ISSN 2278-3091
Volume 8, No.1.5, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse2481.52019.pdf
https://doi.org/10.30534/ijatcse/2019/2481.52019

New Scaling of Bracketing Method for Fourth and Sixth Section

Ruhana Jaafar[1], Nur Aizat Noor Manshah[2], Nurhidayatul Liyana Mohd Mustafa[3], Nur Solihah Khadhiah Abdullah[4]
[1,2,3,4]University Teknologi MARA(UiTM), Faculty of Computer and Mathematical Sciences,
CawanganTerengganu, Kampus Kuala Terengganu
ruhana75@ uitm.edu.my
manshahna@gmail.com
nurhidayatul.liyana@gmail.com
nsolihah@uitm.edu.my

ABSTRACT

Numerical analysis is widely used in the mathematical area for solving root of nonlinear functions. Two methods used in Bracketing Method which is the easiest numerical method for root finding are Fourth Section and Sixth Section. In this research, Fourth Section and Sixth Section were modified by adding a scalar in order to solve nonlinear functions. These nonlinear functions were analysed for their efficiency in terms of number of iterations and CPU time by using the methods. The approximation roots for these methods were compared with the actual roots to acquire their accuracy.

Key words: Bracketing method; Fourth section method; Sixth section method; Scaling; Number of iterations; CPU time

1.INTRODUCTION

Numerical analysis is one of the areas of mathematics and computer science that creates, analyses, and implements algorithms for solving numerical problems of continuous mathematics. There are many equations whose roots cannot be evaluated analytically by any methods. The approximate values of the roots of such equations can be found either by a graphical approach, or the number of iterative methods or by a combination of both processes [2]. This iterative method corresponds to poles and zeros in sequence in an iterative way [1] In numerical methods of solving non-linear equations or root finding, methods used are Bracketing Methods, Open Methods, System of Nonlinear Equations and Roots of Polynomial. In this research, the methods used was n-th Section Method that was modified from Bisection Method which is a part of Bracketing Method.

2.BRACKETING METHOD

Bracketing Method is one of methods for root finding for equations. There are two methods in Bracketing Method which are Bisection Method and False Position Method. [3] stated that both methods have a linear order of convergence, but the False Position Method suffers due to the slow convergence in some cases. [7] explained in their work that a modified Bisection Method was developed to choose a set of data-points at which to perform the numerical simulations and thus decrease the computational effort of this task.

3.BISECTION METHOD

Bisection Method is one of the simplest methods in numerical analysis to find the roots of a non-linear equation. The term 'bisection' indicates the division of two intervals of an equation into two sections as there is a root between the intervals. The division into halves continues to take place until the root is found at a desirable tolerance. [2] as cited in [4] stated that Bisection Method must have opposite sign at both edges of intervals where $f(a) \bullet f(b) < 0$. Why must it be lesser than zero, not equal or more than zero? This is because there is at least one root exists in the interval. Roots will not exist if equal or more than zero is used.

However, the division of interval into halves leads this method to slow convergence even though it is always convergent [2]. This leads to modification of this method which is n-th Section Method to make it faster than before.

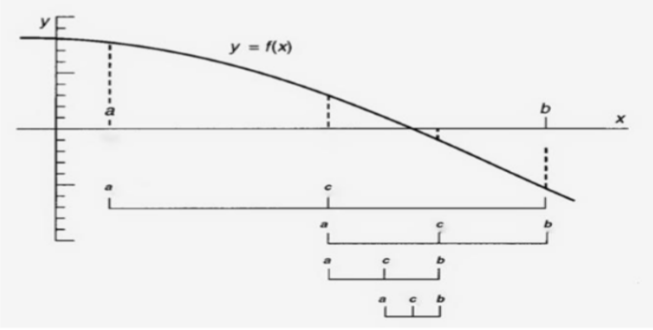

Figure 1: The schematic representation of Bisection Method

4.N-TH SECTION METHOD

N-th Section Method is focussing on Fourth Section and Sixth Section Methods. As they are known as Fourth Section and Sixth Section, the interval is divided into four sections and six sections respectively. The root will be identified in each of these sections, and only $f(n) \bullet f(n+1) < 0$ will be

the existing root. [5] concluded that the execution of this method leads to fast convergence and fast calculation.

These modifications proved that n-th Section Method is faster than Bisection Method in terms of number of iterations, but its CPU time consumption is longer than that of Bisection Method. However, in this project, n-th Section Method was modified to be faster than the origin method by using a scalar during finding roots.

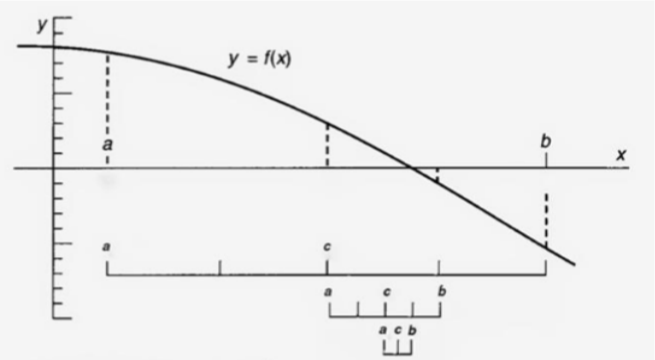

Figure 2: The schematic representation of Fourth Section

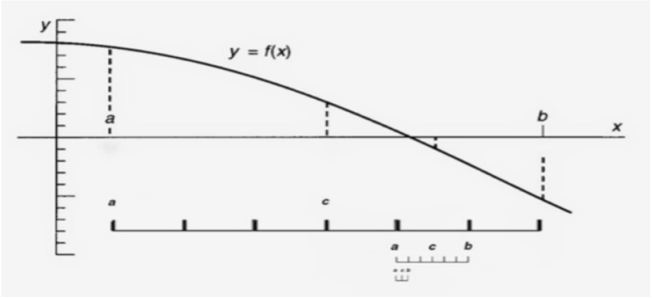

Figure 3: The schematic representation of Sixth Section

5.FOURTH AND SIXTH SECTION METHODS WITH SCALING

Scaling to larger sets of subscribers and resources is one of the importance strategies [6]. In this study, a new modification for Fourth Section and Sixth Section methods was introduced by using a new scaling factor. The new scaling is

$m = c\,\frac{b_i - a_i}{4}$ where $c = \frac{b - a}{4}$ and i is number of iterations

The value for b_i and a_i will change every iteration depending on the interval that has existence of root.

6.METHODOLOGY

Maple 16 was used to solve the Fourth Section and Sixth Section methods because it is the easiest method. Matlab, C++ and Microsoft Excel were not chosen because with the softwares, it was hard to make the programming code for the methods that were used in the study. The CPU processor was used to measure the time taken by the methods in solving for root finding. This section shows the algorithm of scalar and non-scalar methods that were used in this project.

Table 1: Algorithm of Fourth Section

No.	Steps
1	Identify two numbers a and b at which f has different signs.
2	Define midpoint, $c = \frac{b - a}{4}$ and $f(c)$.
3	Determine if root exists $f(a) \cdot f(b) < 0$ then $r \in (a,b)$ or $f(b) \cdot f(c) < 0$ then $r \in (b,c)$ or $f(c) \cdot f(d) < 0$ then $r \in (c,d)$ or $f(d) \cdot f(e) < 0$ then $r \in (d,e)$
4	Choose the desired interval from (i) to (iv)
5	Repeat until the desired iteration/ accuracy is found

Table 2 :Algorithm of Sixth Section

No.	Steps
1	Identify two numbers a and b at which f has different signs.
2	Define midpoint, $c = \frac{b - a}{6}$ and $f(c)$.
3	Determine if root exists $f(a) \cdot f(b) < 0$ then $r \in (a,b)$ or $f(b) \cdot f(c) < 0$ then $r \in (b,c)$ or $f(c) \cdot f(d) < 0$ then $r \in (c,d)$ or $f(d) \cdot f(e) < 0$ then $r \in (d,e)$ or $f(e) \cdot f(f) < 0$ then $r \in (e,f)$ or $f(f) \cdot f(g) < 0$ then $r \in (f,g)$
4	Choose the desired interval from (i) to (vi)

5	Repeat until the desired iteration/ accuracy is found

Table 3: Algorithm of Fourth Section with Scaling

No.	Steps
1	Identify two numbers a and b at which f has different signs.
2	First iteration: $c = \frac{b - a}{4}$ Second iteration until achieve desirable tolerance: $m = c\frac{b_i - a_i}{4}$ and $f(c)$ $c = \text{constant}$
3	Determine if root exists $f(a) \cdot f(b) < 0$ then $r \in (a,b)$ or $f(b) \cdot f(c) < 0$ then $r \in (b,c)$ or $f(c) \cdot f(d) < 0$ then $r \in (c,d)$ or . . . $f(n) \cdot f(n+1) < 0$ then $r \in (n, n+1)$
4	Choose the desired interval from (i) to infinity.
5	Repeat until the desired iteration/ accuracy is found

Table 4: Algorithm of Sixth Section with Scaling

No.	Steps
1	Identify two numbers a and b at which f has different signs.
2	First iteration: $c = \frac{b - a}{6}$ Second iteration until achieve desirable tolerance: $m = c\frac{b_i - a_i}{6}$ and $f(c)$ $c = \text{constant}$
3	Determine if root exists $f(a) \cdot f(b) < 0$ then $r \in (a,b)$ or $f(b) \cdot f(c) < 0$ then $r \in (b,c)$ or $f(c) \cdot f(d) < 0$ then $r \in (c,d)$ or $f(d) \cdot f(e) < 0$ then $r \in (d,e)$ or $f(e) \cdot f(f) < 0$ then $r \in (e,f)$ or . . . $f(n) \cdot f(n+1) < 0$ then $r \in (n, n+1)$
4	Choose the desired interval from (i) to infinity.
5	Repeat until the desired iteration/ accuracy is found

7.TEST FUNCTIONS

The test functions used were mainly from four different non-linear functions which were the exponential, trigonometric, logarithmic and cubic polynomial functions. From each of the non-linear functions, there were two equations selected. The software used to find the roots for every non-linear function was the Maple 16 software.

Test function 1: $f(x): \sin(x) - \frac{x}{2}$ on the interval $[1,2]$

Test function 2: $f(x): \cos(x) - 0.25$ on the interval $[1,2]$

Test function 3: $f(x): \ln(x) + 2x - 6$ on the interval $[2,3]$

Test function 4: $f(x): \log(x) + 2x^2 - 3$ on the interval $[1,2]$

Test function 5: $f(x): x^3 + 4x^2 - 10$ on the interval $[1,2]$

Test function 6: $f(x): x^3 + 6x - 16$ on the interval $[1,2]$

Test function 7: $f(x): e^x + x - 2$ on the interval $[0,1]$

Test function 8: $f(x): e^x - 3x$ on the interval $[1,2]$

8.RESULTS AND DISCUSSION

The result is on the test run by Maple 16 software. Two selected functions from each of nonlinear functions. Then, these methods were compared based on number of iterations, CPU time and error analysis.

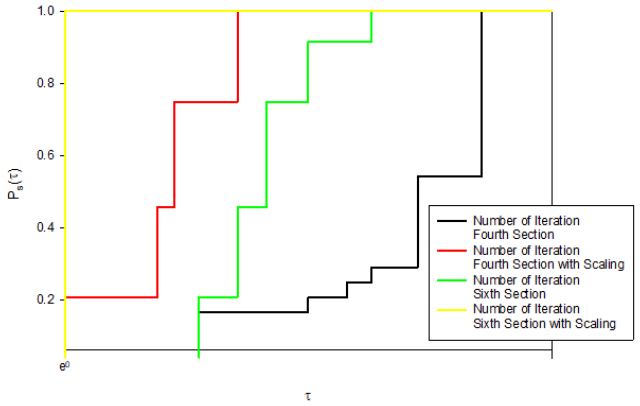

Figure 4: Performance profile of number of iterations

The best method on the number of iterations was determined by choosing the highest curve value based on the left side of the figure (as shown in Figure 4). Sixth Section with Scaling which has the highest curve, was the best method among the four methods followed by Fourth Section with Scaling, Sixth Section and lastly Fourth Section. This is because Sixth Section with Scaling possesses the least number of iterations compared to other methods. When reaching to the right side of the graph, all methods can still be used to solve all problems, but the number of iterations will be greater than Sixth Section with Scaling.

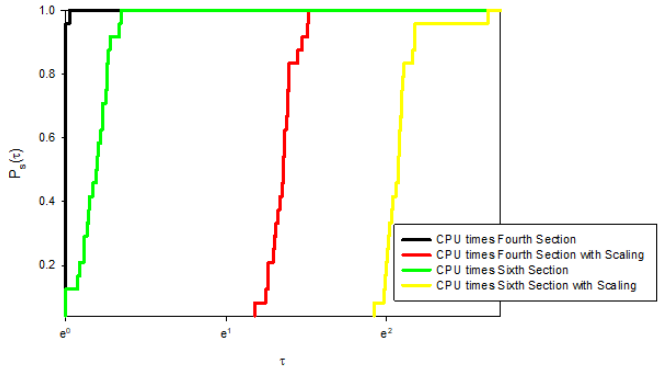

Figure 5: Performance profile of CPU times

As shown in Figure 5, the best method on the CPU time was determined by choosing the highest curve value based on the left side of the figure. At the beginning, Fourth Section which has the highest curve was the best method among the four methods, followed by Sixth Section, Fourth Section with Scaling and lastly Fourth Section. It is because Fourth Section has the fastest CPU time among them. However, when reaching to the right side of the graph, all methods can still solve the problems even though they were slower in terms of CPU time compared to that of Fourth Section.

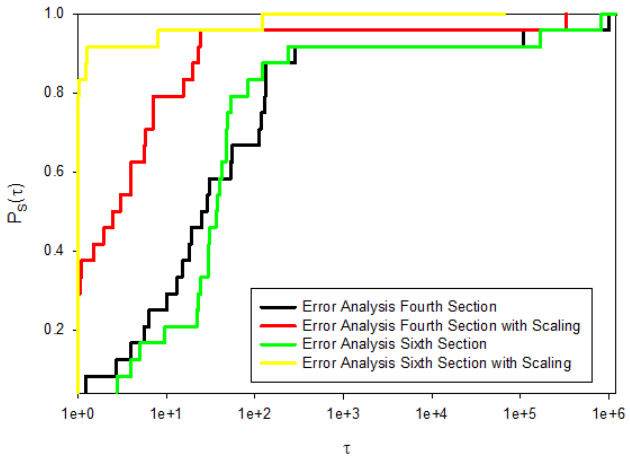

Figure 6 : Performance profile of error analysis

Based on the performance profile of error analysis, the left curve shows that Sixth Section with Scaling was the fastest in solving the problem since it possesses the least of error among other methods (as shown in Figure 6). However, when reaching to right side, it can be presumed that all methods can solve the problem although they were not compatible to Sixth Section with Scaling.

9.CONCLUSION

In this research, Bracketing Method for Fourth Section and Sixth Section Methods was chosen to be the models together with new implemented methods which were the Fourth Section with Scaling and Sixth Section with Scaling. Thus, a total of four methods were used in this research to solve the roots for the non-linear functions. The efficiency of Bracketing Method for Fourth Section and Sixth Section Methods between scalar and non-scalar was analyzed based on the number of iterations, CPU time and error analysis. From the results obtained, it can be concluded that Sixth Section with Scaling is the best method among the four methods in terms of number of iteration as it possesses the lowest number of iterations. However, the results obtained for CPU time showed that Fourth Section came out as the best method as it solved the fastest compared to the other methods. Based on the results obtained in error analysis, Sixth Section with Scaling is the best method compared to the other three methods. This is because it possesses a lower number of errors. To sum up, future researchers are suggested to use the method of Sixth Section with Scaling to find a lesser iteration to solve non-linear functions. The Fourth Section Method or any other Bisection Method can be chosen to solve for faster CPU time.

REFERENCES

1. Africa, A. D. M., Arevalo, P. B. T., Publico, A. P. and Tan, M. A. A., *Time Response Anaylisis of Control Systems.* International Journal of Advanced Trends in Computer Science and Engineering. Vol.8, No. 4, July – August 2019.

2. Ali, M. R. M., Fakhri, M. I., Hayati, N., Ramli, N. A., & Jusoh, I. *The n-th section method: A modification of Bisection.* Malaysian Journal of Fundamental and Applied Sciences, Vol. 13(No. 4), pp 728-731, 2017 https://doi.org/10.11113/mjfas.v0n0.577
3. Alojz, S. *Combined Bracketing Methods for Solving Nonlinear Equations.* Applied Mathematics Letters, Vol. 25, pp 1755-1760, 2012. https://doi.org/10.1016/j.aml.2012.02.006
4. Doron, L. *Introduction to Numerical Analysis.* Departments of Mathematics and Center of Scientific Computation and Mathematical Modelling(CSCAMM)University of Maryland. *Retrieved from http://www2.math.umd.edu/dlevy/books/na.pdf*, 2010.
5. Fakhri, I., Rivaie, M. & Jusoh, I. *An n-th section line search in Conjugate gradient Method for small scale unconstrained optimization.* Malaysian Journal of Fundamental and Applied Sciences, Vol. 13, No. 4, pp 588-592, 2017. https://doi.org/10.11113/mjfas.v0n0.579
6. Kumar, V. and Sharma, P. *Dynamics Elasticity : Cloud Computing.* International Journal of Advanced Trends in Computer Science and Engineering. Vol.5, No. 3, May – June 2016.
7. Yang, K., Arezoomandan, S., & Sensale-Rodriguez, B. *Design of terahertz filters using a modified bisection method*. IEEE 39th International Conference on Infrared, Millimeter, and Terahertz waves (IRMMW-THz) (pp. 1-2). IEEE, Sept. 2014 https://doi.org/10.1109/IRMMW-THz.2014.6956373