Optimal Starting Approximations for Generating Square Root for Slow or No Divide

M. Wayne Wilson
IBM Scientific Center, Houston, Texas

On machines with slow or no division, it is preferable to use an iterative scheme for the square root different from the classical Heron scheme. The problem of optimal initial approximants is considered, and some optimal polynomial initial approximations are tabulated.

KEY WORDS AND PHRASES: square root, Newton-Raphson iteration, optimal approximants
CR CATEGORIES: 5.13

Introduction

A number of papers in the past few years [1, 3, 5, 6, 7] have dealt with the problem of obtaining optimal starting approximations for the Heron or classical Newton-Raphson iteration scheme for square root. For machines where the ratio of multiplication speed to division speed is large, or where there is no divide, the "no division" iteration scheme

$$y_{n+1} = y_n\left(\frac{3}{2} - \left(\frac{x}{2}\right)y_n^{\,2}\right), \qquad n = 0, 1, 2, \cdots \quad (1)$$

which converges to $x^{-\frac{1}{2}}$, if it converges, is preferable. Assuming convergence, $x^{\frac{1}{2}}$ is simply

$$x \cdot \lim_{n\to\infty} y_n .$$

In this paper, optimal starting approximations for (1) and some computational procedures are discussed, and tables of optimal polynomial initial approximants are given.

A number of minor points should be made. In the context of slow or no division, rational initial approximants make little sense. The quantity $(x/2)$ required in the iterations can be obtained once, generally without division. Finally, the iteration scheme (1) is the general Newton-Raphson scheme for finding a zero of $f(y) = x - 1/y^2$.

Added in proof: In papers to appear by D. L. Phillips (*Math. of Comput.*) and G. D. Taylor (*J. of Approx. Theory*), it has been shown that the optimal polynomial above is a constant multiple of the best relative polynomial to $X^{-\frac{1}{2}}$.

Optimal Initial Approximations

On the range $[a, b]$, $0 < a < b < \infty$, let $y_0(x)$ be an approximant of some specified form. The relative error of the nth iterate $y_n(x)$ is then

$$R_n(x) = [y_n(x) - x^{-\frac{1}{2}}]/x^{-\frac{1}{2}} = x^{\frac{1}{2}}\cdot y_n(x) - 1. \quad (2)$$

One can then easily show that

$$R_{n+1}(x) = -\tfrac{1}{2}R_n^{\,2}(R_n + 3) \quad (3)$$

(by manipulating (1) and (2)).

Examining the nature of the function $f(t) = -\frac{1}{2}t^2(t + 3)$ shows:

(i) if $-3 \le R_0(x) \le 1$ then $-2 \le R_1(x) \le 0$;

(ii) if $-2 \le R_n(x) \le 0$ then $-2 \le R_{n+1}(x) \le 0$, $n = 0, 1, 2, \cdots$.

Given some method of measuring the error of an approximant, we say that the approximant (of specified form) is a *best fit* if it minimizes the maximum magnitude of the error curve. As pointed out by Moursund [5], it is logical to choose $y_0(x)$ to be the best fit for the relative error of the Lth iterate (where L is the number of iterations we plan to use). Taking $y_0(x)$ as a best absolute or relative fit to $x^{-\frac{1}{2}}$ (see James and Jarratt [2]) does not do this. On the other hand, an approximant $y_0(x)$ is said to be *optimal* if it is a best fit for each iterate $y_1, y_2, \cdots$.

Theorem. If $y_0(x)$ is a best fit for the relative error of the first iterate, and if $0 \ge R_1(x) \ge -2$ or $1 \ge R_0(x) \ge -2$, then the approximant is optimal.

Proof. The function $f(t) = -\frac{1}{2}t^2(3 + t)$ is monotonic on $[-2, 0]$, mapping $[-2, 0]$ onto $[-2, 0]$. Hence, each point of $[a, b]$ giving a maximum for $|R_1(x)|$ gives a maximum for $|R_k(x)|$, $k = 2, 3, \cdots$. Consequently, if $y_0(x)$ is a best relative fit for $R_1(x)$, it follows from the monotonicity and a contradiction type argument that it is the best relative fit for $R_2(x)$, and similarly, for each succeeding iterate. Thus $y_0(x)$ is optimal under the hypothesis.

Note that, like King and Phillips [3], we can define the logarithmic error of each iterate. The analog of their theorem 2 can then be proved for the iteration scheme (1), but unfortunately, the analog of their theorem 1 does not hold.

Computing Optimal Approximants

Expressing $R_1(x)$ in terms of $y_0(x)$, we obtain

$$R_1(x) = [x^{\frac{1}{2}}y_0(x)/2][3 - xy_0^{\,2}(x)] - 1 \quad (4)$$

from which we can create a generalized weight function and apply the methods of Moursund [4, 5, 6].

However, differentiating, we obtain

$$R_1'(x) = \frac{3}{4x^{\frac{1}{2}}}[1 - xy_0^{\,2}(x)][y_0(x) + 2xy_0'(x)].$$

Setting $1 - xy_0^{\,2}(x) = 0$, we find each root of this equation gives a zero of $R_1(x)$. Looking for maxima of $|R_1(x)|$, we

can ignore these. Candidates for maxima of $|R_1(x)|$ are then the roots of the equation

$$y_0 + 2xy_0' = 0 \tag{5}$$

and the two endpoints a, b of the interval. If we let $y_0(x)$ be a polynomial of degree m, (5) has m roots and we have $m + 2$ maxima (hopefully on $[a, b]$) of $R_1(x)$, nicely matching the $m + 2$ "critical points" of the theory (see Moursund [4, 5]).

We used a Remes-type algorithm for finding the coefficients of the optimal polynomials, leveling the critical point errors to four significant figures. Equation (5) and a polynomial root-finder were used to solve for the new set of "critical points" for each iteration. For a given set of critical points, $a = t_0, t_1, \cdots, t_m, t_{m+1} = b$, we found the new coefficients A_j, $j = 0, 1, \cdots, m$, by solving the nonlinear system of equations

$$R_1(t_i) = r, \qquad i = 0, 1, \cdots, m + 1$$

using a gradient descent method on the function

$$g(A_0, A_1, \cdots, A_m, r) = \sum_{i=0}^{m+1} (R_1(t_i) - r)^2.$$

For the initial approximation of the polynomial and the critical points, we used a polynomial of best relative fit to $x^{-\frac{1}{2}}$ on $[a, b]$ on a set of 64 equidistant discrete points. In all cases reported, we had convergence in at most 5 iterations.

For the constant or degree 0 optimal polynomial, we used the formula

$$y_0 = [(3/(a + a^{\frac{1}{2}}b^{\frac{1}{2}} + b)]^{\frac{1}{2}}$$

with $\max |R'|$ given by

$$|[3(3ab)^{\frac{1}{2}}(a^{\frac{1}{2}} + b^{\frac{1}{2}})/[2(a + a^{\frac{1}{2}}b^{\frac{1}{2}} + b)^{\frac{3}{2}}] - 1|.$$

If we let

$$r_n = \max_{x \in [a,b]} |R_n(x)| \tag{6}$$

then, given r_0, or r_1, the maximum relative errors can be obtained by using the recurrence (derived from (3))

$$r_{n+1} = r_n^2(3 - r_n)/2 \tag{7}$$

(which indicates a slightly lower convergence than that obtained from the classical Heron scheme).

Table I gives the coefficients of the optimal linear, quadratic, and cubic polynomials for the ranges $[\frac{1}{16}, 1]$, $[\frac{1}{4}, 1]$, $[\frac{1}{2}, 1]$. Since we need to know accuracy obtained for subroutine design, Table II gives, for the optimal polynomials of Table I, the values $e_k = -\log_2(r_k)$, the number of "binary figures of agreement" of the kth iterate. The same design philosophy and range reduction schemes utilized for the Heron scheme (see Fike [1, Ch. 2]) are applicable to this iteration.

TABLE I. Coefficients of $Y_0(x) = A_0 + A_1x + A_2x^2 + A_3x^3$

Range	Degree	A_0	A_1	A_2	A_3
[1/16, 1]	1	2.9024186	−2.2113666		
	2	3.7946031	−7.0394729	4.4548726	
	3	4.4623652	−13.969731	20.141076	−9.717320
[1/4, 1]	1	2.1301512	−1.2172292		
	2	2.6705780	−3.2850400	1.6384100	
	3	3.1123485	−5.9108558	6.2298915	−2.4384336
[1/2, 1]	1	1.7875799	−0.80991997		
	2	2.2339432	−2.0662030	0.83544569	
	3	2.6053117	−3.6396485	2.9905309	−0.9566732

TABLE II. Number of Binary Figures of Agreement, e_k

Range	Degree	e_0	e_1	e_2	e_3	e_4	e_5	e_6	e_7
[1/16, 1]	1	1.695	2.961	5.400	10.227	19.869	39.152	77.720	154.85
	2	2.663	4.818	9.069	17.554	34.523	68.461	136.337	
	3	3.580	6.616	12.652	24.720	48.855	97.224	193.663	
[1/4, 1]	1	3.522	6.501	12.422	24.258	47.932	95.279	189.972	
	2	5.372	10.171	19.758	38.932	77.279	153.972		
	3	7.148	13.715	26.846	53.106	105.627			
[1/2, 1]	1	5.484	10.331	20.204	39.823	79.061	157.537		
	2	8.293	16.002	31.418	62.252	123.918			
	3	11.028	21.470	42.356	84.127	167.668			

Finally, it should be pointed out that the iteration scheme

$$y_{n+1} = \tfrac{1}{2}y_n[3 - (y_n^2/z)]$$

also found in the literature, which converges to $z^{\frac{1}{2}}$, if it converges, is essentially an "inverse" image to the scheme (1), in that the transformation $z = 1/x$ yields (1). The recurrence scheme (3), for relative error, is the same. Its optimal polynomials are transforms of those given here. And it costs one initial division to utilize!

Received April, 1970

REFERENCES

1. Fike, C. T. *Computer Evaluation of Mathematical Functions.* Prentice-Hall, Englewood Cliffs, N. J., 1968.
2. James, Wendy, and Jarratt, P. The generation of square roots on a computer with rapid multiplication compared with division. *Math. Comput. 19* (1965), 497–500.
3. King, R. F. and Phillips, D. L. The logarithmic error and Newton's method for the square root. *Comm. ACM 12*, 12 (Dec. 1969), 87–88.
4. Moursund, D. G. Chebyshev approximation using a generalized weight function. *SIAM J. Num. Anal. 3* (1966), 435–450.
5. Moursund, D. G. Optimal starting values for Newton-Raphson calculation of $\sqrt{x}$. *CACM 10* (1967), 430–432.
6. Moursund, D. G. Computational aspects of Chebyshev approximation using a generalized weight function *SIAM J. Num. Anal. 5* (1968), 126–137.
7. Sterbenz, P. H., and Fike, C. T. Optimal starting approximations for Newton's method. *Math. Comput. 23* (1969), 313–318.