Parabola Volume 60, Issue 3 (2024)

A modified method for calculating square roots

Nishant Iyengar[1] and **Anju Iyengar**[2]

1 Introduction

The concept of square root is introduced to students at the middle school level. Square roots of non-negative real numbers, which are not perfect squares, are often irrational, and methods that calculate non-negative square roots of non-perfect positive numbers give the best rational approximations to the square roots.

However, most of the methods to accurately calculate square roots use concepts from advanced high school and undergraduate mathematics. The present authors describe such methods as *top-down methods* since they deploy concepts taught in higher classes for problems introduced in lower classes. The other methods may then be classified as bottom-up methods: these employ concepts at lower or equivalent levels to tackle problems at higher or equivalent levels. The bottom-up methods are mainly pedagogical in nature and are either inaccurate in practice or too slow and incompatible to be of use to advanced computing devices.

The authors of this paper describe a bottom-up method which computes the (positive) square roots of non-negative real numbers. This method employs the modest tools of addition, multiplication and division, which are taught in primary school, to calculate the square roots with extreme accuracy, with the efficiency and computational speed of top-down methods and the ease of use of bottom-up methods. The method could revolutionise the pedagogical approach to square roots.

2 A previous method

In a previous paper [1], the authors proposed a simple interval-weighted denominator method for the rational approximation of square roots of positive real numbers. A quick recap of the method in [1] is given below.

Let a be a positive number. To find a square root approximation of a, the method first requires the knowledge of the perfect squares 1, 4, 9 and so on up to the first perfect square that is larger than a. This is a simple matter to find, for instance by tables or quick multiplications. For instance, there are 31 perfect squares between 1 and 1000:

$$1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256,$$
$$289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900, 961\,.$$

[1]Nishant Iyengar currently resides in New Delhi, India.
[2]Anju Iyengar currently resides in New Delhi, India.

The steps of the method are as follows:

Step a) Determine the perfect squares s^2 and S^2 such that $s^2 \leq a \leq S^2$.

Step b) Calculate $N = a - s^2,\ D = S^2 - s^2$ and $A = N/D$.

Step c) Calculate the rational approximation

$$R = s + \frac{N}{(D-1)+A} \approx \sqrt{a}\,. \tag{1}$$

3 A new method

A new method is proposed below to calculate a rational approximation $R \approx \sqrt{a}$. It is modified from the method in the previous section by iterating the third step:

Step a) Determine the perfect squares s^2 and S^2 such that $s^2 \leq a \leq S^2$.

Step b) Calculate $N = a - s^2,\ D = S^2 - s^2$ and $A = N/D$.

Step c) Iteratively calculate the rational approximation

$$R_n = s + P_n \tag{2}$$

where

$$\begin{aligned} P_1 &= \frac{N}{D-1+A} \qquad \text{and} \\ P_n &= \frac{N}{D-1+P_{n-1}} \quad \text{for} \quad n \geq 2\,. \end{aligned} \tag{3}$$

Iterate Step c) to get the desired accuracy of approximation $R_n \approx \sqrt{a}$.

It is not difficult to prove that R_n converges to $\sqrt{a}$ as n grows large.

4 Examples

Example 1. Suppose that we wish to numerically calculate the square root of $a = 7$. To nine decimal places, the actual value, the approximation R given in Section 2 and the 9th approximation R_9 given in Section 3 are as follows:

$$\begin{aligned} \sqrt{7} &= 2.645751311 \\ R &= 2.652173913 \\ R_9 &= 2.645751311 \end{aligned}$$

Both approximation methods are good but the new method is clearly best.

Example 2. Suppose that we wish to numerically calculate the square root of $a = 70.28$. To nine decimal places, the actual value, the approximation R given in Section 2 and the 5th approximation R_5 given in Section 3 are as follows:

$$\sqrt{70.28} = 8.383316766$$
$$R = 8.383642374$$
$$R_9 = 8.383316766$$

Again, both approximation methods are good but the new method is best, and achieves the accurate value up to nine decimals after only 5 iterations.

5 Results and Discussion

As Examples 1 and 2 demonstrate, our method is very effective. The results are comparable to those obtained from Newton's Method[3] which sets the gold standard in square-root calculation. Unlike that method, however, our new method does not require any knowledge of calculus. Instead, it is based on the iterative techniques that we are so often taught in mathematics and chemistry [4].

Table 5 presents below some of the remarkable results achieved by this simple yet effective method. From the table, it is clear that our iterated method achieves remarkable results with roughly 10 iterations for small single digit numbers which was especially a point of inaccuracy for our previous method. For larger two-digit numbers and higher, square roots can be achieved to accuracy of 12 decimal places in only 4-5 iterations which implies that our method converges faster as the numbers a grow.

a	R	R_n	(n)	Newton's Method	(n)
2	1.428571428	1.414213562	(9)	1.414213562	(4)
7	2.652173913	2.645751311	(9)	2.645751311	(4)
8	2.833333333	2.828427125	(9)	2.828427125	(4)
10	3.162790698	3.162277660	(5)	3.162277660	(4)
13	3.608695652	3.605551275	(5)	3.605551275	(4)
7.5	2.744680851	2.738612788	(11)	2.738612788	(4)
13.5	3.677419355	3.674234614	(10)	3.674234614	(4)
70.28	8.383642375	8.383316766	(5)	8.383316766	(4)
200	14.142156862	14.14213562373	(4)	14.14213562373	(4)

Table 1: A comparison of the square roots of a namely R, R_n and those obtained from Newton's method, and of the number of iterations n required to achieve accuracy up to the number of decimals shown. The numbers (n) in columns 4 and 6 indicate the number of iterations used.

To take advantage of this faster convergence, we can scale up small numbers a, calculate R_n quickly, and then scale down again. For instance, to calculate $R_n \approx \sqrt{2}$, we

[3]For more details about this method, see Subsubsection 6.

can express $\sqrt{2}$ as $\sqrt{200}/10$. As Table 5 shows, it only requires 4 iterations to calculate the value $\sqrt{200} \approx 14.14213562373$, a result that is accurate up to 12 decimal places at the computational expense of just one additional step of division by 10 at the end.

It can be be seen that we have used the middle-school technique of multiplying numerator and denominator keeping the fraction unchanged and have achieved an accuracy of three additional decimal places in 5 fewer steps than the 9 steps required to calculate $\sqrt{2} \approx 1.414213562$.

This is a remarkable result that is unattainable by the current bottom-up methods and, as we shall see later, is at par with top-down approaches but with much less computational cost. Therefore, this method may have revolutionary implications for pedagogy as well as practical applications over the current methods.

6 Existing methods

A brief overview of the most common existing methods for calculating square roots is now given. These can be sorted into two categories: the top-down methods and the bottom-up methods. These methods will be compared with our method.

Top-down methods

These methods define a function as $f(x) = x^2 - a$, where a is the number whose square root is to be found. By solving the equation $f(x) = 0$, the square root $x = \sqrt{a}$ is found. Such methods involve a knowledge of calculus, a concept not generally taught in middle school. Further, these methods involve a quick build-up of huge numbers in the numerator and denominator, which means that they are best performed using advanced computing methods requiring the knowledge of high-level computing languages, further adding to the layers of complexity. Because most of these methods start with an initial guess or two initial guesses, convergence is needed to be proven. The higher the convergence, the faster these methods provide more accurate approximations to the actual roots. Determining accurate enough initial estimates is a trivial matter with the methods available. Three of these methods, namely Newton's method, the Secant Method and the Bisection Method, are discussed below and compared with our method.

Newton's Method

Newton's Method [5], or the Newton-Raphson Method, is a very quick and effective method for solving equations $f(x) = 0$ for differentiable functions f. Applied to the problem of finding square roots, it sets the gold standard. The method starts with a differentiable function $f(x)$, its derivative $f'(x) = 2x$ and an initial guess x_0. For finding the square root of a, the function and its derivative are $f(x) = x^2 - a$ and $f'(x) = 2x$, respectively. Guess an approximation x_0 to the solution $\bar{x}$ (here, $\bar{x} = \sqrt{a}$) such that $f'(x_0) \neq 0$. The tangent line to f in x_0 is $y = f'(x_0)x + f(x_0)$, and this line intersects the x-axis in the point

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}\,. \tag{4}$$

Usually, x_1 is an improved guess over x_0. Repeating this process generates a sequence of hopefully converging approximate solutions $x_n \to \bar{x}$:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}\,.$$

For the purpose of calculating $\bar{x} = \sqrt{a}$, the iterative calculations above simply become

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - a}{2x_{n-1}}\,. \tag{5}$$

The convergence of the method is quadratic, which means that the number of correct digits roughly doubles at every step.

A comparison of Newton's Method with our method is given in Table 5. While Newton's Method converges faster, our method allows a significant increase of speed by the trick of scaling up, calculating and scaling down, as the example of calculating $\sqrt{2}$ as $\sqrt{200}/10$ demonstrated. Moreover, Newton's method requires more calculation operations than those in Step c) of our method; see Equation (3).

The Secant Method

The Secant Method [5] deploys a succession of roots of secant lines to reach the approximation of the solution to $f(x) = 0$ for a differentiable function f, here $f(x) = x^2 - a$. The Secant Method is a quasi-Newton method which assumes that

$$f'(x_{n-1}) = \lim_{\epsilon \to 0} \frac{f(x_{n-1}) - f(x_{n-1} - \epsilon)}{\epsilon} \approx \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}}\,.$$

Plugging this into Equation (4) and simplifying, we get:

$$x_n = \frac{x_{n-2} f(x_{n-1}) - x_{n-1} f(x_{n-2})}{f(x_{n-1}) - f(x_{n-2})} = \frac{x_{n-1} x_{n-2} + a}{x_{n-1} + x_{n-2}}\,. \tag{6}$$

Two initial values x_0 and x_1 are required and the points $(x_0, f(x_0))$ and $(x_1, f(x_1))$ form a secant to the curve $f(x) = x^2 - a$. The point x_2 calculated from Equation (6) above is then used to calculate $f(x_2)$. Then, $(x_1, f(x_1)$ and $(x_2, f(x_2))$ are used to calculate x_3, and so on. The convergence of the Secant Method is slower than the Newton method.

A brief comparison between our method and the Secant Method is given in Table 2. As the table shows, the two methods are comparable in efficiency, when measured by number of iterations. However, our method requires fewer calculations in Equation (3) than in Equation (6) of the Secant Method.

a	R_n	(n)	Secant Method	(n)
2	1.414213562	(9)	1.414213562	(5)
200	14.14213562373	(4)	14.14213562373	(4)
350	18.70828693387	(4)	18.70828693387	(4)
800	28.28427124760	(4)	28.28427124760	(4)

Table 2: A comparison of the square roots obtained from our method (R_n) and the Secant Method. The numbers in brackets indicate the number of iterations used.

For small numbers, the Secant Method is better; for instance it gives the approximation $\sqrt{2} \approx 1.414213562$ in 5 iterations by choosing $x_0 = 1$ and $x_1 = 2$. In comparison, our method reaches this result in 9 iterations. However, as discussed above, the simple step of expressing $\sqrt{2}$ as $\sqrt{200}/10$ leads to an accuracy of 12 decimal places in just 4 iterations. Yet again, the single additional step of division by 10 at the last step makes our method computationally much faster. The same trick may also be applied to the Secant Method and a similar result is obtained in 4 iterations. Thus, for "large enough" two-digit and larger numbers, such as 200, our method is at par with the Secant Method.

Bisection Method

According to the Bisection Method [5], one first guesses an interval $[p, q]$ that might contain a solution to $f(x) = 0$ where $f(x)$ is a continuous function on $[p, q]$; here, $f(x) = x^2 - a$. The numbers p and q must be guessed such that $f(p)$ and $f(q)$ have opposite signs. The method is to bisect this interval to find the midpoint $r = (p + q)/2$; then to calculate $f(r)$ and, finally, to replacing either p or q by r so as to get either $[r, q]$ or $[p, r]$. This process is repeated till $f(r)$ is sufficiently small, depending on the required accuracy. The method suffers from the drawback that it is slow. As an example, it yields the value $\sqrt{5} \approx 2.236068$ after 20 iterations by choosing $p = 2$ and $q = 2.5$. Our method gives the value $\sqrt{5} \approx 2.236068$ after just 5 iterations and is thus superior computationally to the Bisection Method.

Bottom-up methods

Three of bottom-up methods are now presented. The first, generally introduced in Middle-School at the same time as the concept of square roots are introduced is the Long-Division Method of square root calculation, also called *Digit-by-Digit calculation*. The second is the Continued Fraction Expansion Method. Though the deeper concepts in both these methods are subsequently taught in higher classes, educators have ingeniously adapted for square root calculation in middle school. Finally, a discussion is given regarding two recent recursion-based algorithms from the literature which are more or less bottom-up methods in their application.

The Long-Division Method

This method is based on the Binomial Theorem, a staple of high-school mathematics class but well-adapted to middle school where it is packaged analogously to the Long-Form Division Method. This is a method where we solve $(x + y)^2 = x^2 + 2xy + y^2$ in reverse. The Long-Division Method is a slight variation of the Division Method introduced in primary school and follows the simple principle of

$$Dividend = Divisor \times Quotient + Remainder \tag{7}$$

The Dividend in this case is the number a whose square root is to be found, and the quotient is the square root $\sqrt{a}$. Just as the case with the above mentioned square root finding methods, this method first determines the most significant digit of the square root and then the other digits in the decreasing order of significance. Digits in a are paired, starting from right to left, i.e., from least significant to most significant, so that the regular division process may operate from the usual left to right and determines the most significant digits of the square first.

Briefly, the number a is written as the sum of k positive numbers $a_1, a_2, \ldots, a_k$ that indicate the respective digits of the decimal expansion of a. Then

$$a = (a_1 + a_2 + \cdots + a_k)^2\,. \tag{8}$$

By expanding Equation (8) and factoring common terms, we get

$$(a_1+a_2+\cdots+a_k)^2 = a_1^2+(2a_1+a_2)a_2+(2(a_1+a_2)+a_3)a_3+\cdots+\left(2\left(\sum_{i=1}^{k-1} a_i\right)+a_k\right)a_k\,. \tag{9}$$

The approximate square root of a in this method is $\sum_{i=1}^{k-1} a_i$.

Briefly the steps of this method are as follows:

Step a) Starting from the left-most side of the right-hand side of Equation (9), determine a_1, the most significant part of the root. Let M be the left-most digits (or digit) of a; then $a_1^2 \leq M$.

Step b) The remainder from Step a) is placed, and affixed to its right are the next two digits of a, giving the number P. From Equation (9), a_1 is multiplied by 20. Next, a_2 is chosen such that $(20a_1 + a_2)a_2 \leq P$.

These steps are repeated till a desired accuracy is achieved. This method is tedious and expends huge computational cost since it gives only one significant digit for each step.

The Continued Fraction Expansion Method

This bottom-up method utilises the fact that square roots of integers have periodic and repeating expansions. First guess a value x for $\sqrt{a}$. Then $a = x^2 + r$ for some r, and so $a - x^2 = r$. Therefore,

$$(\sqrt{a} + x)(\sqrt{a} - x) = r\,.$$

We can thus express $\sqrt{a}$ as a continued fraction:

$$\sqrt{a} = x + \frac{r}{x + \sqrt{a}} = x + \frac{r}{x + x + \frac{r}{x+\sqrt{a}}} = \cdots = x + \frac{r}{2x + \frac{r}{2x+\frac{r}{2x+\frac{r}{2x+\cdots}}}}\,. \qquad (10)$$

The method proceeds best when the denominator is the largest and r is as small as possible. Therefore, it is important to guess x as accurately as possible.

This method is slower than ours. For instance, it yields the value $\sqrt{2} \approx 1.4142$, accurate to 4 decimal places after 7 steps, whereas our method yields the value $\sqrt{2} \approx 1.414214$ which is accurate to 6 decimal places after 7 iterations. Similarly, the method yields $\sqrt{3} \approx 1.732$, accurate to 3 decimal places after 7 steps, whereas our method yields $\sqrt{3} \approx 1.73206$ which is accurate up to 5 decimal places after 7 iterations.

This method is also difficult to scale-up computationally. Further, our method allows the scaling-up trick for small numbers a, which this method does not.

Other recursion-based algorithms

Our method is compared to two recursion-based iterative algorithms in [4,5]. Series and recursion-based methods are introduced in high school; however, the tools used for the same are developed in primary to middle schools, thus categorizing them as bottom-up methods. The first method is an extension of the well-known Theon's Ladder (MTL) to calculate square root [4]. It may be mentioned here that the Theon's Ladder discussed in [4] is exactly the same as the Continued Fraction Expansion Method for $\sqrt{2}$ and is merely a different representation.

The other is based on Eisenberg's Algorithm [5]. Table 3 shows a comparison of a few numbers by our method as well as the two algorithms mentioned here. A column with the values found by Newton's Method is also included for comparison.

a	R_n	MTL	Eisenberg	Newton
2	1.414213562 (9)	1.414213625 (9)	1.414213552 (9)	1.414213562
3	1.732050808 (9)	1.732057416 (9)	1.732272069 (9)	1.732050808
5	2.236067978 (9)	2.243979058 (9)	2.235621521 (9)	2.236067978

Table 3: A comparison of the square roots obtained from eq. (2), Modified Theon's Ladder and the Eisenberg algorithm. Numbers in brackets denote the number of iterations. The standard Newton's method are included for comparison without mentioning the number of iterations

As can be seen from Table 3, our method is superior to the Modified Theon's Ladder and Eisenberg's Algorithm. Also, these two algorithms involve elaborate construction and large numbers in the numerator and denominator as the number of iterations increase. Further, the divergence from the actual square roots given by Newton's Method gets larger and larger as the numbers increase. Thus, the trick that allowed us to calculate $\sqrt{2}$ as $\sqrt{200}/10$ is clearly not an option for either of these methods. On the other hand, our method predicts very accurate values for larger numbers equally well and with around only half the number of iterations as for single-digit numbers.

7 Conclusions

In conclusion, the authors propose a new method for the extremely accurate computation of non-negative square roots of positive real numbers with accuracies at par with the standard methods such as the Newton's Method or the Secant Method at a fraction of the computation cost. The remarkable iterative modified interval-weighted denominator method may be adapted into school curricula and computing devices as well and has great implications for both educational research and industry.

Acknowledgements

I thank Dr. Thomas Britz for his guidance for the content and writing process and the calm way in which he guided the authors despite our constant queries. Without his support and patience, this publication would not have seen the light of the day.

References

[1] A. Iyengar, A simple and accurate method to calculate square roots and cube roots for chemistry students without calculator, Research Square, last accessed on 2024-12-29.

[2] S.G. Johnson, Square roots via Newton's Method, MIT Course 18.335, last accessed on 2024-12-29.

[3] W. Rudin, *Principles of Mathematical Analysis*, 3rd Edition, McGraw-Hill, New York, 1976.

[4] D.A. McQuarrie, *Quantum Chemistry*, University Science Books, 2003.

[5] R.L. Burden and J.D. Faires, *Numerical Analysis*, 9th Edition, Brooks/Cole ISE, 2012.

[6] S. Giberson and T.J. Osler, Extending Theon's Ladder to any square root, *The College Mathematics Journal* **35** (2004), 222–226.

[7] T. Eisenberg, On an unknown algorithm for computing square roots, *International Journal of Mathematical Education in Science and Technology* **34** (2003), 153–158.