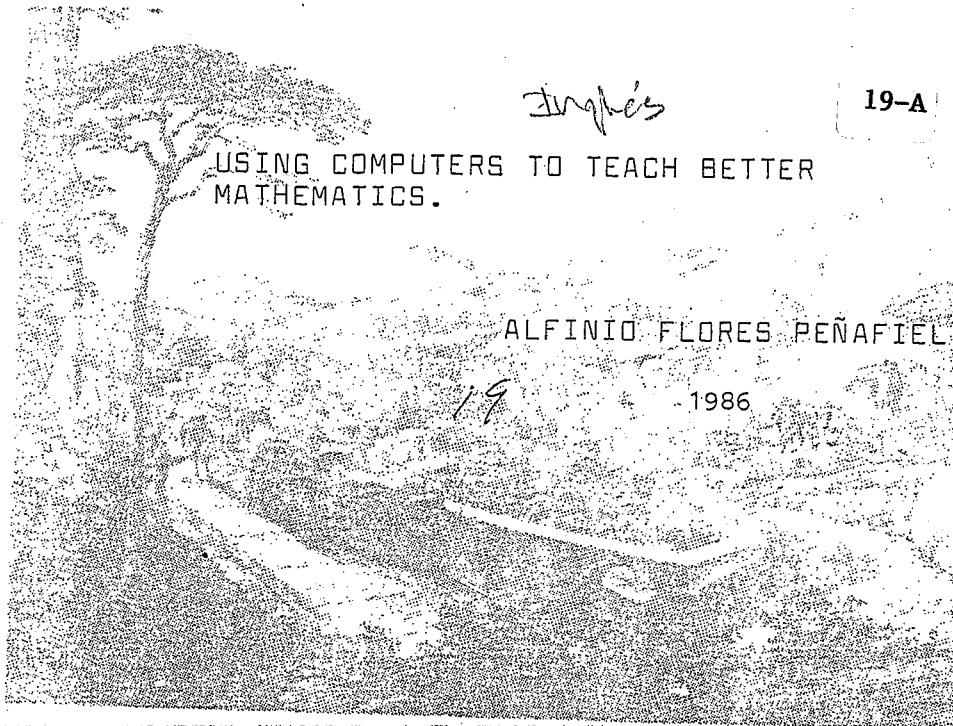


COMUNICACIONES DEL CIMAT



Presentado en: NCTM 64th Annual Meeting, Washington, D. C.,
abril de 1986.

CENTRO DE INVESTIGACION EN MATEMATICAS

Apartado Postal 402

Guanajuato, Gto.

México

Tels. (473) 2-25-50

2-02-58

USING COMPUTERS TO TEACH BETTER MATHEMATICS

Alfinio Flores

Centro de Investigación en Matemáticas, Guanajuato, MEXICO

64th Annual Meeting NCTM, Washington, April 1986

A balanced school curriculum has long been advocated by leading mathematicians. E. H. Moore stated that "the fundamental problem is that of the unification of pure and applied mathematics" (Moore, 1926 p. 45). Felix Klein wrote one of the volumes of his famous lectures for secondary teachers on the mathematics of precision and approximation (Klein, 1928). However, the present high school mathematics lacks completely this approach or any blending between the pure and the applied mathematics. The use of the computer can help to offer a more balanced curriculum. Examples drawn from the realm of algebra will be given to illustrate how teachers can take advantage of the computer to teach better mathematics.

One of the goals of mathematics teaching should be that students learn problem solving methods and skills that are powerful and can be used in a wide range of problems.

Finding roots of polynomials by factoring is a method widely taught, textbooks devote much space to it and classroom teachers a lot of time. Usiskin (1980) lists factoring of polynomials as one of the topics that should not be in the mathematics curriculum. This is an instance of a method that almost never works, except for the examples of the textbook. Very few of the polynomials that appear in applications are easily factorable. Henry Pollak says that, in his long experience of seeing applications of mathematics to the design of real electrical circuits, he never came across a polynomial that could be factored over the integers. An interesting activity, is to randomly generate integers as coefficients of polynomials and check whether the resulting polynomial is factorable over the integers. Run program RANDPOLYNOM for polynomials of the type $x^2 + bx + c$, where b and c are random integers between -100 and 100. The program lists the coefficients of the polynomial and tells if it is factorable. You will be surprised how few of the resulting polynomials are indeed factorable.

Instead of using factoring, one can find roots of a polynomial with methods that illustrate powerful problem solving techniques, such as successive approximations.

The strategy can be described in four steps:

GUESS

CHECK WHETHER YOUR GUESS IS GOOD ENOUGH

GET A BETTER APPROXIMATION FROM THE PREVIOUS ONE

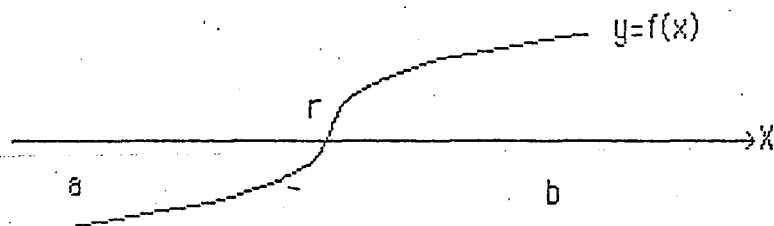
REPEAT THE PROCESS (FROM STEP TWO ON)

The use of computers is particularly well suited for this strategy. They are fast, and it is so easy to tell them to repeat the same procedure again and again.

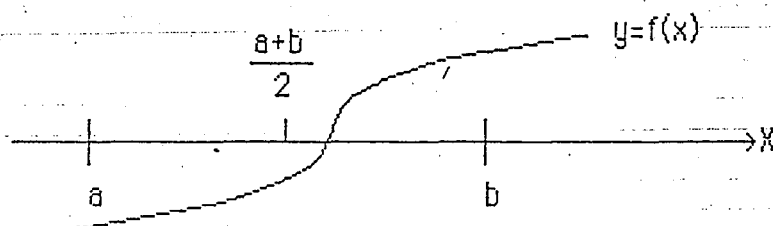
We will discuss two of such methods.

Bisection method (divide and conquer)

For a polynomial p (in fact for any continuous function) if a and b are numbers such that $p(a) < 0$ and $p(b) > 0$, then there is a value r between a and b such that $p(r) = 0$. The number r is called a root of the polynomial.



The numbers a and b are our first guesses to determine an interval where the root is located. To get a closer value for r , we can find the midpoint of the interval and evaluate the polynomial there. If the polynomial is zero, we have a root. If not, we are closer to the solution, because we have the same situation as we started but with an interval of half the length. We can use that half of the interval where the function changes sign, and repeat the procedure. In the case of the example we would take the right half. By getting smaller and smaller intervals we can get the value of the root with any degree of precision.



ACTIVITY 1 Write a computer program (or use program ROOTS1) to approximate the roots of

$$x^3 + 3x^2 + 4x - 11$$

using the bisection method. Run the program and get a value r close enough to a root so that $p(r)$ is in absolute value less than 0.0001. How many times did the computer repeat the procedure to get the desired approximation?

Suppose that now you want more precision, $p(r)$ in absolute value less than 0.0000001. How many more times was the procedure repeated?

The program can be easily modified for the case that $a < b$ and $p(a) > 0$ and $p(b) < 0$. Change your program (or run ROOTS2) to get a root of

$$-x^3 + 3x^2 + 4x - 11$$

Activity 2 Modify your program (or use ROOTS3) so that you can approximate the roots of any polynomial of degree 5 or less.

Approximate the roots of these polynomials.

a) $x^5 + 2x^4 - 4x^3 - 2x^2 + x + 3$

b) $x^5 + x^4 + 1.2x^3 + 0.7x^2 - x + 2$

Our next goal is to write a program to evaluate a polynomial of any degree given its coefficients, at a chosen value x .

Whenever possible it is better to use multiplication instead of exponents because it is more precise. Moreover, some computers do not accept negative values of x when computing x^3 . For higher order polynomials it is too cumbersome to write $x*x*x*x*x*x*x + \dots$

Fortunately there is a way to compute a polynomial without having to multiply so many times. For example, instead of computing

$$6*x*x*x*x + 5*x*x*x*x + 4*x*x + 3*x + 2$$

(9 multiplications and 4 sums), we can tell the computer to compute

$$2 + x*(3 + x*(4 + x*(5 + x*6)))$$

(4 multiplications and 4 sums).

In general, by calculating

$$a_0 + x*(a_1 + x*(a_2 + x*(...)))$$

the number of multiplications grows like n , the degree of the polynomial.

With the usual method the number of multiplications grows like $n^2/2$.

This way of factoring a polynomial is seldom taught in high school.

ACTIVITY 3 Write a computer program using this method to evaluate a polynomial given its coefficients, at any value x . (Or use program POLYNOM1).

ACTIVITY 4 Incorporate the program of activity 3 as a subroutine of the program to approximate roots of polynomials (or use program ROOTS4)

Find at least one root of each of the following polynomials:

a) $x^7 + 2x^6 - 4x^5 - 2x^2 - x^4 - 4x^3 - 2x^2 + x + 3$

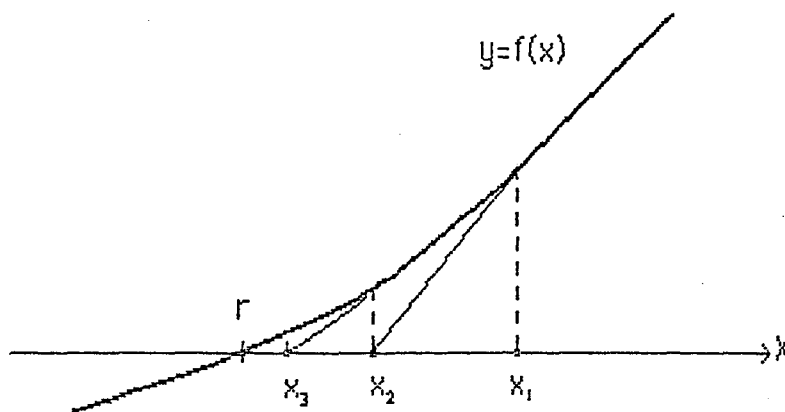
b) $x^5 - 0.2x^4 - 0.4x^3 - 2x^2 + x + 3.14159$

The bisection method to obtain roots of polynomials has several important characteristics. It is an iterative procedure where our approximations get closer and closer to the desired value. We can get the degree of precision wanted by repeating the procedure enough times. This method uses important ideas of mathematics (the theorem of intermediate value and the concept of nested intervals). The method is easy to understand and easy to remember. It is a nice way to teach educated guessing. It exemplifies a powerful idea; namely, guess and have a way so that your next guess is a better one.

One important aspect of computing is that usually there are several procedures to solve a problem, and some of them are more efficient, faster or precise than others. Most people are impressed by how much faster computers are now than 40 years ago. However, there has been an even greater progress in the numerical procedures. In the same span of time where computers evolved to be faster by a factor of 10^5 , in some fields the algorithmic procedures were made faster by a factor of 10^{10} (Boggs, 1981). We will discuss another method to approximate roots of polynomials.

THE NEWTON METHOD

The essence of this method can be illustrated geometrically. If x_1 is a fairly good approximation to a root, then the tangent to the curve at $(x_1, p(x_1))$ will meet the horizontal axis at a much closer approximation.



The slope of the tangent is given by the derivative of the function, $p'(x)$. The equation of the tangent is

$$y - p(x_1) = p'(x_1)(x - x_1)$$

Thus the new approximation is

$$x_2 = x_1 - p(x_1)/p'(x_1)$$

Repeating the procedure we can get better and better approximations.

ACTIVITY 5 Write a program to get a root of $x^3 + 3x^2 + 4x - 11$ using Newton's method (or use program NEWTON1).

Of course the computer can be readily instructed to automatically evaluate the derivative of a polynomial.

ACTIVITY 6 Write a program to obtain the coefficients of the derivative given the coefficients of a polynomial (or use program DERIPOL).

ACTIVITY 7 Incorporate the program of the previous activity as a subroutine

of a program that uses Newton's method to find the roots of a polynomial of any degree.

Use your program (or NEWTON2) to find a root of the following polynomials. Compare how many times the computer repeats the procedure to get the desired precision with Newton's method and with the bisection method.

a) $x^3 + 3x^2 + 4x - 11$

b) $x^5 + x^4 + 1.2x^3 + 0.7x^2 - x + 2$

Of course Newton's method can be used with any function whose derivative can be computed. Write a program (or use program Newton3) to find a zero of $f(x) = \tan(x) - 2*x$

Using Newton's method with calculus students is a nice way to illustrate that computing can gain from mathematics, that learning more advanced mathematics can guide us to find more effective methods. Newton's method is much faster, because for small increments the linear approximation to a function can be pretty close to it. Newton's method is also easy to understand and to recall.

REFERENCES

- Boggs, Paul T. (1981). Mathematical software: how to sell mathematics. In Lynn Arthur Steen (Ed.) Mathematics tomorrow. N.Y.: Springer Verlag. (p. 221-229).
- Fey, James T. (Ed.) (1984). Computing and mathematics: the impact on secondary school curricula. National Council of Teachers of Mathematics.
- Johnson, Jerry (1985, April). Algorithmics and the mathematics curriculum. Paper presented at the annual meeting of the National Council of Mathematics, San Antonio TX.
- Klein, Felix (1928). Elementarmathematik vom höheren Standpunkte aus, Band 3: Präzisions- und Approximationsmathematik. Berlin: Springer.
- Moore, Eliakim Hastings (1926). On the foundations of mathematics. In A general survey of progress in the last twenty-five years, the first yearbook. National Council of Teachers of Mathematics. (p. 32-57).
- Usiskin, Zalman. (1980). What should not be in the algebra and geometry curricula of average college-bound students? The Mathematics Teacher, 73, 413-426.

PROGRAMS

PROGRAM RANDPOLYNOM

```

10 REM RANDPOLYNOM
20 REM GENERATES COEFFICIENTS OF POLYNOMIAL AT RANDOM
30 REM  $X^2+BX+C=0$   $-100 < B,C < 100$ 
40 REM CHECKS WHETHER POLYNOMIAL IS FACTORABLE OVER THE INTEGERS
50 REM *****
60 LET A = 1
70 FOR N = 1 TO 100
80 LET B = INT (200 * RND(1)) - 100
90 LET C = INT (200 * RND(1)) - 100
100 LET D = B*B - 4*A*C
110 IF D < 0 THEN PRINT A;" ";B;" ";C;" COMPLEX ";:GOTO 210
120 LET E = (-B + SQR(D))/2
130 LET F = (-B - SQR(D))/2
140 IF ABS (INT(E) - E) < 0.000001 OR ABS(INT(E)+ 1 - E) < 0.000001 THEN
    LET V1 = 1
150 IF ABS (INT(F) - F) < 0.000001 OR ABS(INT(F)+ 1 - F) < 0.000001 THEN
    LET V2 = 1
160 LET V = V1 *V2
170 IF V = 1 THEN LET M = M+1: PRINT A;" ";B;" ";C;" YES "
180 IF V = 0 THEN PRINT A;" ";B;" ";C,
190 LET V1 = 0
200 LET V2 = 0
210 PRINT N;" "; M / N
220 NEXT N
230 END

```

PROGRAM ROOTS1

```

10 REM ROOTS1
20 REM BISECTION METHOD  $A < B, Y(A) < 0, Y(B) > 0$ 
30 REM *****
40 DEF FN Y(X) = X*X*X + 3*X*X + 4*X - 11
50 INPUT " INTERVAL"; A, B
60 PRINT "R", "Y(R)"
70 LET R = (A + B) / 2
80 PRINT R, FN Y(R)
90 IF ABS (FN Y(R)) < 0.000001 THEN PRINT R : END
100 IF FN Y(R) > 0 THEN LET B = R
110 IF FN Y(R) < 0 THEN LET A = R

```

120 GO TO 70

PROGRAM ROOTS2

```
10 REM ROOTS2
20 REM BISECTION METHOD A < B , Y(A) > 0 , Y(B) < 0
30 REM *****
40 DEF FN Y(X) = -X*X*X + 3*X*X + 4*X - 11
50 INPUT " INTERVAL"; A , B
60 PRINT "R" , "Y(R)"
70 LET R = (A + B) / 2
80 PRINT R , FN Y(R)
90 IF ABS FN Y(R) < 0.000001 THEN END
100 IF FN Y(R) < 0 THEN LET B = R
110 IF FN Y(R) > 0 THEN LET A = R
120 GO TO 70
```

PROGRAM ROOTS3

```
10 REM ROOTS3
20 REM BISECTION METHOD A < B ; Y(A) < 0 , Y(B) > 0
30 REM FOR POLYNOMIALS OF DEGREE UP TO FIVE
40 REM *****
50 REM COEFFICIENTS
60 DIM C(6)
70 FOR M = 1 TO 6
80 PRINT "COEFFICIENT OF X ^ " ; M-1;
90 INPUT C(M)
100 NEXT M
110 REM *****
120 REM MAIN PROGRAM
130 DEF FN Y(X) = C(1) + C(2)*X + C(3)*X*X + C(4)*X*X*X +
    C(5)*X*X*X*X + C(6)*X*X*X*X*X
140 INPUT " INTERVAL"; A , B
150 PRINT "R" , "Y(R)"
160 LET R = (A + B) / 2
170 PRINT R , FN Y(R)
180 IF ABS FN Y(R) < 0.000001 THEN END
190 IF FN Y(R) > 0 THEN LET B = R
200 IF FN Y(R) < 0 THEN LET A = R
210 GO TO 160
```

PROGRAM POLYNOM1

```
10 REM POLYNOM1
20 REM EVALUATE POLYNOMIAL OF DEGREE N AT A POINT X
30 REM *****
40 REM DATA OF POLYNOMIAL
50 INPUT "DEGREE: "; N
60 INPUT "X "; X
70 DIM C(N+1)
80 FOR J = 1 TO N+1
90 INPUT "COEFFICIENT: " C(J)
100 NEXT J
110 REM *****
120 REM EVALUATE
130 LET Y = C(N+1)
140 FOR J = N TO 1 STEP -1
150 LET Y = X * Y + C(J)
160 NEXT J
170 PRINT X , Y
```

PROGRAM ROOTS4

```
10 REM ROOTS4
20 REM BISECTION METHOD FOR A POLYNOMIAL OF ANY DEGREE
30 REM *****
40 REM DATA
50 INPUT "DEGREE: "; N
60 DIM C(N+1)
70 FOR J = 1 TO N+1
80 INPUT "COEFFICIENT: " C(J)
90 NEXT J
100 REM *****
110 REM MAIN PROGRAM
120 INPUT "INTERVAL"; A , B
130 PRINT "R" , "Y(R)"
140 LET R = (A + B) / 2
150 GOSUB 1000
160 IF ABS FN Y(R) < 0.000001 THEN END
170 IF FN Y(R) > 0 THEN LET B = R
180 IF FN Y(R) < 0 THEN LET A = R
190 GO TO 140
200 REM *****
```

```

1000 REM SUBROUTINE TO EVALUATE
1010 LET Y = C(N+1)
1020 FOR J = N TO 1 STEP -1
1030 LET Y = R * Y + C(J)
1040 NEXT J
1050 PRINT R , Y
1060 RETURN

```

PROGRAM NEWTON1

```

10 REM NEWTON1
20 REM *****
30 DEF FN Y(X) = X*X*X + 3*X*X + 4*X - 11
40 DEF FN Z(X) = 3*X*X + 6*X + 4
50 INPUT "YOUR GUESS: ", R
60 PRINT "R" , "Y(R)": PRINT
70 PRINT R , FN Y(R)
80 IF ABS (FN Y(R) ) < 0.000001 THEN PRINT R : END
90 LET R = R - FN Y(R)/ FN Z(R)
100 GOTO 70

```

PROGRAM DERPOL

```

10 REM DERPOL
20 REM DERIVATIVE OF POLYNOMIAL OF ANY DEGREE
30 REM *****
40 REM DATA OF POLYNOMIAL
50 INPUT "DEGREE: "; N
60 DIM C(N+1)
70 FOR J = 1 TO N+1
80 PRINT COEFFICIENT OF X ^ " ; J-1;
90 INPUT C(J)
100 NEXT J
110 REM *****
120 REM DERIVATIVE
130 PRINT "DERIVATIVE: ";
140 DIM B(N)
150 FOR J = 1 TO N
160 LET B(J) = J * C(J+1)
170 PRINT B(J) ; "*X ^ " ; J
180 NEXT J

```

3050 RETURN

PROGRAM NEWTON3

10 REM NEWTON3

20 REM FOR ANY FUNCTION WHOSE DERIVATIVE CAN BE DEFINED

30 REM *****

40 REM MAIN PROGRAM

50 DEF FN Y(X) = TAN(X) - 2*X

60 DEF FN Z(X) = TAN(X) * TAN(X) - 1

70 INPUT "YOUR GUESS: ", R

80 PRINT "R" , "Y(R)"

90 PRINT R , FN Y(R)

100 IF ABS (FN Y(R)) < 0.000001 THEN PRINT R: END

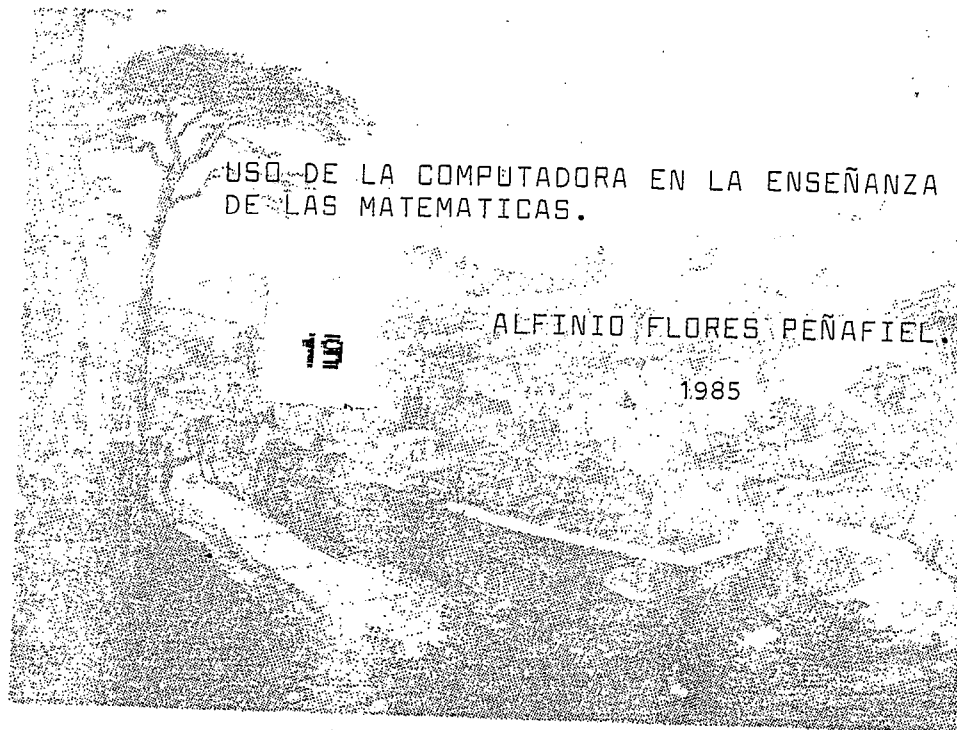
120 LET R = R - FN Y(R)/ FN Z(R)

130 GOTO 90

PROGRAM NEWTON2

```
10 REM NEWTON2
20 REM FOR A POLYNOMIAL OF ANY DEGREE
30 REM*****
40 REM DATA OF POLYNOMIAL
50 INPUT "DEGREE: "; N
60 DIM C(N+1)
70 FOR J = 1 TO N+1
80 INPUT "COEFFICIENT: " C(J)
90 NEXT J
100 GOSUB 2000
110 REM *****
120 REM MAIN PROGRAM
130 INPUT "YOUR GUESS: ", R
140 PRINT "R" , "Y(R)"
150 GOSUB 1000
160 IF ABS(Y) < 0.000001 THEN PRINT R: END
170 GOSUB 3000
180 LET R = R - Y / Z
190 GOTO 150
200 REM *****
1000 REM SUBROUTINE TO EVALUATE
1010 LET Y = C(N+1)
1020 FOR J = N TO 1 STEP -1
1030 LET Y = R * Y + C(J)
1040 NEXT J
1050 PRINT R , Y
1060 RETURN
1070 REM *****
2000 REM SUBROUTINE FOR COEFFICIENTS OF DERIVATIVE
2010 DIM B(N)
2020 FOR J = 1 TO N
2030 LET B(J) = J*C(J+1)
2040 NEXT J
2050 RETURN
2060 REM *****
3000 REM SUBROUTINE TO EVALUATE DERIVATIVE
3010 LET Z = B(N)
3020 FOR J = N-1 TO 1 STEP -1
3030 LET Z = R*Z+B(J)
3040 NEXT J
```

COMUNICACIONES DEL CIMAT



Presentado en el 4º Coloquio de Matemáticas, Taxco, Gro.,
agosto de 1985.

CENTRO DE INVESTIGACION EN MATEMATICAS

Apartado Postal 402

Guanajuato, Gto.

México

Tels. (473) 2-25-50

2-02-58

Uso de la computadora en la enseñanza de las matemáticas

Alfinio Flores Peñafiel
Centro de Investigación en Matemáticas

4^o Coloquio de Matemáticas
Taxco, Gro. 21 de agosto de 1985

Introducción

El uso de la computadora puede modificar profundamente la práctica de la enseñanza de las matemáticas.

La computadora puede ser un medio para:

- lograr una enseñanza de las matemáticas más balanceada
- establecer una conexión entre la teoría y la práctica
- lograr la unificación de las matemáticas puras y aplicadas.

Sin embargo, muchas veces el cambio no es en beneficio de los alumnos, ni de las matemáticas; sino que por el contrario, sucede que el medio se convierte en un fin en sí mismo y el aprendizaje de la computación desplaza al de las matemáticas. Frecuentemente los cursos de computación son impartidos por los maestros de matemáticas durante el tiempo asignado a dicha materia.

A pesar del enorme potencial que ofrece esta herramienta, no es claro cómo y de qué forma aprovecharla. No hay consenso acerca de cuál es la manera de utilizar la computadora para lograr una mejor comprensión de los conceptos y el dominio de las técnicas y habilidades matemáticas. Hay una cantidad de software comercial, con una gran variedad de imágenes y sonidos pero con una gran pobreza conceptual. También hemos observado qué poco se ha podido integrar el uso de la computadora en la enseñanza de las diferentes materias. La enseñanza de la computación se hace sin conexión con las otras asignaturas. Sin embargo, existen ejemplos aislados donde el uso de la computadora se hace de manera integrada (ver Smith, 1984). En algunos casos existen complementos para el libro de texto (Barker y Ward, 1984).

Si bien es cierto que la presencia de la computadora es cada día más frecuente en las escuelas y universidades de nuestro país, la mera accesibilidad a estas máquinas no garantiza su uso adecuado.

Es necesario determinar cuáles son los conceptos fundamentales de las matemáticas, cuáles deben ser enseñados, cuáles de éstos pueden ser mejor tratados con la computadora y de qué manera. Con el acceso de prácticamente todos los alumnos a la computadora en un futuro cercano, es también necesario reexaminar el énfasis que se hace en la enseñanza. Habilidades a las que se dedica mucha atención en el presente, serán sustituidas por otras. (Conference Board for the Mathematical Sciences, 1982; 1984)

Por ejemplo, programar la computadora es una oportunidad magnífica para desarrollar el pensamiento algorítmico, así como para aprender a plantear y resolver problemas.

La preparación de los nuevos matemáticos y profesores de matemáticas debe también reflejar estos cambios. Hay que dar el tiempo y énfasis necesarios para que aprendan cómo usar la computadora para la enseñanza de las matemáticas.

Asimismo, la computadora es capaz de ahorrarnos una cantidad enorme de trabajo. Es evidente que en los cursos de matemáticas se debe aprovechar esta capacidad. Los paquetes de estadística, los de álgebra lineal, los de soluciones numéricas de ecuaciones diferenciales son ejemplos claros. Este es uno de los usos más importantes de la computadora, aunque no se discutirá en el presente trabajo.

A continuación daré algunos ejemplos de cómo la computadora puede ser utilizada dentro de algunos cursos de matemáticas, tratando que cumplan con las siguientes características:

- el alumno tiene un papel activo;
- el alumno le dice a la máquina qué debe hacer, la utiliza como herramienta o le enseña a hacer algo (ver Taylor, 1980);
- se tratan conceptos importantes de las matemáticas;
- no necesitan equipo muy sofisticado y costoso;
- pueden realizarse en condiciones reales;

Daré tres tipos de ejemplos:

- primero: los alumnos escriben sus propios programas;
- segundo: los alumnos utilizan un programa escrito por otros para ilustrar o explorar algún concepto;
- tercero: uso de paquetes "inteligentes".

1 PROGRAMAS ESCRITOS POR LOS ALUMNOS

Algunos investigadores han encontrado paralelismos entre el tipo de pensamiento que se requiere para escribir, probar, corregir y refinar un programa y varios aspectos del pensamiento matemático. Hatfield (1962) menciona los siguientes como algunos procesos del pensamiento matemático que se pueden beneficiar por programar una computadora:

- analizar
- simplificar
- particularizar
- generalizar
- justificar
- conjeturar
- estructurar

Los alumnos pueden escribir programas sencillos y no muy largos que ilustren algunos conceptos claves de los cursos. El tener que explicar a la computadora con precisión cómo utilizar el concepto, es una de las mejores formas de lograr en el alumno una mejor comprensión. Pequeños programas pueden ilustrar grandes ideas. El siguiente programa grafica líneas rectas a través del origen (la computadora Sinclair coloca el origen de coordenadas en la parte inferior izquierda de la pantalla):

```
5 INPUT "pendiente "; A
10 FOR X = 0 TO 255
15 LET Y = A * X
20 PLOT X , Y
30 NEXT X
```

Los alumnos con este programa están trabajando con las ideas fundamentales de la geometría analítica, que son la de asociar puntos en el plano con parejas ordenadas de números, la de hacer corresponder curvas en el plano a ecuaciones de dos variables. Además, en este caso se familiarizan también con el concepto de pendiente de una recta.

Se dan ejemplos para valores de la pendiente de 0, 0.5, 1, 2, 4.

Para escribir un programa que grafique con el origen en el centro de la pantalla, o para cambiar de escala, los alumnos tendrán que entender las transformaciones tales como traslaciones y homotecias. El programa GRAFICAS 1 (ver apéndice) traza la gráfica de la función $y = x + 2$. Con sólo modificar un renglón (el 30) se puede estudiar una gama muy amplia de funciones. Además para cada función se puede ver el resultado de graficarla con distintas escalas.

Programas para cálculo.

El método de incrementos juega un papel fundamental para comprender el concepto de derivada. El siguiente programa calcula la pendientes de secantes a una curva, para intervalos cada vez más pequeños.

```
10 DEF FN Y(X) = X * X
20 INPUT " punto inicial " ; X
30 INPUT " incremento " ; H
40 FOR N = 1 TO 8
50 LET S = ( FN Y(X+H) - FN Y(X) ) / H
60 PRINT H , S
70 LET H = H / 4
80 NEXT N
```

Un alumno que escriba este programa tendrá que prestar atención cuidadosa al cociente de las diferencias para el renglón clave del programa.

Se den dos ejemplos de resultados de este programa, ambos para un valor de $x = 3$, con un incremento positivo en un caso y uno negativo en el otro.

Derivación numérica

Este programa grafica una función $f(x)$ y la aproximación a la derivada de la función dada por el cociente de los incrementos

$$\frac{f(x+h) - f(x)}{h}$$

```
20 DEF FN Y (X) = X*X
30 INPUT "incremento: "; H
40 GOSUB 300
50 FOR X=-3.125 TO 3.25 STEP .0125
60 LET Y = FN Y(X)
70 GOSUB 200
80 LET Y = ( FN Y(X+H) - FN Y(X) ) / H
90 GOSUB 200
100 NEXT X
110 STOP
200 REM subrutina para graficar ***
300 REM *** subrutina para ejes ***
```

Las subrutinas para graficar y para los ejes son las mismas que las del programa GRAFICAS 1 (ver apéndice)

Se anexan ejemplos para las funciones $f(x) = x^2$ y para $f(x) = \text{sen}(x)$

(Para más ejemplos de programas que los alumnos pueden escribir para cálculo ver Smith, 1984; Sagan, 1984; Flores-Peñañiel, 1984)

2 LOS ALUMNOS USAN PROGRAMAS YA HECHOS

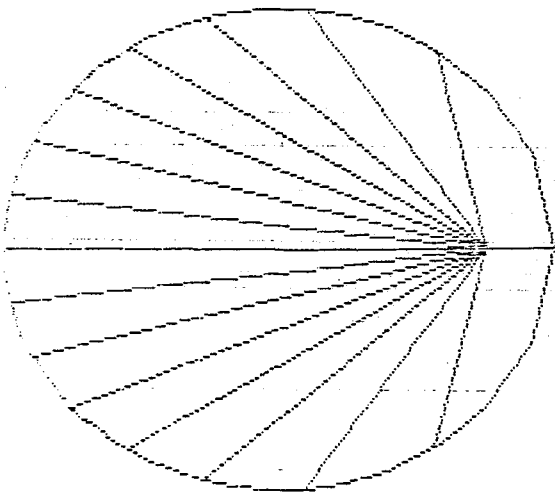
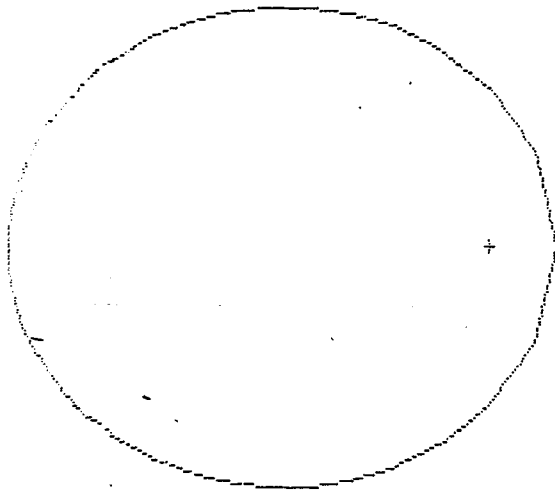
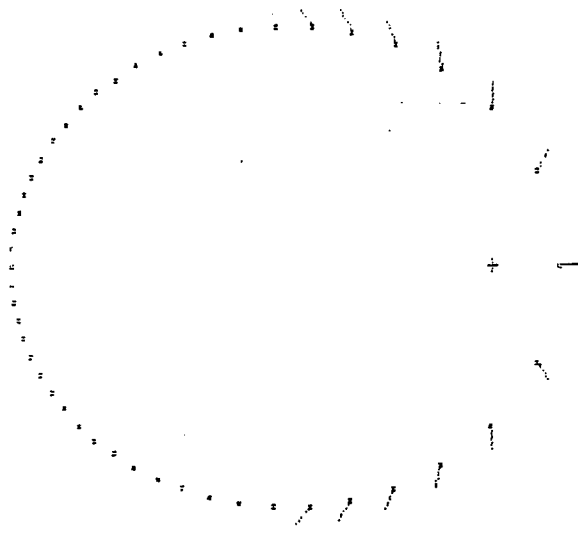
Los alumnos pueden utilizar programas ya hechos para explorar y repasar conceptos matemáticos. En esta modalidad no es necesario que los alumnos comprendan todo el programa. Sólo es necesario que entiendan la parte del programa relacionada directamente con el concepto en cuestión, a fin de que puedan hacer modificaciones sencillas al programa, y en ocasiones ni siquiera tendrán que modificar el programa, sino sólo cambiar datos o parámetros. Esto les dará oportunidad de ver muchos ejemplos, ver qué pasa con las modificaciones, tratar de encontrar generalizaciones a partir de los casos vistos. La computadora se puede convertir así en el corazón de un enfoque "experimental" de las matemáticas. El programa SISTEMAS 1 es un ejemplo de un programa que forma parte de un paquete de actividades para nivel medio superior, que formarán un laboratorio de matemáticas.

Simulación

Representar en forma gráfica o visual un fenómeno o proceso ha sido siempre una de las mejores maneras de lograr una mejor comprensión de este. Muchas veces aún con programas cortos se pueden simular procesos (ver por ejemplo Flores, 1984). La computadora tiene la capacidad de representar procesos dinámicos, de representar cambios conforme transcurre el tiempo. Este es uno de los aspectos que la imagen estática de un libro no puede dar.

Por ejemplo, una función de $\mathbb{R} \rightarrow \mathbb{R}^2$ puede tener la misma imagen que otra, sin embargo recorrerla con velocidades distintas, como $f(t) = (t^3, t^3)$ y $g(t) = (t, t)$ que representan la misma recta, pero ésta se recorre de manera diferente al hacer variar t . El programa ECPARAMETR grafica ecuaciones en forma paramétrica. Al ejecutarlo los alumnos pueden percibir la diferencia de velocidades al recorrer la recta de una forma o de otra. El hecho de que la computadora grafique un conjunto discreto de puntos puede así tener ventajas didácticas.

Al estudiar el movimiento planetario no sólo importa la forma de la órbita, sino cómo es recorrida por el planeta. Este no se mueve alrededor del sol con una rapidez constante, sino que se mueve más rápido cerca del sol. Esto es más notorio mientras mayor sea la excentricidad de la órbita. Un ejemplo particularmente atractivo en este año es el cometa Halley. El programa KEPLERLEY2 simula el movimiento de un cometa en su órbita de modo que se satisfaga la segunda ley de Kepler. (Adaptado del programa "Kepler" (Tattersfield, 1984))



3 EL USO DE PAQUETES "INTELIGENTES"

Hay diversos paquetes con los cuales se pueden enfatizar diferentes conceptos matemáticos. Entre ellos tal vez los más conocidos sean las hojas electrónicas (por ejemplo VisiCalc). Arganbright (1984) da algunas ideas de cómo utilizarlas para la enseñanza de las matemáticas. Tiene también un libro de cómo utilizarlas para técnicas tales como el método de Newton, polinomios de Taylor, valores propios, la regla de Bayes, el algoritmo simplex y otras más (Arganbright, 1985).

Existen otros paquetes tales como TKISolver que resuelven ecuaciones de modo similar a como se hace en álgebra. (Para una reseña de este paquete ver Williams, 1982). De acuerdo a este autor, paquetes como este tienen el potencial de convertirse para el álgebra en lo que las calculadoras son para la aritmética, un medio rápido y preciso de obtener los resultados.

Las computadoras no sólo son capaces de hacer operaciones numéricas, sino que también pueden manipular expresiones simbólicas y dar respuestas con símbolos. Aunque los sistemas simbólicos han existido para computadoras grandes desde hace muchos años (por ejemplo el sistema MACSYMA), hay ahora también paquetes para micro computadoras. Comentaré brevemente acerca de uno de estos, el paquete muMATH, que entre otras cosas hace operaciones aritméticas exactas con números enteros, racionales, algebraicos y complejos, con números enteros hasta de 511 cifras. Por ejemplo puede simplificar fracciones

? 6/8;

@ 3/4

Opera con complejos:

? (3 + 2 * #i) * (5 - #i);

@ 20 + 9 * #i

Puede encontrar máximo común divisor entre dos números:

? GCD(24,60);

@ 12

Encontrar factorial de números y elevarlos a una potencia:

? 10! ^ 10;

@ 39594086612242519324387557078266845776303882240000000000000000000

Puede encontrar límites para los que muchas veces recurrimos a la regla de L'Hôpital, tal como el límite de $x / \sin(x)$ cuando $x \rightarrow 0$

? LIM (X / SIN (X), X, 0);

@ 1

El sistema también resuelve integrales como $\int (x^2 + x \sin(x^2)) dx$.

? INT (C * X ^ 2 + X * SIN (X ^ 2), X);

@ C * X/3 - COS (X ^ 2)/2

Las nuevas versiones de este sistema incluso resuelven ecuaciones diferenciales. (Ver Squire, 1985 para una descripción de las capacidades de este sistema). En algunas universidades ya se usan estos sistemas. Francis Moon, jefe del Departamento de Mecánica Teórica y Aplicada de la Universidad de Cornell, afirma que esta nueva tecnología de software tiene la potencialidad para introducir otra vez el análisis matemático en la investigación industrial junto con el ahora popular CAD (diseño por computadora) y otros métodos numéricos de análisis (SIAM NEWS julio de 1985). Wilf (1982) plantea algunas interrogantes acerca del impacto de estos sistemas en la enseñanza de las matemáticas en la universidad. ¿Nos pasará algo semejante a lo que a muchos maestros de primaria, que todas las operaciones que enseñaban a hacer en largos y penosos meses, ahora un niño es capaz de aprender en unos minutos con una calculadora? ¿Nos condremos en las universidades al uso de estos sistemas, así como en tantas primarias y secundarias se oponen a que los alumnos usen las calculadoras?

Referencias

- Arganbright, Deane. Mathematical applications of an electronic spreadsheet. En Computers in mathematics education. 1984 Yearbook, National Council of Teachers of Mathematics, 1984. p. 184-193.
- Arganbright, Deane. Mathematical applications of electronic spreadsheets. McGraw Hill, 1985.
- Berker, W.H.; Ward, J.E. The calculus companion to accompany Calculus 2nd ed by Howard Anton. Wiley, 1984.
- Conference Board of the Mathematical Sciences. The mathematical sciences curriculum K-12: what is still fundamental and what is not. National Science Foundation, 1982.
- Conference Board of the Mathematical Sciences. New goals for mathematical sciences education. Conference Board of the Mathematical Sciences, 1984.
- Fey, James T.(ed) Computing and mathematics. National Council of Teachers of Mathematics, 1984.
- Flores, Alfinio. "A microcomputer and the law of small numbers." Arithmetic Teacher, 31(7), (marzo 1984) 60-61.
- Flores Peñafiel, Alfinio. "La microcomputadora en la enseñanza del cálculo". Trabajo presentado en el Congreso Anual de la Sociedad Matemática Mexicana, Mérida, noviembre 1984.
- Guidelines and standards for the education of secondary school teachers of science and mathematics. American Association for the Advancement of Science, 1971.
- Hatfield, Larry L. "Instructional computing in mathematics teacher education." Journal of Research Development in Education, 15 (4), (1982) 30-44.
- The muMATH/mu SIMP-80 Symbolic Mathematics System Reference Manual for the Apple II Computer. The Soft Warehouse, 1981.

News Briefs "Cornell will use MACSYMA and muMATH." SIAM News, 18(4), julio 1985, 2.

Recommendations on the mathematical training of teachers. Mathematical Association of America. 1983.

Recommendations for a general mathematical sciences program. Mathematical Association of America. 1981.

Sagan, Hans. Calculus accompanied on the Apple. Reston, 1984.

Smith, David A. Interface: calculus and the computer 2a ed. Saunders College Publishing, 1984.

Squire, William. "muMATH system effective tool for algebra." SIAM News, 17(6), (noviembre 1984) 4.

Tattersfield, Donald. Halley's comet. Blackwell, 1984.

- Wilf, Herbert S. "The disk with the college education." American Mathematical Monthly 89 (1982), 4-8.

Taylor, R. P. The computer in the school: tutor, tool, tutor. New York, Teachers College Pr., 1980.

Williams, Gregg. "Software Arts' TKI Solver." Byte, octubre 1982, 360-76.

PROGRAMA PARA GRAFICAR CON COORDENADAS CARTESIANAS

```
5 REM GRAFICAS 1
8 REM COORDENADAS CARTESIANAS
10 GOSUB 100: REM grafica ejes
15 REM *****
20 REM programa principal
20 FOR X = -3.2 TO 3.175 STEP .025
30 LET Y = X*X
40 GOSUB 200
50 NEXT X
99 STOP
100 REM ***** subrutina para ejes *****
110 PLOT 128 , 0 : DRAW 0 , 175
120 PLOT 0 , 88 : DRAW 255 , 0
130 RETURN
200 REM *****
205 REM subrutina para cambio de coordenadas y graficar
210 LET H = 40 * X + 128
220 LET V = 40 * Y + 88
230 IF V < 0 OR V > 175 THEN RETURN
240 PLOT H , V
250 RETURN
```

PROGRAMA PARA GRAFICAR ECUACIONES PARAMETRICAS

```
5  REM ECPARAMETR
10 GOSUB 100: REM grafica ejes
15 REM *****
20 REM programa principal
40 FOR T = -3.2 TO 3.175 STEP .025
50 LET X = T * T * T
60 LET Y = T * T
70 GOSUB 200 : REM grafico punto
80 NEXT X
99 STOP
100 REM ***** subrutina para ejes *****
110 PLOT 128 , 0 : DRAW 0 , 175
120 PLOT 0 , 68 : DRAW 255 , 0
130 RETURN
200 REM *****
205 REM subrutina para cambio de coordenadas y graficar
210 LET H = 40 * X + 125
220 IF H < 0 OR H > 255 THEN RETURN
230 LET V = 40 * Y - 68
240 IF V < 0 OR V > 175 THEN RETURN
240 PLOT H , V
250 RETURN
```

2a LEY DE KEPLER

```

5 REM KEPLERLEY2
10 PRINT AT 3,0 ; "SIMULACION DE UN COMETA DE PERIODO 12 años,
    EXCENTRICIDAD 0.7, POSICIONES CADA 3 MESES"
15 LET a = 14 : REM semieje mayor
20 LET e = .7 : REM excentricidad
30 LET pp = 1986 : REM paso por perihelio
40 LET ti = 1980 : REM tiempo inicial (año)
50 LET ma = 30 : REM movimiento anual (grados por año)
60 LET u = 48 : REM número de datos
70 LET it = 0.25 : REM intervalo de tiempo (años)
80 DIM k(u+1) : DIM l(u+1) : DIM r(u+1)
90 DIM x(u+1) : DIM y(u+1)
100 LET q = a * (1 - e)
110 LET b = a * SQR (1 - e*e)
120 LET rad = PI / 180
130 LET teta = 0
140 REM ***** calcula posiciones *****
150 FOR n = 1 TO u
160 LET tetac = ma * ( ti + (n - 1) * it - pp ) * rad
170 LET dif = tetac - ( teta - e * SIN teta )
180 LET teta = teta + dif / ( 1 - e * COS teta )
190 IF ABS (dif) > .0001 THEN GO TO 170
200 REM **** convierte polares a cartesianas ****
210 LET x(n) = a * (COS teta - e)
220 LET y(n) = b * SIN teta
230 NEXT n
240 REM ***** escala para la pantalla *****
250 FOR n = 1 TO u
260 LET k(n) = 30 - q + x(n)
270 LET l(n) = 10 + y(n)
280 LET r(n) = SQR ( x(n) * x(n) + y(n) * y(n) )
290 NEXT n
295 CLS
300 PLOT 210, 88 : DRAW 4,0 : PLOT 212, 86 : DRAW 0,4 : REM sol
305 PAUSE 60
310 REM ***** grafica cometa *****
320 FOR n = 1 TO u
330 PLOT 8 * k(n), 8 * (21 - l(n))
340 DRAW -1,0 : DRAW 0,-1 : DRAW 1,0 : DRAW 0,1
350 IF r(n) < a THEN DRAW (k(n) - 212/8)*8/r(n) , (10-l(n))*8/r(n)

```

```

360 PAUSE 25
370 NEXT n
375 STOP
380 CLS
390 PLOT 210, 88 : DRAW 4,0 : PLOT 212, 86 : DRAW 0,4
400 REM ***** órbita *****
405LET k(u+1) = k(1)
410 LET l(u+1) = l(1)
420 FOR n = 1 TO u
430 PLOT 8 * k(n), 8*(21-l(n))
440 DRAW 8*k(n+1) - 8 * k(n), 8*(21-l(n+1))-8*(21-l(n))
450 PAUSE 5
460 NEXT n
470 STOP
475 REM ***** radios vectores *****
480 FOR n = 1 TO u STEP 2
490 PLOT 8 * k(n), 8*(21-l(n))
500 DRAW 212- 8 * k(n), 88-8*(21-l(n))
510 PAUSE 25
520 NEXT n
530 REM KEPLERLEY2 *****

```