Proceedings of the 3rd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/2025.CH23395

Comparison of Three Square Root Algorithms: Theory and Applications

Chenxi Yu

Merchiston International School, Shenzhen, China
chenxi.yu@merchiston.cn

Abstract: This paper examines three different well-known fast square root methods in computer mathematics: Newton's method, binary search method, and inverse square root method. The main goal is to resolve the differences in performance between various methods, especially with regard to their accuracy and convergence properties in numerical applications. The author thoroughly assesses each algorithm's performance across a number of iterations using a strict methodology for numerical analysis. According to the findings, Newton's approach initially performs less accurately than the binary search and inverse square root approaches, but by the fourth and fifth iterations, it has much outperformed the others in terms of convergence speed. This demonstrates how Newton's approach becomes amazingly economical as the estimate gets better, even if it still requires a strong beginning guess. On the other hand, the inverse square root and binary search techniques works well for producing preliminary estimations. The results of this study are important because they help researchers choose the right algorithms for square root computations and offer insights that can bring improvements to computational efficiency in a variety of real-world applications.

Keywords: Square Root Algorithms, Newton's Method, Algorithm Optimization.

1. Introduction

In many fields, such as computer science, engineering, and applied mathematics, the research of quick square root algorithms is essential. Numerous algorithms essential to numerical techniques, graphics rendering, and machine learning applications rely on square root computations. The efficiency of these computations has a significant impact on computing performance and time taken, especially when processing data in real time. Researchers are investigating optimization techniques that can improve the speed and precision of square root computations as a result of the growing need for faster, more effective algorithms brought about by technological advancements.

It is crucial to compare various square root algorithms, especially in view of current developments in numerical analysis. Based on context and processing needs, research shows that approaches like Newton's technique, binary search, and the quick inverse square root method each have unique benefits and drawbacks. To guarantee thorough comprehension and efficient use of these algorithms in real-world scenarios, more research is required. For example, Newton's method is praised for its quick convergence under ideal initial conditions, but it may falter when far from a root1. Additionally, the binary search approach is a dependable substitute for initial estimations since it provides consistent performance without the need for derivatives, even in applications with variable input

DOI: 10.54254/2753-8818/2025.CH23395

characteristics2. By exploring these algorithms, this study seeks to close knowledge gaps and offer insightful information for upcoming algorithm design advancements.

The paper is structured as follows: the technique is described in the next section, with particular attention to the comparative study of the selected algorithm using large numerical experiments. The outcomes of people's evaluations are then presented and discussed, with an emphasis on the advantages and drawbacks of each method. Finally, one can bring a practical recommendation for algorithm selection customized to specific use cases based on the findings gathered from the studies, so adding to the broader mathematical community's effort to optimize computing algorithms.

2. Methodology

Fast square root algorithms have several applications in fields including computer science, engineering, and mathematics, making their study crucial. Many algorithms, particularly those used in machine learning, image processing, and numerical approaches, are based on square root computations. Understanding the efficacy of different algorithms, such as Newton's method, binary search, and improved estimation techniques, helps accelerate computation and provide a path for future developments in algorithm design. Newton's method's rapid convergence properties make it especially useful for locating the roots of real-valued functions. However, its usefulness in complex circumstances may be limited due to its reliance on derivative calculations.

The binary search strategy for square roots, on the other hand, is more straightforward and reliable due to its lack of derivatives. It guarantees consistent performance across different inputs, while having a slower rate of convergence than Newton's approach. Additionally, because of its unique bit manipulation capabilities, the fast inverse square root method has become widely used in computer graphics, frequently beating more conventional methods in terms of efficiency. Numerous researches have examined how well these techniques work in various settings, pointing out both their benefits and drawbacks. People can learn more about each algorithm's performance in real-world applications thanks to this literature.

The function in each data set is iterated five times, and the error at each iteration is calculated. Supposed that the equation is

$$f(x) = x^2 - S = 0 \tag{1}$$

According to Newton method, its root can be calculated as

$$x_{k+1} = x_k - \frac{f_{(x_k)}}{f'_{(x_k)}} = \frac{1}{2}\left(x_k + \frac{S}{x_k}\right) \tag{2}$$

Here, the Percentage Error is given by $Percentage\ Error = |\frac{x_k - \sqrt{S}}{\sqrt{S}}| \times 100\%$. In the Binary Search Method, one has

$$Mid = \frac{x_{low} + x_{high}}{2} \tag{3}$$

If $mid^2 < S^2$,one can update the lower bound's value by $low = mid$. If $mid^2 > S^2$,one can update the higher bound's value by $high = mid$. Specially, if $mid = S$, then this is the real solution and one accidentally finds the answer.

Finally, for the Inverse Square Root Method, it is calculated as

$$y = \frac{1}{\sqrt{x}} = \frac{1}{y^2} \times x \tag{4}$$

DOI: 10.54254/2753-8818/2025.CH23395

One then seeks for using Newton's iteration to find solution $f(y) = 0$, i.e.,

$$y_{k+1} = y_k - \frac{f(y_k)}{f^{'}(y_k)}. \quad (5)$$

According to $f'(y)$, one can find:

$$y_{k+1} = y_k + \frac{y_k^3 - 1/x}{2y_k^2} \quad (6)$$

3. Results and applications

3.1. Results

Newton's approach exhibits a very quick rate of convergence in this graphic. Newton's method is more efficient than the other two since it can rapidly approach the correct solution with the same number of iterations. When it comes to solving particular problems, the inverse square root method's performance could be limited because of its lower error reduction, see Figure 1. Although the error change trend is comparatively constant and the binary search method has the worst convergence, the error reduction is not as substantial as it is for the other two approaches. When compared to Newton's method and the inverse square root approach, the binary search method clearly demonstrates a disadvantage as it is unable to considerably reduce the error after several repetitions. Thus, based on the results, Newton's method is appropriate for optimization problems requiring high accuracy and not only converges quickly but also performs best while achieving minimal errors.

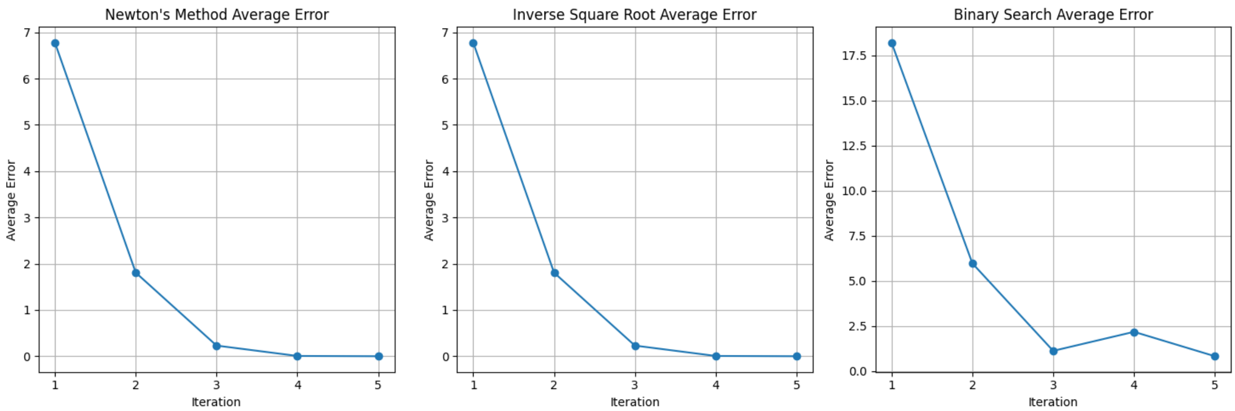

Figure 1: Average errors of different methods

3.2. Example 1: numerical analysis in scientific computing

When it comes to solving complicated equations that come up in physics and engineering simulations, Newton's approach has shown to be quite helpful in scientific computing. According to the research, Newton's approach performs well in subsequent iterations and quickly converges to the correct number, even if it may not be the best option for first estimates. Because of this feature, it is very helpful in iterative solvers for non-linear equations in domains like electromagnetic simulations, fluid dynamics, and structural analysis [1,2]. Researchers can greatly cut down on computing time for high-precision computations by utilizing Newton's method's rapid convergence in later stages, which allows for more intricate and precise simulations.

DOI: 10.54254/2753-8818/2025.CH23395

By implementing adjustments like adaptive step sizes and smoothing techniques, Newton's method's efficiency can be further increased in practice. According to recent research, for example, modified Newton methods that use the nullspace of the Jacobian are better able to handle singular nonlinear equations [3]. In addition to increasing convergence rates, this flexibility increases Newton's method's usefulness in a wider range of scientific fields. Furthermore, sparse polynomial systems can be solved by combining Newton's approach with other numerical techniques, such as homotopies or optimization algorithms. This is very important for contemporary computational issues [4].

These developments have significant ramifications since they enable academics to study more complex models that were previously computationally unfeasible. Newton's approach speeds up calculations, which promotes improvements in physical phenomenon simulations, more precise forecasts, and more intelligent engineering designs. In fluid dynamics, for instance, the capacity to solve non-linear equations quickly results in simulations that are more responsive and realistic, which eventually helps in the construction of effective aerodynamic structures [5,6].

All things considered, Newton's method continues to be a pillar of scientific computing, and research is constantly improving its use and efficiency in resolving challenging mathematical issues.

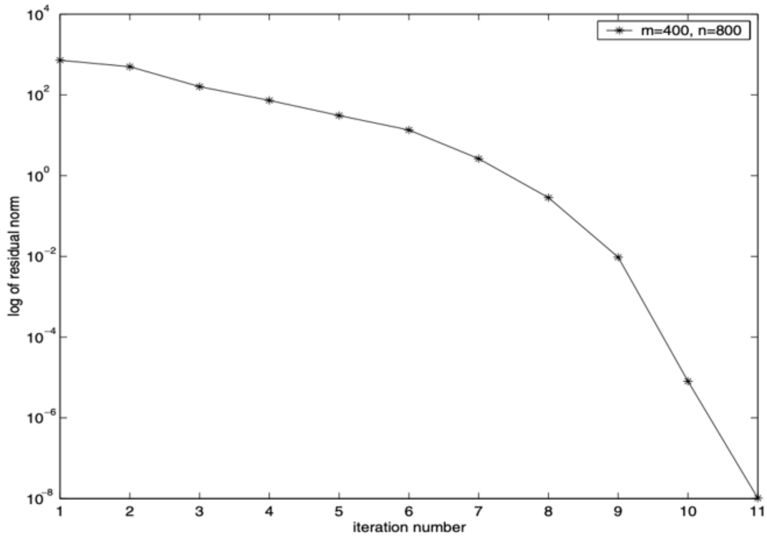

Figure 2: The logarithm of residual norm by iterations [2]

In the smoothing Newton methods, the Figure 2 shows that the residual norm decreases considerably in the first few iterations (1–3) and then gradually in the following iterations (4–11), yet it still declines [2]. This demonstrates that the algorithm's convergence is quadratic, meaning that it converges more quickly in the early iterations and steadily stabilizes in the later ones.

Table 1: Results for Rayleigh's problem (version 2) with different step sizes [2]

N	CPU time/s	Iterations	N	CPU time/s	Iterations
500	0.130	13	5000	1.190	14
1000	0.240	13	8000	2.120	15
2000	0.470	13	12800	3.140	14
4000	1.080	15	25600	5.990	14

In the modified Newton methods, the number of iterations needed is largely constant regardless of how the step size (N) is altered. This demonstrates that Newton's approach may converge rapidly

DOI: 10.54254/2753-8818/2025.CH23395

under various step size conditions and is insensitive to the problem's meshes, which is highly useful. The CPU time grows linearly as N rises, see Table 1. This indicates that Newton's method outperforms other direct discretization techniques and that the algorithm remains effective even when handling increasingly complicated issues. Additionally, it demonstrates that Newton's method can converge without depending on a reliable starting guess—a crucial feature for intricate optimization problems. In real-world applications, this aspect increases the method's dependability and usability.

Table 2: Convergence analysis [7]

	IT	NFE	Xn
F1x0=4			
NM	20	53	3.000000000000000
SM	13	84	3.000000000000000
Algorithm2	6	18	3.000000000000000
F2x0=1			
NM	7	12	-1.40449164821534
SM	5	21	-1.40449164821534
Algorithm2	3	15	-1.40449164821534
F3x0=0.7			
NM	8	12	0.804133097503664
SM	6	18	0.804133097503664
Algorithm2	4	12	0.804133097503664
F4x0=4			
NM	6	12	0.257530285439861
SM	4	18	0.257530285439861
Algorithm2	3	12	0.257530285439861
F5x0=3			
NM			Divergence
SM			Divergence
Algorithm2	4	12	1.5598640272247e-14

At the same time, the Table 2 shows that even though Newton's method uses a very large number of iterations, it is still less than other algorithms. Newton's method also has the feature of local quadratic convergence, which makes it converge relatively quickly, especially when it is close to the solution [7]. In most situations, Newton's approach can find more precise solutions, but in other cases, additional iterations are needed.

This demonstrates this algorithm's supremacy even further. It is also evident that Newton's approach might have to compute the Jacobian matrix and inverse operations, which would greatly raise the computational load and time complexity of high-dimensional issues. The table shows that the number of function evaluations (NFE) computed may be higher than with other approaches for more complex functions (like F1 and F3).

3.3. Example 2: computer graphics and 3D rendering

The inverse square root approach has been widely used in the realm of computer graphics and gaming. Normalized vectors, which are crucial for 3D graphics engines' lighting and shading computations, are very easy to compute with this approach. This method's rapid convergence, particularly in its

DOI: 10.54254/2753-8818/2025.CH23395

latter iterations, enables fast and precise vector normalization, which enhances rendering performance and produces more realistic lighting effects in real-time graphics applications [8,9].

Furthermore, the algorithm's effectiveness has prompted its integration into a number of computational systems outside of graphics, such as machine learning methods and physics simulations, where quick computations of vector orientations are essential [10]. For example, it has been shown that the approach improves the performance of neural networks that need vector normalization during training stages in addition to speeding up computations in visual rendering [11]. The inverse square root method's applicability in many fields highlights its adaptability and influence across academic fields.

Moreover, recent studies have focused on optimizing the inverse square root procedure to further reduce mathematical costs. Innovations such as modified magic stable techniques have demonstrated improved accuracy while maintaining and reducing computation time [12]. These advancements are critical as the demand for real-time processing in gaming and simulation environments persists to grow, pushing the limits of existing computational capabilities.

In conclusion, the inverse square root approach is a fundamental technique in computer graphics that helps to improve the visual content's aesthetic appeal through lighting improvements and shading while simultaneously promoting more effective computing methods in wider applications. This algorithm's performance and applicability are expected to increase with more research and development.

4. Conclusion

To sum up, this work thoroughly investigated three well-known fast square root algorithms: the fast inverse square root method, the binary search method, and Newton's approach. The author evaluated their performance over a range of iterations using rigorous numerical analysis, paying particular attention to convergence speed and accuracy. The results showed that Newton's approach was quite effective when it came to an accurate estimate, even though it initially lacked precision due to its quick convergence in subsequent iterations. For initial estimates, on the other hand, the binary search approach demonstrated greater stability, guaranteeing consistent performance over a range of inputs. Furthermore, because of its novel bit manipulation strategies, the fast inverse square root method demonstrated remarkable efficiency, particularly in computational graphics applications. In the future, this study creates opportunities for more research. Future research could look into hybrid strategies that combine these algorithms' advantages, which could improve their applicability and convergence rates in more challenging situations. Furthermore, resolving the drawbacks of Newton's method's reliance on early estimations may offer insightful information for practical uses. In the end, the findings of this study not only advance knowledge of square root calculation algorithms but also work as a manual for professionals trying to choose the best methods for particular computational requirements.

References

[1] Zavalani, G. (2014). A modification of Newton method with third-order convergence. Applied Mathematics and Computation, 2(4), 98-101.

[2] Chen, X. D., Sun, D., & Sun, J. (2003). Complementarity functions and numerical experiments on some smoothing Newton methods for second-order-cone complementarity problems. Computational optimization and applications, 25, 39-56, page53

[3] Shen, Y. Q., & Ypma, T. J. (2005). Newton's method for singular nonlinear equations using approximate left and right nullspaces of the Jacobian. Applied numerical mathematics, 54(2), 256-265.

[4] Verschelde, J., Verlinden, P., & Cools, R. (1994). Homotopies exploiting Newton polytopes for solving sparse polynomial systems. SIAM Journal on Numerical Analysis, 31(3), 915-930.

DOI: 10.54254/2753-8818/2025.CH23395

[5] *Cartis, C., Gould, N. I., & Toint, P. L. (2010). On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. Siam journal on optimization, 20(6), 2833-2852.*

[6] *Chen, J., & Gerdts, M. (2012). Smoothing technique of nonsmooth Newton methods for control-state constrained optimal control problems. SIAM Journal on Numerical Analysis, 50(4), 1982-2011.*

[7] *Kou, J., Li, Y., & Wang, X. (2006). A modification of Newton method with third-order convergence. Applied Mathematics and Computation, 181(2), 1106-1111.*

[8] *Moroz, L., Samotyy, V., Horyachyy, O., & Dzelendzyak, U. (2019, September). Algorithms for calculating the square root and inverse square root based on the second-order householder's method. In 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 1, pp. 436-442). IEEE.*

[9] *Ercegovac, M. D., Lang, T., Muller, J. M., & Tisserand, A. (2000). Reciprocation, square root, inverse square root, and some elementary functions using small multipliers. IEEE Transactions on computers, 49(7), 628-637.*

[10] *Zheglova, P., & Farquharson, C. (2012, November). Level set method in seismic inversion: 2D reconstruction of boundaries. In SEG International Exposition and Annual Meeting (pp. SEG-2012). SEG.*

[11] *Kock, N., & Hadaya, P. (2018). Minimum sample size estimation in PLS-SEM: The inverse square root and gamma-exponential methods. Information systems journal, 28(1), 227-261.*

[12] *Frommer, A., Hashemi, B., & Sablik, T. (2014). Computing enclosures for the inverse square root and the sign function of a matrix. Linear Algebra and its Applications, 456, 199-213.*