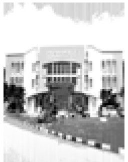
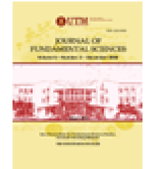
Journal of Fundamental Sciences Vol. 7, No. 1 (2011) 35- 40



ISSN 1823-626X

Journal of Fundamental Sciences

available online at http://jfs.ibnusina.utm.my



A numerical algorithm for n^{th} root

Natarajan Murugesan [1], A.M.S. Ramasamy [2]

[1] *11 Fair Mount Crescent, Toronto, Ontario, M4L 2H1, Canada.*
[2] *Department of Mathematics, Pondicherry University, Pondicherry - 605 014, India.*

Received 25 February 2011, Revised 5 April 2011, Accepted 10 May 2011, Available online 20 May 2011.

ABSTRACT

Presently a direct analytical method is available for the digit-by-digit extraction of the square root of a given positive real number. To calculate the n^{th} root of a given positive real number one may use trial and error method, iterative method, etc. When one desires to determine the n^{th} root, it is found that such methods are inherent with certain weaknesses like the requirement of an initial guess, a large number of arithmetic operations and several iterative steps for convergence, etc. There has been no direct method for the determination of the n^{th} root of a given positive real number. This paper focuses attention on developing a numerical algorithm to determine the digit-by-digit extraction of the n^{th} root of a given positive real number up to any desired accuracy. Examples are provided to illustrate the algorithm.

| n^{th} root | perfect n^{th} power | non-perfect n^{th} power | blocks | concatenation |

http://dx.doi.org/10.11113/mjfas.v7n1.207

1. INTRODUCTION

Let n be any natural number > 1. Certain methods are currently available to find real positive n^{th} root of a given positive real number. Historically speaking, the Babylonian method was developed in 1800 B.C. for the extraction of the square root of a number. One can apply the concept of logarithm introduced by Napier [1(a), (b)] to evaluate the root. However the logarithmic method does not yield accurate results due to truncation errors. It is observed that the conventional long-division square root method is accurate. An algorithm for finding the square root of a number has been described by R.G.Dromey in [2]. For the approximation of quadratic irrationals by rationals and the application of Pell's equation for the extraction of square root, one may refer Niven and Zuckerman [3]. For the determination of cube root, fourth root, etc., of a given number, one may employ an appropriate numerical method, for e.g, Newton's method. Determination of the n^{th} root reduces to solving a non-linear equation of a single variable for which the methods available may be categorized as direct analytical method, graphical method, trial and error method, iterative method, etc. Gower [4] has described an iterative method for the determination of roots. For the iteration methods like bisection method, false position method, Taylor series method, Newton-Raphson method, Muller's method, etc, one may see Burden and Faires [5]. These methods are inherent with certain weaknesses when one desires to utilize them for the extraction of the n^{th} root such as the requirement of an initial guess, a large number of arithmetic operations and several iterative steps for convergence, etc. For example, in Newton-Raphson method, a previously guessed value is required and an iterative method may not converge if the initial guess differs very much from the exact root. In [6] Matthews has discussed the computation of n^{th} root of positive integers. Newton`s method in the extraction of n^{th} root has been demonstrated by Priestley [7]. An algorithm for finding the root with a five-function calculator employing logarithm has been furnished by Muench and Wildenberg in [8].

However, there is no direct analytical method for the extraction of the n^{th} root of a given number. Developing an algorithm to find the n^{th} root has remained an interesting and challenging problem. To mitigate the drawbacks in the existing methods, there is a crucial need to develop a new efficient method. Under this background, the present paper focuses attention on a computationally simple numerical algorithm for the digit-by-digit determination of the real positive n^{th} root of a given positive real number up to any desired accuracy, by introducing three functions U_i, V_i and T_i and establishing a relationship involving them.

2. NUMBER OF DIGITS DUE TO EXPONENTIATION

First we need a basic result for our algorithm. Let $\mathbb{N}$ and W denote the sets of natural and whole numbers respectively. Let α be a given positive integer with k digits, $k \in \mathbb{N}$. Let us consider the number of digits occurring while α is raised to n^{th} power.

Corresponding author at: Ramanujan School of Mathematical Sciences, *Pondicherry University, Pondicherry, India-605014*
E-mail address: ramasamy.mat@pondiuni.edu.in

Theorem 1

If m denotes the number of digits in α^n then
$(k - 1)\, n + 1 \le m \le kn$, for all $n, k \in \mathbb{N}$. (2.1)

Proof

Case (i) First let us consider $n = 2$. If $k = 1$,

Then, $m = \begin{cases} 1, & \text{for } 1 \le \alpha \le 3 \\ 2, & \text{for } 4 \le \alpha \le 9 \end{cases}$ (2.2)

If k > 1, then we have

$$m = \begin{cases} 2k - 1, & \text{for } 10^{k-1} \le \alpha \le 3(10^{k-1}) \\ 2k, & \text{for } 4(10^{k-1}) \le \alpha \le 10^k - 1 \end{cases} \quad (2.3)$$

and $2k - 1 \le m \le 2k$
for $3(10^{k-1}) + 1 \le \alpha \le 4(10^{k-1}) - 1$ (2.4)
Equations (2.2), (2.3) and (2.4) imply that (2.1) holds for n = 2.

Case (ii) Next suppose that $n > 2$. If $k = 1$, then $\alpha < 10$ or $\alpha^n < 10^n$. This implies that α^n contains utmost n digits. Therefore
$1 \le m \le n$ (2.5)

Consider a k-digit number α with $k > 1$. Then we have $10^{k-1} \le \alpha < 10^k$. Hence, $10^{(k-1)n} \le \alpha^n < 10^{kn}$. This implies that
$(k\text{-}1)\, n + 1 \le m \le kn$ (2.6)

Hence the equation (2.5) and (2.6) imply that (2.1) holds for $n > 2$. Thus the theorem holds for all values of $n, k \in \mathbb{N}$.

3. NUMERICAL ALGORITHM

Suppose it is required to find the n^{th} root of a given positive integer M. First we present a method when M is a perfect n^{th} power of a positive integer. The general case is postponed to section 5.

Step 1

Starting from the unit place of M, split the digits of M into a maximum possible number of blocks each of size n. In the process, if certain left most digits of M are still remaining, then form another block with these digits. Let the total number of blocks, so formed with the digits of M, be k. Then, by Theorem 1, the real positive n^{th} root of M consists of k-digits, say $a_1, a_2, \dots, a_k$ starting from the unit place of it.

Let us denote M by $M(k, n)$. Then
$M(k, n) = (10^{k-1} a_k + 10^{k-2} a_{k-1} + \dots + a_1)^n$ (3.1)
Name the blocks of $M(k, n)$, starting from the right of $M(k, n)$, as $B_1, B_2, \dots, B_k$. Let $|B_i|$ denote the size of B_i. Then
$|B_i| = n$, for i=1, 2, ..., k-1 (3.2)
and $1 \le |B_k| \le n$ (3.3)

Determine the maximum possible value of $a_k \in \mathbb{N}$ such that
$(a_k)^n \le B_k$ (3.4)
Let $R_k = a_k$ (3.5)
If $k = 1$, then R_k gives the required real positive n^{th} root of $M(k, n)$. If $k \ne 1$, then go to step 2.

Step 2

Form the block $D_k = B_k - a_k^n$ (3.6)
Define M_{k-1} such that
$M_{k-1} = M(k, n) - 10^{(k-1)n} a_k^n$ (3.7)
Let $U_k = a_k$ (3.8)
and $V_k = 0$ (3.9)

Starting from the unit place of M_{k-1}, split the digits of M_{k-1} into (k-2) blocks each of size n and form another block with the remaining left most digits of M_{k-1}. Denote the left most block of M_{k-1} by $B_{k-1}(M_{k-1})$. Then we have
$B_{k-1}(M_{k-1}) = D_k \bullet B_{k-1}$ (3.10)
where $D_k \bullet B_{k-1}$ is defined as the concatenation (denoted by '•') of D_k and B_{k-1}.

Take $U_{k-1} = 10\,(U_k + V_k)$ (3.11)
Determine the maximum value of $V_{k-1} \in W$ such that
$V_{k-1}\, T_{k-1} \le B_{k-1}(M_{k-1})$ (3.12)
where $T_{k-1} = \sum_{r=1}^{n} \binom{n}{r} U_{k-1}^{n-r} V_{k-1}^{r-1}$ (3.13)
and $\binom{n}{r}$ denotes the number of combinations of n objects taken r at a time.

Let $a_{k-1} = V_{k-1}$ (3.14)
Define $R_{k-1} = R_k \circ V_{k-1}$ so that
$R_{k-1} = 10\, a_k + a_{k-1}$ (3.15)
If $k = 2$, then R_{k-1} is the required root. If $k \ne 2$, then go to step 3.

Step 3

Define $M_{k-2} = M_{k-1} - 10^{(k-2)n} V_{k-1} T_{k-1}$ (3.16)
Repeat the process as in step 2, where M_{k-2} consists of (k-2) blocks.

Find U_{k-2} and V_{k-2} by following the similar procedures in equations (3.11) and (3.12) respectively. Let $a_{k-2} = V_{k-2}$ and define R_{k-2}, etc., in a similar way. Repeat the process until k reduces to 1. Now $R_1 = 10^{k-1} a_k + 10^{k-2} a_{k-1} + \dots + a_1$ is the n^{th} root of M.

4. PROOF OF THE METHOD

We shall validate the method presented in section 3, by the principle of mathematical induction.

Lemma 1

$1 \le a_k \le 9$

Proof

Using (3.3) and (3.4), we obtain $a_k^n \le B_k < 10^n$. Hence $a_k < 10$. $a_k \ge 1$, since $B_k \ne 0$ which imply that $1 \le a_k \le 9$.

Lemma 2

$0 \le a_i \le 9$, $i = 1, 2, \dots, k\text{-}1$.

Proof

Employing (3.6) we get $10^n D_k = 10^n B_k - 10^n a_k^n$. It follows that $10^n a_k^n + 10^n D_k + B_{k-1} = 10^n B_k + B_{k-1}$. Using (3.8) and (3.10) we get $10^n U_k^n + B_{k-1}(M_{k-1}) = B_k \bullet B_{k-1}$. Because of (3.12) this relation implies that $10^n U_k^n + V_{k-1} T_{k-1} \le B_k \circ B_{k-1}$. In view of (3.2) and (3.3) we obtain $|10^n U_k^n + V_{k-1} T_{k-1}| \le 2n$. Now (3.9) and (3.11) imply that $|U_{k-1}^n + V_{k-1} T_{k-1}| \le 2n$. Using Binomial theorem, we get $|(U_{k-1} + V_{k-1})^n| \le 2n$. Therefore $|(10\ U_k + V_{k-1})^n| \le 2n$.

In view of $U_k = a_k \neq 0$ and the maximum choice of a_k, by Theorem 1 it follows that $0 \le V_{k-1} \le 9$. Using (3.14) we obtain $0 \le a_{k-1} \le 9$.

Similarly we can prove that $0 \le a_i \le 9$, $i = 1, 2,..., k-2$.

Let $M(k, n)$ be the perfect n^{th} power of a positive integer, say λ. Then $M(k, n) = \lambda^n$ (4.1) where $\lambda = 10^{k-1} a_k + 10^{k-2} a_{k-1} + ... + a_1$, using (3.1).

From equations (3.11) and (3.14) and step 3 of the algorithm it follows that

$$\left.\begin{array}{l} V_i = a_i,\ i = 1, 2, \dots, k-1 \\ U_i = 10(U_{i+1} + V_{i+1}),\ i = 1, 2, \dots, k-1 \end{array}\right\} \quad (4.2)$$

Equation (3.13) and step 3 of algorithm imply that

$$T_i = \sum_{r=1}^{n} \binom{n}{r} U_i^{n-r} V_i^{r-1},\ i = 1, 2, \dots, k-1 \quad (4.3)$$

It is seen that

$$U_i = \sum_{j=i+1}^{k} a_j 10^{j-i},\quad i = 1, 2, \dots, k-1 \quad (4.4)$$

from equations (3.8), (3.9) and (3.11) and step 3 of algorithm. For i = 1, 2,..., $k-2$, step 3 and equations (3.7) and (3.16) imply that $M_i = M(k,n) - \{a_k^n\ 10^{(k-1)n} + \sum_{j=i+1}^{k-1} V_j\ T_j 10^{(j-1)n}\}$.

From equations (3.5) and (3.15) and step 3, we obtain $R_i = \sum_{j=i}^{k} a_j 10^{j-i}$, i = 1, 2, ..., k.

For i = 1, 2, ..., k -1, relation (4.4) implies that $U_i = 10 \sum_{j=i+1}^{k} a_j 10^{j-(i+1)} = 10\ R_{i+1}$. Hence from equations (3.8), (3.9) and (3.11) and step 3 of the algorithm we obtain $U_i = 10\ R_{i+1}$, $i = 1, 2, \dots, k-1$.

For i = 1, 2, ..., $k-2$, step 3 and equations (3.7) and (3.16) imply that $M_i = M(k,n) - \{a_k^n\ 10^{(k-1)n} + \sum_{j=i+1}^{k-1} V_j\ T_j 10^{(j-1)n}\}$.

From equations (3.5) and (3.15) and step 3, we get $R_i = \sum_{j=i}^{k} a_j 10^{j-i}$, i = 1, 2, ..., k.

For i = 1, 2, ..., k -1, relation (4.4) implies that $U_i = 10 \sum_{j=i+1}^{k} a_j 10^{j-(i+1)} = 10\ R_{i+1}$. Hence $U_i = 10\ R_{i+1}$, $i = 1, 2, \dots, k-1$.

Theorem 2

Given n $\in \mathbb{N}$, the $U_i(i = 1, 2, ..., k)$, $V_i(i = 1, 2, ..., k)$ and $T_i(i = 1, 2, ..., k - 1)$ satisfy the relation

$$U_k^n\ 10^{(k-1)n} + \sum_{i=1}^{k-1} V_i\ T_i 10^{(i-1)n} = M(k,n) \quad (4.5)$$

for all $k \in \mathbb{N}$.

Proof

From (3.1), it is seen that the relation (4.5) holds for k = 1.

Assume that the relation (4.5) holds for $k \in \mathbb{N}$. Let us consider $M(k + 1, n)$. Then the relations (3.8), (3.9), (4.2) and (4.3) hold with k increased by 1.

Now $U_{k+1}^n 10^{kn} + \sum_{i=1}^{k} V_i T_i 10^{(i-1)n} = U_{k+1}^n 10^{kn} + \sum_{i=2}^{k} V_i T_i 10^{(i-1)n} + V_1 T_1 = \left(U_{k+1}^n\ 10^{(k-1)n} + \sum_{i=2}^{k} V_i\ T_i 10^{(i-2)n}\right)10^n + V_1 T_1$
$= (10^{k-1} a_{k+1} + 10^{k-2} a_k + \dots + a_2)^n 10^n + V_1 T_1$
by using induction assumption for U_i(i = 2, 3, ..., k+1), V_i(i = 2, 3, ..., k+1) and T_i(i = 2, 3, ..., k).

Hence $U_{k+1}^n 10^{kn} + \sum_{i=1}^{k} V_i T_i 10^{(i-1)n} = (10^k a_{k+1} + 10^{k-1} a_k + \dots + 10 a_2)^n + V_1 T_1$ (4.6)

Using the relation (4.5) applicable for (k + 1), the right side of (4.6) becomes $U_1^n + V_1 T_1 = U_1^n + V_1 \left[\binom{n}{1} U_1^{n-1} + \binom{n}{2} U_1^{n-2} V_1 + \dots + \binom{n}{n} V_1^{n-1}\right] = (U_1 + V_1)^n$, using Binomial theorem
$= (10^k a_{k+1} + 10^{k-1} a_k + \dots + 10 a_2 + a_1)^n$
which yields $M(k + 1, n)$. Hence the theorem follows by induction on k.

5. n^{th} ROOT IN GENERAL CASE

In general one may require to find the n^{th} root of a positive integer, which may not be a perfect n^{th} power or n^{th} root of a positive real number. Then the algorithm presented in section 3 has to be adopted with certain modifications as indicated below.

Let M be a positive real number. It may consist of integral and decimal parts. Suppose that the root is required upto h places of decimals. Then multiply the given number M by 10^{hn} and follow the algorithm in section 3 for $[10^{hn}M]$, wherein R_1 would be the integral part of the n^{th} root of $10^{hn}M$. Now divide R_1 by 10^h to obtain the n^{th} root of M up to h places of decimals, since $\sqrt[n]{M} = \frac{1}{10^h} \sqrt[n]{10^{hn}M}$.

Let us take some examples.

5.1 Case *M* is a perfect n^{th} power

Example 1. Find $\sqrt[3]{16457616482180544}$.

Step 1

Let M = 16457616482180544. Since n = 3 split the digits of M as described in section 3. We have

B_6	B_5	B_4	B_3	B_2	B_1
16	457	616	482	180	544

where k = 6. We assert that $a_6 = 2$ is the maximum value such that $a_6^3 \le 16$. It is seen that $R_6 = a_6 = 2$. Since $k \neq 1$, go to step 2.

Step 2

Form the block $D_6 = B_6 - \alpha_6^3 = 8$. Define $M_5 = M - 10^{5\cdot3}\alpha_6^3$ = 8 457 616 482 180 544. Let $U_6 = \alpha_6 = 2$ and $V_6 = 0$. It is seen that $B_5(M_5) = D_6 \bullet B_5 = 8457$.

Split the digits of M_5 as

$B_5(M_5)$	B_4	B_3	B_2	B_1
8457	616	482	180	544

Take $U_5 = 10\,(U_6 + V_6) = 20$. Assigning the values of 0, 1, 2, 3, 4, 5 and 6 to V_5, we get V_5T_5 = 0, 1261, 2648, 4167, 5824, 7625 and 9576 respectively, where $T_5 = \sum_{r=1}^{3}\binom{3}{r}U_5^{3-r}V_5^{r-1}$. Hence the maximum value of $V_5 \in$ W, such that $V_5T_5 \leq 8457$, is 5 where $T_5 = 1525$. Let $\alpha_5 = V_5 = 5$ and $R_5 = 10\,\alpha_6 + \alpha_5 = 25$.

For the subsequent steps, the results are presented in the following table.

Step 3

i	M_i	U_i	$B_i(M_i)$	V_i	T_i	V_iT_i	α_i	R_i
4	832616482180544	250	832616	4	190516	762064	4	254
3	70552482180544	2540	70552482	3	19377669	58133007	3	2543
2	12419475180544	25430	12419475180	6	1940512476	11643074856	6	25436
1	776400324544	254360	776400324544	4	194100081136	776400324544	4	254364

Hence $R_1 = 10^5\alpha_6 + 10^4\alpha_5 + 10^3\alpha_4 + 10^2\alpha_3 + 10\alpha_2 + \alpha_1 = 254364$ is the cube root of M. It is seen that $U_6^3 10^{5\cdot3} + \sum_{i=1}^{5} V_i T_i 10^{(i-1)\cdot3} = 254364^3$ = M.

For a given problem, the above stepwise procedure can be followed. However, for a simpler presentation, another procedure may prove handy, as illustrated in the following example

Example 2. Find $\sqrt[4]{16192865729296}$.

	α_4	α_3	α_2	α_1		
	U_4	V_3	V_2	V_1		
	2	0	0	6		
	16	1928	6572	9296	$U_4 = 2$, $V_4 = 0$	$R_4 = 2$
2^3	16				$\longleftarrow U_4^4$	
	0	1928			$U_3 = 10(U_4+V_4) = 20$, $V_3 = 0$	$R_3 = 20$
$T_3 \longrightarrow$ 32000		0			$\longleftarrow V_3T_3$	
		1928	6572		$U_2 = 10(U_3+V_3) = 200$, $V_2 = 0$	$R_2 = 200$
$T_2 \longrightarrow$ 32000000			0		$\longleftarrow V_2T_2$	
		1928	6572	9296	$U_1 = 10(U_2+V_2) = 2000$, $V_1 = 6$	$R_1 = 2006$
$T_1 \longrightarrow$ 32144288216		1928	6572	9296	$\longleftarrow V_1T_1$	
				0		

$T_3 = \sum_{r=1}^{4}\binom{4}{r}U_3^{4-r}V_3^{r-1} = 32{,}000$

$T_2 = \sum_{r=1}^{4}\binom{4}{r}U_2^{4-r}V_2^{r-1} = 32{,}000{,}000$

$T_1 = \sum_{r=1}^{4}\binom{4}{r}U_1^{4-r}V_1^{r-1} = 32{,}144{,}288{,}216$. Hence 2006 is the required root.

5.2 Case M is not a perfect nth power

Example 3. Find $\sqrt[6]{175}$ correct to two places of decimals**.**

$$\sqrt[6]{175} = \frac{1}{10^3}\sqrt[6]{175 \times 10^{18}}$$

	2	3	6	5		
	175	000000	000000	000000	$U_4 = 2,\ V_4 = 0$	$R_4 = 2$
2^6	64				$\longleftarrow U_4^6$	
	111	000000			$U_3 = 10(U_4+V_4) = 20,\ V_3 = 3$	$R_3 = 23$
$T_3 \longrightarrow 28011963$	84	035889			$\longleftarrow V_3T_3$	
	26	964111	000000		$U_2 = 10(U_3+V_3) = 230,\ V_2 = 6$	$R_2 = 236$
$T_2 \longrightarrow 4122596132256$	24	735576	793536		$\longleftarrow V_2T_2$	
	2	228534	206464	000000	$U_1 = 10(U_2+V_2) = 2360,\ V_1 = 5$	$R_1 = 2365$
$T_1 \longrightarrow 441582605197453125$	2	207913	025987	265625	$\longleftarrow V_1T_1$	
		20621	180476	734375		

Hence $\sqrt[6]{175} \cong 2.365 \cong 2.37$

Example 4. Find $\sqrt[3]{35.66\ldots}$ correct to three places of decimals.

$$\sqrt[3]{35.66\ldots} = \frac{1}{10^4}\sqrt[3]{35.66\ldots \times 10^{12}}$$

	3	2	9	1	7		
	35	666	666	666	666	$U_5 = 3,\ V_5 = 0$	$R_5 = 3$
3^3	27					$\longleftarrow U_5^3$	
	8	666				$U_4 = 10(U_5+V_5) = 30,\ V_4 = 2$	$R_4 = 32$
$T_4 \longrightarrow 2884$	5	768				$\longleftarrow V_4T_4$	
	2	898	666			$U_3 = 10(U_4+V_4) = 320,\ V_3 = 9$	$R_3 = 329$
$T_3 \longrightarrow 315921$	2	843	289			$\longleftarrow V_3T_3$	
		55	377	666		$U_2 = 10(U_3+V_3) = 3290,\ V_2 = 1$	$R_2 = 3291$
$T_2 \longrightarrow 32482171$		32	482	171		$\longleftarrow V_2T_2$	
		22	895	495	666	$U_1 = 10(U_2+V_2) = 32910,\ V_1 = 7$	$R_1 = 32917$
$T_1 \longrightarrow 3249895459$		22	749	268	213	$\longleftarrow V_1T_1$	
			146	227	453		

Hence $\sqrt[3]{35.66\ldots} \cong 3.2917 \cong 3.292$

6. CONCLUSION

The analytic method contained in this paper would enable one to carry out digit-by-digit extraction of the n^{th} root of a given positive real number which can be directly implemented. When one uses Newton's method for the determination of the n^{th} root, he shall find out an initial solution and then he has to resort to differentiation. On the other hand, the method presented in this paper uses simple arithmetic operations only and an initial solution is not necessary. Since it is computationally feasible, it may be built in electronic devices to determine the n^{th} root of a given positive real number without demanding more of memory space.

ACKNOWLEDGEMENT

The authors sincerely thank the referee for the several suggestions towards the improvement of the paper.

REFERENCES

[1] (a) J. Napier, *Mirifici Logarithmorum Canonis Descriptio (1614)* and E. Wright (English Translation from Latin), *A Description of the Admirable Table of Logarithms* (1616).
(b) J. Napier (1969), *A Description of the Admirable Table of Logarithms, London 1616*, Amsterdam; New York, N.Y.: Theatrum Orbis Terrarum; Da Capo Press.
[2] R.G.Dromey, *How to solve it by computer*, Pearson Education, New Delhi, 2008.
[3] I. Niven and H.Zuckerman, *An introduction to the theory of numbers*, Wiley Student Edition, New Delhi, 1991.
[4] J. Gower, *A note on an iterative method for root extraction,* The Computer J., 1(3) (1958), 142-143.
[5] R. Burden and D. Faires, *Numerical analysis*, Thomson Asia, Bangalore, 2005.
[6] K. Matthews, *Computing* m^{th} *roots*, The College Math. J., 19 (1988), 174- 176.
[7] W. Priestley, *From square roots to* n^{th} *roots*, Newton`s method in disguise, The College Math. J., 30:5 (1999), 387-388.
[8] D. Muench and G. Wildenberg, *A logarithm algorithm for a five-function calculator*, The Two-year College Math. J., 14 (4) (1983), 324-326.