Computers Math. Applic. Vol. 19, No. 3, pp. 13–34, 1990
Printed in Great Britain

0097-4943/90 $3.00 + 0.00
Pergamon Press plc

MICROCOMPUTER LABORATORIES IN MATHEMATICS EDUCATION†

S. BREUER, J. GAL-EZER‡ and G. ZWAS
School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv, Israel

Abstract—This article discusses the mathematical-educational potential of a computational laboratory at the pre-calculus and co-calculus levels. The laboratory envisaged is based on a set of microcomputers, whose use plays a central role in the teaching process, with particular emphasis on algorithmization. A new role for the mathematics teacher and professor is layed out, augmenting the "chalk and talk" methods by active participation as a laboratory instructor. Following a brief description of the integration of such a laboratory into the mathematical education, seven appropriate subjects are discussed, including some new relevant elementary proofs and worked out examples. Emphasis is placed upon the mathematical-educational byproducts (such as error bounds, ill-conditioning, complexity, rate of convergence, etc.) accompanying the implementation of these seven modules. Special attention is given to the removal of "black box" procedures and to the construction of "numerical methods that work". Extensions and generalizations to more advanced topics are indicated, especially where the results in our modules may serve as points of departure in that direction.

1. THE MATHEMATICAL LABORATORY AND ITS POTENTIAL

Accumulated experience has shown that early emphasis on algorithmic thinking, augmented by actual computing, is indispensable in mathematical education. Recognizing the cardinal importance of the individual, active involvement of each and every student in the computational activity (as opposed to a mere demonstration by the teacher), we advocate the use of mathematical laboratories, based upon a set of microcomputers. Under optimal conditions a special room should be set aside for a mathematical laboratory. Failing that, the physics or chemistry laboratories may be used, since they tend to create the proper atmosphere. Each pair of students is assigned to a microcomputer, very much like the microscopes in the biology laboratory. A few hours out of the students' weekly mathematical training should be spent in the laboratory, most of which being devoted to working with its microcomputers.

The mere presence of an increasing number of microcomputers in various educational institutions, even those at which a programming language such as True-BASIC or Pascal is taught, in no way constitutes a new mode of teaching and learning. The full potential of microcomputers, along with proper courseware, should be harnessed to improve the state-of-the-art in education. Moreover, a new role is to be played by the mathematics teacher in accordance with this objective, since his previous "chalk and talk" methods must henceforth be augmented by active participation as a *laboratory instructor.*

The following points will serve to bring out the educational potential of the mathematical laboratory:

Concretization of abstract ideas

Abstract mathematical concepts may be made concrete and thus are likely to be vividly grasped and understood. The limit concept, for example, is a case in point. Furthermore, the computational approach leads to an interplay between theoretical and numerical ideas which undoubtedly improves the teaching process.

Creativity coupled with delightful learning

In the laboratory, a higher percentage of the students will be active (at their own pace) than under traditional learning circumstances. Creativity is stimulated through the fun-filled, gratifying

†Dedicated to Eugene Isaacson on his 70th birthday.
‡Presently at The Open University of Israel.

dialogue with the computer. The student will experience the truth of the saying "mathematics is like kissing—the only way to discover its delights is by doing it".

Individual pace

The laboratory environment enables every pair of students to progress at a pace compatible with their ability (subject, of course, to some minimum goal required of everybody). The better pairs of students will tackle some of the "starred" assignments, without being held back and bored by the traditional pace of an average class. "Double starred" assignments will be designed to challenge one or two pairs of superior students to come up with their own original ideas, generalizations, improved algorithms, etc. Their progress should *not* be channelled towards the next assignment but rather towards a more profound mastery of the current subject. In this way every student can realize his full potential.

No more tables

The laboratory is the most natural means of doing away with tables, which have traditionally been used as "black boxes" without the faintest understanding of their construction. It is to be emphasized that we are not advocating the introduction of new "black boxes" by using, say, the logarithmic built-in function of the microcomputer. The student will learn just what is behind such built-in functions as part of the material covered in the mathematical laboratory.

Getting down to earth

As pointed out by William E. Milne, "many know how to solve a probelm but can't do it". The mathematical laboratory serves to educate the student not to fall into this category, but rather to train him to translate his theoretical ideas into practical algorithms which actually work. At the same time the algorithmic thinking of the students is cultivated and this, according to some researchers (see Ref. [1]), should be placed at the center of mathematical education.

In-depth learning

While teaching a certain subject, we often find out that we never fully understood certain subtle points. Writing a computer program will usually show us that there are even more fundamental issues we never paid attention to. Being a dummy (though a fast and powerful one), the computer will carry out our instructions precisely—but blindly. Lack of total, in-depth understanding of the problem at hand may lead to an imperfect program, which will break down exactly when an unforseen situation occurs.

Learning by discovery

The mathematical laboratory offers virtually unlimited opportunities to the art of learning by discovery. The microcomputer enables the user to set up mathematical "experiments", to test various conjectures, to check non-trivial particular cases of a general proposition, etc. It goes without saying that not every given group of students can be expected to take full advantage of this method of learning. On the other hand, the better students can and will. Even in modern mathematical research, there are examples of outstanding results whose origin can be traced to computational experimentation. The discovery of solitons is a case in point. Every student of mathematics should experience the gratifying feeling inherent in the discovery of *some* mathematical rule by himself. This method clearly enhances intuitive thinking, an essential component of the learning of mathematics.

New vital aspects in the teaching of mathematics

The mathematical laboratory introduces, and in fact stresses, some new and vital concepts conspicuously absent from standard curricula. Some examples are *approximate* solutions with error control, computational efficiency and complexity, the influence of small changes in the data on the overall solution, and the characterization of situations where the effect of round-off errors is critical.

We find concurrence with our ideas in the fact that two of the main recommendations for school mathematics in the 1980s put forth by the American NCTM were that problem solving be stressed and that calculators and computers be used to full advantage at all levels.

An efficient way of implementing the ideas outlined above is the introduction of, say, nine computational models, seven of which have been completed by us as described below. A conscious effort has been made in each module to construct elementary proofs, at the pre-calculus or co-calculus level, sacrificing complete generality at times but maintaining full rigor throughout. We found this to be a highly satisfactory compromise, allowing the use of much simpler proofs. These proofs are then followed by a statement of the type "the results of this theorem can be generalized so as to be valid for a wider class of functions, i.e. all functions . . .".

We now turn to the actual discussion of our seven modules, including worked out examples.

2. AREA APPROXIMATION IN THE MATHEMATICAL LABORATORY

In our first typical laboratory module [2] we study the approximation of areas to a prescribed accuracy, including the appropriate error analysis. Different types of approximations are examined, with progressively increasing efficiency, and their laboratory implementation discussed. Extensions and generalizations are indicated for more advanced mathematical education.

To begin with, the rectangle method for approximating the area under the graph of a positive function should be introduced, including a pre-calculus error analysis for monotonic functions. This method requires considerable microcomputer time for high accuracy calculations, and thus motivates a search for better methods (see Ref. [2]). Since the rectangle method is based upon rectangles, which are simple geometric figures meeting the curve at one point per strip, it is only natural to try and use a set of trapezoids which meet the curve at *two points* per strip.

In order to maintain the advocated rigorous approach even at the pre-calculus level, we shall limit the generality, to start with, to *convex* (or concave) functions $f(x)$. These concepts should be defined geometrically by using the property that any secant connecting two arbitrary points on such a curve is completely above (or completely below) the curve.

It is good to start with a specific example such as $f(x) = 1/x$ (convex), for $1 \leqslant x \leqslant 2$. Alternatively, $f(x) = \sqrt{x}$ (concave) might be used. The sum of all the trapezoidal areas is given by

$$T_n = h[\tfrac{1}{2}f(x_0) + f(x_1) + \cdots + f(x_{n-1}) + \tfrac{1}{2}f(x_n)], \tag{1}$$

where $x_0 = a$, $x_n = b$, $h = (b - a)/n$ and $x_j = a + jh$.

In line with our spirit of teaching via the mathematical laboratory, we shall make an effort to estimate the error incurred by the approximation T_n in a vivid geometrical way. This will include the construction of a teaching aid for every student to see (and even build for himself). It turns out, as we shall see below, that *concave* functions are more convenient for this construction.

Let us draw the function $f(x)$ and its associated trapezoids. Next we continue the "roof" of each trapezoid to the left, say, until it covers the adjacent trapezoid (see Fig. 1). In addition, we draw

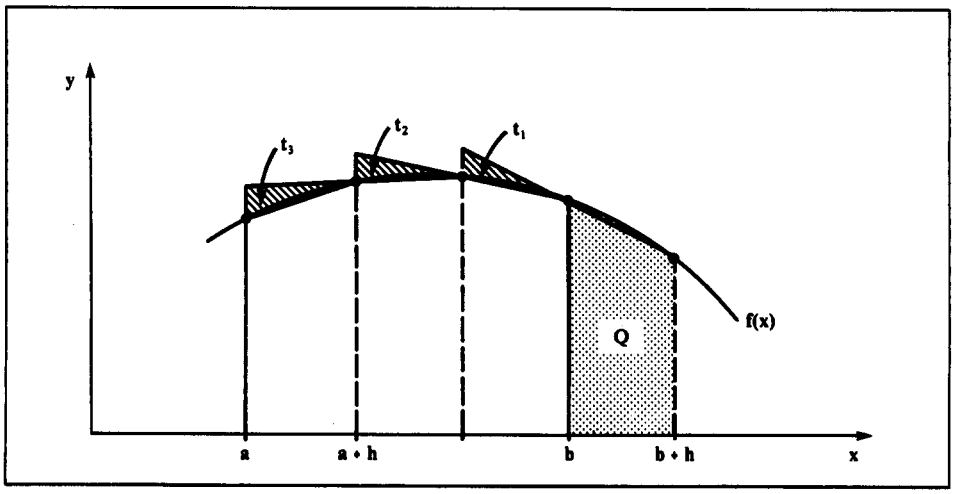

Fig. 1

an extraneous trapezoid Q, which fits into the extension of $f(x)$ to the right, from $x = b$ to $x = b + h$, as seen in the figure. Had we extended the roofs to the right, we would now be adding an extra trapezoid to the left of $x = a$. We now observe that the "local error" (per strip) incurred by the trapezoidal approximation is the difference between the area under the curve $f(x)$ and the area of the corresponding trapezoid. This error, in turn, is seen in Fig. 1 to be less than the triangle (such as t_1, t_2 or t_3) sitting on that very trapezoid. If the function $f(x)$ is convex (and not concave as in the figure), we would obtain a similar situation except that the roles of the two non-vertical sides of the little triangles would be reversed.

Now we take the extreme triangle on the right (t_1 in Fig. 1) and *slide* it leftward until it sits precisely on top of its adjacent triangle (t_2 in Fig. 1). Our construction of these triangles enables us to do just that. Next we take t_1 and t_2, together, and slide them leftward until they sit right on top of t_3 (see Fig. 2). In general there will be n strips, and we may carry out the sliding process just described $(n - 1)$ times. In this way we shall obtain a composite triangle, composed of n little triangles with no overlapping and no holes. The situation for three strips ($n = 3$) is shown in Fig. 2, in which the final, composite triangle has vertices U, V and W. A teaching aid designed to demonstrate this sliding process can easily be constructed and successfully employed in the mathematical laboratory.

Since the composite triangle consists of the sum of triangles each of which represents a bound on the local error, its area B gives us a bound for the global error $E = |S - T_n|$, where S is the area under the curve. Thus we have

$$E = |S - T_n| < B = \frac{h}{2}|z - f(a)|, \tag{2}$$

in which z is the distance of the point V from the x-axis. Clearly, the area B of the composite triangle is given by one-half of its height h times its base $|z - f(a)|$. The absolute value is used here since for a convex function $z < f(a)$.

In order to calculate the value of z, we write down the equation of the straight line through U and V. This line goes through the point $(a + h, f(a + h))$, so that we have

$$y = m[x - (a + h)] + f(a + h). \tag{3}$$

A little reflection will show that the effect of the sliding process is to cause the slopes of the tops of the triangles t_1, $t_1 + t_2$, etc., to be equal, indeed equal to the slope of the roof of the extraneous trapezoid Q. Consequently,

$$m = \frac{f(b + h) - f(b)}{h}. \tag{4}$$

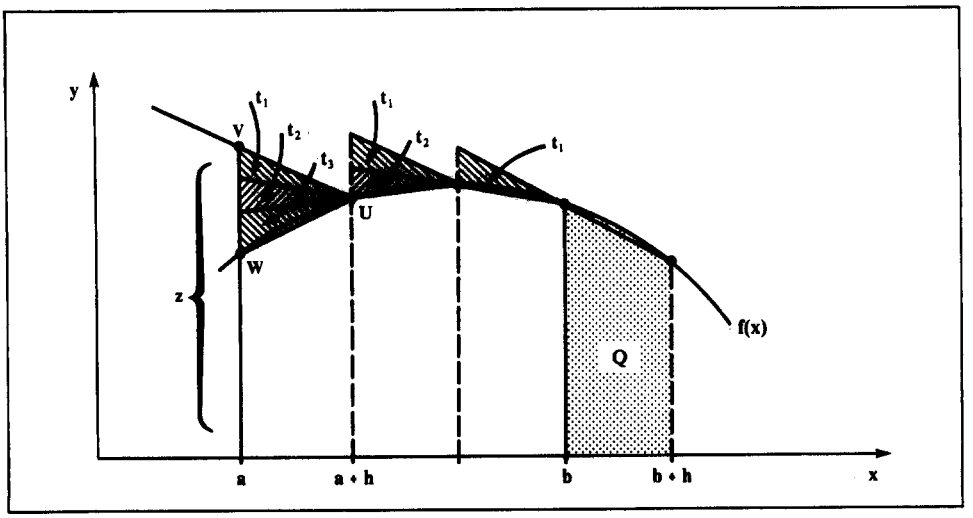

Fig. 2

Substituting $x = a$ in equation (3) and using the value of m from equation (4), we reach

$$z = -f(b+h)+f(b)+f(a+h), \tag{5}$$

from which

$$B = \frac{h}{2}\left|-f(b+h)+f(b)+f(a+h)-f(a)\right|. \tag{6}$$

For future convenience, we shall rewrite equation (6) in the form

$$B = \frac{h^2}{2}\left|\frac{f(b+h)-f(b)}{h} - \frac{f(a+h)-f(a)}{h}\right|, \tag{7}$$

so that inside the absolute value symbol we have the difference of two slopes of the type (4). Using $h = (b-a)/n$ in the factor $h^2/2$ in equation (7), we reach

$$B = \frac{K}{n^2}, \tag{8}$$

where

$$K = \frac{(b-a)^2}{2}\left|\frac{f(b+h)-f(b)}{h} - \frac{f(a+h)-f(a)}{h}\right|. \tag{9}$$

We shall now investigate the nature of K in a typical problem by considering $f(x) = 1/x$ for $1 \leqslant x \leqslant 2$. For this case,

$$\begin{aligned} K &= \frac{(2-1)^2}{2}\left|\frac{1}{h}\left(\frac{1}{2+h}-\frac{1}{2}\right)-\frac{1}{h}\left(\frac{1}{1+h}-1\right)\right| \\ &= \frac{1}{2}\left|\frac{1}{1+h}-\frac{1}{2(2+h)}\right| < \frac{1}{2}\left(\frac{1}{1+h}\right) \leqslant \frac{1}{2}. \end{aligned} \tag{10}$$

It follows that in this case K does not exceed $1/2$, regardless of the number of strips used in the approximation. The global error, therefore, is bounded by $(1/2)/n^2$. Whereas the rectangle approximation yields an error bound inversely proportional to n, the trapezoidal approximation gives an error bound inversely proportional to n^2, at least for $f(x) = 1/x$.

More generally, a method of approximation with an error bound inversely proportional to n^p is called a method of order p. Thus the rectangle method is of first order, whereas the trapezoidal method seems to be of second order.

In our particular case, $f(x) = 1/x$ for $1 \leqslant x \leqslant 2$, the attainment of q correct decimal figures requires that $(1/2)/n^2 \leqslant (1/2)10^{-q}$, so that $n = 10^{q/2}$ will do. Thus, four correct decimal figures necessitate 100 strips, in contradistinction to 10,000 strips that are needed when the rectangle method is used for the same purpose. More generally, the use of a second order method is seen to save a lot of computational effort (for a given accuracy), since the accompanying error decreases at a faster rate with refinement of the partition (increasing n).

For $f(x) = 1/x$ we obtained that $K \leqslant 1/2$, independently of n, where K as given by equation (9) is the numerator of the error bound (8). In general, for a given convex or concave $f(x)$, it is only necessary to analyze equation (9) and determine a constant, say $\tilde{K}$, such that $K \leqslant \tilde{K}$ on the underlying interval for all n. Except for the multiplying factor $(b-a)^2/2$, the right-hand side of equation (9) is just the difference of the slopes of the secants associated with the interval endpoints. The only situation that may present a difficulty would be the case of a function $f(x)$ possessing vertical (i.e. unbounded) slopes near one or both endpoints of the interval, such as $\sqrt{1-x^2}$ near $x = 1$. If we agree to exclude such cases from our considerations, we may sum up our findings as follows:

> Given a function $f(x)$, which is continuous, positive and convex or concave on $[a, b]$. If we approximate the area S under the curve by T_n given by equation (1), then the error incurred $|S - T_n|$ will not exceed $\tilde{K}/n^2$, where $\tilde{K}$ is a constant independent of n (and h), satisfying
>
> $$\tilde{K} \geqslant K = \frac{(b-a)^2}{2}\left|\frac{f(b+h)-f(b)}{h} - \frac{f(a+h)-f(a)}{h}\right|. \tag{11}$$

In many cases, the given function may be neither convex nor concave over the underlying interval, but may enjor *one* of these properties in suitably constructed subintervals. By applying the above analysis to each subinterval, we are able to enlarge the class of functions to which our results are applicable.

Other examples of using this error bound, as well as details concerning the implementation of this laboratory assignment, may be found in Ref. [2].

3. MATHEMATICAL-EDUCATIONAL ASPECTS OF THE COMPUTATION OF π

In this module [3], four methods of computing π are discussed at the pre-calculus and co-calculus levels. Emphasis is placed upon the mathematical-educational by-products accompanying the implementation of these methods. Thus it is shown how to expose the laboratory participants to concepts such as error bounds and control, ill-conditioned algorithms, rates and acceleration of convergence, probabilistic reasoning and computer simulation—while their attention is focused upon the computation of π.

We first introduce the historically important method of Archimedes, using perfect polygons inscribed in and circumscribed about a unit circle. The loss of accuracy due to round-off errors in this method is demonstrated and analyzed. Table 1 shows typical results obtained with a microcomputer. In this table, n denotes the number of sides of the polygon and P is the approximation of π, computed by averaging the semi-perimeters of the inscribed and circumscribed n-gons. We left a few blank spaces between the last accurate figure of π (obtained via better methods) and the first inaccurate one, in order to emphasize the increasing, and then decreasing accuracy, inherent in this method.

These results should be compared with the results obtained by computing the area under the graph of the function $y = \sqrt{1 - x^2}$. This computation has been carried out, using the trapezoidal method (for $0 \leqslant x \leqslant \frac{1}{2}$) with 800 strips, and yielded the approximation 3.14159. (For details see Ref. [3].)

Next, we offer the laboratory participants to tackle the so-called Buffon's needle-tossing method, named after Count Buffon who discovered it in the eighteenth century. Unlike the previous two methods, the present one calls for some rudimentary knowledge of integral calculus (see Refs [3, 4]). Typical results are given in Table 2.

The subtleties inherent in the use of random number generators (especially for a vast number of times) are left in the background. A closer examination will show that if our only interest were to calculate π with ever increasing accuracy, we would end up testing the quality of the generator rather than approximating π. Moreover, Table 2 shows the convergence to be deplorably slow. Nevertheless, we highly recommend including the Buffon method among the assignments of the mathematical laboratory. It has a considerable mathematical-educational value in that it introduces probabilistic methods into the realm of computing, and those in turn lead to the well known Monte-Carlo methods. In addition, the laboratory participants are given the opportunity to become acquainted with the concept of simulation.

The last method described in this module is based upon the expansion of $\arctan x$ in a power series and is intended therefore for co-calculus level laboratory participants. Following John Machin we use the arctan addition formula

$$\arctan u + \arctan v = \arctan \frac{u + v}{1 - uv}, \tag{12}$$

to obtain

$$\frac{\pi}{4} = \arctan \frac{1}{2} + \arctan \frac{1}{3}. \tag{13}$$

With this formula, 10 terms in the power series will supply us with six correct decimal figures of π. (For details see Ref. [3], where a brief historical sketch of developments in computing π may be found.)

Table 1

n	P	
12	3.1	60609
24	3.14	6144
48	3.14	2718
96	3.141	873
192	3.141	663
384	3.1415	98
768	3.14159	3
1536	3.1415	83
3072	3.141	487
6144	3.141	862
12288	3.14	4864
24576	3.1	56844

Table 2

No. of tosses	Approximation
10^4	3.1521
10^5	3.1382
10^6	3.1409
10^7	3.1416

The authors' own experience in implementing the computation of π in a mathematical laboratory has been very gratifying indeed. Most participants were very enthused as their activity centered around a well known yet tantalizing assignment, while being exposed to a variety of mathematical ideas. Such a gratifying success may be achieved if the laboratory instructor constantly bears in mind that teaching mathematics—and not the computation of π *per se*—is the ultimate goal.

4. CONVERGENCE ACCELERATION AS A COMPUTATIONAL ASSIGNMENT

In Ref. [5] we treat methods of accelerating the convergence of infinite series, in a way appropriate for a mathematical-educational laboratory. Following a discussion concerning rate of convergence, closed-form sums, and the use of upper and lower remainder estimates, the acceleration methods of Kummer and Euler as well as a method based on an approximate recursion relation are presented. Only rudimentary concepts from calculus are used. The actual reduction of the number of terms needed to achieve a desired accuracy, is vividly demonstrated via runs on a typical laboratory microcomputer. Here we will demonstrate the method based on an "approximate" recursion relation, using the geometric series as a point of departure.

The terms of a geometric series Σa_n satisfy the relation $a_{n+1} = qa_n$. If $|q| < 1$, the sum of the infinite series is given by

$$S = \frac{a_0}{1-q} = \frac{a_0}{1 - a_1/a_0} = \frac{a_0^2}{a_0 - a_1}. \tag{14}$$

A closed-form formula can also be obtained for series whose terms satisfy

$$a_{n+1} = \alpha_1 a_n + \alpha_2 a_{n-1}. \tag{15}$$

This can be seen by summing up both sides of equation (15) for $n = 1, 2, \ldots,$ to obtain

$$S - a_0 - a_1 = \alpha_1(S - a_0) + \alpha_2 S, \tag{16}$$

so that for this *convergent* series we have

$$S = \frac{a_0 + a_1 - \alpha_1 a_0}{1 - \alpha_1 - \alpha_2}. \tag{17}$$

If the values of α_1 and α_2 are known, the sum can be computed from equation (17). Otherwise, α_1 and α_2 can be found from equation (15) for $n = 1, 2$, yielding

$$S = \frac{(a_0 + a_1)(a_1^2 - a_0 a_2) - a_0(a_2 a_1 - a_0 a_3)}{a_1^2 + a_2^2 + a_0(a_3 - a_2) - a_1(a_3 + a_2)}. \tag{18}$$

We have thus expressed S in terms of a_0, a_1, a_2 and a_3. In general, if we have a convergent series whose terms satisfy the relation

$$a_{n+k-1} = \alpha_1 a_{n+k-2} + \cdots + \alpha_k a_{n-1}. \tag{19}$$

for all n, then its sum can be expressed in terms of $a_0, a_1, \ldots, a_{2k-1}$. The resulting formulas, similarly to equation (18), are a natural extension of the geometric series formula.

The method presented in this section makes use of the above, to accelerate the convergence of series whose terms *approximately* satisfy a relation of the form of equation (19).

The following example will serve to show what is meant by "approximately satisfy". Let us consider the alternating series

$$\sum_{n=0}^{\infty} a_n = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}. \tag{20}$$

In order to discover the relation between three consecutive terms of equation (20), let us examine the following linear combination:

$$\begin{aligned} a_{n+1} - \alpha a_n - \beta a_{n-1} &= \frac{(-1)^{n+1}}{2n+3} - \alpha \frac{(-1)^n}{2n+1} - \beta \frac{(-1)^{n-1}}{2n-1} \\ &= (-1)^n \frac{4(\beta - \alpha - 1)n^2 + 4(2\beta - \alpha)n + (3\beta + 3\alpha + 1)}{(4n^2-1)(2n+3)}. \end{aligned} \tag{21}$$

We would like to choose α and β, so that the right-hand side will vanish, or at least be as small as possible for large values of n. Choosing $\alpha = -2$ and $\beta = -1$ so that

$$\begin{aligned} \beta - \alpha &= 1, \\ 2\beta - \alpha &= 0, \end{aligned} \tag{22}$$

causes the coefficients of n^2 and n in the numerator to vanish, and we are left with

$$a_{n+1} + 2a_n + a_{n-1} = (-1)^n \frac{-8}{(4n^2-1)(2n+3)}. \tag{23}$$

In other words, we found that

$$a_{n+1} = -2a_n - a_{n-1} + O\left(\frac{1}{n^3}\right). \tag{24}$$

This is what we meant by saying that the terms "approximately satisfy" a relation of the form (19), i.e. a recursion relation that holds up to terms of order $1/n^p$, $p > 0$.

Now, let us sum equation (23) for $n = 1, 2, \ldots,$ to obtain

$$S - a_0 - a_1 = -2(S - a_0) - S - 8 \sum_{n=1}^{\infty} \frac{(-1)^n}{(4n^2-1)(2n+3)}. \tag{25}$$

Since $a_0 = 1$ and $a_1 = -1/3$, we find that

$$S = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \tfrac{2}{3} + 2 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{(4n^2-1)(2n+3)}. \tag{26}$$

As a result, the method enables us to replace the original series by a faster converging series, as will be seen below.

Both series in equation (26) alternate in sign, and thus (see Ref. [5]) their remainders satisfy

$$\left| \sum_{n=k+1}^{\infty} \frac{(-1)^n}{2n+1} \right| \leqslant \frac{1}{2k+3} < \frac{1}{2k}, \tag{27}$$

$$2\left| \sum_{n=k+1}^{\infty} \frac{(-1)^{n+1}}{(4n^2-1)(2n+3)} \right| \leqslant \frac{2}{(4k^2+8k+3)(2k+5)} < \frac{1}{4k^3}. \tag{28}$$

Now, to compute S so that a tolerance of $(1/2)10^{-4}$ can be assured, 10,001 terms are needed in the original series, while only 18 terms are needed after applying this method as shown. We see that a considerable acceleration has been achieved.

The method can be extended further, by examining four consecutive terms, rather than three as in equation (21). By doing so we get

$$a_{n+2} - \alpha a_{n+1} - \beta a_n - \gamma a_{n-1} = \frac{(-1)^n}{(2n+3)(2n+5)(4n^2-1)}[8(\alpha - \beta + \gamma + 1)n^3 + 4(5\alpha - 7\beta + 9\gamma + 3)n^2 + 2(-\alpha - 7\beta + 23\gamma - 1)n + (-5\alpha + 15\beta + 15\gamma - 3)]. \quad (29)$$

Reasoning as before, we choose $\alpha = -3$, $\beta = -3$ and $\gamma = -1$, so that the coefficients of n^3, n^2 and n will vanish. This leads to

$$a_{n+2} = -3a_{n+1} - 3a_n - a_{n-1} + (-1)^{n+1}\frac{48}{(2n+3)(2n+5)(4n^2-1)}. \quad (30)$$

Summing up both sides of equation (30) for $n = 1, 2, \ldots$, and using the values of a_0, a_1 and a_2, we end up with

$$S = \tfrac{11}{15} + 6\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{(2n+3)(2n+5)(4n^2-1)}. \quad (31)$$

Checking the remainder this time, we find that it is bounded by $0.375/k^4$ and thus only eight terms are needed to assure the tolerance of $(1/2)10^{-4}$.

The laboratory participants will observe, that the additional preparatory work, was not worth while, particularly since the terms of the new series are computationally more complex. The advantage of equation (31) over equations (21) will be more pronounced when higher accuracy is required. Nevertheless, adding eight terms of equation (31), we found $S = 0.7854$ to four decimals. Adding 18 terms in equation (21) yielded the same result. Just as a check, we summed 10,001 terms in the original series using double precision, and also found 0.7854 to four decimals. We had to use double precision since the accuracy is otherwise damaged due to round-off errors creeping into the computation and accumulating for such large values of n. The need to avoid such accumulations emphasizes even more the importance of acceleration methods. The laboratory participants should be asked to apply the method to other series, such as

$$\sum_{n=1}^{\infty}\frac{(-1)^n}{n^2} \quad \text{and} \quad \sum_{n=1}^{\infty}\frac{q^n}{2n-1}, \quad -1 \leqslant q < 1.$$

The method presented above bears some resemblance to the ϵ-method, which is beyond our scope here. The precise conditions under which the ϵ-method is effective, as well as a presentation of another variant called the ρ-method, may be found, for example, in Ref. [6]. A presentation of Kummer's and Euler's methods, in a way appropriate for a microcomputer educational laboratory, can be found in Ref. [5].

This laboratory assignment not only shows how to improve computational efficiency, but rather emphasizes the difference between a theoretical proof that a series converges, and an actual computation of its sum to a given accuracy.

5. AN ALGORITHMIC APPROACH TO LINEAR SYSTEMS

As another mathematical laboratory subject, an algorithm for the solution of algebraic linear systems is presented. No knowledge of matrices, vectors and the underlying theory is assumed. Pedagogical considerations guided the choice of material, style and level of presentation, while emphasizing the learning process in a mathematical laboratory environment. Special attention is given to possible loss of accuracy, sensitivity to minor changes in the data, pivoting, pre-scaling and computational efficiency. Since the laboratory participants are introduced to these concepts without using matrix theory, the laboratory sessions can be carried out even before the study of linear algebra.

We suggest the introduction of Gaussian elimination via the use of a concept such as the "coefficient-table" (see Ref. [7, p. 503]).

The concept of computational complexity should be discussed in the laboratory in connection with the estimate of computational work in the elimination as a function of N, the number of unknowns. We show that the naive elimination process has a computational complexity of N^3 (for details see Ref. [7, p. 508]). Some available methods for solving N linear equations have a complexity of $N!$ so that they are of course exceedingly less efficient than the Gaussian elimination. (The laboratory participants should compare these complexities for increasing values of N, in order to get the proper feeling of the difference.)

The laboratory instructor should not fail to emphasize at the very beginning that we are interested not only in constructing a solution algorithm that works, but also in its efficiency. This consideration will appear time and again in the laboratory's activities, and the participants should be made aware of it right from the start. This tends to make them more critical and willing to compare various methods of completing a given assignment. We feel that this is a relevant point in the difference between *mathematical education* and learning mathematics.

Pivoting is shown to be essential [7, pp. 509–512], since naive elimination does not take into account a zero or a very small (in absolute value) pivot element.

It should be emphasized that having computational difficulties (due to small pivot elements, say) depends crucially on the accuracy of the computing-device used. To demonstrate this, let us consider the following system:

$$\left[\begin{array}{ccc|c} -1.41 & 2 & 0 & 1 \\ 1 & -1.41 & 1 & 1 \\ 0 & 2 & -1.41 & 1 \end{array}\right], \tag{32}$$

whose solution (correct to three significant figures) is $X(1) = X(2) = X(3) = 1.69$. Let us now make the artificial assumption that we are equipped with a three-decimal-digit computer. Thus we will round-off *every* computer result to three significant figures in order to simulate our assumed computing-device. In order to achieve triangularization, we multiply the first row by $1/1.41 = 0.709$ and add it to the second. Then we multiply the new second row by $-2/0.01 = -200$ and add it to the third, to obtain

$$\left[\begin{array}{ccc|c} -1.41 & 2 & 0 & 1 \\ 0 & 0.01 & 1 & 1.71 \\ 0 & 0 & -201 & -341 \end{array}\right]. \tag{33}$$

The back-solution now yields $X(3) = 1.70$, $X(2) = 1.00$ and $X(1) = 0.709$, which is manifestly incorrect. Had we used the pivoting strategy, the correct solution (to three significant figures) would have been found. Using pivoting would *not* have been necessary, had we used a five-decimal-digit computer. With such a computer, we start by multiplying the first row by $1/1.41 = 0.70922$ and adding it to the second, then multiplying the second by $-2/0.00844 = -236.97$ and adding it to the third. This leads to

$$\left[\begin{array}{ccc|c} -1.41 & 2 & 0 & 1 \\ 0 & 0.00844 & 1 & 1.7092 \\ 0 & 0 & -238.38 & -404.03 \end{array}\right]. \tag{34}$$

The back-solution gives: $X(3) = 1.6949$, $X(2) = 1.6943$ and $X(1) = 1.6940$ which, when rounded finally to three significant figures, is correct (to this accuracy).

This opportunity can be used by the laboratory instructur to introduce the concept of double precision, and point out that increasing the accuracy of the computations from three to five digits,

in our example, is roughtly like a change from single to double precision. Thus, the laboratory participants should reach the following conclusions:

(i) The degree of computational difficulty depends on the computational device used.
(ii) Changing single precision to double precision sometimes contributes to overcoming the computational difficulty.

When we are actually solving a given linear system on a computer, we find that round-off errors are always present in the data stored or in intermediate results. In addition, the data may contain errors stemming from the use of measurement equipment with limited accuracy, etc. It is necessary to investigate the influence of such errors on the computed solution. In particular, we would like to beware of systems in which small errors in the data will cause large changes in the computed solution, i.e. "ill-conditioned" systems.

Consider, for example, the system (suggested by Dancis [8]) whose augmented table is given by

$$\left[\begin{array}{rrrrr|r} 0.1 & -1 & 0 & 0 & 0 & 0.1 \\ 0 & 0.1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0.1 & -1 & 0 & 0.1 \\ 0 & 0 & 0 & 0.1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right]. \tag{35}$$

By back-solving we obtain $X(5) = 1$, $X(4) = 0$, $X(3) = 1$, $X(2) = 0$ and $X(1) = 1$. If we change the right-hand side a little, by changing the bottom element of the sixth column 1 to 1.01, yet leaving all the other entries unchanged, the solution obtained is $X(5) = 1.01$, $X(4) = 0.1$, $X(3) = 2$, $X(2) = 10$ and $X(1) = 101$. A change of 1% in one entry caused a very considerable change in the solution. System (35) has an obvious pattern, needs no triangularization, and thus the laboratory participants can construct larger, similar systems where the above phenomenon is even more pronounced.

The laboratory instructor should emphasize the distinction between errors stemming from the very nature of a given system (ill-conditioning) and errors resulting from the method of solution. The latter can be dealt with by modifying the algorithm, while ill-conditioning is intrinsic and cannot be removed even by a sophisticated modification. In an ill-conditioned case the solution is hypersensitive to changes in the data. Furthermore, if we substitute the computed solution in the given equations, we will find that even though the equations are satisfied to a high accuracy, the computed solution column might differ considerably from the true solution column.

The possibility of ill-conditioning impairs the credibility of the computed solution, and thus we would like to have some *a priori* indication on the conditioning of a given system (see Ref. [7, p. 517]).

We would like to emphasize that in this module we had no intention of presenting the entire subject of solving linear systems, with all the underlying theory. Nor has our intention been to present a numerical approach for teaching linear algebra. Our goal has been to show how to introduce these subjects to students *unfamiliar with matrix theory*, but willing to take advantage of the new computing opportunities. These opportunities are created by an increasing number of microcomputers appearing in various educational institutions.

6. COMPUTER ROOT EXTRACTION BY *A PRIORI* DESIGN

In this module we show how to construct iterative formulas for the extraction of kth roots which are of built-in desired order by *a priori design*. Contrary to the usual practice in which a certain iterative procedure is suggested and then analyzed for its order, we are taking the reverse point of view. That is to say, our point of departure is a desired, prescribed order of convergence, and we employ purely algebraic means to construct an iterative procedure possessing this order.

The accompanying error decay is shown to equal $(1/2)^r$ for iterations of pre-designed order r. The computer-educational aspects are stressed throughout, as is the suitability of the algorithms for a computer library.

We start by showing that for the extraction of the square root of $s > 0$, it suffices to consider the reduced range $1 \leqslant s < 4$, so that $1 \leqslant \sqrt{s} < 2$ (for details of this range reduction see Ref. [9, p. 305]. Initialization of the iterations with $x_0 = 1.5$ will thus guarantee that $|x_0 - \sqrt{s}| \leqslant \frac{1}{2}$.

As mentioned above, our point of view is opposite to the usual practice. That is to say, we wish to construct iterations x_n approximating $\sqrt{s}$ which are to obey the inequalities

$$|x_{n+1} - \sqrt{s}| \leqslant K|x_n - \sqrt{s}|^2, \quad K \text{ constant}, \tag{36}$$

i.e. be of second order. For simplicity we assume $K = 1$. Let us investigate the consequences of inequality (36) and how they compare with the performance of bisection. In this way the laboratory participants will be motivated to search for higher order methods even without knowing *a priori* what specific form they might take. Repeated use of inequality (36) with $K = 1$ yields

$$|x_{n+1} - \sqrt{s}| \leqslant |x_n - \sqrt{s}|^2 \leqslant |x_{n-1} - \sqrt{s}|^4 \leqslant |x_{n-2} - \sqrt{s}|^8 \leqslant \cdots \leqslant |x_0 - \sqrt{s}|^{2^{n+1}}, \tag{37}$$

leading to

$$|x_n - \sqrt{s}| \leqslant (1/2)^{2^n}. \tag{38}$$

We may compare inequality (38) with the corresponding result for bisection in which the right-hand side is $(1/2)^n$. For example, $n = 5$ iterations make $|x_5 - \sqrt{s}| \leqslant (1/2)^{32} < (1/2)10^{-9}$, whereas $n = 31$ iterations are required to achieve that accuracy with bisections (see Ref. [9, p. 311]).

With a view towards constructing a second order method, let us write

$$x_{n+1} - \sqrt{s} = \frac{(x_n - \sqrt{s})^2}{\alpha}, \tag{39}$$

and try to choose α judiciously. We may write equation (39) in the form

$$x_{n+1} - \sqrt{s} = \frac{x_n^2 + s}{\alpha} - \frac{2x_n}{\alpha}\sqrt{s}, \tag{40}$$

and observe that if we choose $\alpha = 2x_n$, the quantity $\sqrt{s}$ disappears from equation (40). Thus we are left with the iterative formula

$$x_{n+1} = \frac{x_n^2 + s}{2x_n} = \frac{1}{2}\left(x_n + \frac{s}{x_n}\right), \tag{41}$$

which has the *built-in* property

$$x_{n+1} - \sqrt{s} = \frac{(x_n - \sqrt{s})^2}{2x_n}, \tag{42}$$

by equation (39). The laboratory instructor should point out that equation (41) has first been obtained by Heron of Alexandria (about 100 B.C.) by an entirely different approach. Using methods of differential calculus, Newton and Raphson arrived at the same result as a special case of their method of tangents. A method based upon the comparison of the areas of a rectangle and a square can be found in Ref. [10]. All the above approaches lead to some iterative formula which is *a posteriori* shown to be of second order. As mentioned before, our approach goes precisely in the opposite direction in that it advocates construction of higher order iterations by *a priori* design.

Reference to equation (42) shows that its left-hand side is non-negative, for all values of n, once we have chosen $x_0 > 0$. It follows that $x_n \geqslant \sqrt{s}$ for all values of n. On the other hand $1 \leqslant \sqrt{s} < 2$ so that $x_n \geqslant 1$, and we have

$$0 \leqslant x_{n+1} - \sqrt{s} \leqslant \tfrac{1}{2}(x_n - \sqrt{s})^2 \leqslant (x_n - \sqrt{s})^2. \tag{43}$$

Taking stock of what we have done, we find that our goal expressed by inequality (36) has actually been realized, with $K = 1$ (we could even take $K = 1/2$ if we wanted to). Accordingly, iterations (41) supply us an even better version of inequality (38) in the form

$$0 \leqslant x_n - \sqrt{s} \leqslant (1/2)^{2^n}. \tag{44}$$

Had we not been so generous in inequalities (43) and taken $K = 1/2$, the resulting error estimate would have been somewhat sharper but more cumbersome. For the purpose of comparison with the estimate for bisection, the laboratory participants will find the form (44) much more convenient since it juxtaposes the rapid decay $(1/2)^{2^n}$ with the relatively slow $(1/2)^n$.

Another point worthwhile mentioning is the sensitivity of the quality of approximations to the initial value x_0. Had we somehow been able—without excessive computational labor—to find a better initial guess x_0, satisfying $|x_0 - \sqrt{s}| \leqslant \lambda < 1/2$, the above analysis would have led to

$$|x_n - \sqrt{s}| \leqslant \lambda^{2^n}, \tag{45}$$

with correspondingly fewer iterations for a prescribed accuracy. As a "starred" exercise, the better laboratory participants should be asked to show that the choice $x_0 = (8s + 17)/24$ satisfies $|x_0 - \sqrt{s}| \leqslant 1/24 = \lambda$ for $1 \leqslant s < 4$, so that the error decay is very fast indeed.

We note that since $|x_0 - \sqrt{s}| \leqslant 1/2$, one correct binary digit in our initial approximation is guaranteed. By inequalities (43), the number of correct digits is *doubled* in each iteration, which serves again to explain why few iterations suffice. The motivation to search for methods of even higher order is thus very natural, as are generalizations to cube roots and kth roots (see Refs [9, 16]).

7. POLYNOMIAL FUNCTION APPROXIMATIONS

Approximations by polynomial interpolation is discussed at the pre-calculus level. Special attention is given to the removal of "black box" procedures, and inherent concepts such as quality of approximation, relative error and computational efficiency are examined. A suitable example is given, demonstrating the actual construction of computer library functions. This subject, in particular, sheds light on the "mystery" confronting practically every user of computers: when a computer is instructed to evaluate a given complicated expression, how does it come up with the required answer so accurately and so fast.

In the following let us suppose that we are faced with the problem of evaluating a function $f(x)$ for various values of x in the interval $[a, b]$. We shall assume that we know the values of $f(x)$ at $(n + 1)$ specific points $x_0, x_1, x_2, \ldots, x_n$ in that interval. It is best, in the laboratory, to proceed with a concrete example familiar to the students, $f(x) = \sin x$. The interval $[a, b]$ will be $[0, \pi/2]$, so that the evaluation of $y = \sin x$ in the interval will furnish us with the values of $\sin x$, for all x (in radians) via trigonometric identities familiar to the students.

For definiteness let us choose $n = 6$ (i.e. seven points), $x_0 = 0$ so that $y_0 = \sin x_0 = 0$, and $x_6 = \pi/2$ so that $y_6 = \sin \pi/2 = 1$. The remaining five points, $x_1, \ldots, x_5$, will be distributed evenly over the interval $[0, \pi/2]$, in the absence (at this stage) of any motivation to do otherwise. Moreover, we now have $x_1 = \pi/12$, $x_2 = \pi/6$, $x_3 = \pi/4$, $x_4 = \pi/3$, $x_5 = 5\pi/12$, and the corresponding values of $y_1, \ldots, y_5$ can be easily obtained. Thus $y_2 = \sin \pi/6 = \sin 30°$ and $y_4 = \sin \pi/3 = \sin 60°$ are available from the 30°–60°–90° triangle, and $y_3 = \sin \pi/4 = \sin 45°$ is known from the isosceles right triangle. Furthermore, $y_1 = \sin \pi/12 = \sin 15°$ can be obtained by using the half-angle formula, and we get $y_1 = \sqrt{\{(1 - \cos 30°)/2\}} = \sqrt{(2 - \sqrt{3}/2}$. Finally, $y_5 = \sin 5\pi/12 = \sin 75° = \cos 15° = \sqrt{\{(1 + \cos 30°)/2\}} = \sqrt{(2 + \sqrt{3})/2}$, again by using the half-angle formula for the cosine. In this way we obtain the following seven pairs of values (correct to 10 decimal figures):

$$x_0 = 0.0000000000, \qquad y_0 = 0.0000000000$$

$$x_1 = 0.2617993878, \qquad y_1 = 0.2588190451$$

$$x_2 = 0.5235987757, \qquad y_2 = 0.5000000000$$

$$x_3 = 0.7853981635, \qquad y_3 = 0.7071067813$$

$$x_4 = 1.0471975513, \qquad y_4 = 0.8660254039$$

$$x_5 = 1.3089969392, \qquad y_5 = 0.9659258263$$

$$x_6 = 1.5707963270, \qquad y_6 = 1.0000000000. \tag{46}$$

The use of standard trigonometric identities enables us to add more points to the seven given above, if we so desire. We shall, however, confine the present discussion to the seven specific points displayed above (i.e. $n = 6$), which will henceforward be referred to as *nodes*.

Our objective is to evaluate the function $y = \sin x$ when its values at the points $x_0, x_1, \ldots, x_6$ are known. The simplest procedure towards attaining this goal is to approximate $\sin x$ by a polygonal line composed of line segments joining successive pairs of nodes. The equation of the segment joining (x_j, y_j) and (x_{j+1}, y_{j+1}) is readily found to be

$$P_1(x) = \frac{x - x_{j+1}}{x_j - x_{j+1}} y_j + \frac{x - x_j}{x_{j+1} - x_j} y_{j+1}, \quad x_j \leqslant x \leqslant x_{j+1}. \tag{47}$$

Equation (47) holds of course for $j = 0, 1, \ldots, 6$, and the notation $P_1(x)$ refers to the fact that equation (47) represents, for each j, a segment of a straight line, i.e. a polynomial of the first degree. With a view towards generalizations of equation (47) in the sequel, let us observe that for $x = x_j$ the second term on the right of equation (47) vanishes, while the coefficient of y_j in the first term equals unity, and thus $P_1(x_j) = y_j$. Analogously, $P_1(x_{j+1}) = y_{j+1}$. Clearly, seven expressions of type (47), one for each value of j, will represent the required polygonal approximation to $\sin x$.

An equation of type (47) is referred to as an *interpolation* formula, since it furnishes us with approximate intermediate values of the function we wish to evaluate. Thus the approximation generated by equation (47) constitutes a piecewise linear interpolation.

When carrying out the above procedure in the mathematical laboratory, the question should be raised as to the possibility of using the data in equations (46) to construct a better approximation to $\sin x$ via a method which yields a tighter local fit. The most natural thing to do is to take the nodes three at a time, and pass a parabola through each triplet of points. In our case we would have three such triplets of the form $(x_{j-1}, y_{j-1}), (x_j, y_j), (x_{j+1}, y_{j+1})$, for $j = 1, 3, 5$, respectively. The equation of each such parabola should be an extension of equation (47). There is a unique parabola passing through a given triplet of points (two distinct such parabolas would imply three distinct roots of a quadratic equation, which is clearly impossible). In order to obtain this requested extension of formula (47), we shall use a sum of three terms corresponding to the three y_js, such that the coefficient of each y_j is a quotient of quadratic rather than linear expressions. Moreover, each coefficient should vanish for two of the x_js and equal unity for the third one, corresponding to its y_j. We are thus led to

$$P_2(x) = \frac{(x - x_j)(x - x_{j+1})}{(x_{j-1} - x_j)(x_{j-1} - x_{j+1})} y_{j-1} + \frac{(x - x_{j-1})(x - x_{j+1})}{(x_j - x_{j-1})(x_j - x_{j+1})} y_j$$

$$+ \frac{(x - x_{j-1})(x - x_j)}{(x_{j+1} - x_{j-1})(x_{j+1} - x_j)} y_{j+1}, \quad x_{j-1} \leqslant x \leqslant x_{j+1}. \tag{48}$$

Each term above represents a parabola, hence so does the sum, which is therefore denoted by $P_2(x)$. As a check, if we substitute $x = x_{j-1}$, say, we find that the coefficient of y_{j-1} equals unity while the two others vanish; hence $P_2(x_{j-1}) = y_{j-1}$. Similarly, we find $P_2(x_j) = y_j$ and $P_2(x_{j+1}) = y_{j+1}$.

Summing up, we may now construct three consecutive parabolic arcs, of form (48), which together gives us a piecewise quadratic interpolation for $\sin x$ with the aid of the seven nodes in equation (46).

At this point the instructor in the mathematical laboratory should encourage the question of extending the process described heretofore beyond quadratic interpolation. Could one pass two successive cubic polynomials through the first four nodes and the last four nodes, respectively? This would lead to the following general question: why not pass one polynomial of nth degree through all $(n+1)$ nodes and use it to approximate the given function $f(x)$? (In our case $f(x) = \sin x$ and $n = 6$.) It turns out that such a polynomial does indeed exist and, moreover, is unique. One denotes this polynomial by $P_n(x)$ and refers to it as the interpolating polynomial of degree n [coinciding with the given $f(x)$ at the nodes]. We note that just as in the case of linear and quadratic approximations, the general interpolating polynomial $P_n(x)$ is of degree n which is one less than the number of given nodes. Thus in our specific case $y = \sin x$ with seven nodes, we shall eventually seek the polynomial of degree six which will do the job.

Returning to the general case it seems reasonable, in view of equations (47) and (48), to look for an nth degree polynomial of the form

$$P_n(x) = \sum_{j=0}^{n} \frac{N_j(x)}{D_j} y_j, \tag{49}$$

where $N_j(x)$ is an nth degree polynomial such that $N_j(x_j)/D_j = 1$ and $N_j(x_k) = 0$ for $k \neq j$. If we look again at equations (47) and (48), we may conclude that

$$\frac{N_j(x)}{D_j} = \frac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}. \tag{50}$$

We verify that equation (50) indeed represents an nth degree polynomial (with the required properties) since the factor $(x - x_j)$ is missing in the numerator. For $j = 0$ and $j = n$, the first and last factors, respectively, of numerator (and denominator) will be the missing ones. The interpolation polynomial given by equations (49) and (50) is due to J. L. Lagrange (1736–1813) and bears his name.

For computational purposes when $(n+1)$ specific nodes are given, we can rewrite equations (49) and (50) in the form

$$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0, \tag{51}$$

in which the coefficients $a_0, a_1, \ldots, a_n$ are expressed in terms of the x_js and y_js. While this preliminary work is perhaps considerable, we must impress upon the student that it is done only *once*. Thus, using equation (51) for repeated computations in order to approximate $f(x)$ for various values of x, is computationally far more efficient than using equations (49) and (50).

Up to this point we have not yet discussed the cardinal question of the quality of approximation we can expect from $P_n(x)$. That is to say, when using $P_n(x)$ as an approximation to $f(x)$, for various values of x in $[a, b]$, what are the errors incurred, and what can be said about their magnitude?

We are now ready to use the seven nodes in equations (46) and actually construct in the mathematical laboratory the polynomial $P_6(x)$ which approximates $\sin x$ in $[0, \pi/2]$. This $P_6(x)$ is then evaluated for values of x increasing from $x = 0$ to $x = \pi/2$ with increments of $\pi/180$, say, which correspond to increments of one degree. The values thus obtained are compared with the corresponding values of $\sin x$ given by the computer's built-in sine function, and the differences

$$R(x) = \sin x - P_6(x) \tag{52}$$

are recorded and plotted (on the screen and on hard copy if desired). The function $R(x)$ is referred to as the remainder, and it should be plotted using an appropriate scale which accentuates its behavior and thereby the quality of approximations. We actually carried out the computations indicated above and found that

$$|R(x)| = |\sin x - P_6(x)| < \tfrac{5}{4}10^{-6}, \quad 0 \leqslant x \leqslant \pi/2. \tag{53}$$

This means that a sixth degree polynomial, based upon just seven given nodes, furnishes us with accuracy between five and six decimal figures. We verified the claim made in equation (53) by repeating these computations in the mathematical laboratory, using increments of $\pi/360$ and $\pi/720$.

The use of yet smaller increments will not change the picture. In order to see the behavior of the remainder $R(x)$ in a vivid way, we plotted $10^6R(x)$ against x and obtained the graph in Fig. 3. The accuracy obtained by approximating $\sin x$ with $P_6(x)$ is so high that had we plotted $\sin x$ and $P_6(x)$ simultaneously, the two graphs would have been virtually identical, and in any case practically indistinguishable. It is only when we plot the magnified remainder $R(x)$ that we are able to see it altogether, and study its behavior.

At this point the laboratory instructor must not fail to underscore the fact that $P_6(x)$ is compared to the computer's built-in sine function which by itself, in turn, is based upon *some* approximation. However, since the built-in sine function carries a very high accuracy of, say, 10 decimal figures, its comparison with $P_6(x)$ can be regarded as the comparison of $P_6(x)$ with the true sine function. Moreover, this sheds light upon the methods by which built-in functions can be constructed.

We turn now to a closer study of the behavior of $R(x)$ in the graph, particularly the error defined by $|R(x)|$.

(a) $R(x)$ vanishes, naturally, at our seven equidistant nodes. Elsewhere $R(x)$ oscillates between positive and negative values, reflecting the fact that $P_6(x)$ winds and wraps itself around $\sin x$.

(b) If we were to add two more nodes, say, it is suggested we place them near the interval endpoints, say at $x = \pi/24$ and $x = 11\pi/24$ (corresponding to 7.5° and 82.5°). This would tend to decrease the error $|R(x)|$ where the graph shows its magnitude to be larger than elsewhere.

(c) For the same reason we could redistribute our seven nodes if we must do with just seven. It stands to reason to shift the second (and perhaps also the third) node towards the left endpoint and act analogously with the sixth (and seventh) node towards the right. The purpose of doing that is to try to "smear" the error as uniformly as possible throughout the interval and thus decrease its maxima.

(d) If the situation described above for $\sin x$ is any indication of what happens in general (with more advanced methods it can be shown that indeed it is), then one should choose a higher density of nodes towards the interval's endpoints, and a lower density around its center.

(e) We stress that the attempt to smear the error uniformly over the interval does not require placing the first and last nodes at the endpoints of the interval. Allowing the positions of the first and last nodes to vary as well, may help to smear the error more uniformly.

At this juncture the laboratory participants should be encouraged to experiment by trial and error with the location of the nodes, given their number. A few more nodes may also be added but their number should be limited. The guiding principle is that *efficiency* be maintained. That is to say, the attainment of higher accuracy must not be offset by unreasonable, additional computational effort caused by adding too many nodes. For example, it is certainly unreasonable to use an approximating polynomial of degree 100 in order to obtain highly accurate values of $\sin x$.

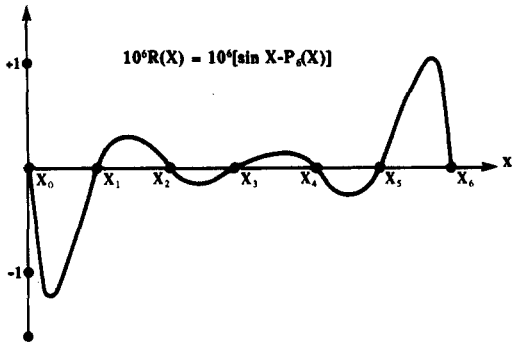

Fig. 3

The study of this section serves as a natural point of departure for a number of profound mathematical questions whose full treatment is way beyond our scope. Given a function $f(x)$ over an interval $[a, b]$ which we wish to approximate by an interpolation polynomial $P_n(x)$ for a given n. Then,

(a) Do there exist *optimal* nodes, in the sense that the maximum value of $|R(x)| = |f(x) - P_n(x)|$ for all x in the interval is minimized? In other words, can we minimize the maximum deviation of $P_n(x)$ from $f(x)$?
(b) If such an optimal polynomial exists, is it unique? If so,
(c) does this optimal polynomial possess characteristic properties, leading to some procedure by which it can be constructed?

The Russian mathematician P. Chebyshev showed that under rather general conditons the answers to the questions above are in the affirmative. This polynomial is called the corresponding *minimax polynomial* since it minimizes the maximum deviation, and indeed it oscillates about $f(x)$ in a way that smears the error uniformly (equal ripple property). These results may be verified in the mathematical laboratory for a very simple case—the approximation of a convex (concave) curve by a straight line.

Actually, for purposes of a computer library function, the relative error should be controlled rather than the absolute error. This important topic should definitely be part of this laboratory assignment (see Ref. [11]).

The subject matter covered in this module appears to us to be extremely suitable for the mathematical laboratory. It sheds light on the "mystery" confronting practically every user of computers: how, when a computer is instructed to evaluate a given complicated expression, does it come up with the required answer so accurately and so rapidly?

8. THE COMPUTATIONAL POTENTIAL OF RATIONAL APPROXIMATIONS

In this laboratory assignment, rational approximations are introduced via typical significant examples, and are based on very rudimentary concepts from calculus. Using interpolative techniques, approximations with nearly equal ripple errors are constructed. The advantage of rational over polynomial approximations—when more than two parameters are involved—is demonstrated, revealing the computational potential of rational approximations.

We suggest the introduction of rational approximations, within the framework of the mathematical laboratory, via specific functions, such as $\ln x$ and $\sqrt{x}$. Besides being well known to the laboratory participants, and part of every computer library, they possess two desirable properties:

(a) Range reduction can be easily performed, i.e. in order to compute them for any positive argument t, it suffices to approximate these functions in a relatively small interval. This is so, since they satisfy

$$\ln t \equiv \ln(x \cdot 2^m) = \ln x + m \cdot \ln 2, \tag{54}$$

$$\sqrt{t} \equiv \sqrt{x \cdot 4^m} = \sqrt{x} \cdot 2^m, \tag{55}$$

where x is the appropriate number in $[1, 2)$, in the case of $\ln x$, and in $[1, 4)$ for the square-root (m is an appropriate integer in each case). In addition to approximating $\ln x$ in $[1, 2]$, the computation must include the multiplication of m by the constant $\ln 2$, which has to be precomputed once and for all. Following the approximation of $\sqrt{x}$ in $[1, 4]$, a binary shift of m places is necessary.

(b) Both functions are concave, i.e. a chord connecting any two points on the graph of each function (in the relevant interval) lies entirely below the graph. This property enables us to use simpler and more elementary proofs later on. This is part of our general philosophy in the mathematical laboratory: sacrifice some generality in order to gain simplicity in the mathematical proofs, but maintain rigor throughout.

At this point a close look should be taken at the approximation of $\ln x$ near $x = 1$, since in this neighborhood a small absolute error does not imply a small relative error. If the laboratory participants are not thoroughly familiar with the concept of relative error and its practical importance, this is the time to acquaint themselves with the subject. Assuming rudimentary knowledge of calculus, including

$$\lim_{x \to 1} \frac{\ln x}{x = 1} = 1, \tag{56}$$

the approximation of $\ln x$ near $x = 1$ can be handled in one of the following two ways:

(a) Approximate $\ln x/(x - 1)$ in $[1, 2]$, and then multiply the approximant by $(x - 1)$. This method is equivalent to removing the root of the relevant function before the construction of the approximation.
(b) Approximate $\ln x$ in $[1, 2]$ and use this approximant only in the interval $[1 + \delta, 2]$. In the interval $[1, 1 + \delta]$, for a properly chosen δ, use $(x - 1)$ to approximate $\ln x$. If the laboratory participants have the appropriate background, they may use additional powers of $(x - 1)$ as well.

In the computational assignment presented in this module it is not our intention to introduce the subject of rational approximations with all generality, even for the two above mentioned functions. We will concentrate on rational approximations with two or three parameters, which will be determined so as to obtain the "best" possible approximations in a sense relevant for a computer library. By "best" for a computer library we mean minimizing (or nearly minimizing) the maximal deviation in $[a, b]$ of the approximant $A(x)$ from the function $F(x)$. In other words we are interested in reducing the error

$$\max_{a \leqslant x \leqslant b} |F(x) - A(x)| \tag{57}$$

as much as possible.

The laboratory instructor will of course realize that we are actually laying the foundations of approximations in the maximum norm, for a possible study later on.

We will demonstrate the computational potential of rational approximations with three-parameter approximations of $\ln x$, $1 \leqslant x \leqslant 2$. Such approximations can be represented in the following three main forms:

Parabola:

$$(px + q)x + r, \tag{58}$$

Hyperbola:

$$u + \frac{v}{x + w}, \tag{59}$$

"Inverse parabola":

$$\frac{a}{(x + b)x + c}. \tag{60}$$

In each of these forms the sum of the degrees of the numerator and the denominator is 2. All three are represented in a computationally economic form. The evaluation of the first involves 2 additions and 2 multiplications, the second—2 additions and 1 division and the third—2 additions, 1 multiplication and 1 division. In all those approximations, three interpolation points and four critical points (two of which are the interval endpoints) are expected (see Ref. [12]). The best approximation in each of the three forms can not be found analytically, so we will adopt the interpolative approach. In this approach, we will construct approximations that will concide with the relevant function at three prescribed points. These points will be chosen from a table of abscissas at which the values of the relevant function are accurately known. To clarify this idea,

let us examine the function $y = \ln x$ in the interval [1, 2]. For the preparation of this table, we calculate once the constant $t = 2^{1/32}$, by five successive square-root extractions, as described in Ref. [9]. In addition, we compute the constant $z = (\ln 2)/32$ to a desirable high accuracy. Now, the x values in the table will be: $t^0, t^1, t^2, \ldots, t^{32}$, and the corresponding known values of $y = \ln x$ are: $0, z, 2z, \ldots, 32z$. From this table we will select our interpolation points. At first, it would seem natural to choose initial interpolation abscissas close to 1.25, 1.50 and 1.75. However, from our experience with two-parameter approximations (see Ref. [12]) we know that it is preferable to choose the first and last abscissa closer to the interval endpoints. Hence we recommend choosing initial interpolation points (from the table) whose x values are nearly equal to 1.1, 1.5 and 1.9. From Table 3 it is clear that the initial values of x should therefore be

$$x_1 = 1.090508, \qquad x_2 = 1.509164, \qquad x_3 = 1.915206. \tag{61}$$

Let us start the three-parameter approximations with parabola (58), i.e. $px^2 + qx + r$. The collocation of this parabola with $\ln x$ at (x_1, y_1), (x_2, y_2) and (x_3, y_3) requires

$$\begin{aligned} p &= \left(\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}\right) \Big/ (x_3 - x_1), \\ q &= \frac{y_2 - y_1}{x_2 - x_1} - p(x_1 + x_2), \\ r &= y_1 - \frac{y_2 - y_1}{x_2 - x_1} x + p x_1 x_2. \end{aligned} \tag{62}$$

The laboratory instructor will observe that p, q and r given by equations (62), actually stem from Newton's form of the interpolation polynomial of order 2.

As pointed out above we start with x_1, x_2, x_3 given in equations (61), i.e. the points numbered 4, 19 and 30 shown in Table 3. Next, the main program should be run, tabulating the differences between $\ln x$ of the computer's library and our parabolic approximation. These differences represent the error function for values of x increasing from 1 to 2 with a desired increment (0.005 for example). This table should be accompanied by a corresponding graph.

After inspection of the resulting table, we changed the interpolation points (using Table 3) so as to get smaller error ripples. Thus, the resulting approximation gradually approached the state of having the equal ripple property with four critical points. Two of those points are in the interior, the two others being the endpoints of the relevant interval (1 and 2 in our case). The best interpolation points in Table 3 were thus found to be points No. 2, 17 and 30, for which the error ripples are closest to being equal. With those best interpolation points, the values of the error function

$$E_1(x) = \ln x - (px^2 + qx + r), \tag{63}$$

at the four critical points, were found to be

$$\begin{aligned} E_1(1.00) &= -0.0027, \\ E_1(1.20) &= +0.0035, \\ E_1(1.72) &= -0.0036, \\ E_1(2.00) &= +0.0038. \end{aligned} \tag{64}$$

Inspecting equations (64), we decided to further improve the error ripples, by using interpolation points from a denser table, generated with $t = 2^{1/128}$ and $z = \ln 2/128$. Since this new table is quite long, we show in Table 4 only its relevant parts.

Using this table we continued to "improve" the approximation. The best interpolation points in Table 4 found in this manner are points No. 9, 69 and 121.

Table 3

k	$x = t^k$	$y = kz$
0	1.000000	0.000000
1	1.021897	0.021661
2	1.044274	0.043322
3	1.067141	0.064983
4	1.090508	0.086643
5	1.114387	0.108304
6	1.138789	0.129965
7	1.163725	0.151626
8	1.189207	0.173287
9	1.215248	0.194948
10	1.241858	0.216608
11	1.269052	0.238269
12	1.296840	0.259930
13	1.325237	0.281591
14	1.354256	0.303252
15	1.383911	0.324913
16	1.414214	0.346574
17	1.445182	0.368234
18	1.476827	0.389895
19	1.509164	0.411556
20	1.542212	0.433217
21	1.575982	0.454878
22	1.610492	0.476539
23	1.645757	0.498199
24	1.681794	0.519860
25	1.718621	0.541521
26	1.756254	0.563182
27	1.794711	0.584843
28	1.834010	0.606504
29	1.874170	0.628165
30	1.915206	0.649825
31	1.957146	0.671486
32	2.000000	0.693147

Table 4

k	$x = t^k$	$y = kz$
0	1.00000	0.000000
⋮	⋮	⋮
6	1.033025	0.032491
7	1.038634	0.037906
8	1.044273	0.043322
9	1.049944	0.048737
10	1.055645	0.054152
⋮	⋮	⋮
63	1.406572	0.341158
64	1.414209	0.346574
65	1.421888	0.351989
66	1.429609	0.357404
67	1.437371	0.362819
68	1.445176	0.368234
69	1.453023	0.373650
70	1.460913	0.379065
71	1.468845	0.384480
72	1.476821	0.389895
73	1.484840	0.395310
⋮	⋮	⋮
118	1.894565	0.638995
119	1.904853	0.644410
120	1.915196	0.649825
121	1.925595	0.655241
122	1.936050	0.660656
⋮	⋮	⋮
128	2.000000	0.693147

The resulting error function was found to have the following four extremal values:

$$\begin{aligned} E_1(1.00) &= -0.0030, \\ E_1(1.22) &= +0.0035, \\ E_1(1.72) &= -0.0036, \\ E_1(2.00) &= +0.0033. \end{aligned} \tag{65}$$

Obviously these results can be further improved by using even denser interpolation points. However, equations (65) suffice for our purposes, as will be seen in Fig. 4.

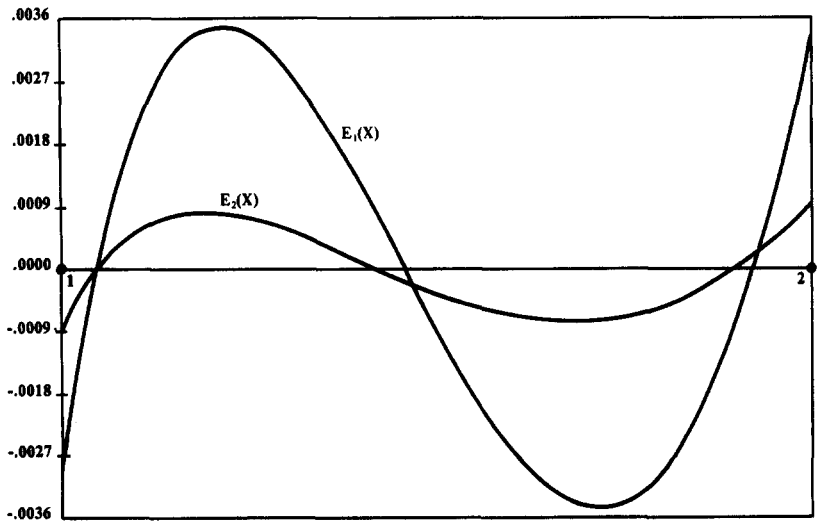

Fig. 4

The deviations in equations (65) give a clear indication of the quality of parabolic approximation to $\ln x$ in [1, 2]. The approximating parabola with error ripples given by equations (65) is

$$P(x) = -0.240035x^2 + 1.406915x - 1.163814. \tag{66}$$

We will now turn our attention to the construction of a rational approximation of form (59), i.e. $u + v/(x + w)$. By requiring this hyperbola to coincide with $y = \ln x$ at (x_1, y_1), (x_2, y_2) and (x_3, y_3), we get the following formulas for the coefficients u, v, w:

$$w = \frac{(x_3 - x_2)(x_2y_2 - x_1y_1) - (x_2 - x_1)(x_3y_3 - x_2y_2)}{(y_3 - y_2)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_2)},$$

$$u = \frac{(y_3 - y_2)(x_2y_2 - x_1y_1) - (y_2 - y_1)(x_3y_3 - x_2y_2)}{(y_3 - y_2)(x_2 - x_1) - (y_2 - y_1)(x_3 - x_2)}, \tag{67}$$

$$v = (y_1 - u)(x_1 + w).$$

As our initial interpolation points, we took points 9, 69, 121 from Table 4 (which yielded the best parabolic approximation). After improving our hyperbolic approximation using Table 4, we found the best points in the table to be points No. 9, 64, 119. The hyperbolic approximation with these points was found to be

$$R(x) = 2.361334 - \frac{5.698596}{x + 1.414210}, \tag{68}$$

with extremal error values of

$$\begin{aligned} E_2(1.00) &= -0.0009, \\ E_2(1.19) &= +0.0008, \\ E_2(1.68) &= -0.0008, \\ E_2(2.00) &= +0.0009. \end{aligned} \tag{69}$$

Thus the maximal error of our rational-hyperbolic approximation, with almost equal error ripples, is about (1/4) of the corresponding error of the polynomial-parabolic approximation. This result demonstrates the potential of rational approximations, which will be further emphasized below.

For completeness we repeated the whole process with the approximation of the form (60), i.e. $a/(x^2 + bx + c)$. In this case, even after improvement of the choice of interpolation points, the sizes of the error ripples were about 30 times those of the parabolic approximation. Thus we have decided to exclude the error function corresponding to this third approximation from Fig. 4, in which we display

$$\begin{aligned} E_1(x) &= \ln x - P(x), \\ E_2(x) &= \ln x - R(x), \end{aligned} \tag{70}$$

where $P(x)$ and $R(x)$ are given in equations (66) and (68), respectively.

Although our results have been limited to a three-parameter approximation of $\ln x$, the rational approximation technique has emerged as a powerful computational tool, whose advantages are even more pronounced when additional parameters are introduced (see Ref. [13, p. 161]).

The superiority of rational approximations over polynomial ones, demonstrated above for $\ln x$, is not just a lucky strike. Corroborating evidence for the computational potential of rational approximations may be found in Refs [12, 15] (see also Ref. [13, Chap. 9, in particular Table 9.3.1]).

9. CONCLUDING REMARKS

We believe that the teaching of mathematics via computational laboratory modules, as described heretofore, supplies the student with a fertile ground for mathematical "experiences", never available before the personal computer era, tending to enhance and cultivate his mathematical

intuition. Moreover, when exposed to these and similar modules, the student is molded in the spirit of numerical applied mathematics at an early stage—so crucial to his entire mathematical point of view.

Numerical experiments should be part of modern mathematical education. Indeed, there has always been an experimental side to mathematics (see Ref. [14, p. 163]). As Euler insisted, "the properties of numbers have usually been discovered by observation, well before their validity has been confirmed . . .". Euler stated also that "it is by observation that we increasingly discover new properties, which we next do our utmost to prove". Computers have greatly increased our capabilities of observation and experimentation in mathematics [14]. It is in this spirit that we suggest the introduction of the described modules into the mathematical laboratory. No one doubts the indispensability of a series of laboratory assignments for the completion of an education in, say, physics or biology. The microcomputer laboratory, we maintain, plays a similar role in mathematical education.

REFERENCES

1. A. Engel, Outline of a problem oriented, computer oriented and application oriented High-School mathematics Course. *Int. J. Math. Educ. Sci. Technol.* **4,** 452 (1973).
2. S. Breuer and G. Zwas, Area approximations in the mathematical laboratory. *Int. J. Math. Educ. Sci. Technol.* **14,** 373 (1983).
3. S. Breuer and G. Zwas, Mathematical-educational aspects of the computation of π. *Int. J. Math. Educ. Sci. Technol.* **15,** 231 (1984).
4. G. Gamow, *One Two Three Infinity*. The American Library—Mentor Books, New York (1953).
5. J. Gal-Ezer and G. Zwas, Convergence acceleration as a computational assignment. *Int. J. Math. Educ. Sci. Technol.* **18,** 15 (1987).
6. A. S. M. Halliday, A review of methods for the acceleration of convergence of infinite series. Math. Report 8D-3, UWIST, Cardiff (1980).
7. J. Gal-Ezer and G. Zwas, An algorithmic approach to linear systems. *Int. J. Math. Educ. Sci. Technol.* **15,** 501 (1984).
8. J. Dancis, The effects of measurement errors on systems of linear algebraic equations. *Int. J. Math. Educ. Sci. Technol.* **15,** 485 (1984).
9. S. Breuer and G. Zwas, Computer root extraction by *a priori* design. *Computers Educ.* **8,** 305 (1984).
10. A. I. Forsythe *et al.*, *Computer Science a First Course*, Chap 7. Wiley, New York (1969).
11. S. Breuer and G. Zwas, Function approximations in the mathematical laboratory. *Int. J. Math. Educ. Sci. Technol.* **14,** 507 (1983).
12. J. Gal-Ezer and G. Zwas, The computational potential of rational approximations. *Computers Educ.* **11,** 33 (1987).
13. C. T. Fike, *Computer Evaluation of Mathematical Functions*. Prentice-Hall, Englewood Cliffs, N.J. (1968).
14. R. F. Churchhouse *et al.*, The influence of computers and informatics on mathematics and its teaching. *Enseign. Math.* **30,** 161 (1984).
15. J. Gal-Ezer and G. Zwas, Computational aspects of rational vs polynomial interpolations. *Int. J. Math. Educ. Sci. Technol.* **19,** 567 (1988).
16. S. Breuer and G. Zwas, Polynomial iterations for root extraction. *Computers Educ.* **12,** 289 (1988).