An Alternate Derivation of the General Square Root Algorithm

Update Version: June 21, 2006

Ron Doerfler (http://www.myreckonings.com)

In Chapter 3 of my book, ***Dead Reckoning: Calculating Without Instruments***, a general algorithm is described for calculating square roots to arbitrary precision. Each decimal group of the root is found by summing the contributions to that decimal group from various iterations of the Newton-Raphson method for approximating the square root. The derivation provided in the book for the algorithm is not intuitive and is rather complicated. In a separate discussion on the topic of checksums for the algorithm, however, John McIntosh (www.urticator.net) introduced a box diagram that lends itself to a much more understandable, if less rigorous, derivation of the algorithm itself. The notation in this derivation is somewhat different from the book to make it more straightforward; these differences will be explained near the end of this paper. Extension of this algorithm to cube roots is discussed in the final section.

Let **a** be the square root we are seeking for a number N less than 1. If N is greater than one, we can shift the decimal point in N to the left (by two places at a time) to get N less than 1, and then shift the decimal point right (one place at a time) in **a** to get the actual square root. We separate **a** into single digits: $a=0.a_1a_2a_3a_4\ldots a_n$. The negative of the subscript provides the power of 10 for each digit, so when we refer to a_2, we know that this is a single-digit number $x10^{-2}$.

We can multiply out $a^2 = (a_1+a_2+a_3+a_4+\ldots)^2$ to get the terms in the boxes in Figure 1. Then $N = a^2$ is the sum of all the boxes. The value of each box (which can be one or two digits long) will have a power of 10 given by the negative of the sum of the two subscripts.

	a_1	a_2	a_3	a_4	a_5	…
a_1	a_1a_1	a_1a_2	a_1a_3	a_1a_4	a_1a_5	…
a_2	a_2a_1	a_2a_2	a_2a_3	a_2a_4	a_2a_5	…
a_3	a_3a_1	a_3a_2	a_3a_3	a_3a_4	a_3a_5	…
a_4	a_4a_1	a_4a_2	a_4a_3	a_4a_4	a_4a_5	…
a_5	a_5a_1	a_5a_2	a_5a_3	a_5a_4	a_5a_5	…
…	…	…	…	…	…	…

Figure 1

Step 1. Find a_1:

We estimate the closest single-digit a_1 for the square root of N. We will denote the remainder $R_1 = N-a_1^2$. This remainder is the sum of all the white boxes in Figure 2, while the a_1a_1 box is colored grey.

Now let's separate N into single digits as $N=0.n_1n_2n_3n_4n_5\ldots$, where again the multiple of 10 is the negative of the subscript for each digit. Since the upper left box is the only one whose subscripts add to 2 (each a_1 being multiplied by 10^{-1}), this is the main box that will affect n_2 in N (if a_1a_1 is a two-digit result, then the upper digit will be n_1; otherwise, $n_1=0$).

	a_1	a_2	a_3	a_4	a_5	…
a_1	a_1a_1	a_1a_2	a_1a_3	a_1a_4	a_1a_5	…
a_2	a_2a_1	a_2a_2	a_2a_3	a_2a_4	a_2a_5	…
a_3	a_3a_1	a_3a_2	a_3a_3	a_3a_4	a_3a_5	…
a_4	a_4a_1	a_4a_2	a_4a_3	a_4a_4	a_4a_5	…
a_5	a_5a_1	a_5a_2	a_5a_3	a_5a_4	a_5a_5	…
…	…	…	…	…	…	…

Figure 2

a_1 = closest one-digit square root of N, with remainder R_1

We now know $a=0.a_1$ and the remainder R_1 that is actually $0.00n_3n_4n_5\ldots$.

Step 2. Find a_2:

For the second step, we subtract from R_1 those terms that are related to the n_3 digit in N, these being the diagonal boxes in yellow in Figure 3 whose subscripts add to 3.

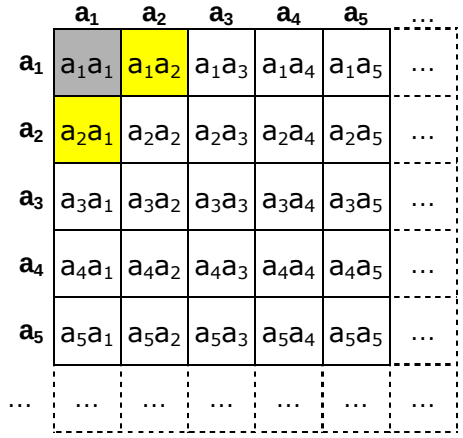

Figure 3

For the new remainder R_2 being the white boxes,

$$R_1 - 2a_1a_2 = R_2$$

Substituting $R_1 = N-a_1^2$ and bringing the 2 up into the numerator to divide by the smaller a_1, we get:

$$a_2 + \frac{R_2/2}{a_1} = \frac{(N - a_1^2)/2}{a_1}$$

We will divide the right-hand side to the second decimal place for a_2, and the rest is assigned to the term $R_2/2a_1$. Note that if $2a_1a_2$ has two digits, a_2 will have two digits, and the upper digit will simply add to n_2.

We can simplify our later work if we realize that $R_2/2$ is actually the remainder after dividing the right side to the second decimal place. We call this r_2 and we simply find the value of the remainder of the division rather than the whole $R_2/2a_1$, and we have:

$$a_2 = \frac{(N - a_1^{2)}/2}{a_1}$$ with remainder r_2 after the second decimal place

We now know $a = 0.a_1a_2$ and the remainder $r_2=R_2/2$.

Step 3. Find a_3:

We subtract from R_2 those terms that are related to the n_4 digit in N, these being the diagonal boxes in purple in Figure 4 whose subscripts add to 4.

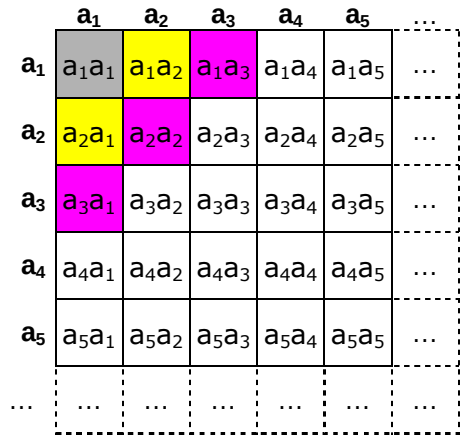

Figure 4

For the new remainder R_3 being the white boxes,

$$R_2 - 2a_1a_3 - a_2^2 = R_3$$

or,

$$a_3 + \frac{R_3/2}{a_1} = \frac{R_2/2 - a_2^2/2}{a_1}$$

and using the division remainders:

$$a_3 = \frac{r_2 - a_2^2/2}{a_1}$$ with remainder r_3 after the third decimal place

We now know $a = 0.a_1a_2a_3$ and the remainder $r_3=R_3/2$.

Step 4. Find a_4:

We subtract from R_3 those terms that are related to the n_5 digit in N, these being the diagonal boxes in blue in Figure 5 whose subscripts add to 5.

For the new remainder R_4 being the white boxes,

$$R_3 - 2a_1a_4 - 2a_2a_3 = R_4$$

or,

$$a_4 + \frac{R_4/2}{a_1} = \frac{R_3/2 - a_2a_3}{a_1}$$

and using the division remainders:

	a_1	a_2	a_3	a_4	a_5	…
a_1	a_1a_1	a_1a_2	a_1a_3	a_1a_4	a_1a_5	…
a_2	a_2a_1	a_2a_2	a_2a_3	a_2a_4	a_2a_5	…
a_3	a_3a_1	a_3a_2	a_3a_3	a_3a_4	a_3a_5	…
a_4	a_4a_1	a_4a_2	a_4a_3	a_4a_4	a_4a_5	…
a_5	a_5a_1	a_5a_2	a_5a_3	a_5a_4	a_5a_5	…
…	…	…	…	…	…	…

Figure 5

$$a_4 = \frac{r_3 - a_2a_3}{a_1}$$ with remainder r_4 after the fourth decimal place

We now know $a = 0.a_1a_2a_3a_4$ and the remainder $r_4=R_4/2$.

Step 5. Find a_5:

We subtract from R_4 those terms that are related to the n_5 digit in N, these being the diagonal boxes in green in Figure 6 whose subscripts add to 6:

For the new remainder R_5 being the white boxes,

$$R_4 - 2a_1a_5 - 2a_2a_4 - a_3^2/2 = R_5$$

or,

$$a_5 + \frac{R_5/2}{a_1} = \frac{R_4/2 - a_2a_4 - a_3^2/2}{a_1}$$

and using the division remainders:

	a_1	a_2	a_3	a_4	a_5	…
a_1	a_1a_1	a_1a_2	a_1a_3	a_1a_4	a_1a_5	…
a_2	a_2a_1	a_2a_2	a_2a_3	a_2a_4	a_2a_5	…
a_3	a_3a_1	a_3a_2	a_3a_3	a_3a_4	a_3a_5	…
a_4	a_4a_1	a_4a_2	a_4a_3	a_4a_4	a_4a_5	…
a_5	a_5a_1	a_5a_2	a_5a_3	a_5a_4	a_5a_5	…
…	…	…	…	…	…	…

Figure 6

$$a_5 = \frac{r_3 - a_2a_4 - a_3^2/2}{a_1}$$ with remainder r_5 after the fifth decimal place

We now know $a = 0.a_1a_2a_3a_4a_5$ and the remainder $r_5=R_5/2$.

Later Steps. Find a_n:

We can see the pattern for finding each a_n by looking at the diagonal slices we subtract in later steps. We take the remainder from the previous division for a_{n-1}, and then subtract pair-wise multiplications of previous digits of **a**, working from the outside digits inward. If there is a single digit left in the middle, we subtract half of its square. Then we divide by a_1 to find a_n and a new remainder r_n.

Relating this Derivation to the Book Notation

The book notation is a little different from that used in this derivation:

- The value of a_n is given as $a_0.b_0b_1b_2b_3b_4\ldots$ in the book, and each variable is two digits long rather than one, since we can multiply and divide by two-digit numbers fairly easily.
- Each step is structured so the two-digit b_n that is calculated is the integer portion of the division, so the previous remainder is multiplied by 100 in each step and the exponents of the b's are not considered. The remainder in the each step is therefore found after the division stops at the decimal point.
- The first remainder $N-a_0^2$ can have many digits if N has many digits, but it would be impractical to mentally carry all these digits through the remainders of all the steps until they become relevant. In fact, these extra digits are simply the later digits of N. Therefore, the first remainder in the algorithm in the book is limited to a two-digit value based on the first 4 digits of N. Then in each later step the next digit pair of N is added to the numerator, but halved because the 2 in the denominator is moved into the numerator in our algorithm.
- The b_n calculated in each step is rounded to the nearest positive or negative integer. If the magnitude of b_{n+1} calculated in the next step exceeds 50, then b_n is adjusted again to obtain a new remainder that will lower the magnitude of b_{n+1}. This is done to limit the effects of large b's in later calculations.

With these differences in mind, the algorithm given in the book is equivalent to our derivation above:

1. Move the decimal point two places at a time in N so the value to the left of the decimal point will have a two-digit root—we will move the decimal point back one place at a time in the final square root to correct for this initial shift. Then take this N and find the nearest square root and the remainder $N-a_0^2$, using only the integer portion of N.
2. Take 100 times the remainder and divide by 2, then add ½ the next two digits of N if they exist. Divide by a_0 to get the next two-digit number b_0 and remainder. The current value is now $a_0.b_0$ with a remainder.
3. Take 100 times the remainder from the previous step and add ½ the next two digits of N if they exist. Consider the values of b following the original estimate a_0. Starting from the

outside two b's, subtract pair-wise multiplications of these as you work inward to the middle (the first time there will be no pairs, just b_0). If there is a b left in the middle after the pairs of b are multiplied and subtracted, subtract the square of that b divided by 2. Divide by a_0 to get a new b (which can be positive or negative) and a new remainder. We can adjust the quotient to get a different remainder if it is needed to force the magnitude of the next b to be less than or equal to 50.

4. Repeat step 3 until you decide to quit or you can't remember any more b's.
5. When you quit, merge the final $a_0.b_0b_1b_2b_3\ldots$ (in which b's can be positive or negative) into a real number and place the decimal point in the correct position.

In addition to the example calculations in the book, there are papers on the Dead Reckoning: Chapter 3 page of http://www.myreckonings.com that provide numerical examples of finding square roots using the book notation.

Can this Derivation be Extended to Cube Roots?

We can use the principles of this derivation to derive an algorithm for cube roots, but the terms are so intermingled that the final algorithm is more complicated than for the square root (in the same way that manually taking a cube root is more complicated than a manual square root). Let's take a look at the first few terms only.

If we separate the cube root **c** of N into single digits: $c=0.c_1c_2c_3c_4\ldots c_n$ and multiply out $c^3 = (c_1+c_2+c_3+c_4+\ldots)^3$, we find that the terms contributing to the lowest three powers of 10 in N are given in Table 1.

Terms	Power of 10
c_1^3	3
$3c_1^2c_2$	4
$3c_1^2c_3 + 3c_1c_2^2$	5

Table 1

We will not stop at single digits and carry remainders as we did in the square root, but rather each value of **c** is found to a few decimal places and is added to the previous result. (This simplifies the division for c_3, so this term is divided by c_1 rather than c_1^2.) Thus, the cube root equals $c_1+c_2+c_3+\ldots$ where each c_n has multiple digits. Following the procedures in the earlier sections, but without remainders, we would find:

c_1 = initial estimate of the cube root of N

$$c_2 = \frac{(N - c_1^3)}{3c_1^2}$$

$$c_3 = -\frac{c_2^2}{c_1}$$

Later **c** values become too difficult to perform mentally, in my opinion.

Let's try an example of the cube root of 119. On pages 99-100 of the book, this cube root was initially estimated as $a_0=5$, then the Newton-Raphson method is used to obtain a better estimate of $a_1=4.92$, and finally the Chebyshev correction for a cube root is subtracted to end with $a_1'=4.91872$ compared to the actual value of 4.9186847…

Here we again start with c_1=5. Then,

$$c_2 = \frac{(119 - 5^3)}{3(5)^2} = -0.08$$

so 5 - 0.08 = 4.92 is our current estimate. The fact that this is the same as the a_1 result in the book that uses the Newton-Raphson method is not too surprising, since that method was the origin of the general square root derivation in the book.

Now,

$$c_3 = -\frac{c_2^2}{c_1} = (-0.08)^2/5 = -0.00128$$

so 4.92 – 0.00128 = 4.91872 is our current estimate, which exactly matches the result a_1' obtained after the Chebyshev correction in the book. This turns out to be no accident—the complicated Chebyshev correction in the book is algebraically equivalent to $(a_1-a_0)^2/a_0$ in the book's notation, where (a_1-a_0) is the difference in the first two estimates, which is c_2 here.

So the general cube root algorithm taken to a practical number of steps is no better than using the Newton-Raphson method with the Chebyshev correction. However, the formula for the Chebyshev correction that is given in the book is rather complicated, so thinking in terms of these formulas for **c** can be an advantage.