MATHEMATICS OF COMPUTATION, VOLUME 35, NUMBER 152
OCTOBER 1980, PAGES 1221–1230

Optimal Partitioning of Newton's Method for Calculating Roots

By Günter Meinardus and G. D. Taylor*

Abstract. In this paper, an algorithm is given for calculating roots via Newton's method initialized with a piecewise best starting approximation. The piecewise best starting approximation corresponds to an optimal partitioning of the interval of the domain of Newton's method. Explicit formulas are given when piecewise linear polynomials are used for the best starting approximations. Specific tables are given for square roots, cube roots and reciprocal square roots.

1. Introduction. An effective algorithm for calculating roots is Newton's method initialized with a best starting approximation [11], [12]. Recently [2], [3], this procedure has been modified, in that a piecewise best starting approximation was used for initializing the Newton iteration. This is equivalent to subdividing the interval of application of the Newton iteration into subintervals and applying the theory of best starting approximations to each subinterval. In this paper, we shall describe how this subdivision can be done in an optimal manner.

The theory of best starting approximations for calculating roots was first studied by Moursund [11] for the special case of square roots. This theory was extended to general roots by Moursund and Taylor in [12]. Subsequent studies found that the best starting approximation for calculating roots via Newton's method is independent of the number of iterations to be used and is, in fact, a multiple of the best relative approximation to the root [8], [13], [14], [15], [16], [18]. Surprisingly, it was also shown [8], [13], [15] that one of the square root subroutines in use prior to the development of this theory [5] was, in fact, the method of Moursund.

This theory allows considerable leeway in designing a specialized root routine. In the case of large scale computers, it is possible to design routines that return a predetermined accuracy and will require less execution time than library routines of the system. This time differential will be especially dramatic when full machine accuracy is not required. One such case, where this approach was taken, was in the development of a reciprocal square root routine using a divide-free Newton iteration for inclusion in the particle moving section of a relativistic plasma code on an IBM 360/91. The design constraints in this case were a required accuracy of 10^{-5} after one Newton iteration

Received May 31, 1977; revised July 24, 1978.

1980 *Mathematics Subject Classification.* Primary 65D20, 41A30.

Key words and phrases. Computation of roots, optimal initialization of Newton's method for computing roots, best piecewise starting approximations.

* Research sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant no. AFOSR-76-2878.

initialized with a linear polynomial on [1/8, 1/2]. Here the interval [1/8, 1/2] was divided into five subintervals in order to satisfy these constraints.

For the case of microcomputers, these algorithms can be incorporated as firmware to calculate roots. In [2], [3] square root routines based on this theory were developed for 8-bit and 16-bit microprocessors. For the 16-bit routine, the goal was to develop an algorithm which would give 15 bits of accuracy after one Newton iteration initialized with a linear polynomial and have as its domain of application all numbers of the form $X = \{j/2^{16}\}_{j=2^{14}+1}^{2^{16}}$. In order to accomplish this, it was necessary to partition the point set into $X = X_1 \cup X_2 \cup X_3$, $X_1 = X \cap (1/4, c_1]$, $X_2 = X \cap (c_1, c_2]$ and $X_3 = X \cap (c_2, 1]$ for appropriately chosen c_1 and c_2. Treating each of these three sets independently, it is possible to develop a square root algorithm satisfying the constraints listed above (with the exception of the domain constraint). Thus, the algorithm used a piecewise best linear polynomial starting approximation. Specifically, given $y > 0$, this algorithm would first compute an integer k and a real number $x \in X$ so that $y = 2^{2k} \cdot x$ (argument reduction). Next, it would determine which subset (X_1, X_2 or X_3) contains x and then evaluate the appropriate piece of the best linear starting approximation at x. This value is then used to initialize one Newton iteration for calculating square roots. The result of this iteration is then multiplied (shifted) by 2^k, and this value is returned as the desired square root. This algorithm was compared with the corresponding direct and Cordic type of methods [18]) including Chen's modified version [4], and was found to be preferable for the types of architectures considered.

The organization of this paper is the following: Section 2 contains a summary of the definitions and basic theoretical results of best starting approximations, Section 3 gives our general results, and Section 4 gives specific examples of this theory for calculating square roots, reciprocal square roots using a divide-free iteration, and cube roots, each subject to certain design constraints. The major intent of this paper is to provide guidance for the implementation of these ideas as mathematical software.

2. **Definitions and Basic Notions.** Let $[a, b]$ be a fixed interval with $0 < a < b$ and set $R_k^m[a, b] = \{R = P/Q\colon\ P \in \Pi_m, Q \in \Pi_k, Q(x) > 0$ for all $x \in [a, b], (P, Q) = 1\}$, where Π_k denotes the class of all real algebraic polynomials of degree less than or equal to k, and (P, Q) denotes the greatest common (polynomial) divisor of P and Q. Fix n a real number, $n \neq 0$ or ± 1, and define $N\colon\ C^+[a, b] \longrightarrow C[a, b]$, where $C^+[a, b]$ denotes the class of all continuous positive functions defined on $[a, b]$ by

$$N(h)(x) = \frac{1}{n}\left[(n-1)h(x) + \frac{x}{h^{n-1}(x)}\right].$$

Observe that $N(h)(x)$, for fixed x, is simply the result of one Newton iteration for calculating $x^{1/n}$ with $h(x)$ as its starting approximation (or initial guess). That is, the formula for N is simply the result of applying Newton's method to $y^n - x = 0, x$ fixed. As usual, we also define N^ν by $N^\nu(h)(x) = N(N^{\nu-1}(h))(x)$, the result of ν Newton iterations. Then $R^* \in R_k^m[a, b]$ is said to be the best (relative) starting approximation from $R_k^m[a, b]$ for calculating nth roots on $[a, b]$, provided

$$(1)\quad \eta[a, b] = \left\| \frac{x^{1/n} - N(R^*)(x)}{x^{1/n}} \right\|_{[a,b]} = \min_{R \in R_k^m[a,b]} \left\| \frac{x^{1/n} - N(R)(x)}{x^{1/n}} \right\|_{[a,b]},$$

where $\|f(x)\|_{[a,b]} = \max\{|f(x)|: x \in [a, b]\}$ for $f \in C[a, b]$. We shall suppress the subscript $[a, b]$ on $\|\cdot\|$ whenever the meaning is clear. Thus, the relative error of approximating $x^{1/n}$ with one Newton iteration is minimized on the interval $[a, b]$, if $R^*(x)$ is used as the initial guess. It is shown in [6], [10] that R^* exists, is unique and is a multiple (depending upon n, $[a, b]$, m and k) of the best relative approximation $\widetilde{R}(x)$ to $x^{1/n}$ from $R_k^m[a, b]$, i.e.,

$$(2)\quad \left\| \frac{x^{1/n} - \widetilde{R}(x)}{x^{1/n}} \right\| = \min_{R \in R_k^m[a,b]} \left\| \frac{x^{1/n} - R(x)}{x^{1/n}} \right\|.$$

From the general theory of uniform relative approximation [1], it is known that $\widetilde{R}(x)$ exists, is unique and can be calculated by various methods; see, for example, [8]. In fact [7], [11], if $\|(x^{1/n} - \widetilde{R}(x))/x^{1/n}\| = \lambda$, then $R^*(x) \equiv \gamma\widetilde{R}(x)$, where

$$(3)\quad \gamma = [((1+\lambda)^{n-1} - (1-\lambda)^{n-1})/2(n-1)\lambda(1-\lambda^2)^{n-1}]^{1/n},$$

and this $R^*(x)$ is the best starting approximation from $R_k^m[a, b]$ for ν Newton iterates, i.e.

$$\eta^\nu[a, b] = \left\| \frac{x^{1/n} - N^\nu(R^*)(x)}{x^{1/n}} \right\| = \min_{R \in R_k^m[a,b]} \left\| \frac{x^{1/n} - N^\nu(R)(x)}{x^{1/n}} \right\|.$$

In closing this section, we would like to remark that a theory of best (absolute) starting approximations for calculating roots, i.e. $\inf_{R \in R_k^m[a,b]} \|x^{1/n} - N^\nu(R)(x)\|$, ν a positive integer is neither as well developed nor as rich as the corresponding relative theory. It is known [10] that best absolute starting approximations exist, are unique and can (in theory) be calculated by a Remes-type algorithm or a generalized differential correction algorithm [7]. Whether or not best absolute starting approximations are a multiple of some other well-known approximation to $x^{1/n}$ is not known (they are not a multiple of the best uniform approximation to $x^{1/n}$) and optimal partitioning results, corresponding to what we shall prove for the relative case, are not known. Thus, unless explicitly stated to the contrary, we shall be concerned with the relative theory in what follows.

3. Theoretical Results. In this setting, we wish to first state a lemma that is well known for the case of best relative approximations for roots.

LEMMA 1. *If $R^* \in R_k^m[a, b]$ is the best starting approximation from $R_k^m[a, b]$ for calculating nth roots on $[a, b]$, then $R(t) = \rho^{1/n}R^*(t/\rho)$, $\rho a \leqslant t \leqslant \rho b$, $\rho > 0$ is the best starting approximation from $R_k^m[\rho a, \rho b]$ for calculating nth roots on $[\rho a, \rho b]$. Furthermore, $\eta[a, b] = \eta[\rho a, \rho b]$.*

Proof. This follows via the change of variable $x = t/\rho$. □

Using this result, we are able to prove our optimal partitioning result. The flavor of this result is the following. Suppose that one wishes to subdivide the interval $[a, b]$, $0 < a < b$, into ν subintervals and calculate nth roots on $[a, b]$ by actually calculating nth roots independently on each subinterval. Then it turns out that there exists a unique partitioning of $[a, b]$ into ν subintervals such that the relative error of approximating $x^{1/n}$ with a Newton iterate initialized on each subinterval with the best starting approximations for that subinterval is minimal over all such partitions of $[a, b]$ and, in fact, the relative error on each subinterval is the same. This result is an extension of a result of James and Jarratt [9]. In this paper, a similar result is proven for computing square roots via Newton's method initialized with the best relative approximation to the square root (two other initializations were also considered).

THEOREM 1. *Let* $\mathcal{P} = \{\mathbf{d}\colon \mathbf{d} = \{d_0, d_1, \ldots, d_\nu\}$ *with* $a = d_0 < d_1 < d_2 < \cdots < d_{\nu-1} < d_\nu = b\}$ *be the set of all partitions of* $[a, b]$ *into* ν *subintervals. Then, there exists one and only one partition,* $\mathbf{c} = \{c_0, c_1, \ldots, c_\nu\} \in \mathcal{P}$, *for which*

$$\max_{0 \leqslant i \leqslant \nu-1} \eta[c_i, c_{i+1}] = \min_{\mathbf{d} \in \mathcal{P}} \max_{0 \leqslant i \leqslant \nu-1} \eta[d_i, d_{i+1}].$$

This unique partition is given by the formulas $c_j = f^j a$, $j = 0, 1, \ldots, \nu$, *where* $f = (b/a)^{1/\nu}$. *In addition, this theorem holds with* η *replaced by* η^l *for* l *a positive integer, and* $\eta^l[c_i, c_{i+1}] = \eta^l[c_{i+1}, c_{i+2}]$ *for all* l *and* i.

Proof. First, observe that for the partition $\mathbf{c} = \{c_j\}_{j=0}^{\nu}$, we have that $[c_\mu, c_{\mu+1}] = [\rho_\mu c_0, \rho_\mu c_1]$, where $\rho_\mu = f^\mu$, for $\mu = 1, \ldots, \nu - 1$. Thus, by Lemma 1, $\eta[c_\mu, c_{\mu+1}] = \eta[c_0, c_1]$ for $\mu = 1, \ldots, \nu - 1$.

To prove the min-max statement of Theorem 1, (which will also establish the uniqueness claim), we prove the following result first which follows via a straightforward zero-counting argument. Namely, if $[a, b]$, $0 < a < b$, and $[c, d]$, $0 < c < d$, are any two intervals and $a/b > c/d$, then $\eta[a, b] > \eta[c, d]$. Now, by Lemma 1, we can replace $[c, d]$ by $[\rho c, \rho d]$, where $\rho = a/c$ and $\eta[c, d] = \eta[\rho c, \rho d]$. Thus, setting $e = \rho d < b$, we shall prove that $\eta[a, b] > \eta[a, e]$, which will establish this result. To do this, let $\widetilde{R}(x)$ be the best relative approximation to $x^{1/n}$ on $[a, b]$ from $\mathcal{R}_k^m[a, b]$. Define the defect, $\tilde{d}$, of $\widetilde{R}(x)$ by $\tilde{d} = \min(m - \partial\widetilde{P}, k - \partial\widetilde{Q})$, where $\widetilde{R}(x) = \widetilde{P}(x)/\widetilde{Q}(x)$, and ∂P denotes the exact degree of the polynomial P. Then, by the standard theory of best relative approximation [1], there exists at least $N = k + m + 2 - \tilde{d}$ extreme points, $a \leqslant x_1 < x_2 < \cdots < x_N \leqslant b$, on which the error curve $E(x) = 1 - \widetilde{R}(x)/x^{1/n}$ alternates, i.e., $|E(x_i)| = \|E\|$, $i = 1, \ldots, N$, and $E(x_i) = -E(x_{i+1})$, $i = 1, 2, \ldots, N - 1$. Since $E \in C^1[a, b]$, we must have that $E'(x_i) = 0$ for at least $i = 2, \ldots, N - 1$. Thus, $E'(x)$ must have at least $k + m - \tilde{d}$ zeros. Now

$$E'(x) = \frac{x^{(1/n)-1}\,[x\widetilde{Q}(x)\widetilde{P}'(x) - \widetilde{P}(x)\,(\widetilde{Q}(x)/n + x\widetilde{Q}'(x))]}{(x^{1/n}\widetilde{Q}(x))^2},$$

and since $a > 0$, and the degree of the polynomial in the brackets in the numerator is less than or equal to $(\partial\widetilde{P} + \partial\widetilde{Q})$, which is less than or equal to $k + m - 2\tilde{d}$, we see that $E'(x)$ can have at most $k + m - 2\tilde{d}$ zeros in $[a, b]$. Comparing these two zero counts, we see that we must have $\tilde{d} = 0$, $\partial\tilde{P} = m$, $\partial\tilde{Q} = k$, $x_1 = a$, $x_N = b$, and that $E(x)$ must have precisely $N = k + m + 2$ extreme points in $[a, b]$. Here, $y \in [a, b]$ is said to be an extreme point if $|E(y)| = \|E\|$. Thus, $\tilde{R}$ is not the best relative approximation to $x^{1/n}$ on $[a, e]$ from $\mathcal{R}_k^m[a, e]$, since $E(x)$ does not have the necessary alternating behavior on $[a, e]$. Let $R_1 \in \mathcal{R}_k^m[a, e]$ be the unique best relative approximation to $x^{1/n}$ on $[a, e]$. We claim that R_1 is not a multiple of $\tilde{R}$. This follows from the above zero-counting argument, since, for any real $c \neq 0$, $E_c(x) = 1 - c\tilde{R}(x)/x^{1/n}$ must be such that $E_c'(x)$ vanishes in $[a, b]$ only where $E'(x)$ vanishes in $[a, b]$, implying that $c\tilde{R}$ does not have the necessary number of alternations to be the best relative approximation to $x^{1/n}$ in $[a, e]$. Since the best starting approximations for calculating nth roots from $\mathcal{R}_k^m[a, e]$ and $\mathcal{R}_k^m[a, b]$ on $[a, e]$ and $[a, b]$ are multiples of R_1 and $\tilde{R}$, respectively, and $\tilde{R} \in \mathcal{R}_k^n[a, e]$, we must have that

$$\eta[a, e] = \min_{R \in \mathcal{R}_k^m[a,e]} \left\| \frac{x^{1/n} - N(R)(x)}{x^{1/n}} \right\|_{[a,e]} < \eta[a, b]$$

by uniqueness.

Now let $\mathbf{d} = \{d_j\}_{j=0}^{\nu}$, $a = d_0 < d_1 < \cdots < d_\nu = b$, be a different partition of $[a, b]$ than the one given by the formulas in the hypothesis of Theorem 1. Then we claim that for some j, $0 \leqslant j \leqslant \nu - 1$, $ad_{j+1}/d_j > c_1$ must hold. Indeed, if $ad_{j+1}/d_j \leqslant c_1$ for all j, then there must be an index j_1, $0 \leqslant j_1 \leqslant \nu - 1$, at which strict inequality holds as the partitions are distinct. Let j_1 be the first index where strict inequality holds. This in turn implies (proof following) that $d_j \leqslant c_j$ for $0 \leqslant j \leqslant j_1$, and $d_j < c_j$ for $j_1 < j \leqslant \nu$, which is a contradiction as $d_\nu = c_\nu = b$. We prove that $d_j \leqslant c_j$ for $0 \leqslant j \leqslant j_1$ and $d_j < c_j$ for $j_1 < j \leqslant \nu$ by an inductive argument, as follows. Note that $d_0 = c_0 = a$. Assume that $d_j \leqslant c_j$ holds for some j, $0 \leqslant j \leqslant j_1 - 1$. Then $d_{j+1} \leqslant c_1 d_j/a$ by our original assumption so that $d_{j+1} \leqslant c_0^{-1/\nu} c_\nu^{1/\nu} d_j \leqslant c_0^{-1/\nu} c_\nu^{1/\nu} c_j = c_{j+1}$. Thus, for $0 \leqslant j \leqslant j_1$ we have $d_j \leqslant c_j$. For $j = j_1$, the assumption $ad_{j+1}/d_j < c_1$ implies $d_{j_1+1} < c_1 d_{j_1}/a \leqslant c_0^{-1/\nu} c_\nu^{1/\nu} c_{j_1} = c_{j_1+1}$. Thus, by induction again, we find that $d_j < c_j$ for $j_1 < j \leqslant \nu$, as claimed. Hence, $ad_{j+1}/d_j \leqslant c_1$ cannot hold for all j.

Let j, $0 \leqslant j \leqslant \nu - 1$, be an index for which $ad_{j+1}/d_j > c_1$ holds. Then the interval $[d_j, d_{j+1}]$ is such that $[ad_j/d_j, ad_{j+1}/d_j] \equiv [a, e_j]$ with $e_j > c_1$. Hence, by our preceding work, we have that $\eta[d_j, d_{j+1}] > \eta[a, c_1]$, as desired.

From this it follows that $\max_{0 \leqslant i \leqslant \nu-1} \eta[c_i, c_{i+1}] < \max_{0 \leqslant i \leqslant \nu-1} \eta[d_i, d_{i+1}]$ for each $\mathbf{d} \in \mathcal{P}$ with $\mathbf{d} \neq \mathbf{c}$. To see that this is also true for η replaced with η^l, l a positive integer, one need only observe that N is a strictly pointwise monotone one-sided operator [10]. What this implies is that $N(R)\,(x) > x^{1/n}$ for $x \in [a, b]$ and $R(x) \neq x^{1/n}$, and if $N(R_1)\,(y) > N(R_2)\,(y)$ for some $y \in [a, b]$, then $N^l(R_1)\,(y) > N^l(R_2)\,(y)$ for l a positive integer. From this observation the final result readily follows. □

Before applying this theory to some specific examples, we wish to discuss the problem of finding best starting approximations from $\mathcal{R}_0^1[a, b] \equiv \Pi_1$ for calculating nth roots on $[a, b]$. In this very simple case, it is possible to give analytical formulas for the best relative approximation to $x^{1/n}$ from Π_1 on $[a, b]$ and, therefore, also for the best starting approximation from Π_1 for calculating nth roots on $[a, b]$. See reference [14], where this result has appeared. We shall simply summarize the situation here.

THEOREM 2. *Fix the interval* $[a, b]$, $0 < a < b$. *Then the best* (*linear*) *relative approximation to* $x^{1/n}$ *on* $[a, b]$ *from* $\Pi_1, \tilde{p}(x) = \alpha x + \beta$, *is given by*

$$\alpha = \frac{(b^{1/n} - a^{1/n})(1 - \lambda)}{b - a}, \tag{4}$$

$$\beta = \frac{(ba^{1/n} - ab^{1/n})(1 - \lambda)}{b - a}, \tag{5}$$

where

$$\lambda = \left\| \frac{x^{1/n} - \tilde{p}(x)}{x^{1/n}} \right\| = \min_{p \in \Pi_1} \left\| \frac{x^{1/n} - p(x)}{x^{1/n}} \right\| = \frac{w - 1}{w + 1}, \tag{6}$$

and

$$w = \frac{n}{n - 1} \left(\frac{ba^{1/n} - ab^{1/n}}{b - a} \right) \left(\frac{(n - 1)(b^{1/n} - a^{1/n})}{ba^{1/n} - ab^{1/n}} \right)^{1/n}. \tag{7}$$

Proof. By referring to the proof of Theorem 1, we know that the error curve $E(x) = 1 - \tilde{p}(x)/x^{1/n} = 1 - (\alpha x + \beta)/x^{1/n}$ must have precisely three extreme points, $a, \xi, b, a < \xi < b$, and that $\tilde{p}$ will satisfy the following system (the unknowns are α, β, ξ and λ)

$$\begin{aligned} 1 - a^{-1/n}(\alpha a + \beta) &= \lambda, \\ 1 - \xi^{-1/n}(\alpha \xi + \beta) &= -\lambda, \\ 1 - b^{-1/n}(\alpha b + \beta) &= \lambda, \\ (n - 1)\alpha \xi^{-1/n} - \beta \xi^{-(n+1)/n} &= 0, \end{aligned}$$

where the fourth equation is the derivative of the error curve at ξ set equal to 0. Also, since the best linear relative approximation is unique, we have that there exists one and only one solution to this system. Solving simultaneously for α and β in terms of $1 - \lambda$ in equations 1 and 3 give (4) and (5) of the theorem. Substituting this in equation 4 gives an expression for ξ, and then substituting all these values in equation 2 gives the formula for λ. □

COROLLARY 1. *The best starting approximation from* Π_1 *for calculating nth roots is* $p^*(x) \equiv \gamma \tilde{p}(x)$, *where* $\tilde{p}$ *is defined in Theorem* 3 *and* γ *is given by* (3). □

4. Examples.

In this section, we give specific examples of the above theory for computing square roots, reciprocal square roots using a divide-free Newton iteration and, finally, cube roots. In the first two examples we shall only use best starting approximations from Π_1 and consider what happens when at most two Newton iterations are required. For the cube root case we shall also consider other classes of rational functions for the initialization of the Newton iteration.

TABLE 1

(a, 4a]	subinterval(s)	$p^*(x)$	η	η^2
$(\frac{1}{2}, 2]$	$(\frac{1}{2}, 2]$	.4848608528(x + 1)	$3.96\text{x}10^{-4}$	$7.84\text{x}10^{-8}$
$(\frac{1}{2}, 2]$	$(\frac{1}{2}, 1]$	.5901785321x+.4173192421	$2.79\text{x}10^{-5}$	$3.8\text{x}10^{-10}$
	$(1, 2]$	.4173192421x+.5901785321	$2.79\text{x}10^{-5}$	$3.8\text{x}10^{-10}$
$(\frac{1}{4}, 1]$	$(\frac{1}{4}, 4^{-2/3}]$	.8879377727x+.2796828727	$5.54\text{x}.0^{-6}$	$1\text{x}10^{-11}$
	$(4^{-2/3}, 4^{-1/3}]$	.7047566772x+.3523783386	$5.54\text{x}10^{-6}$	$1\text{x}10^{-11}$
	$(4^{-1/3}, 1]$	.5593657454x+.4439688863	$5.54\text{x}10^{-6}$	$1\text{x}10^{-11}$

A. *Square Roots.* To develop an algorithm for computing square roots based upon the preceding theory, we must first select an interval of application. Any interval of the form $(a, 4a]$, $a > 0$, will do; however, reasonable choices are intervals such as $(1/8, 1/2]$, $(1/4, 1]$ or $(1/2, 2]$. An algorithm for calculating square roots based on $(a, 4a]$ will have the following components. First of all, the algorithm will have a scaling feature. Thus, to find $\sqrt{y}$, $y > 0$, the algorithm will first scale y; that is, it will calculate m, an integer for which $y = 2^{2m}x$ and $x \in (a, 4a]$. Then, if a subdivison of the interval $(a$, 4a] is being used, the algorithm will determine which subinterval contains x and evaluate the appropriate piece of the best piecewise starting approximation. Next, it will compute one or more Newton iterates, $N(h)\ (x) = \frac{1}{2}(h(x) + x/h(x))$, to calculate $\sqrt{x}$ using the above theory to get a best starting approximation on $(a, 4a]$. It will then multiply (shift) this final value by 2^m and return this for the value $\sqrt{y}$. For $n = 2$ the formulas of Theorem 2 and (3) for the interval $[a, b]$ reduce to

$$\lambda = -\left[\frac{b^{1/4} - a^{1/4}}{b^{1/4} + a^{1/4}}\right]^2,$$

$$\alpha = \frac{1 - \lambda}{b^{1/2} + a^{1/2}},$$

$$\beta = a^{1/2}b^{1/2}\alpha,$$

$$\gamma = (1 - \lambda^2)^{-1/2}.$$

In Table 1 we give three examples of this theory for computing square roots using best piecewise linear starting approximations. In this table, the column headed by $p^*(x)$ gives the best piecewise starting approximation corresponding to the subinterval on which it is defined (which appears in the same row and to the right of $p^*(x)$). The (absolute relative) error of approximation after one and two Newton iterations is given in the η and η^2 columns, respectively. All calculations were done on a hand calculator (Texas Instrument SR-56) and all digits occurring at the end of the calculation are given for the coefficients, whereas the values for η and η^2 have been rounded to three places.

As remarked earlier, the third case in Table 1 was used as a guide for constructing a 16-bit microprocessor square root routine [3]. Since the domain of application was to be $X = \{j/2^{16}\}_{j=2^{14}+1}^{2^{16}}$, this example implies that X should be partitioned into $X_1 \cup X_2 \cup X_3$, where $X_1 = \{j/2^{16}\}_{j=2^{14}+1}^{26,007}$, $X_2 = \{j/2^{16}\}_{j=26,008}^{41,284}$, and $X_3 = \{j/2^{16}\}_{41,285}^{2^{16}}$. This partition was then modified in [3] to the partition $X = Y_1 \cup Y_2 \cup Y_3$, where $Y_1 = (1/4, 7/16] \cap X$, $Y_2 = (7/16, 3/4] \cap X$, and $Y_3 = (3/4, 1] \cap X$. One reason for this was that we wanted the coefficients of the x term in each initialization piece to have nonzero bits in at most its three leading bits when written in binary. This allows the product of this coefficient and the inputed x value to be calculated by at most three shifts and two adds, which is a significant improvement over a full multiply in a microprocessor environment. (See [3] for a more complete discussion.)

TABLE 2

(a, 4a]	subinterval(s)	$p^*(x)$	η	η^2
$(\frac{1}{2}, 2]$	$(\frac{1}{2}, 2]$	-.4314166817x+1.509958386	$1.05\text{x}10^{-2}$	$1.64\text{x}10^{-4}$
$(\frac{1}{2}, 2]$	$(\frac{1}{2}, 1]$	-.8100537518x+1.787875129	$7.38\text{x}10^{-4}$	$8.16\text{x}10^{-7}$
	(1, 2]	-.2863972505x+1.264218627	$7.38\text{x}10^{-4}$	8.16x10-7
$(\frac{1}{2}, 2]$	$(\frac{1}{2}, 2^{-2/3}]$	-1.184260206x+2.002810852	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$
	$(2^{-2/3}, 2^{-1/3}]$	-.8373984226x+1.784301621	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$
	$(2^{-1/2}, 1]$	-.5921301032x+1.589632027	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$
	$(1, 2^{1/3}]$	-.4186992113x+1.416201135	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$
	$(2^{1/3}, 2^{2/3}]$	-.2960650516x+1.261691776	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$
	$(2^{2/3}, 2]$	-.2093496057x+1.124039586	$9.35\text{x}10^{-6}$	$1.35\text{x}10^{-10}$

B. *Reciprocal Square Roots–Divide-Free Iteration.* This amounts to applying the above theory with $n = -2$ to obtain approximations for $1/\sqrt{x}$, using an iteration that requires no divides. In this case, $N(h)(x) = h(x)\,[3 - x \cdot h^2(x)]/2$. In developing an algorithm using this iteration, we must again scale numbers as in the square root case. Thus, we shall assume that the scaling will be done with respect to an interval of the form $(a, 4a]$. In Table 2, we shall give the best linear starting approximations for this algorithm when one uses the interval $(1/2, 2]$, subdivides it into two subintervals, and subdivides it into six subintervals.

Now, for this particular iteration on $[a, b]$, the formulas of Theorem 2 and (3) reduce to

$$\lambda = \frac{2(b + a^{1/2}b^{1/2} + a)^{3/2} - 3^{3/2}a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}{2(b + a^{1/2}b^{1/2} + a)^{3/2} + 3^{3/2}a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}, \tag{12}$$

$$\alpha = -\frac{(1 - \lambda)}{a^{1/2}b^{1/2}(b^{1/2} + a^{1/2})}, \tag{13}$$

$$\beta = -(b + a^{1/2}b^{1/2} + a)\alpha, \tag{14}$$

$$\gamma = \left(\frac{3}{3 - \lambda^2}\right)^{1/2}. \tag{15}$$

C. *Cube Roots.* For this particular example, we will also consider some nonlinear best starting approximations. In designing a cube root routine of this sort, one must first select an interval of application. The interval we shall use is $(1/8, 1]$. Thus, the final algorithm for computing $\sqrt[3]{y}$, y real, would have a scaling routine which (i) changes the sign of y if $y < 0$, and also changes the sign (to negative) of the computed cube root of $-y$, prior to returning a final approximation, and (ii) scales y (assume $y > 0$), i.e., compute $y = 2^{3m}x$, where m is an integer and $x \in (1/8, 1]$. Next, the cube root of x is calculated according to the theory presented here, and this result is multiplied (shifted) by 2^m and returned for $\sqrt[3]{|y|}$. Finally, a sign change is performed, if necessary, as noted in (i). In Table 3, for the interval $(1/8, 1]$ and the partition of this interval into three subintervals, we shall give the best starting approximations from R_n^m, $0 \leqslant m, n$; $m + n = k$, $k = 1, 2, 3$, where m and n for fixed k are chosen so that the best starting approximation from $R_{\bar{n}}^{\bar{m}}$, $(\bar{m}, \bar{n}) \neq (m, n)$, $0 \leqslant \bar{m}, \bar{n}$; $\bar{m} + \bar{n} = m + n$ does not give a better relative approximation for $\sqrt[3]{x}$ after one Newton iteration. These best starting approximations were calculated on a CDC-6400 where we found the best relative approximation to $\sqrt[3]{x}$ on the interval in question discretized into equally spaced mesh points with a step size of $h = 1/256$ using [8].

TABLE 3

$(\frac{1}{8}, 1]$	subinterval(s)	$p^*(x)$	η	η^2
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, 1]$	$.6055481056x+.4541610792$	$3.30x10^{-3}$	$1.09x10^{-5}$
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, 1]$	$1.477484521 - \frac{.8414788493}{.7387462419+x}$	$4.23x10^{-5}$	$1.78x10^{-9}$
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, 1]$	$.2437995493x+.8898929185 - \frac{.1698975861}{.2796064148+x}$	$8.44x10^{-7}$	$0.0\ (\sim 10^{-14})$
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, \frac{1}{4}]$	$1.046616906x+.3725069311$	$4.41x10^{-5}$	$1.94x10^{-9}$
	$(\frac{1}{4}, \frac{1}{2}]$	$.6593273358x+.4693293238$	$4.41x10^{-5}$	$1.94x10^{-9}$
	$(\frac{1}{2}, 1]$	$.4153501946x+.5913178943$	$4.41x10^{-5}$	$1.94x10^{-9}$
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, \frac{1}{4}]$	$1.128076154 - \frac{.3016177699}{.3553223735+x}$	$6.50x10^{-8}$	$0.0\ (\sim 10^{-16})$
	$(\frac{1}{4}, \frac{1}{2}]$	$1.421286893 - \frac{.7600291547}{.710644747+x}$	$6.50x10^{-8}$	$0.0\ (\sim 10^{-16})$
	$(\frac{1}{2}, 1]$	$1.790709274 - \frac{1.915153461}{1.421289494+x}$	$6.50x10^{-8}$	$0.0\ (\sim 10^{-16})$
$(\frac{1}{8}, 1]$	$(\frac{1}{8}, \frac{1}{4}]$	$.4190115298x+.6904625373-\frac{.0646502159}{.1412333954+x}$	$1.5x10^{-10}$	$0.0\ (\sim 10^{-20})$
	$(\frac{1}{4}, \frac{1}{2}]$	$.2639607233x+.8699282849-\frac{.1629083358}{.2824667908+x}$	$1.5x10^{-10}$	$0.0\ (\sim 10^{-20})$
	$(\frac{1}{2}, 1]$	$.1662848358x+1.096040958-\frac{.4105032829}{.5649335816+x}$	$1.5x10^{-10}$	$0.0\ (\sim 10^{-20})$

We then multiplied this function by the appropriate γ, given by (3), using the respective λ. Finally, it should be noted that argument reduction can be done relative to intervals other than the form $(a, 8a]$, $a > 0$. Another, frequently used, interval is $(1/2, 1]$. In this case, the post scaling phase will include a multiplication by $2^{1/3}$, $2^{2/3}$ or 1.

Acknowledgement. The authors wish to express their gratitude to the referee for many excellent comments regarding the organization of this paper.

Fachbereich Mathematik
University of Siegen
D-5900 Siegen 21, West Germany

Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523

1. N. I. ACHIESER, *Theory of Approximations*, Ungar, New York, 1956.

2. M. ANDREWS, S. F. McCORMICK & G. D. TAYLOR, "Evaluation of functions on microprocessors: Square root," *Comput. Math. Appl.*, v. 4, 1978, pp. 359–367.

3. M. ANDREWS, S. F. McCORMICK & G. D. TAYLOR, *Evaluation of the Square Root Function on Microprocessors*, Proc. of ACM, Houston, Texas, October 1976, pp. 185–191.

4. T. C. CHEN, *The Automatic Computation of Exponentials, Logarithms, Ratios and Square Roots*, IBM Research Memo, RJ970 (#16884), San Jose, Calif., February 1972.

5. W. J. CODY, "Double-precision square root for the CDC-3600," *Comm. ACM*, v. 7, 1964, pp. 715–718.

6. J. B. GIBSON, "Optimal rational starting approximations," *J. Approximation Theory*, v. 12, 1974, pp. 182–189.

7. E. H. KAUFMAN, JR. & G. D. TAYLOR, "Uniform rational approximation of functions of several variables," *Internat. J. Numer. Methods Engrg.*, v. 9, 1975, pp. 297–323.

8. R. F. KING & D. L. PHILLIPS, "The logarithmic error and Newton's method for the square root," *Comm. ACM*, v. 12, 1969, pp. 87–88.

9. W. JAMES & P. JARRATT, "The generation of square roots on a computer with rapid multiplication compared with division," *Math. Comp.*, v. 19, 1965, pp. 497–502.

10. G. MEINARDUS & G. D. TAYLOR, "Optimal starting approximations for iterative schemes," *J. Approximation Theory*, v. 9, 1970, pp. 1–19.

11. D. G. MOURSUND, "Optimal starting values for the Newton-Raphson calculation of $\sqrt{x}$," *Comm. ACM*, v. 10, 1967, pp. 430–432.

12. D. G. MOURSUND & G. D. TAYLOR, "Optimal starting values for the Newton-Raphson calculation of inverses of certain functions," *SIAM J. Numer. Anal.*, v. 5, 1968, pp. 138–150.

13. I. NINOMIYA, "Best rational starting approximations and improved Newton iteration for the square root," *Math. Comp.*, v. 24, 1970, pp. 391–404.

14. D. L. PHILLIPS, "Generalized logarithmic error and Newton's method for the *m*th root," *Math. Comp.*, . v. 24, 1970, pp. 383–389.

15. P. H. STERBENZ & C. T. FIKE, "Optimal starting approximations for Newton's method," *Math. Comp.*, v. 23, 1969, pp. 313–318.

16. G. D. TAYLOR, "Optimal starting approximations for Newton's method," *J. Approximation Theory*, v. 3, 1970, pp. 156–163.

17. J. S. WALTHER, *A Unified Algorithm for Elementary Functions*, Proc. AFIPS, Spring Joint Computer Conference, v. 38, 1971, pp. 379–385.

18. M. W. WILSON, "Optimal starting approximations for generating square root for slow or no divide," *Comm. ACM*, v. 13, 1970, pp. 559–560.