DOCUMENT RESUME

ED 218 126 SE 038 236

AUTHOR Rheinboldt, Werner C.
TITLE Horner's Scheme and Related Algorithms. Applications of Computer Science. [and] Algorithms for Finding Zeros of Functions. Computer Science. Modules and Monographs in Undergraduate Mathematics and Its Applications Project. UMAP Units 263 and 264.
INSTITUTION Education Development Center, Inc., Newton, Mass.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 80
GRANT SED-76-19615-A02
NOTE 59p.

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Algorithms; *College Mathematics; Computer Programs; *Computer Science; *Computer Science Education; Higher Education; Instructional Materials; *Learning Modules; *Mathematical Applications; Problem Solving; Supplementary Reading Materials
IDENTIFIERS *BASIC Programing Language

ABSTRACT

This material contains two units which view applications of computer science. The first of these examines Horner's scheme; and is designed to instruct the user on how to apply both this scheme and related algorithms. The second unit aims for student understanding of standard bisection, secant, and Newton methods of root finding and appreciation of their limitations and strong points. An introduction to more recent root finding methods is also provided. Both modules contain exercises, and answers to these problems are given at the conclusion of each unit. (MP)

ED218126

UMAP MODULE 263

MODULES AND MONOGRAPHS IN UNDERGRADUATE MATHEMATICS AND ITS APPLICATIONS

Horner's Scheme and Related Algorithms

by Werner C. Rheinboldt

$x_0 = -1$

$p(x)$	1	-2	0	1	-1	
	0	-1	3	-3	2	
$q_1(x)$	1	-3	3	-2	1	$= p(-1)$
	0	-1	4	-7		
$q_2(x)$	1	-4	7	-9		$= p'(-1)$
	0	-1	5			
$q_3(x)$	1	-5	12			$= \frac{1}{2!} p''(-1)$
	0	-1				
$q_4(x)$	1	-6				$= \frac{1}{3!} p'''(-1)$

$= \frac{1}{4!} p^{(4)}(-1)$

Applications of Computer Science

Birkhäuser Boston Inc.
380 Green Street
Cambridge, MA 02139

2

SE 038 236

Intermodular Description Sheet: *UMAP Unit 263*

Title: HORNER'S SCHEME AND RELATED ALGORITHMS

Author: Werner C. Rheinboldt
Department of Mathematics
and Statistics
University of Pittsburgh
Pittsburgh, PA 15261

Review Stage/Date: IV 7/30/80

Classification: APPL COMP SCI

Prerequisite Skills:
1. Definition of a derivative.
2. High school algebra.
3. Fundamentals of how to draw a flowchart.

Output Skills:
1. Be able to apply Horner's and the related algorithms of the unit.

The Project would like to thank Charles Votaw of Fort Hays State University and Brian J. Winkel of Albion College for their reviews, and all others who assisted in the production of this unit.

This material was field-tested and/or student reviewed in preliminary form in class by Jonathan Choate of The Groton School; Gene B. Chase of Messiah College; Donald G. Malm of Oakland University, and; Mary Hayward of Roberts Wesleyan College, and has been revised on the basis of data received from these sites.

This material was prepared with the partial support of National Science Foundation Grant No. SED76-19615 A02. Recommendations expressed are those of the author and do not necessarily reflect the views of the NSF or the copyright holder.

HORNER'S SCHEME AND RELATED ALGORITHMS

by

Werner C. Rheinboldt
Department of Mathematics
and Statistics
University of Pittsburgh
Pittsburgh, PA 15261

TABLE OF CONTENTS

1. INTRODUCTION

Our basic problem is the computational evaluation of a polynomial

(1) $p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_2 x^2 + a_1 x + a_0$

and its derivatives $p'(x), p''(x), \dots, p^{(n)}(x)$ for a prescribed value $x = x_0$. Here the coefficients a_n, $a_{n-1}, \dots, a_0$ are given real numbers.

Let us consider first a simple cubic polynomial

(2) $p(x) = 3x^3 - 4x^2 + 2x - 3.$

For any given number x_0 the evaluation of $p(x_0)$ does not present any principal difficulties. We may compute x_0^2 and x_0^3 and then combine them together appropriately. In an informal programming language this may be written as the following algorithm:

(3)
1. Input $\{x_0\}$
2. $u := x_0 x_0$
3. $v := u x_0$
4. $p := 3v - 4u + 2x_0 - 3$
5. Output $\{x_0, p\}$.

(:= is used to represent assignment of a value to a variable.)

Altogether there are five multiplications and three additions (or subtractions). For the general polynomial (1) this approach would require the computations:

(4)
$$u_1 = x_0,\ u_2 = u_1 x_0,\ u_3 = u_2 x_0,\ \dots,\ u_n = u_{n-1} x_0,$$
$$p = a_n u_n + a_{n-1} u_{n-1} + \dots + a_1 u_1 + a_0.$$

Thus altogether we need 2n-1 multiplications and n additions. Suppose that a particular computer uses a sec and μ sec for any addition (or subtraction) and multiplication, respectively. Then our method (4) takes at least $(2n-1)\mu + n\alpha$ seconds. Without question a practical computer program would run longer than that, since it takes time to fetch and store the data, to control the loop involved in (4), and to perform the input and output. But the overall time should be proportional to $((2n-1)\mu + n\alpha)$. The next section shows that we can do better than that.

2. HORNER'S SCHEME

How can we reduce the number of arithmetic operations in the evaluation of a polynomial? The clue is a suitable factoring of p(x). In fact, (2) can be written as follows:

$$p(x) = ((3x-4)x+2)x-3.$$

Now there are only three multiplications and three additions. That does not appear to be much of a savings in this case but it does represent a big savings when the degree of p(x) goes up.

We shall discuss this approach of evaluating a polynomial in the form of a scheme for hand computation. Let a general cubic polynomial be given:

$$(5) \qquad p(x) = a_3x^3 + a_2x^2 + a_1x + a_0.$$

We may rewrite it as

$$p(x) = ((a_3x+a_2)x+a_1)x+a_0.$$

To evaluate this for $x = x_0$ we use a table with three rows and four columns. Into the first row we write the four coefficients a_3, a_2, a_1, a_0 (in that order), and into column 3, row 2 we place a zero. The rest of the boxes are initially blank.

(6)

	Col 3	Col 2	Col 1	Col 0
Row 1	a_3	a_2	a_1	a_0
Row 2	0	a_3x_0	$(a_3x_0+a_2)x_0$	$((a_3x_0+a_2)x_0+a_1)x_0$
Row 3	a_3	$a_3x_0+a_2$	$(a_3x_0+a_2)x_0+a_1$	$((a_3x_0+a_2)x_0+a_1)x_0+a_0$

The computation proceeds from the left-most column to the right and consists of looping through the following two steps for $k = 3,2,1,0$:

(a) Add the numbers in rows 1 and 2 of column k and write the result into row 3 of the same column.

(b) For $k \geq 1$ multiply the number in row 3 of column k by x_0 and place the result into row 2 of column $k-1$.

This process is indicated by arrows in the Table (6) and the results are indicated in each field. The final result in row 3 of the last column is the value of p at the point x_0. This method of computing the value of a polynomial is called *Horner's Scheme*.

We give Horner's scheme for our special polynomial (4) and two different values of x_0:

(7)

$x_0 = 2$

3	-4	2	-3	
0	6	4	12	
3	2	6	9	$p(2) = 9$

$x_0 = -1$

3	-4	2	-3	
0	-3	7	-9	
3	-7	9	-12	$p(-1) = -12$

As another example, consider the quartic polynomial

(8) $$p(x) = x^4 - 2x^3 + x - 1.$$

Note here that the coefficient of x^2 is zero; it should be included in the computation with that value.

$x_0 = -1$

	1	-2	0	1	-1	
(9)	0	-1	3	-3	2	
	1	-3	3	-2	1	$p(-1) = 1$

Exercises

1. Evaluate (2) at $x_0 = 1$, $x_0 = -1$, $x_0 = 10$. Check your answers.
2. Evaluate (8) at $x_0 = 0$, $x_0 = 1$, $x_0 = 2$. What can you say about the behavior of $p(x)$ in the intervals $-1 \le x \le 0$ and $1 \le x \le 2$?
3. Differentiate the polynomial (8) and evaluate the resulting cubic polynomial at $x_0 = -1$ and $x_0 = 2$.

---

3. IMPLEMENTATION OF HORNER'S SCHEME

How can we program (6) for a general polynomial (1)? Assume that the coefficients $a_0, a_1, \ldots a_n$ are stored in an array of length n+1. If we are in column k of Table (6) and the number in row 3 of that column is stored in p, then the numbers in rows 2 and 3 of column k-1 will be px_0 and $px_0 + a_{k-1}$, respectively. (In the left-most column the corresponding numbers are, of course, 0 and a_n.) Thus we can write the overall process in the form of a simple loop:

```
       1.   Input {a0, a1, ..., an, x0}
       2.   p := an
(10)   3.   For k = n-1, n-2, ..., 0 do
            3.1  p := px0 + ak
       4.   Output {x0, p}.
```

Each execution of step 3 involves one multiplication and one addition, that is, altogether there are n multiplications and additions each requiring $n(\mu+\alpha)$ seconds.

This represents a considerable saving over the $(2n-1)\mu+n\alpha$ seconds needed for (4).

Exercises

4. Draw a flow chart for the algorithm (10).

5. If a programmable calculator or computer is available, implement (10) for cubic and quartic polynomials. Test your program with the polynomials (3) and (8) and the values of x used in Exercises 1 and 2 of Section 2.

6. Do a hand calculation or run your program on another polynomial such as $p(x) = x^3 + x^2 + x + 1$ at $x_0 = 1$, $x_0 = -1$, $x_0 = 10$.

---

4. CONVERSION TO DECIMAL REPRESENTATION

As an application of the Horner scheme, consider the question of finding the decimal representation of some integer $N = (a_n a_{n-1} \cdots a_0)_b$ given in base b notation. For example, let b = 2 and N the binary number

$$N = 110011. \tag{11}$$

Generally, the notation $N = (a_n a_{n-1} \cdots a_0)_b$ means that

$$N = a_n b^n + a_{n-1} b^{n-1} + \ldots + a_1 b + a_0 .$$

In other words, if we introduce the polynomial (1), then we have $N = p(b)$. Thus, in the case of (11), we need to evaluate the polynomial

$$p(x) = x^5 + x^4 + x + 1$$

at $x_0 = 2$. The Horner scheme for this has the form

$\underline{x_0 = 2}$

1	1	0	0	1	1
0	2	6	12	24	50
1	3	6	12	25	51

and hence our binary number 110011 has the decimal representation 51.

Exercises

7. Convert the binary integers 10101, 1111, 10000 to decimal representations.

8. Convert the integers $(74013)_8$, $(112101)_3$, $(4401)_5$ to decimal representations.

---

5. HORNER'S SCHEME AND POLYNOMIAL DIVISION

We return to our algorithm (10), but this time we retain the successive values of the variable p in steps 2 and 3.1, that is, the entries in row 3 of Table (6). We rewrite the algorithm as follows:

(12)

1. Input $\{a_0, \ldots, a_n, x_0\}$
2. $p := a_n$
3. For $k = n-1, n-2, \ldots, 0$ do
   3.1 $q_k := p$
   3.2 $p := px_0 + a_k$
4. Output $\{x_0, p, q_0, \ldots, q_{n-1}\}$

Hence between the coefficients $a_0, \ldots, a_n, q_0, \ldots, q_{n-1}$ and p we have the relations:

(13)

$$
\begin{aligned}
q_{n-1} &= a_n \\
q_{n-2} &= q_{n-1} x_0 + a_{n-1} \\
q_{n-3} &= q_{n-2} x_0 + a_{n-2} \\
&\;\vdots \\
q_1 &= q_2 x_0 + a_2 \\
q_0 &= q_1 x_0 + a_1 \\
p &= q_0 x_0 + a_0 .
\end{aligned}
$$

Evidently in Table (6) q_2, q_1, q_0, p are the numbers in row 3.

We now introduce the new polynomial

(14) $$q(x) = q_{n-1}x^{n-1} + \dots + q_1x + q_0 .$$

It is related to $p(x)$ via a simple formula. In fact, using the formulas (13) we find that

$$\begin{aligned} q(x)(x-x_0)+p &= q_{n-1}x^n + q_{n-2}x^{n-1} + \dots + q_1x^2 + q_0x + p \\ &\quad - q_{n-1}x_0x^{n-1} - \dots - q_2x_0x^2 - q_1x_0x - q_0x_0 \\ &= q_{n-1}x^n + (q_{n-2} - q_{n-1}x_0)x^{n-1} + \dots \\ &\quad + (q_1 - q_2x_0)x^2 + (q_0 - q_1x_0)x + (p - q_0x_0) \\ &= a_nx^n + a_{n-1}x^{n-1} + \dots + a_2x^2 + a_1x + a_0 \end{aligned}$$

and hence because of $p = p(x_0)$ that

(15) $$p(x) = q(x)(x-x_0) + p(x_0).$$

Thus, $q(x)$ is the result of the division of the polynomial $p(x)$ by the linear factor $x-x_0$ and $p(x_0)$ is the remainder. Horner's scheme is indeed only a slightly condensed form of the standard division of a polynomial by a linear factor. To illustrate this we write this process in its familiar form for the cubic polynomial (5) (recall $q_2 = a_3$):

$$\begin{array}{r|l} & q_2x^2 + q_1x + q_0 \\ \hline x - x_0 & a_3x^3 + a_2x^2 + a_1x - a_0 \\ & \underline{q_2x^3 - q_2x_0x^2} \\ & \qquad q_1x^2 + a_1x \\ & \qquad \underline{q_1x^2 - q_1x_0x} \\ & \qquad\qquad q_0x + a_0 \\ & \qquad\qquad \underline{q_0x - q_0x_0} \\ & \qquad\qquad\qquad p \end{array}$$

7

which means that

$$a_3x^3+a_2x^2+a_1x+a_0 = (x-x_0)(q_2x^2+q_1x+q_0)+p$$

in agreement with (15).

More generally, this long division may be applied to divide any polynomial by a nonzero polynomial of lower degree. For instance, the division of (8) by $u(x) = x^2-x+2$ proceeds as follows:

$$
\begin{array}{rl}
 & \underline{\quad x^2-x-3} \\
x^2-x+2 & \big)\, x^4-2x^3+0\cdot x^2+x-1 \\
 & \underline{x^4-\;x^3+\;2x^2} \\
 & \quad -x^3-2x^2+x \\
 & \quad \underline{-x^3+\;x^2-2x} \\
 & \qquad -3x^2+3x-1 \\
 & \qquad \underline{-3x^2+3x-6} \\
 & \qquad\qquad\qquad 5
\end{array} \tag{17}
$$

which means that

$$x^4-2x^3+x-1 = (x^2-x+2)(x^2-x-3) + 5 .$$

For the general polynomial (1) and any

(18) $u(x) = u_mx^m+u_{m-1}x^{m-1}+ \ldots +u_1x+u_0, \; m\le n, \; u_m\neq 0$

this division algorithm may be written in the following form:

(19)

1. Input $\{a_0, \ldots, a_n, u_0, \ldots, u_m\}$
2. For $j = 0, 1, \ldots, n$ do $r_j := a_j$
3. For $k = n-m, n-m-1, \ldots, 0$ do
   - 3.1 $q_k := r_{m+k}/u_m$
   - 3.2 For $j = m+k-1, m+k-2, \ldots, k$ do
     - 3.2.1 $r_j := r_j-q_ku_{j-k}$
4. Output $\{q_0, \ldots, q_{n-m}, r_0, \ldots, r_{m-1}\}$.

For the resulting polynomials

(20) $q(x) = q_{n-m}x^{n-m}+\ldots+q_1x+q_0, \; r(x) = r_{m-1}x^{m-1}+\ldots+r_1x+r_0$

we then have

(21) $p(x) = q(x)u(x) + r(x)$.

In the special case of $u(c) = x-x_0$, that is, $m=1$, $u_1=1$, $u_0=-x_0$, the algorithm reduces essentially to (12). The only difference is that the remainder is now a polynomial which we initialize in step 2 as $r(x) = p(x)$. Formerly we knew that the remainder is a constant which may be initialized as $p = a_n$.

The algorithm (19) is probably easiest to understand by going in detail through the following ~~long~~-hand division process:

$$\begin{array}{ll} & q_2x^2+q_1x+q_0 \\ u_2x^2+u_1x+u_0 & \overline{)\,r_4x^4+r_3x^3+r_2x^2+r_1x+r_0} \end{array}$$

$$\left.\begin{array}{l} r_4=q_2u_2, \\ r_3=r_3-q_2u_1,\ r_2=r_2-q_2u_0 \end{array}\right\} \qquad \begin{array}{r} \underline{q_2u_2x^4+q_2u_1x^3+q_2u_0x^2} \\ r_3x^3+r_2x^2+r_1x \end{array}$$

$$\left.\begin{array}{l} r_3=q_1u_1 \\ r_2=r_2-q_1u_1,\ r_1=r_1-q_1u_0 \end{array}\right\} \qquad \begin{array}{r} \underline{q_1u_2x^3+q_1u_1x^2+q_1u_0x} \\ r_2x^2+r_1x+r_0 \end{array}$$

$$\left.\begin{array}{l} r_2=q_0u_2 \\ r_1=r_1-q_0u_1,\ r_0=r_0-q_0u_0 \end{array}\right\} \qquad \begin{array}{r} \underline{q_0u_2x^2+q_0u_1x+q_0u_0} \\ r_1x+r_0 \quad \text{remainder} \end{array}$$

Exercises

9. Perform the division (16) for the polynomial (2) and $x_0 = 2$. Compare your results with those of (7).

10. As in (17) divide $p(x) = x^6+x^5-x^4+2x^3-x+2$ by $u(x) = 2x^3+2x^2-x+3$. Then follow the same steps in Algorithm (19).

11. If a programmable calculator or computer is available, implement (19) for reasonable values of $n\geq m>0$. Test your program with the polynomials of (17) and Exercise 2 above.

12. (Optional) Show that there is only one pair of polynomials $q(x)$, $r(x)$ with degree $r(x)$ < degree $u(x)$ that satisfies (21).

---

6. HORNER'S SCHEME AND THE DERIVATIVES

We return to the basic formula (15). Since $q(x)$ turns out to be the difference quotient

$$q(x) = \frac{p(x)-p(x_0)}{x-x_0} ;$$

we expect that $q(x_0)$ is the value of the derivative of p at x_0. In fact, by applying the product rule to (15), we obtain

$$p'(x) = q'(x)(x-x_0) + q(x), \tag{22}$$

whence indeed

$$p'(x_0) = q(x_0) . \tag{23}$$

Thus $p'(x_0)$ may be evaluated by applying Horner's scheme to q. For our example (2) and $x_0 = 2$ this looks as follows:

$$\begin{array}{l|llll}
\underline{x_0 = 2} & & & & \\
 & 3 & -4 & 2 & -3 \\
 & 0 & 6 & 4 & 12 \\
\hline
(24) & 3 & 2 & 6 & 9 = p(2) \\
 & \underline{0} & \underline{6} & \underline{16} & \\
 & 3 & 8 & 22 = p'(2) &
\end{array}$$

To implement this, note that each column of (24) can be computed from the column on its left. Thus, we don't have to complete the evaluation of p(2), i.e., fill in all of row 3, before finding the value of p'(2). However, observe also that in the last column only p itself is evaluated. Thus we may extend (10) as follows:

$$
(25)\quad
\begin{array}{ll}
1. & \text{Input } \{a_0, \ldots, a_n, x_0\} \\
2. & p := a_n \\
3. & p' := p \\
4. & \text{For } k = n-1, n-2, \ldots, 1 \text{ do} \\
 & \quad 4.1\ \ p := px_0 + a_k \\
 & \quad 4.2\ \ p' := p'x_0 + p. \\
5. & p := px_0 + a_0 \\
6. & \text{Output } \{x_0, p, p'\}.
\end{array}
$$

The process may be extended to higher derivatives. For this note that the repeated application of the Horner scheme results in a sequence of divisions:

$$
(26a)\quad
\begin{aligned}
p(x) &= q_1(x)(x-x_0) + p(x_0) \\
q_1(x) &= q_2(x)(x-x_0) + q_1(x_0) \\
q_2(x) &= q_3(x)(x-x_0) + q_2(x_0) \\
&\ \ \vdots \\
q_k(x) &= q_{k+1}(x)(x-x_0) + q_k(x_0)
\end{aligned}
$$

where $q_1(x)$ denotes our original $q(x)$. At each step the degree of the q's decreases exactly by one; that is, the degree of $q_1(x)$ is $n-1$; for $q_2(x)$ it is $n-2$; and generally $q_k(x)$ has degree $n-k$. Thus $q_n(x)$ is a constant and we have $q_{n+1}(x) \equiv 0$, and our sequence of equations (26) ends with

$$
(26b)\quad
\begin{aligned}
q_{n-1}(x) &= q_n(x)(x-x_0) + q_{n-1}(x_0) \\
q_n(x) &= q_n(x_0).
\end{aligned}
$$

We multiply the k^{th} equation by $(x-x_0)^k$, $k = 0, 1, \ldots, n$, and add all of them together. Then for $k = 1, \ldots, n$, the term $q_k(x)(x-x_0)^k$ arising on the left of the k^{th} equation cancels against the same term on the right in the $(k-1)^{st}$ equation. Hence we obtain

(27) $p(x) = p(x_0)+q_1(x_0)(x-x_0)+q_2(x_0)(x-x_0)^2+ \ldots + q_n(x_0)(x-x_0)^n$

By differentiating this k times, $0 \le k \le n$, the first $(k-1)^{st}$ terms disappear, the k^{th} term becomes $k!q_k(x_0)$, and all subsequent terms still have a nonzero power of $(x-x_0)$ as a factor. Thus for $x = x_0$ these terms become zero and we find that

(28) $$q_k(x_0) = \frac{1}{k!} p^{(k)}(x_0), \quad k = 0, 1, \ldots, n.$$

Moreover, (27) becomes

(29) $$p(x) = p(x_0)+p'(x_0)(x-x_0) + \frac{1}{2!} p''(x_0)(x-x_0)^2+ \ldots + \frac{1}{n!} p^{(n)}(x_0)(x-x_0)^n.$$

This is the Taylor expansion of $p(x)$ at $x = x_0$.

The sequence of divisions (26a/b) is, of course, computed by means of repeated application of the Horner scheme. For example, in the case of the quartic polynomial (8) we obtain for $x_0 = -1$ the following results:

	$x_0 = -1$						
	$p(x)$	1	-2	0	1	-1	
		0	-1	3	-3	2	
	$q_1(x)$	1	-3	3	-2	1	$= p(-1)$
		0	-1	4	-7		
(30)	$q_2(x)$	1	-4	7	-9	$= p'(-1)$	
		0	-1	5			
	$q_3(x)$	1	-5	12	$= \frac{1}{2!} p''(-1)$		
		0	-1				
	$q_4(x)$	1	-6	$= \frac{1}{3!} p'''(-1)$			
		$= \frac{1}{4!} p^{(4)}(-1)$					

and thus

$$p(x) = 1 - 9(x+1) + 12(x+1)^2 - 6(x+1)^3 + (x+1)^4. \tag{31}$$

Besides providing us with a simple method for the evaluation of the derivatives of $p(x)$ at a given point $x = x_0$, we have obtained here also an algorithm for rewriting $p(x)$ in terms of the powers of $(x-x_0)$ instead of those of x.

In extension of the algorithm (25) the entire process can be written as follows:

(32)

1. Input $\{a_0, \ldots, a_n, x_0\}$.
2. For $k = 0, 1, \ldots, n$ do $p_k := a_n$
3. For $k = n-1, n-2, \ldots, 0$ do
   - 3.1 $p_0 := p_0 x_0 + a_k$
   - 3.2 For $j = 1, \ldots, k$ do
     - 3.2.1 $p_j := p_j x_0 + p_{j-1}$.
4. Output $\{x_0, p_0, \ldots, p_n\}$.

The resulting values are

$$p_k = \frac{1}{k!} p^{(k)}(x_0), \quad k = 0, 1, \ldots, n,$$

and hence are exactly the coefficients of the "shifted" polynomial (31).

As (25) the algorithm (32) computes the data column-wise from left to right. The computational process is easily followed in the next table.

	a_3	a_2	a_1	a_0
p_0	$a_3 \rightarrow$	$a_3x_0+a_2 \rightarrow$	$a_3x_0^2+a_2x_0+a_1 \rightarrow$	$a_3x_0^3+a_2x_0^2+a_1x_0+a_0$
p_1	$a_3 \rightarrow$	$2a_3x_0+a_2 \rightarrow$	$3a_3x_0^2+2a_2x_0+a_1$	
p_2	$a_3 \rightarrow$	$3a_3x_0+a_2$		
p_3	a_3			

Exercises

13. Verify by direct differentiation and evaluation of the resulting polynomials, the results given in (24), (30), and (31).

14. Follow in detail the algorithm (32) on the example (30).

15. If a programmable calculator or computer is available, implement (32) for reasonable values of n. Test your program with the data in the example (24) and (30).

16. Write out in detail the proof of (27) and (28).

17. Compute the coefficients of $p(x-1)$ for

$$p(x) = x^6 - 6x^5 + 15x^4 - 10x^3 - 15x^2 + 4x - 9$$

7. OUTLOOK

The basic method named in the title of this unit was given by W. G. Horner in the early nineteenth century in connection with an efficient method for finding the coefficients of $p(x-x_0)$, [Philosophical Transactions, Royal Society of London 109, 1819, 308-335]. But the factorization

$$p(x) = (\ldots((a_n x_0 + a_{n-1})x_0 + a_{n-1})x_0 + \ldots)$$

on which it is based was already used by Isaac Newton some hundred years earlier [Analysis per Quantitatem Series, London, 1711].

We saw earlier that Horner's method uses fewer operations than, for instance, the approach indicated in (4). It can be shown that when the inputs to our algorithm (10) are arbitrary constants, that is, when we are not using any further information about them, then there is no other algorithm which computes p with less than n multiplications and n additions.

In practice the computations in all our algorithms are performed in floating point arithmetic on some computer. Then round-off errors are introduced and the question arises how they affect the results. For instance, it turns out that with increasing $|x_0^*|$, (absolute value of x_0), the result of the Horner algorithm (10) may be increasingly inaccurate. The situation is more complex when it comes to the other algorithms given here.

8. ANSWERS TO EXERCISES

1. $\underline{x_0 = 1}$

3	-4	2	-3
0	3	-1	1
3	-1	1	-2

$p(1) = -2$

$\underline{x_0 = -1}$

3	-4	2	-3
0	-3	7	-9
3	-7	9	-12

$p(-1) = -12$

$\underline{x_0 = 10}$

3	-4	2	-3
0	30	260	2620
3	26	262	2617

$p(10) = 2617$

2. $\underline{x_0 = 0}$

1	-2	0	1	-1
0	0	0	0	0
1	-2	0	1	-1

$p(0) = -1$

$\underline{x_0 = 1}$

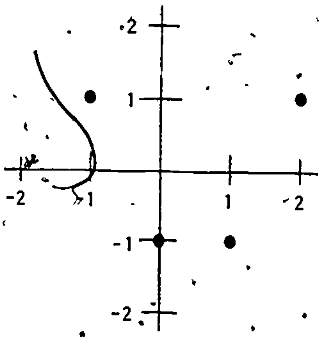
1	-2	0	1	-1
0	1	-1	-1	0
1	-1	-1	0	-1

$p(1) = -1$

$x_0 = 2$

1	-2	0	1	-1
0	2	0	0	2
1	0	0	1	1

$p(2) = 1$



$-1 \leq x \leq 0$ Since $p(-1) > 0$, $p(0) < 0$ $p(x)$ must cross the x-axis, i.e. have a zero, between -1 and 0.

$1 \leq x \leq 2$ Since $p(1) < 0$, $p(2) > 0$ $p(x)$ must cross the x-axis between 1 and 2.

3. $p'(x) = 4x^3 - 6x^2 + 1$

$x_0 = -1$

4	-6	0	1
0	-4	10	-10
4	-10	10	-9

$p(-1) = -9$

$x_0 = 2$

4	-6	0	1
0	8	4	8
4	2	4	9

$p(2) = 9$

5. For quartic polynomials:

```
10 DIM A(5)
20 MAT INPUT A
30 INPUT X
40 LET P = A(5)
50 FOR K=4 TO 1 STEP -1
60 LET P = P*X +A(K)
70 NEXT K
80 PRINT X,P
90 END
```

6. $\underline{x_0 = 1}$

1	1	1	1
0	1	2	3
1	2	3	4

$p(1) = 4$

$\underline{x_0 = -1}$

1	1	1	1
0	-1	0	-1
1	0	1	0

$p(-1) =$ [illegible]

$\underline{x_0 = 10}$

1	1	1	1
0	10	110	1110
1	11	111	1111

$p(10) = 1,111$

7. $\underline{x_0 = 2}$

1	0	1	0	1
0	2	4	10	20
1	2	5	10	21

$(10101)_2 = (21)_{10}$

$\underline{x_0 = 2}$

1	1	1	1
0	2	6	14
1	3	7	15

$(1111)_2 = (15)_{10}$

$\underline{x_0 = 2}$

1	0	0	0	0
0	2	4	8	16
1	2	4	8	16

$(10000)_2 = (16)_{10}$

8. $\underline{x_0 = 8}$

7	4	0	1	3
0	56	480	3840	30728
7	60	480	3841	30731

$(74013)_8 = (30731)_{10}$

$\underline{x_0 = 3}$

1	1	2	1	0	1
0	3	12	42	129	387
1	4	14	43	129	388

$(112101)_3 = (388)_{10}$

$\underline{x_0 = 5}$

4	4	0	1
0	20	120	600
4	24	120	601

$(4401)_5 = (601)_{10}$

9.

$$
\begin{array}{r|l}
 & 3x^2 + 2x + 6 \\
\hline
x - 2 & 3x^3 - 4x^2 + 2x - 3 \\
 & \underline{3x^3 - 6x^2} \\
 & \qquad 2x^2 + 2x \\
 & \qquad \underline{2x^2 - 4x} \\
 & \qquad\qquad 6x - 3 \\
 & \qquad\qquad \underline{6x - 12} \\
 & \qquad\qquad\quad + 9
\end{array}
$$

$$3x^3 - 4x^2 + 2x - 3 = (x - 2)(3x^2 + 2x + 6) + 9$$

10.

$$
\begin{array}{r|l}
 & \frac{1}{2}x^3 \qquad - \frac{1}{4}x + \frac{1}{2} \\
\hline
2x^3 + 2x^2 - x + 3 & x^6 + x^5 - x^4 + 2x^3 \quad - x \quad + 2 \\
 & \underline{x^6 + x^5 - \frac{1}{2}x^4 + \frac{3}{2}x^3} \\
 & \qquad - \frac{1}{2}x^4 + \frac{1}{2}x^3 \qquad - x \\
 & \qquad \underline{- \frac{1}{2}x^4 - \frac{1}{2}x^3 + \frac{1}{4}x^2 - \frac{3}{4}x} \\
 & \qquad\qquad x^3 - \frac{1}{4}x^2 - \frac{1}{4}x + 2 \\
 & \qquad\qquad \underline{x^3 + x^2 - \frac{1}{2}x + \frac{3}{2}} \\
 & \qquad\qquad - \frac{5}{4}x^2 + \frac{1}{4}x + \frac{1}{2}
\end{array}
$$

$$x^6 + x^5 - x^4 + 2x^3 - x + 2 = (2x^3 + 2x^2 - x + 3)(\tfrac{1}{2}x^3 - \tfrac{1}{4}x + \tfrac{1}{2})$$
$$+ (-\tfrac{5}{4}x^2 + \tfrac{1}{4}x + \tfrac{1}{2}).$$

11. For $n = 4$, $m = 2$:

```
10 DIM A(5),U(3),Q(3),R(2)
20 MAT INPUT A
30 REM THE ROUTINE DESTROYS A
40 MAT INPUT U
50 FOR K = 3 TO 1 STEP -1
60 LET Q(K) = A(K+2)/U(3)
70 FOR J = K+! TO K STEP -1
80 LET A(J) = A(J) - Q(K)*U(J-K+1)
90 NEXT J
100 NEXT K
110 LET R(1) = A(1)
120 LET R(2) = A(2)
130 MAT PRINT Q
140 MAT PRINT R
150 END
```

12. Suppose there were more than one pair.

That is suppose $p(x) = q(x)u(x) + r(x)$ where either $r(x) = 0$ or $\deg r(x) < \deg u(x)$

$p(x) = q^*(x)u(x) + r^*(x)$ where either $r^*(x) = 0$ or $\deg r^*(x) < \deg u(x)$.

$\Rightarrow (q(x) - q^*(x))u(x) = r(x) - r^*(x)$

$\Rightarrow u(x)$ divides $r(x) = r^*(x)$ and thus $\deg u(x) \leq \deg[r(x) - r^*(x)]$ or $[r(x) - r^*(x)] = 0$.

But $0 \leq \deg [r(x) - r^*(x)] < \deg u(x)$ so we must have $r(x) - r^*(x) =$ or $r(x) = r^*(x)$.

$\Rightarrow q(x)u(x) = q^*(x)u(x)$

And hence, $q(x) = q^*(x)$.

Therefore there was really only one pair, i.e., the $q(x)$ and $r(x)$ are unique.

15. For n = 4:

```
10  DIM A(5),P(5)
20  MAT INPUT A
30  INPUT X
40  FOR K=1 TO 5
50  LET P(K) = A(5)
60  NEXT K
70  FOR K = 5 TO 1 STEP -1
80  LET P(1) = P(1)*X + A(K)
90  FOR J = 2 TO K
100 LET P(J) = P(J)*X + P(J-1)
110 NEXT J
120 NEXT K
130 PRINT X
140 MAT PRINT P
150 END
```

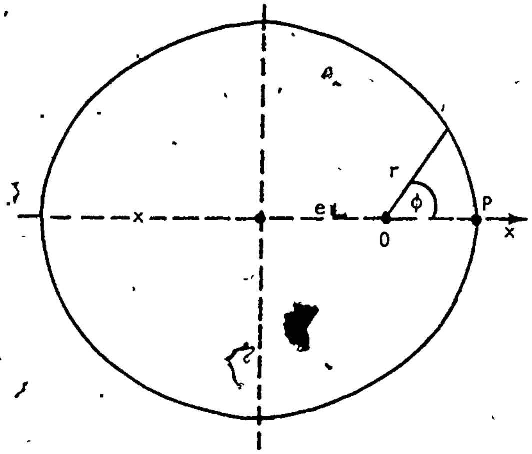
umap

UNIT 264

MODULES AND MONOGRAPHS IN UNDERGRADUATE
MATHEMATICS AND ITS APPLICATIONS PROJECT

ALGORITHMS FOR FINDING ZEROS OF FUNCTIONS

by Werner C. Rheinboldt



COMPUTER SCIENCE

edc/umap / 55chapel st./newton.mass 02160

ALGORITHMS FOR FINDING ZEROS OF FUNCTIONS

by

Werner C. Rheinboldt
Department of Mathematics and Statistics
University of Pittsburgh
Pittsburgh, PA 15260

Table of Contents

Intermodular Description Sheet: *UMAP Unit 264*

Title: ALGORITHMS FOR FINDING ZEROS OF FUNCTIONS

Author: Werner C. Rheinboldt
Department of Mathematics and Statistics
University of Pittsburg
Pittsburg, PN 15260

Review Stage/Date: III 12/15/78

Classification: COMP SCI

Suggested Support Materials:

References: See Section 7 of text.

Prerequisite Skills:
1. Be familiar with the Mean Value Theorem.
2. Be familiar with the Intermediate Value Theorem.
3. Be able to differentiate elementary functions.
4. Be familiar with making estimates using absolute value notation.

Output Skills:
1. Understanding of standard bisection, secant, and Newton methods of root finding and appreciation of their limitations and strong points. Introduction to more recent root finding methods.

Other Related Units:

Horner's Scheme and Related Algorithms *(Unit 263)*

MODULES AND MONOGRAPHS IN UNDERGRADUATE MATHEMATICS AND ITS APPLICATIONS PROJECT (UMAP)

The goal of UMAP is to develop, through a community of users and developers, a system of instructional modules in undergraduate mathematics and its applications which may be used to supplement existing courses and from which complete courses may eventually be built.

The Project is guided by a National Steering Committee of mathematicians, scientists and educators. UMAP is funded by a grant from the National Science Foundation to Education Development Center, Inc., a publicly supported, nonprofit corporation engaged in educational research in the U.S. and abroad.

The Project would like to thank Charles Votaw and Douglas F. Hale for their reviews, and all others who assisted in the production of this unit.

This material was prepared with the support of National Science Foundation Grant No. SED76-19615 A02. Recommendations expressed are those of the author and do not necessarily reflect the views of the NSF, nor of the National Steering Committee.

ALGORITHMS FOR FINDING ZEROS OF FUNCTIONS

by

Werner C. Rheinboldt
Department of Mathematics and Statistics
University of Pittsburgh
Pittsburgh, PN 15260

1. SOME MODEL PROBLEMS

Let f be some real function of a real variable x. We want to find a real solution (zero, root) x^* of the equation

$$(1) \qquad f(x) = 0.$$

Only in a few cases, such as for linear or quadratic functions f, are there any explicit formulas for such a solution. Hence we will have to be satisfied with computing x^* approximately.

Nonlinear equations arise frequently in applications. For later use we give here a few simple examples.

Van der Waal's equation of state for an imperfect gas has the form

$$(2) \qquad (p + \frac{a}{v^2})(v - b) = RT.$$

Here p [atm] is the pressure, v[liters/mole] the molal volume (volume/mass), T[°K] the absolute temperature, R = 0.082054 [liter atm/mole °K] the gas constant, and a [liter^2 atm/mole^2], b [liter/mole] constants dependent on the particular gas. For instance, for carbon dioxide we have a = 3.592, b = 0.04267 and for helium

a = 0.03412, b = 0.02370.

For given values of p, T, a, b we want to compute the corresponding value(s) of v for which Equation (2) holds. This is a problem of the form of Equation (1). More specifically, after multiplying by v^2, the desired values are the solutions of the cubic polynomial in v

$$(3) \qquad pv^3 - (pb + RT)v^2 + av - ab = 0.$$

As another example consider the motion of a particle of mass m which is attracted to a fixed center O by a Newtonian force $\mu m/r^2$ with constant $\mu > 0$. Kepler's first law then states that the particle moves on a conic section with eccentricity e with one focus at O. Thus for $0 < e < 1$ the orbit is an ellipse, for $e = 1$ a parabola, and for $e > 1$ a branch of a hyperbola.

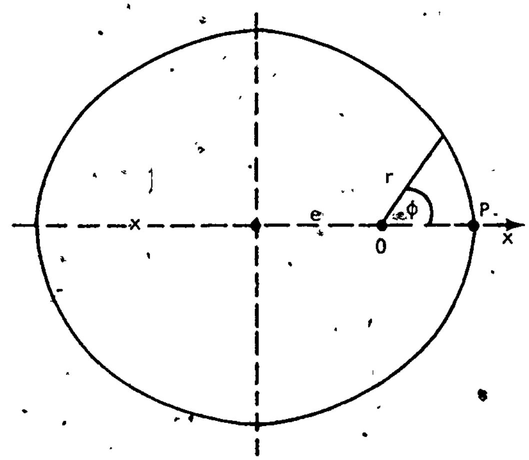


Figure 1.

More specifically, let P be the pericenter, that is the point on the orbit closest to O, and introduce the polar coördinates (r,ϕ) with center at O and the direction of $\overline{OP}$ as the x-axis. Then for $e > 0$, $e \neq 1$ the orbit is given by

(4) $r = \frac{a|1 - e^2|}{1 + e\cos\phi}$

Now let T be the time when the particle is at the pericenter, then its position at time t is determined by the Kepler equations

(5) (a) $n(t - T) = u - e\sin u$, if $0 < e < 1$
(b) $n(t - T) = e\sinh u - u$, if $e > 1$.

The variable u is called the eccentric anomaly; it relates to r by the equations

(6) $r = a(1 - e\cos u)$, if $0 < e < 1$
$r = a(e\cosh u - 1)$, if $e > 1$.

The parameter n is the mean motion, that is, in the case of an ellipse $n = 2\pi/p$ where p is the period. For given a, e, n, T the problem of determining the position of the particle at time t now requires the solution u of the corresponding Equation (5). Then r can be found from Equations (6) and ϕ from Equation (4).

Some values of the parameters for the case of the earth's orbit around the sun are a = 1.000, e = .017, n = .01720, and T = Jan 1, 1900. In this elliptic case Equation (5)(a) is unchanged if we add or subtract a multiple of 2π from $\ell = n(t - T)$ and u. Hence we may always reduce the left side such that $-\pi < \ell \leq \pi$.

Exercises

1. Let the function g be continuous on the closed interval $a \leq x \leq b$ and differentiable on $a < x < b$. Then the Mean Value Theorem ensures the existence of at least one value x^* such that

$$g(b) - g(a) = (b - a)\, g'(x^*), \quad a < x^* < b.$$

Thus to find x^* we need to solve some nonlinear equation of the form of Equation (1). Write down this equation in the following cases:

(a) $g(x) = (x + 1)e^{-x}$, $a = -1$, $b = 0$.
(b) $g(x) = x^{20}$, $a = 0$, $b = 20^{1/19} \doteq 1.1708$.

2. For a continuous function g on the interval $a \leq x \leq b$ there exists at least one value x^* such that

$$\int_a^b g(x)dx = (b - a)\, g(x^*), \; a < x^* < b.$$

This is the integral mean value theorem. Write down the resulting equation for x^* in the cases

(a) $\int_2^3 \frac{dx}{x \log x}$, (b) $\int_0^1 (x + e^x)\, dx$.

2. EXISTENCE QUESTIONS

Before we look at methods for solving a given Equation (1), it is important to realize that there may be no solution at all or there may exist any number of them. The examples in Table 1 illustrate some of the possibilities.

TABLE 1

f(x)	No. of zeros	Zeros
e^x	none	—
$\frac{1}{2}x - 1$	one	$x^* = 2$
$x^2 - 1$	two	$x_1^* = 1,\ x_2^* = -1$
tan x	countably many	$x_k^* = k\pi,\ k=0, \pm 1, \pm 2, \ldots$
$\begin{cases} (x+1)^2 & \text{for } x \leq -1 \\ 0 & \text{for } -1 \leq x \leq 1 \\ (x-1)^2 & \text{for } x \geq 1 \end{cases}$	a continuum	any x^* with $-1 \leq x^* \leq +1$

This indicates that it is advisable to begin an investigation of a particular equation by localizing a suitable interval $a \leq x \leq b$ in which the desired solution exists. A simple approach for the construction of such an interval is to plot the graph of $f(x)$. For instance in the case of the simple Kepler equation

$$(7) \qquad f(x) = x - 1 - \frac{1}{2}\sin x = 0$$

this may provide the results shown in Table 2 and Figure 2.

TABLE 2

x	f(x)
0	-1.0000
0.5	-0.7397
1.0	-0.4207
1.5	$0.1252 \cdot 10^{-2}$
2.0	0.5454
2.5	1.2008
3.0	1.9294

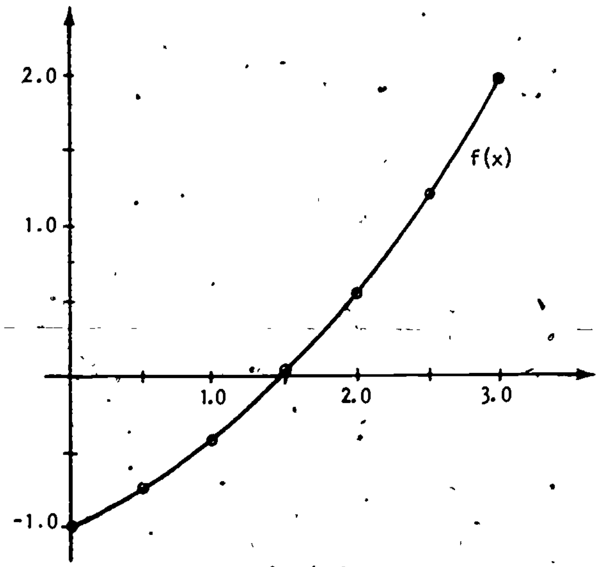


Figure 2.

The drawing indicates that a zero of f should be contained in the interval $1 \leq x \leq 1.5$.

The theoretical foundation for this conclusion is the following basic theorem:

Intermediate Value Theorem: Let f be a continuous function on the interval $a \leq x \leq b$, $a < b$. If

$$(8) \qquad \text{sign } f(a) \neq \text{sign } f(b)$$

then $f(x) = 0$ has a solution in the interval $a \leq x \leq b$.

The proof of this theorem is not entirely simple and uses some fundamental properties of the real number system. But intuitively the result is clear. A continuous function might be characterized as a function with a graph that can be drawn without lifting the pen from the paper. The condition (8) implies that at two endpoints of the interval the function values are on opposite sides of the x-axis; hence when drawing the graph of f we need to cross the x-axis somewhere in that interval.

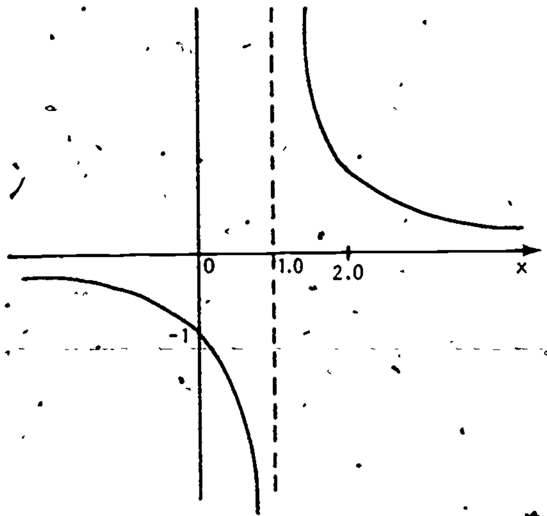


Figure 3.

This observation already indicates that without the continuity assumption the result is not necessarily valid. Indeed, for instance, the function $f(x) = 1/(x - 1)$ has the values $f(0) = -1$ and $f(2) = 1$ yet there is no zero in the interval $0 \le x \le 2$. (See Figure 3.)

We note also that the theorem does not say anything about the number of solutions in the interval $a \le x \le b$. In fact, there may be any number of distinct roots. For instance, in the interval $-2 \le x \le 2$ the equation

(9) $$f(x) = x^3 - \frac{1}{2}x^2 + a = 0$$

has for $a = -1/2$ a single root $x^* = 1$, for $a = 0$ the two roots $x^* = 0$ and $x^* = 1/2$, and finally for $0 < a < 1/54$ exactly three roots.

The case $a = 0$ is exceptional. Generally, a root x^* of the Equation (1) is called a root of multiplicity m if for x near x^* the function f can be written in the form

(10) $$f(x) = (x - x^*)^m g(x)$$

with some function g that is continuous near x^* and satisfies $g(x^*) \neq 0$. In our case, Equation (9) has for $a = 0$ the form $f(x) = x^2(x - \frac{1}{2})$ which shows that $x^* = 0$ is a root of multiplicity two while $x^* = \frac{1}{2}$ has multiplicity one. Thus counting multiplicities we really have three roots. Generally, the following result holds:

> Theorem: Under the conditions of the intermediate value theorem, the interval $a \le x \le b$ contains either infinitely many solutions or finitely many solutions for which the sum of their multiplicities is an odd number.

Exercises:

1. Determine the solutions of the function

$$f(x) = \begin{cases} 0 & \text{for } x = 0 \\ x \sin \frac{1}{x} & \text{for } 0 < x \le 1 \end{cases}$$

in the interval $0 \le x \le 1$.

2. For any $0 < e < 1$ and $-\pi < \ell < \pi$ determine an interval which contains a root of the Kepler equation $\ell = u - e \sin u$.

3. Show that for the so called critical values

$$p_c = \frac{a}{27b^2}, \quad T_c = \frac{8a}{27Rb}$$

the van der Waal Equation (3) has only one root $v_c = 3b$ of multiplicity three. Show that the constants a, b, R can be expressed in terms of the critical values as

$$a = 3p_c v_c^2, \quad b = \frac{1}{3}v_c, \quad R = \frac{8}{3}\frac{p_c v_c}{T_c}.$$

Use this to show that with the dimension-less variables $\hat{p} = p/p_c$, $\hat{v} = v/v_c$, $\hat{T} = T/T_c$ the equation assumes the form

$$(\hat{p} + \frac{3}{\hat{v}^2})(3\hat{v} - 1) = 8\hat{T}.$$

3. THE BISECTION METHOD

Suppose that an interval $a \le x \le b$ has been found where the conditions of the intermediate value theorem are satisfied. Then we know that there is at least one solution x^* of Equation (1) between a and b. For the midpoint $m = a + (b - a)/2$ we test now the condition sign $f(m) \neq$ sign $f(a)$. If it holds then the intermediate value theorem guarantees that there is a root in the interval $a \le x \le m$, otherwise, we have sign $f(m)$ = sign $f(a) \neq$ sign $f(b)$ and hence there must be a root in $m \leq x \leq b$. In either case, the length of the interval has been halved. By repeating the process we can decrease the interval-length below a prescribed tolerance and hence approximate a root of f arbitrarily closely.

In an informal programming language this algorithm may be written in the following form.

1. Input {a, b, kmax, tol};
2. k := 0;
3. If (sign f(a) = sign f(b)) then error return 1: "Wrong interval";
4. Print {k, a, b};
5. If |b-a| ≤ tol then normal return;
6. k := k+1;
7. m := a + (b-a)/2;
8. If (sign f(a) ≠ sign f(m)) then b=m else a=m;
9. If k < kmax then go to 4 else error return 2 "kmax exceeded";

Step 2 has been included to verify that at all times the basic condition (8) is satisfied. If it holds for the input interval then theoretically it will remain valid for all subsequent intervals. But in practice this may well not be the case due to round-off errors. All iterative methods should include a count k of the number of steps taken and use it to stop the process when a given maximum count kmax has been exceeded. This is done here in step 9.

Table 3 shows the results of the first five steps when the algorithm is applied to the Kepler Equation (7) on the interval $1 \leq x \leq 2$.

Table 3

k	a	b	b-a	f(a)	f(b)
0	1	2	1	-0.4207	0.5454
1	1	1.5	0.5	-0.4207	$0.1252\cdot10^{-2}$
2	1.25	1.5	0.25	-0.2245	$0.1252\cdot10^{-2}$
3	1.375	1.5	0.125	-0.1154	$0.1252\cdot10^{-2}$
4	1.4375	1.5	0.0625	-0.05806	$0.1252\cdot10^{-2}$
5	1.46875	1.5	0.08125	-0.02865	$0.1252\cdot10^{-2}$

We shall see later that, with eight digit accuracy, the exact root is $x^* = 1.4987011$. Obviously, our algorithm is not particularly fast.

The interval decreases at each step by a factor of two. Hence the kth interval has the length $(b - a)/2^k$. If the tolerance is, say, 10^{-t} then we require that

$$(b-a)2^{-k} \leq 10^{-t}$$

or

$$2^k \geq (b-a)\,10^t,$$

that is

(11) $$k \geq \log_2[(b - a)10^t].$$

In the example of Table 3 this means that we need $k \doteq 23$ to obtain seven digit accuracy.

Exercises

1. Draw a flow chart for the bisection algorithm.
2. If a programmable calculator or computer is available, implement the algorithm for the general Kepler Equations (5) with given $-\pi < \ell = n(t - T) \leq \pi$ and $e > 0$.
3. For the cubic polynomial $f(x) = x^3 - 2x + 2$ determine an interval containing (only) root and apply the bisection algorithm to approximate the root to four digit accuracy.

4. SOME LINEARIZATION METHODS

In order to overcome the relatively slow convergence of the bisection method, we turn now to a different principle for computing solutions of Equation (1). It is based on the idea of replacing the function $f(x)$ by a succession of linear functions $g_k(x) = a_k x + b_k$, $k = 1,2,\ldots$, such that their zeros $-b_k/a_k$, $k = 1,2,\ldots$, approximate the desired solution x^* of Equation (1).

A linear function is determined by its values at two distinct points. Suppose that we are at the kth step of our process and that the approximations $x_0, x_1, \ldots, x_k$ of x^* have been computed already. If $k \geq 1$ and $x_k \neq x_{k-1}$ then we may construct the linear function $g_k(x)$ which agrees with $f(x)$ at x_k and x_{k-1}, namely

(12) $$g_k(x) = f(x_k) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} (x - x_k).$$

For $f(x_k) \neq f(x_{k-1})$ the zero of this function, that is

(13) $$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k),$$

is taken as the next approximation of x^* and in this way the process is continued. This is called the secant method. Clearly the process will fail when two equal function values are encountered. But if this does not happen then we shall see that the method is considerably faster than the bisection method.

A linear function is also defined by its function value and its slope at a given point. Hence, suppose that at x_k, $k \geq 0$, we are able to compute not only $f(x_k)$ but also the value of the derivative $f'(x_k)$. Then we can replace the secant line (12) by the tangent line

(14) $$g_k(x) = f(x_k) + f'(x_k)(x - x_k).$$

If $f'(x_k) \neq 0$ then the zero of Equation (14) is

(15) $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

which becomes our next approximation of x^*. The resulting process is called Newton's method. It will fail whenever a zero derivative value is encountered, but otherwise it turns out to be even faster than the secant method.

As an example, we consider the computation of the positive square root of some positive number $a > 0$. In other words, we wish to find the positive root of $f(x) = x^2 - a$. In that case, we have

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = \frac{x_k^2 - x_{k-1}^2}{x_k - x_{k-1}} = x_k + x_{k-1}$$

and hence the secant method assumes the form

(16) $$x_{k+1} = x_k - \frac{1}{x_k + x_{k-1}}(x_k^2 - a) = \frac{x_k x_{k-1} + a}{x_k + x_{k-1}}.$$

On the other hand, Newton's method becomes

(17) $$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

For $a = 10$ we start (16) with $x_0 = 11$, $x_1 = 10$ and (17) with $x_1 = 10$. The resulting first few steps are given in Table 4.

Table 4

k	Secant Method		Newton's Method	
	x_k	$f(x_k)$	x_k	$f(x_k)$
0	11	111		
1	10	90	10	90
2	5.71429	-22.6531	5.5	20.25
3	4.27273	8.25620	3.65909	3.38895
4	3.44603	1.87515	3.19601	0.214448
5	3.20310	0.259849	3.16246	$.112557\cdot 10^{-2}$
6	3.16402	$0.110211\cdot 10^{-1}$	3.16228	$0.2\cdot 10^{-7}$
7	3.16229	$0.7064\cdot 10^{-4}$		
8	3.16228	$0.2\cdot 10^{-7}$		

Exercises.

1. Apply the secant method to the Kepler Equation (7) starting with $x_0 = 2$, $x_1 = 1.5$.
2. Use Newton's method starting from $x_0 = -1$ to solve the equation specified by problem 1(a) of Section 1.
3. If a programmable calculator or computer is available implement Newton's method for the computation of the square root of any positive number a. Use $x_0 = a$ as starting point.
4. For polynomial equations the value of the function and its derivative at any given point may be computed simultaneously by means of Horner's Scheme.* Draw a flow chart of the resulting process. If a programmable calculator or computer is available implement the method for cubic and quartic polynomials and test it on several equations, such as the polynomial (3).

* For explanation of Horner's Scheme, see UMAP #263.

5. RATES OF CONVERGENCE

The bisection method generates a sequence of intervals $a_k \le x \le b_k$, $k = 0, 1, \ldots$ which contain the desired root x^*. Any point in the kth interval may be considered as the kth approximation of x^*; for the moment let us consider the midpoint $m_k = a_k + (b_k - a_k)/2$ for that purpose. Since at each step the interval is halved we then have the obvious relation

$$(18) \qquad |m_{k+1} - x^*| \le \tfrac{1}{2}|m_k - x^*|, \quad k = 0, 1, \ldots .$$

In other words, the errors converge to zero at least as fast as the geometric sequence $|m_0 - x^*|/2^k$, $k = 0, 1, \ldots .$

Now suppose that Newton's method is used and produces a sequence of points $x_0, x_1, x_2, \ldots$ which converge to the solution x^* of Equation (1). Moreover, assume that the x_k are all contained in some interval $a \le x \le b$ where

$$(19) \qquad \text{(i)} \quad |f'(x)| \ge \alpha > 0, \qquad \text{(ii)} \quad |f''(x)| \le \beta, \quad \text{for } a \le x \le b.$$

Then it follows by Taylor's formula that -- with certain ξ_k in our interval --

$$0 = f(x^*) = f(x_k) + f'(x_k)(x^* - x_k) + \tfrac{1}{2}f''(\xi_k)(x^* - x_k)^2$$
$$= [f(x_k) + f'(x_k)(x_{k+1} - x_k)]$$
$$+ f'(x_k)(x^* - x_{k+1}) + \tfrac{1}{2}f''(\xi_k)(x^* - x_k)^2 .$$

By the definition (15) of Newton's method the term in the square bracket is zero whence

$$0 = f'(x_k)(x^* - x_{k+1}) + \tfrac{1}{2}f''(\xi_k)(x^* - x_k)^2$$

or, because of condition (19),

$$(20) \qquad |x^* - x_{k+1}| = \frac{1}{2}\cdot\frac{|f''(\xi_k)|}{|f'(x_k)|}\,|x^* - x_k|^2 \le \frac{1}{2}\frac{\beta}{\alpha}\,|x^* - x_k|^2 .$$

In other words, in this case, the error is squared at each step (up to some factor). Thus if, say, the k^{th} error is proportional to 10^{-t} then the $(k + 1)$st error is proportional to 10^{-2t}. This is clearly seen in the third column of Table 4 where for $k \geq 3$ the number of correct digits doubles with each step.

For the secant method a similar estimate can be derived.* It turns out that if again Condition (19) holds and all x_k remain in $a \leq x \leq b$ then we have -- with some constant $\gamma > 0$ --

$$(21) \quad |x^* - x_{k+1}| \leq \gamma \cdot |x^* - x_k|^t, \quad k = 0, 1, \ldots, \quad t = \tfrac{1}{2}(1 + \sqrt{5}) \doteq 1.6180.$$

Thus the errors tend to zero somewhat more slowly than in the case of Newton's method but certainly faster than with the bisection method.

These results are somewhat deceptive. First of all, the condition (19)(i) is fairly strong since it implies that there is only one root in the interval $a \leq x \leq b$ and this root must have multiplicity one. In fact, by the mean-value theorem we have

$$|f(x)| = |f(x) - f(x^*)| = |f'(\xi)(x-x^*)| \geq \alpha |x - x^*|, \quad a \leq x \leq b.$$

Hence $f(x) = 0$ for any $a \leq x \leq b$ implies that $x = x^*$. Moreover, if f can be written in the form of Equation (10) with $g(x^*) \neq 0$ then the left side of

$$|x - x^*|^{m-1} \cdot |g(x)| \geq \alpha > 0$$

tends to zero as x goes to x^* unless $m = 1$.

For zeros of multiplicity greater than one, Newton's method indeed converges much more slowly. For example, in the simple case

$$(22) \qquad f(x) = (x - 1)^m = 0, \quad x^* = 1$$

Newton's method has the form

$$x_{k+1} = x_k - \frac{(x_k - 1)^m}{m(x_k - 1)^{m-1}} = \frac{m - 1}{m}(x_k + 1)$$

whence

$$x_{k+1} - 1 = \frac{m - 1}{m}(x_k - 1).$$

In other words, for $m = 2$ the convergence is here as slow as that of the bisection method, and for $m > 2$ it is even slower. The secant method shows a similar behavior; in fact, not only the rate of convergence deteriorates but arbitrarily close to the root we may encounter $f(x_k) = f(x_{k-1})$ in which case the method fails completely.

There are further problems with the function (22). In fact, we see that $|f(x)| \leq \varepsilon$ implies that

$$1 - \varepsilon^{1/m} \leq x \leq 1 + \varepsilon^{1/m}.$$

Hence, say, for $m = 10$ we have $|f(x)| \leq 10^{-6}$ for $.75 \leq x \leq 1.25$. In other words, with six digit accuracy any point in this interval can be called a zero of f. Unless higher accuracy is used any iterative process entering this uncertainty interval will, by necessity, show erratic behavior. A root of this kind is called ill-conditioned. It turns out that also roots of multiplicity one may be ill-conditioned.

Even if the conditions (19) hold the estimate (20) for Newton's method may be very misleading. Consider for example the equation

$$(23) \qquad x^{19} - 1 = 0$$

we encountered in Exercise 1(b) of Section 1. Here clearly $f'(x) > 0$ for $x > 0$ and for any interval $0 < a \leq x \leq b$ containing $x^* = 1$ the estimate (20) holds. But the factor $\beta/2\alpha$ will be very large unless a and b are very close to one. This reflects difficulties with Newton's method, and in fact for $x_0 = 1/2$ we have

$x_1 = 13{,}797.53$ which is certainly a much worse approximation of x^* than x_0. The subsequent iterates decrease monotonically, but very slowly, to one. Only very close to one the expected rapid convergence sets in.

Exercises

1. Show that when Newton's method is used for solving $x^2 - a = 0$, $a > 0$, starting from any $x_0 > 0$, $x_0 \neq \sqrt{a}$, the iterates satisfy

$$x_1 > x_2 > \ldots > x_k > x_{k+1} > \sqrt{a}, \quad k \geq 1$$

and

$$x_{k+1} - \sqrt{a} = \frac{1}{2x_k}(x_k - \sqrt{a})^2.$$

2. Apply the secant method to the equation $f(x) = x^2 - a = 0$, $a > 0$, starting from $x_0 > x_1 > \sqrt{a}$. Show that

$$x_{k+1} - \sqrt{a} = \frac{(x_k - \sqrt{a})(x_{k-1} - \sqrt{a})}{x_k + x_{k-1}}$$

and

$$x_0 > x_{k-1} > x_k > x_{k+1} > \sqrt{a}, \quad k \geq 1.$$

3. Apply Newton's method to Equation (23) starting from $x_0 = 1/2$. Show that

$$x_{k+1} = \frac{18}{19}x_k + \frac{1}{19x_k^{18}} \doteq \frac{18}{19}x_k.$$

How many iteration steps are needed to reach $|x_k - 1| \leq 1.1$?

---

6. A PRACTICAL ALGORITHM

The results of the previous sections show that none of the methods discussed here is entirely satisfactory. The bisection method is fairly reliable but slow, the Newton and secant method are both much more rapid in certain cases but show unreliable behavior in many others.

We discuss now an algorithm which combines the bisection and secant methods to bring out their best features. Again we work with a sequence of intervals of decreasing length for which the intermediate value theorem holds. If, say, $a \leq x \leq b$ is the kth interval, then we set

$$x_k = a,\ y_k = b \text{ if } |f(a)| \leq |f(b)|$$

$$x_k = b,\ y_k = a \text{ if } |f(a)| > |f(b)|.$$

Thus x_k may be considered the current best approximation of the root in the kth interval.

A step of the algorithm now consists in determining a new point between x_k and y_k, called w for the moment, which will become either x_{k+1} or y_{k+1}. For this we introduce the point

$$z_k = \begin{cases} x_{k-1} & \text{if } k \geq 1 \text{ and } y_k = y_{k-1} \\ y_k & \text{otherwise} \end{cases}$$

and consider first the secant step

(24) $$s = x_k - \frac{(x_k - z_k)f(x_k)}{f(x_k) - f(z_k)}$$

provided it gives a better result than the bisection step

$$m = x_k + (y_k - x_k)/2.$$

In other words, since x_k is the current best approximation, s has to be between x_k and m. At the same time, since x_k

is not yet within a given tolerance of the root, s should differ from x_k at least by that tolerance.

Before we discuss the choice of s or m, a few words about the definition of z_k may be useful. The normally expected choice would be $z_k = y_k$. Here s represents a secant step based on the two current endpoints of the interval where the function values have different signs. This is called a regula-falsi step. Unless round-off interferes, such steps do not lead out of the interval and, since they have no subtractive-cancellation problem in the denominator of (24), they are generally rather stable. But in situations, such as that shown in Figure 4, regular falsi steps may give very poor improvements of the interval. For this reason, we use in the case of $y_k = y_{k-1}$, $k \geq 1$, a secant step based on x_{k-1} and x_k. Such a step may lead entirely out of the interval and hence has to be carefully controlled, but it certainly guarantees that there will be no long sequence of small steps of the type shown in the figure.

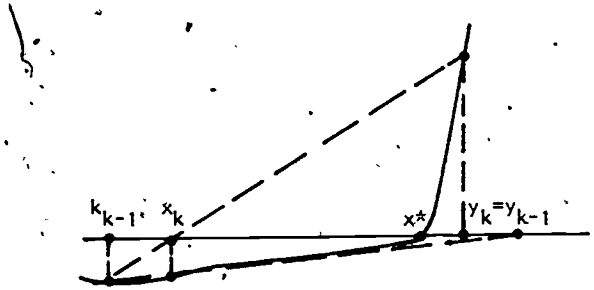


Figure 4.

In order to test convergence we use the tolerance function

$$tol(x) = \varepsilon|x| + \delta$$

where $\varepsilon \geq 0$, $\delta \geq 0$ are given constants with $\varepsilon + \delta > 0$. For $\varepsilon = 0$ the condition $|x - x^*| < tol(x)$ requires the absolute error $|x - x^*|$ to be below δ while for $\delta = 0$ it forces the relative error $|x - x^*|/|x|$ to be less than ε.

With this we set now w = s if s is between $t = x_k + \text{sign}(y_k - x_k)\, tol(x_k)$ and m, and w = t if s is between x_k and t. In all other cases, w = m is chosen, that is, we take a bisection step.

Thus in either case we have settled on a value of w. If sign $f(w) \neq$ sign $f(y_k)$ then the interval between w and y_k is our new interval, otherwise the interval between x_k and w is chosen. This completes one step of the process.

We terminate the algorithm if the length of the interval between x_k and m is less than $tol(x_k)$, that is, if $|y_k - x_k| \leq 2\,tol(x_k)$. This fits with our choice of the minimum step w = t when s is between x_k and t. In fact, if we have not yet converged then

$$|m - x_k| > tol(x_k) = |t - x_k|$$

and thus also in this case w is always between x_k and m.

For the implementation of the process we have to take care that the division in the secant step (24) does not produce overflow or underflow. For this we compute the numerator and denominator

$$p = (x_k - y_k)f(x_k), \quad q = f(y_k) - f(x_k)$$

separately and then test

$$\tfrac{1}{2}|y_k - x_k||q| \geq |p| \geq |q|\,tol(x_k)$$

to determine whether s will be between t and m.

The overall algorithm can now be formulated as follows.

1. __Input__ {x,y,ε,δ,kmax};
2. z:=y;
3. k:=0;
4. __If__ (|f(x)|>|f(y)|) then z:=x; x:=y; y:=z;
5. __Print__ {k,x,y};
6. tol:=ε|x| + δ;

7. If $(|x-y| \le 2\,\text{tol})$ then normal return;
8. If $(k > k\,\text{max})$ then error return "k max exceeded";
9. $k:=k+1$
10. $p:=(x-z)f(x)$;
11. $q:=f(z) - f(x)$;
12. If $(p < 0)$ then $p:=-p$; $q:=-q$;
13. $z:=x$;
14. If $(p \le |q|\text{tol})$ then $x:=x + \text{sign}(y-x)\text{tol}$
    else if $(p < \frac{1}{2}(y-x)q)$ then $x:=x + p/q$
    else $x:=x + (y-x)/2$;
15. If (sign $f(x)$ = sign $f(y)$) then $y:=z$;
16. Go to 4;

As an example we give in Table 5 the solution of Equation (23) with $\varepsilon = 0$, $\delta = 0.5 \times 10^{-6}$, and the starting interval $x = 0.5$, $y = 2.0$. The straight bisection method uses in this case about 21 steps.

Table 5

Interval		F(x)
0.500000	2.00000	-0.999998
0.500000	1.250000	-0.999998
0.510811	1.250000	-0.999997
0.880406	1.250000	-0.911084
0.880406	1.06520	-0.911084
0.932504	1.06520	-0.734927
0.998854	1.06520	$-0.215500 \cdot 10^{-1}$
0.998854	1.00086	$-0.215500 \cdot 10^{-1}$
0.999991	1.00086	$-0.168309 \cdot 10^{-3}$
1.00000	1.00086	$-0.127405 \cdot 10^{-5}$
1.00000	1.00000	$-0.127405 \cdot 10^{-5}$

The process does not always perform that well. In fact, for instance, for Equation (22) with $m = 19$ and $x = 0$, $y = 10$, $\varepsilon = 0$, $\delta = 0.5 \cdot 10^{-6}$ it requires several thousand steps. (Even for $\delta = 0.5 \cdot 10^{-3}$ a total of 121 steps are taken). The reason is that for a long time the algorithm uses only the minimal steps of length tol (x_k).

Various remedies have been proposed for this problem. The easiest approach is to force periodically some bisection steps. For instance, we may simply add a test between steps 12. and 13. which forces a bisection step if k is a multiple of some fixed period M and then bypasses step 13. More sophisticated is a test every M steps which determines whether the interval at the beginning of the period has been reduced at least by the factor 2^M corresponding to M bisection steps. We leave the details as an exercise.

Exercise

1. If a computer is available implement the process described in this section. Test it with the example of Table 5, then use it to solve the equations of Section 1.
2. Introduce in your program a forced bisection step when the iteration index k is a multiple of some integer $M > 1$. Apply the resulting process to Equation (22) and experiment with different values of M.
3. Instead of the approach of Exercise 2 introduce a periodic check comparing the actual reduction of the interval with the expected reduction by means of the bisection method. Compare the performance with that of the process developed in Exercise 2 above.

---

7. REFERENCES

The literature on the numerical solution of nonlinear equations is very extensive. Most standard texts on numerical analysis contain material on the topic and give further references. Some monographs solely devoted to iterative methods for nonlinear equations in one as well as several variables are [1]-[4].

The method described in Section 6 was originally developed by T. J. Dekker [5]. Various improvements and modification, including those mentioned at the end of the section are discussed, for example, in [6]-[8].

[1] A. S. Householder, The numerical treatment of a single nonlinear equation, McGrawhill, New York 1970.

[2] J. Ortega, W. Rheinboldt, Iterative methods for nonlinear equations in several variables, Academic Press, Inc. 1970.

[3] A. Ostrowski, Solution of Equations in Euclidean and Banach Spaces, Academic Press, Inc. 1973.

[4] J. Traub, Iterative methods for the solution of equations, Prentice Hall, Inc. 1964.

[5] T. J. Dekker, Finding a zero by means of successive linear interpolation, in "Constructive Aspects of the Fundamental Theorem of Algebra" ed by B. Dejon, P. Henrici, Wiley-Interscience, London 1969.

[6] R. P. Brent, An algorithm with guaranteed convergence for finding a zero of a function, Computer Journal 14, 4 1971, 422-425.

[7] J. C. P. Bus, T. J. Dekker, Two efficient algorithms with guaranteed convergence for finding a zero of a function, ACM Trans. on Math. Software 1, 4, 1975, 330-345.

[8] G. H. Gonnet, On the structure of zero finders, BIT 17, 1977, 170-183.

8. ANSWERS TO EXERCISES

Section 1

1. (a) $1 + xe^{-x} = 0$.
   (b) $x^{19} = 1$.

2. (a) $\frac{1}{x \log x} = \log \frac{\log 3}{\log 2} \doteq 10.63289$.
   (b) $x + e^{x} = e - \frac{1}{2} \doteq 2.21828$.

Section 2

1. $x^* = \frac{1}{k\pi}$, $k = 1, 2, \ldots$, and $x^* = 0$.

2. For $f(u) = \ell - u + e \sin u$ we have $f(0) = \ell$ and $f(-\pi) = \ell + \pi > 0$, $f(\pi) = \ell - \pi < 0$. Thus for $0 < \ell < \pi$ we may use the interval $0 \le u \le \pi$ and for $-\pi < \ell < 0$ the interval $-\pi \le u \le 0$.

3. For the critical calues p_c and T_c the van der Waal polynomial (3) reduces to
$$0 = v^3 - 9bv^2 + 27b^2v - 27b^3 = (v-3b)^3$$
and thus has only one triple root $v_c = 3b$. From $v_c = 3b$ it follows that $b = v_c/3$ and hence from the formula for p_c we obtain $a = 27p_c(v_c/3)^2 = 3p_cv_c^2$. Now the expression for T_c gives $R = 24p_cv_c^2/(9T_cv_c) = (8/3)p_cv_c/T_c$. By substituting these quantities into (2) we find that
$$\left(p + \frac{3p_cv_c^2}{v^2}\right)\left(v - \frac{v_c}{3}\right) = \frac{8}{3} p_c v_c \frac{T}{T_c}$$
which leads to the stated dimensionless equation after multiplication by $3/(p_cv_c)$.

Section 3

3. $f(-2) = -2$, $f(-1) = 3$, $x^* \doteq -1.7693$.

Section 4

1. The iterates are in sequence

   2.0, 1.5, 1.4988490, 1.4987012, 1.4987011

   and the last number is correct to eight digits.

2. The iterates are in sequence

   -1., -.68393972, -.57745448, -.56722974, -.56714330

   and the last number is correct to eight digits.

3. A simple informal program for this might look as follows:

```
 1. Input {x, n, a_0, a_1, ..., a_n, kmax, big, tol};
 2. k:= 0;
 3. Print {k,x};
 4. k:= k+1;
 5. p:= a_n;
 6. pprime:= p;
 7. For i = n-1, n-2, ..., 1 do 7.1 p:= p × x + a_k;
                                 7.2 pprime:= pprime × x + p;
 8. p:= p × x + a_0;
 9. If (|p| > |pprime| × big) then error return 1:
                                  "Excessive step";
10. If |p| < |pprime| × tol) then normal return;
11. x:= x - p/prime;
12. If k < kmax then go to 3
            else error return 2: "kmax exceeded";
```

Section 5

1. By (17) we have

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right)$$

and thus

$$x_{k+1} - \sqrt{a} = \frac{1}{2x_k}(x_k^2 - 2x_k\sqrt{a} + a) = \frac{1}{2x_k}(x_k - \sqrt{a})^2$$

as well as

$$x_k - x_{k+1} = \frac{1}{2}\left(x_k - \frac{a}{x_k}\right) = \frac{1}{2x_k}(x_k^2 - a).$$

Hence for all $k \geq 0$ we have the implications

$$x_k > 0,\ x_k \neq \sqrt{a} \Rightarrow x_{k+1} > \sqrt{a}$$

$$x_k > \sqrt{a} \Rightarrow x_k > x_{k+1}$$

from which the stated inequalities follow directly by induction.

2. By (16) we have

$$x_{k+1} = \frac{x_k x_{k-1} + a}{x_k + x_{k-1}}$$

and thus

$$x_{k+1} - \sqrt{a} = \frac{1}{x_k + x_{k-1}}[x_k x_{k-1} - \sqrt{a}(x_k + x_{k-1}) + a]$$

$$= \frac{1}{x_k + x_{k-1}}(x_k - \sqrt{a})(x_{k-1} - \sqrt{a})$$

as well as

$$x_k - x_{k+1} = \frac{1}{x_k + x_{k-1}}(x_k^2 - a).$$

Hence for $k \geq 1$ we have here the implication

$$x_{k-1} > \sqrt{a},\ x_k > \sqrt{a} \Rightarrow x_k > x_{k+1} > \sqrt{a}$$

and the stated result follows directly by induction.

3. Newton's method here has the form

$$x_{k+1} = x_k - \frac{x_k^{19} - 1}{19x_k^{18}} = \frac{18x_k^{19} + 1}{19x_k^{18}} = \frac{18}{19}x_k + \frac{1}{19x_k^{18}}.$$

Even for $x_k = 1.1$ the second term on the right is only of the order of 1/100, and hence until then the principal reduction comes principally from the first term. For $x_1 = 13{,}797.53$ we have

$$\left(\frac{18}{19}\right)^k \cdot x_1 \leq 1.1$$

for $k = 175$.

Section 6

3. An informal program incorporating such a periodic check might look as follows:

```
1.  Input {x, y, ε, δ, kmax};
2.  m:=0;
3.  length:=|y-x|;
4.  z:=y;
5.  k:=0;
6.  If (|f(x)| > |f(y)|) then z:=x; x:=y, y:=z;
7.  Print {k, x, y};
8.  tol:=ε|x| + δ;
9.  If (|x-y| ≤ 2·tol) then normal return;
10. If (k > kmax) then error return "kmax exceeded";
11. k:=k+1
12. p:=(z-x)f(x)
13. q:=f(z) - f(x)
14. If (p < o) then p:=-p; q:=-q;
15. z:=x;
16. m:=m+1
17. If (m ≥ 4) then if (16|y-x| ≥ length)
                              then x:=x+½(y-x); go to 20;
                              else m:=0; length: = |y-x|
18. If (p ≤ |q|tol) then x:=x + sign(y-x)tol
                    else if (p < ½(y-x)q)
                             then x:=x + p/q
                             else x:=x + ½(y-x)
19. If (sign f(x) = sign f(y)) then y:=z;
20. Go to 6
```

STUDENT FORM 1

Request for Help

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Student: If you have trouble with a specific part of this unit, please fill out this form and take it to your instructor for assistance. The information you give will help the author to revise the unit.

Your Name______________________ Unit No.______

Page______ ○ Upper ○ Middle ○ Lower	OR	Section______ Paragraph______	OR	Model Exam Problem No.______ Text Problem No.______

Description of Difficulty: (Please be specific)

Instructor: Please indicate your resolution of the difficulty in this box.

○ Corrected errors in materials. List corrections here:

○ Gave student better explanation, example, or procedure than in unit. Give brief outline of your addition here:

○ Assisted student in acquiring general learning and problem-solving skills (not using examples from this unit.)

Instructor's Signature______________________

Please use reverse if necessary.

STUDENT FORM 2

Unit Questionnaire

Return to:
EDC/UMAP
55 Chapel St.
Newton, MA 02160

Name________________ Unit No.__________ Date__________

Institution________________ Course No.________________

Check the choice for each question that comes closest to your personal opinion.

1. How useful was the amount of detail in the unit?

____Not enough detail to understand the unit
____Unit would have been clearer with more detail
____Appropriate amount of detail
____Unit was occasionally too detailed, but this was not distracting
____Too much detail; I was often distracted

2. How helpful were the problem answers?

____Sample solutions were too brief; I could not do the intermediate steps
____Sufficient information was given to solve the problems
____Sample solutions were too detailed; I didn't need them

3. Except for fulfilling the prerequisites, how much did you use other sources (for example, instructor, friends, or other books) in order to understand the unit?

____A Lot ____Somewhat ____A Little ____Not at all

4. How long was this unit in comparison to the amount of time you generally spend on a lesson (lecture and homework assignment) in a typical math or science course?

____Much Longer ____Somewhat Longer ____About the Same ____Somewhat Shorter ____Much Shorter

5. Were any of the following parts of the unit confusing or distracting? (Check as many as apply.)

____Prerequisites
____Statement of skills and concepts (objectives)
____Paragraph headings
____Examples
____Special Assistance Supplement (if present)
____Other, please explain________________

6. Were any of the following parts of the unit particularly helpful? (Check as many as apply.)

____Prerequisites
____Statement of skills and concepts (objectives)
____Examples
____Problems
____Paragraph headings
____Table of Contents
____Special Assistance Supplement (if present)
____Other, please explain________________

Please describe anything in the unit that you did not particularly like.

Please describe anything that you found particularly helpful. (Please use the back of this sheet if you need more space.)

Corrections

UMAP Module 264

p. 20, line-12

$$p = (x_k - z)f(x_k), \; q = f(z) - f(x_k)$$

p. 21, lines 1 to 11

7. <u>If</u> (|x-y| ≤ 2 tol) <u>then</u> normal return;
8. <u>If</u> (k > kmax) <u>then</u> error return "kmax exceeded";
9. k:=k+1
10. p:=(x-z)f(x);
11. q:=f(z) - f(x);
12. <u>If</u> (p < 0) <u>then</u> p:=-p; q:=-q;
13. z:=x;
14. <u>If</u>(p ≤ |q|tol) <u>then</u> x:=x + sign(y-x)tol
    <u>else</u> <u>if</u> (p < $\frac{1}{2}$ (y-x)q) <u>then</u> x:=x + p/q
    <u>else</u> x:=x + (y-x)/2;
15. <u>If</u> (sign f(x) = sign f(y)) <u>then</u> y:=z;
16. <u>Go</u> <u>to</u> 4;

p. 21, line 13, Equation (23) with $\varepsilon = 0$,...

p. 27, line 11-23,

11. k:=k+1
12. p:=(z-x)f(x)

13. q:=f(z) - f(x)

14. <u>If</u> (p < o) <u>then</u> p:=-p; q:=-q;

15. z:=x;

16. m:=m+1

17. <u>If</u> (m > 4) <u>then</u> <u>if</u> (16|y-x| $\geq$ length)

<u>then</u> x:=x + $\frac{1}{2}$(y-x); <u>go</u> <u>to</u> 20;

<u>else</u> m:=0; length: = |y-x|

18. <u>If</u> (p $\leq$ |q|tol) <u>then</u> x:=x + sign(y-x)tol

<u>else</u> <u>if</u> (p < $\frac{1}{2}$(y-x)q)

<u>then</u> x:=x + p/q

<u>else</u> x:=x + $\frac{1}{2}$(y-x)

19. <u>If</u> (sign f(x) = sign f(y)) <u>then</u> y:=z;

20. <u>Go</u> <u>to</u> 6