Choosing Starting Values for Newton-Raphson Computation of Reciprocals, Square-Roots and Square-Root Reciprocals

Peter Kornerup, Jean-Michel Muller

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Choosing Starting Values for Newton-Raphson Computation of Reciprocals, Square-Roots and Square-Root Reciprocals

Peter Kornerup and Jean-Michel Muller

No 4687

Janvier 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4687--FR+ENG

Choosing Starting Values for Newton-Raphson Computation of Reciprocals, Square-Roots and Square-Root Reciprocals

Peter Kornerup and Jean-Michel Muller

Thème 2 — Génie logiciel
et calcul symbolique
Projet Arénaire

Rapport de recherche n°4687 — Janvier 2003 — 13 pages

Abstract: We aim at finding the best possible seed values when computing reciprocals, square-roots and square-root reciprocals in a given interval using Newton-Raphson iterations. A natural choice of the seed value would be the one that best approximates the expected result. It turns out that in most cases, the best seed value can be quite far from this natural choice. When we evaluate a monotone function $f(a)$ in the interval $[a_{min}, a_{max}]$, by building the sequence x_n defined by the Newton-Raphson iteration, the natural choice consists in choosing x_0 equal to the arithmetic mean of the endpoint values. This minimizes the maximum possible distance between x_0 and $f(a)$. And yet, if we perform n iterations, what matters is to minimize the maximum possible distance between x_n and $f(a)$.

Key-words: Computer arithmetic, Newton-Raphson iteration, Division, Square-Root, Square-Root Reciprocal

(Résumé : tsvp)

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN (France)
Téléphone : 04 76 61 52 00 - International : +33 4 76 61 52 00
Télécopie : 04 76 61 52 52 - International : +33 4 76 61 52 52

Sur le choix du point de départ lorsque l'on calcule des inverses, des racines carrées et des inverses de racines carrées par la méthode de Newton-Raphson

Résumé : On cherche à obtenir les meilleurs points de départ possibles lorsque l'on calcule des inverses, des racines carrées, et des inverses de racines carrées par la méthode de Newton-Raphson. Lorsque l'on évalue une fonction monotone $f(a)$ dans l'intervalle $[a_{min}, a_{max}]$, en construisant une suite x_n à l'aide de la méthode de newton-Raphson, un choix naturel est de prendre comme point de départ la moyenne des valeurs de f en a_{min} et a_{max}. Ceci minimise la plus grande distance possible entre x_0 et $f(a)$. Cependant, si on effectue n itérations, ce qui est réellement important est de minimiser la plus grande distance possible entre x_n et $f(a)$

Mots-clé : Arithmétique des ordinateurs, Itération de Newton-Raphson, Division, Racine Carrée, Racine Carrée Inverse

1 Introduction

Newton-Raphson iteration is a well-known and useful technique for finding zeros of functions. It was first introduced by Newton around 1669 [3], to solve polynomial equations (without explicit use of the derivative), and generalized by Raphson a few years later [7]. NR-based division and/or square-root have been implemented on many recent processors [5, 2, 6, 4, 1].

As a matter of fact, the classical "Newton-Raphson" iteration for evaluating square-roots (deduced from the general iteration by looking for the zeros of function $x^2 - a$) goes back to much earlier. It was already used by Heron of Alexandria (this is why it is frequently quoted as "Heron iteration"), and seems to have been known by the Babylonians 2000 years before Heron.

Heron's idea (with modern notations) was the following. Assume that you want to evaluate $\sqrt{a}$, and that you already know some number b_0 that is close to the desired square-root. If b_0^2 is less than a, then b_0 is less than $\sqrt{a}$, therefore $B = a/b_0$ is more than $\sqrt{a}$. On the other hand, if b_0^2 is more than a, then $B = a/b_0$ is less than $\sqrt{a}$. Therefore, we know two approximations to $\sqrt{a}$, namely b_0 and B, and we know that $\sqrt{a}$ is between them. Hence, a natural choice is to try to approximate $\sqrt{a}$ by the average value of b_0 and B, that is:

$$b_1 = \frac{1}{2}\left(b_0 + \frac{a}{b_0}\right).$$

Let us now turn to the modern Newton-Raphson (NR) iteration. Assume we want to compute a root α of some function ϕ. The NR iteration consists in building a sequence

$$x_{n+1} = x_n - \frac{\phi(x_n)}{\phi'(x_n)}. \tag{1}$$

If ϕ has a continuous derivative and if α is a single root (that is, $\phi'(\alpha) \neq 0$), then the sequence converges quadratically to α, provided that x_0 is close enough to α.

The NR iteration is frequently used for evaluating some arithmetic and algebraic functions. For instance,

- by choosing
$$\phi(x) = \frac{1}{x} - a$$
one gets
$$x_{n+1} = x_n(2 - ax_n).$$
This sequence goes to $1/a$: hence it can be used for computing reciprocals;
- by choosing
$$\phi(x) = x^2 - a$$
one gets
$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right).$$
This sequence goes to $\sqrt{a}$.

- by choosing
$$\phi(x) = \frac{1}{x^2} - a$$
one gets
$$x_{n+1} = \frac{x_n}{2}\left(3 - ax_n^2\right).$$
This sequence goes to $1/\sqrt{a}$. It is also frequently used to compute $\sqrt{a}$, obtained by multiplying the final result by a.

In the following, we focus on these 3 iterations. Assume we want to evaluate $f(a) = 1/a$, $\sqrt{a}$ or $1/\sqrt{a}$. To make the iterations converge quickly, we have to make sure that x_0 is close enough to the wanted result. It is also important to make sure that the number of required iterations is a small constant. This is frequently done by using the first, say k, bits of the input value a to address a table of suitable initial values. Hence, for all the input values with the same first k bits (they constitute some interval $[a_{min}, a_{max}]$), the iterations will be started with the same x_0. A natural choice consists in choosing the value of x_0 that minimizes
$$\max_{a\in[a_{min},a_{max}]} |f(a) - x_0|.$$
If f is monotone, this is done by taking x_0 equal to the arithmetic mean
$$\frac{1}{2}\left(f(a_{min}) + f(a_{max})\right).$$
As said above, this minimizes the maximum possible distance between x_0 and $f(a)$. And yet, if we perform n iterations, what really matters is to minimize the maximum possible distance between x_n and $f(a)$. In the following, we develop expressions for starting values for a specific number of iterations. These choices turns out to be much better than the natural choice. In the case of reciprocation, we actually provide optimal choices.

2 Newton-Raphson Reciprocation

As mentioned above, Newton-Raphson iteration for computing the reciprocal of a number a consists in performing the iteration
$$x_{n+1} = x_n(2 - ax_n) \tag{2}$$
In practice, when we wish to compute the reciprocal of a number a that will be assumed to be between 1 and 2, the first k bits of the binary representation of $a-1$ (the "implicit one" being omitted) are used as address bits to find in a table an adequate value of the *seed* x_0. This means that the same x_0 will be used for all values of a in an interval
$$[a_{min}, a_{max}],$$
with $a_{max} - a_{min}$ of the form 2^{-k} in the most frequent cases. Fig. 1 shows that the choice of the starting point can have a huge influence on the final approximation error.

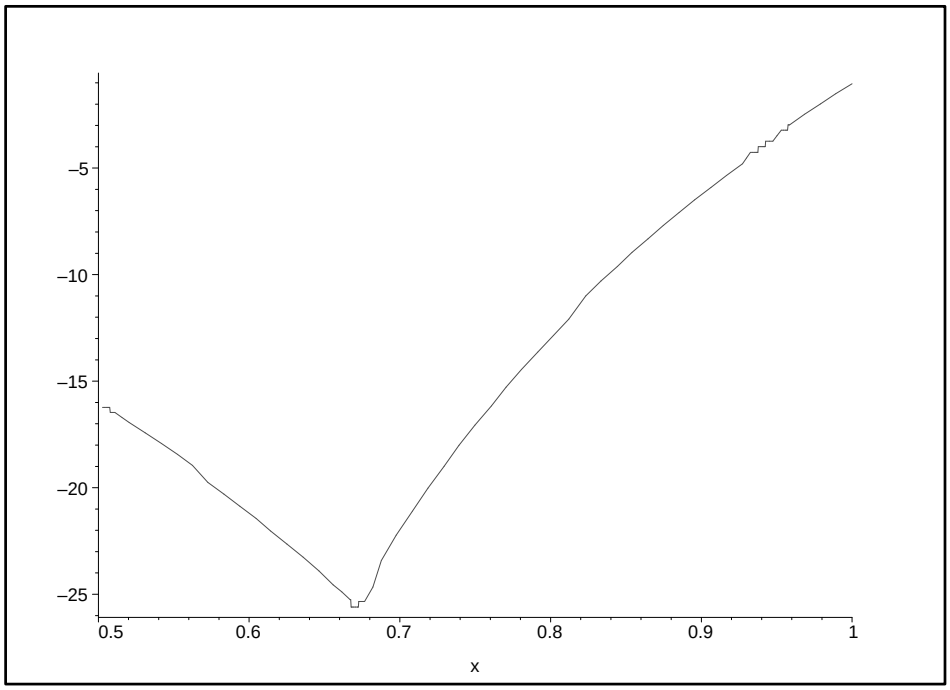


Figure 1: Radix-2 logarithm of the maximum distance (for all a in $[1, 2]$) between iterate x_4 and $1/a$, depending on the choice of x_0 in $[1/2, 1]$.

2.1 Choosing the Best Starting Point

As said in the introduction, it is frequently suggested to choose the arithmetic mean

$$\beta_0 = \frac{1}{2}\left(\frac{1}{a_{min}} + \frac{1}{a_{max}}\right).$$

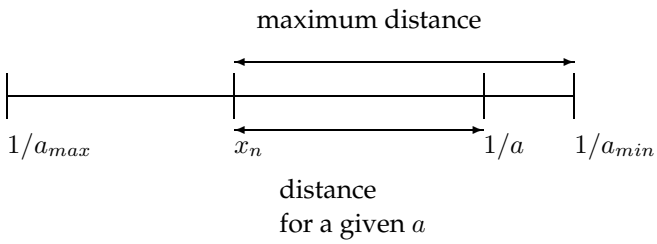
Let us try to minimize the distance between x_n and $1/a$. First, let us compute that distance. From (2), we get

$$x_{n+1} - \frac{1}{a} = 2x_n - ax_n^2 - \frac{1}{a} = -a\left(x_n - \frac{1}{a}\right)^2.$$

Hence, by induction

$$x_n - \frac{1}{a} = -a^{2^n-1}\left(x_0 - \frac{1}{a}\right)^{2^n}. \tag{3}$$

What we now have to find is the value x_0 (between $1/a_{min}$ and $1/a_{max}$) such that the maximum value (for a between a_{min} and a_{max}) of $|x_n - 1/a|$ is as small as possible. For a given x_0 (and, hence, a given x_n), that maximum value is obviously obtained for $a = a_{min}$ or $a = a_{max}$ (see figure below).



That is, the maximum error is either

$$E_1 = a_{min}^{2^n-1}\left(x_0 - \frac{1}{a_{min}}\right)^{2^n}$$

or

$$E_2 = a_{max}^{2^n-1}\left(x_0 - \frac{1}{a_{max}}\right)^{2^n}.$$

This maximum value will be minimized when $E_1 = E_2$ (see figure above). This gives an equation that x_0 must satisfy to be the best starting point for n iterations

$$a_{min}^{2^n-1}\left(x_0 - \frac{1}{a_{min}}\right)^{2^n} = a_{max}^{2^n-1}\left(x_0 - \frac{1}{a_{max}}\right)^{2^n}. \tag{4}$$

To solve this equation define

$$\lambda_n = a_{min}^{(2^n-1)/2^n}$$

and

$$\mu_n = a_{max}^{(2^n-1)/2^n}.$$

From (4) we get

$$\left[\lambda_n x_0 - \frac{\lambda_n}{a_{min}}\right]^{2^n} = \left[\mu_n x_0 - \frac{\mu_n}{a_{max}}\right]^{2^n}.$$

And, since

$$\frac{1}{a_{max}} \leq x_0 \leq \frac{1}{a_{min}}$$

this gives

$$\lambda_n x_0 - \frac{\lambda_n}{a_{min}} = \frac{\mu_n}{a_{max}} - \mu_n x_0.$$

This is now very easily solved, and gives

$$x_0 = \frac{\frac{\mu_n}{a_{max}} + \frac{\lambda_n}{a_{min}}}{\lambda_n + \mu_n}.$$

From that we deduce the following result

Theorem 1 *The maximum possible distance between x_n and $1/a$ is smallest when x_0 is equal to the number*

$$\beta_n = \frac{a_{max}^{(2^n-1)/2^n-1} + a_{min}^{(2^n-1)/2^n-1}}{a_{max}^{(2^n-1)/2^n} + a_{min}^{(2^n-1)/2^n}}. \tag{5}$$

Some values of β_n are of particular interest:

- β_0 is the arithmetic mean of $1/a_{min}$ and $1/a_{max}$: we find again (which is not surprising) the value that minimizes the maximum distance between $1/a$ and x_0;
- β_1 is the geometric mean of $1/a_{min}$ and $1/a_{max}$, that is,

$$\beta_1 = \frac{1}{\sqrt{a_{min}a_{max}}}.$$

- the limit value (when $n \to \infty$) of β_n is

$$\beta_\infty = \frac{2}{a_{min} + a_{max}}$$

that is, the reciprocal of the midpoint of the interval $[a_{min}, a_{max}]$. This shows (and this will be confirmed, in the next section, by the experiments) that this "naive" choice for x_0 is far from being naive, and turns out to be a much better choice than the sophisticated value β_0 that minimizes the maximum distance between $1/a$ and x_0.

2.2 Some experiments

2.2.1 First example: $a_{min} = 1$ and $a_{max} = 2$.

This example corresponds to the direct computations of reciprocals of mantissas of floating-point numbers without any tabulation. By (5) we find the following starting values

$$\begin{cases} \beta_0 &= 3/4 \\ \beta_1 &= 1/\sqrt{2} \\ \beta_2 &= 0.68644\cdots \\ \beta_3 &= 0.67642\cdots \\ \beta_\infty &= 2/3 \end{cases}$$

We get, depending on the choice of x_0, the following approximation errors:

x_0	$\max\|x_1 - 1/a\|$	$\max\|x_2 - 1/a\|$	$\max\|x_3 - 1/a\|$	$\max\|x_4 - 1/a\|$
β_0	0.125	0.031	0.00195	7.63×10^{-6}
β_1	$\mathbf{0.086}$	0.0147	0.000433	3.75×10^{-7}
β_2	0.098	$\mathbf{0.0097}$	0.000187	6.98×10^{-8}
β_3	0.104	0.0108	$\mathbf{0.000120}$	2.88×10^{-8}
β_4	0.107	0.0115	0.000133	$\mathbf{1.83 \times 10^{-8}}$
β_∞	0.110	0.0122	0.000150	2.25×10^{-8}

Choosing β_4 as a starting point for performing 4 NR iterations is approximately 415 times more accurate than choosing β_0. This corresponds to an improvement of more than 8.5 bits of accuracy, for the same number of iterations.

2.2.2 Second example: $a_{min} = 3/2$ and $a_{max} = 7/4$

Of course, when $a_{max} - a_{min}$ decreases, the difference tends to be reduced (since the interval where x_0 can lie shrinks). This shows in the following figures.

x_0	$\max\|x_1 - 1/a\|$	$\max\|x_2 - 1/a\|$	$\max\|x_4 - 1/a\|$
β_0	0.0040	0.000028	3.09×10^{-18}
β_1	$\mathbf{0.0037}$	0.000023	1.65×10^{-18}
β_2	0.0038	$\mathbf{0.000022}$	1.19×10^{-18}
β_3	0.0038	0.000022	1.01×10^{-18}
β_4	0.0039	0.000023	$\mathbf{9.32 \times 10^{-19}}$
β_∞	0.0039	0.000023	9.39×10^{-19}

3 Square-root (direct iteration)

We now consider the NR iteration for square-root

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right). \tag{6}$$

Again, if x_0 is close enough to $\sqrt{a}$, we will have quadratic convergence to $\sqrt{a}$.

3.1 Finding a Good Starting Point

We can easily show

$$x_{n+1} - \sqrt{a} = \frac{1}{2x_n}\left(x_n - \sqrt{a}\right)^2, \tag{7}$$

and from this we derive

$$x_2 - \sqrt{a} = \frac{1}{2x_1}\frac{1}{(2x_0)^2}\left(x_0 - \sqrt{a}\right)^4.$$

The general solution by induction is not easily usable for our purpose. We will simplify it by assuming that x_1, x_2, etc. are close enough to $\sqrt{a}$ so that we can replace them by $\sqrt{a}$ in the term that is before $(x_0 - a)^{2^n}$. This gives

$$x_n - \sqrt{a} \approx \frac{1}{2^{2^n-1}} \left(\frac{1}{x_0}\right)^{2^{n-1}} \left(\frac{1}{\sqrt{a}}\right)^{2^{n-1}-1} \left(x_0 - \sqrt{a}\right)^{2^n}. \tag{8}$$

Again, the best starting point is obtained by saying that the last equation takes the same value for $a = a_{min}$ and for $a = a_{max}$. We solve the obtained equation in a way similar to what we did for reciprocation by defining

$$\lambda_n = \left(\frac{1}{\sqrt{a_{min}}}\right)^{\frac{2^{n-1}-1}{2^n}} \quad \text{and} \quad \mu_n = \left(\frac{1}{\sqrt{a_{max}}}\right)^{\frac{2^{n-1}-1}{2^n}}.$$

We find the best starting point for n iterations of (6):

$$\beta_n = \frac{\mu_n \sqrt{a_{max}} + \lambda_n \sqrt{a_{min}}}{\lambda_n + \mu_n} \tag{9}$$

with the limit value

$$\beta_\infty = \frac{a_{max}^{1/4} + a_{min}^{1/4}}{a_{max}^{-1/4} + a_{min}^{-1/4}}.$$

The above formula for β_n is not valid for $n = 0$. In this case the maximum distance between $\sqrt{a}$ and x_0 is obtained if x_0 is equal to

$$\beta_0 = \frac{1}{2}\left(\sqrt{a_{min}} + \sqrt{a_{max}}\right).$$

An interesting observation is that (9) gives a value for β_1 equal to this β_0.

3.2 Example

With $a_{min} = 1$ and $a_{max} = 2$ we obtain:

x_0	$\max\lvert x_1 - \sqrt{a}\rvert$	$\max\lvert x_2 - \sqrt{a}\rvert$	$\max\lvert x_3 - \sqrt{a}\rvert$	$\max\lvert x_4 - \sqrt{a}\rvert$
β_0	0.018	0.00015	1.20×10^{-8}	7.22×10^{-17}
β_1	**0.018**	0.00015	1.20×10^{-8}	7.22×10^{-17}
β_2	0.019	**0.00013**	8.72×10^{-9}	3.79×10^{-17}
β_3	0.020	0.00014	$\mathbf{7.38 \times 10^{-9}}$	2.72×10^{-17}
β_4	0.021	0.00015	7.89×10^{-9}	$\mathbf{2.29 \times 10^{-17}}$
β_∞	0.021	0.00016	8.60×10^{-9}	2.61×10^{-17}

As for the computations of reciprocals, there is a difference in the final approximation error, depending on the value of the initial starting point. For performing 4 iterations, the choice $x_0 = \beta_4$ gives a final error 3.15 times better than the choice $x_0 = \beta_0$. And yet, the factor is much smaller than for the reciprocal function. This seems due to two facts:

1. By chance, in this case, β_0 happens to be equal to β_1. This means that β_0 is a much better starting point than should be expected in the general case;
2. We have only solved an *approximation* to the problem: Eqn. (8) is not exact. It is quite possible that with the "exact" values of the β_i's the factor would be larger.

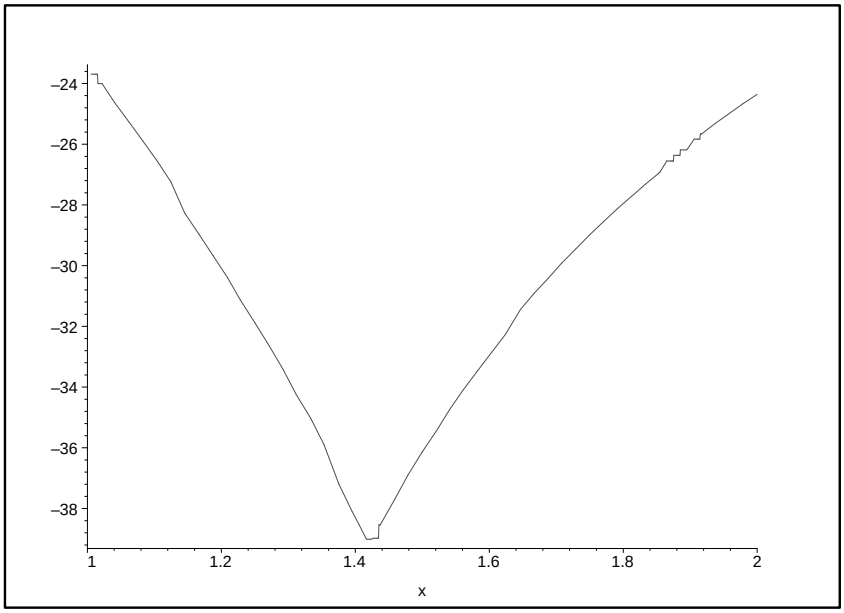


Figure 2: Radix-2 logarithm of the maximum distance (for all a in $[1, 4]$) between iterate x_4 and $\sqrt{a}$, depending on the choice of x_0 in $[1, 2]$.

4 Square-Root Reciprocal

The conventional iteration (6) for square root is not frequently used, since it requires a division at each step, and division is significantly slower than multiplication on almost all systems. Hence one may prefer the following iteration:

$$x_{n+1} = \frac{x_n}{2}\left(3 - a x_n^2\right), \tag{10}$$

converging to $1/\sqrt{a}$. To get $\sqrt{a}$ it suffices to multiply the final result by a.

4.1 Getting a Good Starting Point

From (10), we get

$$x_{n+1} - \frac{1}{\sqrt{a}} = -\left[3\frac{\sqrt{a}}{2} + \frac{a}{2}\left(x_n - \frac{1}{\sqrt{a}}\right)\right] \times \left(x_n - \frac{1}{\sqrt{a}}\right)^2 \tag{11}$$

As for the "direct" square root iteration (6), we cannot get an exact induction formula simple enough to deduce useful information. And yet, if we consider in Eqn (11) the factor:

$$\left[3\frac{\sqrt{a}}{2} + \frac{a}{2}\left(x_n - \frac{1}{\sqrt{a}}\right)\right]$$

it makes sense (since $x_n \to 1/\sqrt{a}$) to assume that as soon as $n \geq 1$, the term

$$\frac{a}{2}\left(x_n - \frac{1}{\sqrt{a}}\right)$$

becomes negligible compared to

$$3\frac{\sqrt{a}}{2}.$$

This remark leads us to the following result

$$x_n - \frac{1}{\sqrt{a}} = -\left(3\frac{\sqrt{a}}{2}\right)^{2^{n-1}-1}\left[3\frac{\sqrt{a}}{2} + \frac{a}{2}\left(x_0 - \frac{1}{\sqrt{a}}\right)\right]^{2^{n-1}}\left(x_0 - \frac{1}{\sqrt{a}}\right)^{2^n} \tag{12}$$

Let us try to find the seed as an x_0 for which the largest value of this expression is minimal when a varies. Equating the values for $a = a_{min}$ and for $a = a_{max}$, defining

$$\lambda = a_{min}^{(2^{n-1}-1)/2^n}$$

and

$$\mu = a_{max}^{(2^{n-1}-1)/2^n},$$

we obtain

$$\left[\lambda a_{min} - \mu a_{max}\right] x_0^3 - 3(\lambda - \mu)x_0 + 2\left[\frac{\lambda}{\sqrt{a_{min}}} - \frac{\mu}{\sqrt{a_{max}}}\right] = 0. \tag{13}$$

Giving a closed formula for the solutions to this 3rd degree polynomial equation is possible, but useless. In practice, one will numerically solve the equation. And yet, we can easily find the limit value (as $n \to \infty$) of the solution. When n goes to infinity, $\lambda \to \sqrt{a_{min}}$ and $\mu \to \sqrt{a_{max}}$, so that (13) becomes

$$\left(a_{min}^{3/2} - a_{max}^{3/2}\right) x_0^3 - 3\left(\sqrt{a_{min}} - \sqrt{a_{max}}\right) x = 0$$

whose only positive solution is

$$\beta_\infty = \sqrt{\frac{3(\sqrt{a_{max}} - \sqrt{a_{min}})}{\left(a_{max}^{3/2} - a_{min}^{3/2}\right)}}.$$

Let us also deal with the case $n = 0$. Because of the approximation, (13) is not valid for $n = 0$. The choice that minimizes the maximum distance between x_0 and $1/\sqrt{a}$ is obviously obtained by choosing x_0 equal to

$$\beta_0 = \frac{1}{2}\left(\frac{1}{\sqrt{a_{min}}} + \frac{1}{\sqrt{a_{max}}}\right) \tag{14}$$

4.2 Example

Let us focus on the computation of $1/\sqrt{a}$ for a in the interval $[1, 4)$. With $a_{min} = 1$ and $a_{max} = 4$, (13) becomes

$$\left(1 - 4 \times 4^{\frac{2^{n-1}-1}{2^n}}\right) x^3 - 3\left(1 - 4^{\frac{2^{n-1}-1}{2^n}}\right) x + 2 - 4^{\frac{2^{n-1}-1}{2^n}} = 0. \tag{15}$$

- From (14) we deduce $\beta_0 = 3/4$.
- in the case $n = 1$, (15) becomes
$$-3\,x^3 + 1 = 0$$
which gives $\beta_1 = 0.6933612744$.
- in the case $n = 2$, (15) becomes
$$\left(1 - 4\sqrt[4]{4}\right) x^3 - 3\left(1 - \sqrt[4]{4}\right) x + 2 - 1/2\,4^{3/4} = 0$$
which gives $\beta_2 = 0.6735060405$.
- in the case $n = 3$, (15) becomes
$$\left(1 - 4 \times 4^{3/8}\right) x^3 - 3\left(1 - 4^{3/8}\right) x + 2 - 1/2\,4^{\frac{7}{8}} = 0$$
which gives $\beta_3 = 0.6639422646$
- our formula for β_∞ gives $\beta_\infty = 0.6546536707$

We have run Newton-Raphson iterations with these starting points. The table below shows the largest obtained approximation errors.

x_0	$\max\lvert x_1 - 1/\sqrt{a}\rvert$	$\max\lvert x_2 - 1/\sqrt{a}\rvert$	$\max\lvert x_3 - 1/\sqrt{a}\rvert$	$\max\lvert x_4 - 1/\sqrt{a}\rvert$
β_0	0.217	0.121	0.0407	0.00484
β_1	**0.127**	0.0436	0.00553	0.0000914
β_2	0.142	**0.0290**	0.00232	0.0000161
β_3	0.150	0.0322	**0.00154**	6.47×10^{-6}
β_4	0.154	0.0339	0.00171	$\mathbf{4.35 \times 10^{-6}}$
β_∞	0.158	0.0356	0.00188	5.29×10^{-6}

In this case we can notice that the choice consisting in minimizing the maximum distance between the seed value x_0 and $1/\sqrt{a}$ (that is, the choice $x_0 = \beta_0$) is extremely poor. The ratio between the obtained approximation errors for 4 iterations, choosing either $x_0 = \beta_0$ or $x_0 = \beta_4$, is close to 1113, corresponding to a difference of more than 10 bits of accuracy.

Conclusion

We have suggested a strategy for getting optimal starting points for Newton-Raphson-based division, and good starting points for approximating square-root and square-root reciprocals. In many cases choosing these values, result in much smaller approximation errors, than using traditional seed values.

References

[1] P. Markstein. *IA-64 and Elementary Functions : Speed and Precision*. Hewlett-Packard Professional Books. Prentice Hall, 2000. ISBN: 0130183482.

[2] P. W. Markstein. Computation of elementary functions on the IBM risc system/6000 processor. *IBM Journal of Research and Development*, 34(1):111–119, January 1990.

[3] I. Newton. *Methodus Fluxionem et Serierum Infinitarum*. 1664-1671.

[4] Stuart F. Oberman. Floating-point division and square root algorithms and implementation in the AMD-k7 microprocessor. In Koren and Kornerup, editors, *Proceedings of the 14th IEEE Symposium on Computer Arithmetic (Adelaide, Australia)*, pages 106–115, Los Alamitos, CA, April 1999. IEEE Computer Society Press.

[5] C. V. Ramamoorthy, J. R. Goodman, and K. H. Kim. Some properties of iterative square-rooting methods using high-speed multiplication. *IEEE Transactions on Computers*, C-21:837–847, 1972. Reprinted in E. E. Swartzlander, *Computer Arithmetic*, Vol. 1, IEEE Computer Society Press Tutorial, Los Alamitos, CA, 1990.

[6] David Rusinoff. A mechanically checked proof of IEEE compliance of a register-transfer-level specification of the AMD-k7 floating-point multiplication, division, and square root instructions. *LMS Journal of Computation and Mathematics*, 1:148–200, 1998.

[7] P. Sebah and X. Gourdon. Newton's method and high order iterations. Technical report, 2001. http://numbers.computation.free.fr/Constants/Algorithms/newton.html.

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399