algebra; the notation of the predicate calculus; introduction to the verification of algorithms.

iii) Limits and Summation. Topics: the notion of infinite processes; ideas of convergence and limits; limits of discrete functions; summation.

iv) Mathematical Induction. Topics: the principles of induction; examples of induction proofs.

v) The Discrete Number System. Topics: real numbers and finite number systems; definition and laws of the discrete number system; number bases other than 10.

vi) Basic Combinatorial Analysis. Topics: the binomial theorem and Stirling numbers; permutations and combinations; simple combinatorial algorithms.

vii) Difference Equations and Generating Functions. Topics: recurrence relations; linear difference equations and their solution; generating functions.

viii) Discrete Probability. Topics: basic laws; discrete probability distributions; random number generation; queueing theory; probability and algorithm analysis.

ix) Graphs and Trees. Topics: basic definitions and theorems of graph theory; path and colouring problems; tree enumeration and binary trees.

x) Basic Recursion and Automata Theory. Topics: basic definitions; recursive algorithms; recursive functions; regular sets and expressions; finite state machines; languages and grammars; Turing machines.

In connection with this list, the following points should be noted:

(A) These topics are only suggestions. Moreover, it is assumed by Ralston that they will be presented in some combination with abstract algebra, linear algebra, analysis etc., for in [4] it is observed that "... there are numerous areas of computer science where calculus plays an important role..." Moreover, a better balanced curriculum is being argued for, but not a complete reversal in favour of discrete mathematics.

(B) These topics are, with the possible exception of some in viii), mathematics subjects and as such are best taught by mathematicians.

(C) Due to the differences between the educational systems here and in America, certain additions and subtractions might need to be made were these proposals to be adapted to fit into our context (Probably extra more advanced material such as more computability theory or computational complexity could be added for, say, honours students).

(D) These proposals are at least worthy of consideration, for Professor Ralston has wide experience in both computer science and mathematics and backs up his suggestions with an exhaustive study.

More questions are asked here than are answered. For example, consideration needs to be given to the feasibility of such topics for various types of student, ranging from students of management through to honours mathematics students. But space permits no more comment, and for answers to such questions the reader must either consult [4] and [5] or, if Ralston [5] page 484 is correct, undertake experiment for himself or herself.

Educational problems are not usually very well defined; they are likely to be controversial and to raise temperatures. Indeed it may be that Ralston's criticism does not apply here and that all is well. If not, and this article creates some discussion or starts people thinking about the problems raised here, then it will have achieved its purpose. We hardly need reminding in 1983 that computer science is a major undergraduate subject. But what has perhaps not been widely recognised yet is the fact that the next generation of students will be taught computer science in secondary schools by those currently studying it at third-level. Future incoming students may therefore elect to study computer science "because it is familiar" just as many do now, I suspect, in the case of mathematics.

References

[1]. E.W. Dijkstra, Programming as a Discipline of Mathematical Nature, Amer. Math. Monthly, 81 (1974), 608-612.

[2]. S.P. Gordon, A Discrete Approach to Computer Oriented Calculus, Amer. Math. Monthly, 86 (1979), 386-391.

[3]. D.E. Knuth, Computer Science and its Relation to Mathematics, Amer. Math. Monthly, 81 (1974), 323-343.

[4]. A. Ralston, Computer Science, Mathematics and the Undergraduate Curricula in Both, Technical Report 161, Department of Computer Science, SUNY at Buffalo, 1980.

[5]. ———— Computer Science, Mathematics and the Undergraduate Curricula in Both, Amer. Math. Monthly, 88 (1981), 472-485.

Department of Mathematics,
University College,
Cork.

USING MICROCOMPUTERS IN UNIVERSITY MATHEMATICS TEACHING

A.W. Wickstead

Until the last few years, most attempts to involve computers in the teaching process involved sitting a student at the terminal of a large computer and attempting to perform the whole teaching process by presenting information which was then tested. Apart from being very limited in what could be taught in this way, such projects have tended to be very expensive. They involve relatively large computers, many terminals to them and a lot of (expensive) programming effort. All of this expenditure has to be incurred before the technique can be tried, so its use has, of course, been very limited.

The advent of cheap microcomputers in recent years has enabled the less well financed University Mathematics department to acquire a computer and, experiment with its use in teaching. With such limited resources (and no programming assistance) it would be foolish to hope to emulate the large