

The n -th section method: A modification of Bisection

Mohd Rivaie Mohd Ali ^{a,*}, Muhammad Imza Fakhri ^b, Nujma Hayati ^a, Nurul Atikah Ramli ^b, Ibrahim Jusoh ^a

^a Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi Mara Terengganu, 21300 Kuala Terengganu, Terengganu, Malaysia

^b Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi Mara Shah Alam, 40450 Shah Alam, Selangor, Malaysia

* Corresponding author: rivaie75@yahoo.com

Article history

Received 18 February 2017
 Accepted 4 October 2017

Abstract

Bisection method is the easiest method to find the root of a function. This method is based on the existence of a root on a specified interval. This interval is then halved or divided into two parts. The root is known to be lying in either one of these interval. The iterative sequence is continued until a desired stopping criterion is reached. In this research, a new modification of bisection method namely fourth section and sixth section methods are introduced. These methods are tested for several selected functions by using Maple software. The results are then analyzed based on the number of iterations and the CPU times. Based on the results, it is shown that when the interval increases, the CPU times will also increase. However, the number of iterations is reduced significantly.

Keywords: Bisection method; Roots finding; Number of iterations; CPU times

© 2017 Penerbit UTM Press. All rights reserved

INTRODUCTION

Numerical analysis is one of the areas of mathematics and computer science that create, analyses, and implements algorithms for solving numerically 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 number of iterative methods or by a combination of both processes. In numerical methods of solving non-linear equations or root finding, the most popular methods are Bisection method, Newton's method and Secant method. Several studies shown that many researchers are interested in the development and the application of Bisection method such as Muller method by Park and Hltotumatu (1987) and bisection-exclusion method by Yakoubsohn (2005).

BISECTION METHOD

Bisection method is the easiest method using an interval which is divided into two or half. The root is known to be lying in either one of these interval. The division of this interval into two continue until the root is found, up to certain degree of accuracy. Doron (2010) assume that this method has opposite signs at both edges of the intervals where $f(a)f(b) < 0$. Then, it is known that $f(x)$ has at least one root in the interval, $[a, b]$. The existing method proceeds and continue its iteration until it converges to a point within the tolerance range and finds the value of x such that $f(x) = 0$ or approximately 0 (Solanki, 2014). A better approximation is obtained if a point $(c, 0)$ is found where the

secant line L joining the points $(a, f(a))$ and $(b, f(b))$ crosses the x -axis (Mathews *et. al*, 2004). The main advantage of Bisection method is that the behaviour itself is always convergent since the method brackets the root much more quickly than the incremental search method does. Hence, Bisection method is easy to use, apply and has wide range of applications in other developments.

However, the division of the interval into two section leads to slow convergence of Bisection method. Hence, a new method is proposed named as n -th section method. This methods are compared with the original Bisection method based on number iterations, CPU times and accuracy. The bisection algorithm is the most primitive but robust for finding a real root because it is always converges to the root (Burden *et. al*, 1993).

THE N -TH SECTION METHOD

The n -th section method is a modification of classical Bisection method which is fourth and sixth section method. Bisection method which divides the interval into two section leads to slow convergence. This new scheme divided the interval into four and six section. The root is then identify either in the first, second, third, fourth, fifth or sixth interval. This will lead to faster convergence of the root and provide faster calculation of the roots. For an even number of n , the algorithm will be similar to Bisection method. However, for an odd number of n the algorithm needs to be modified. Therefore, in this research only the even number of n is considered and compared with the Bisection method. A comparison between bisection method and n -th section method will be analysed based on its efficiency in terms of number of

iterations and CPU times. Fig. 1, 2 and 3 show the schematic representation of Bisection and n -th section line search method.

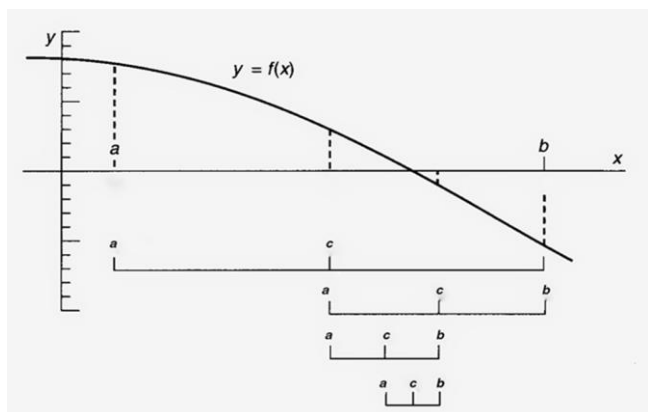


Fig. 1 The schematic representation of bisection method

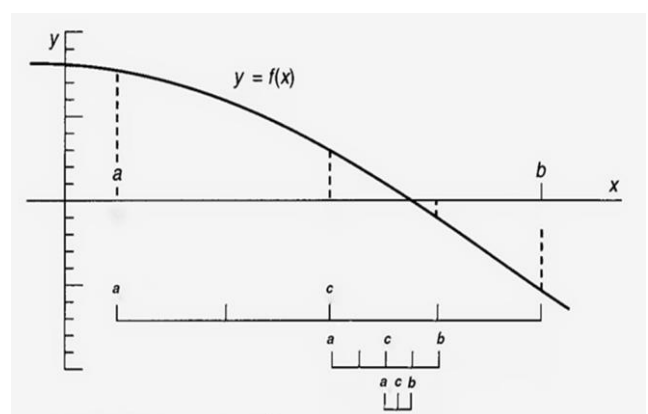


Fig. 2 The schematic representation of fourth section method

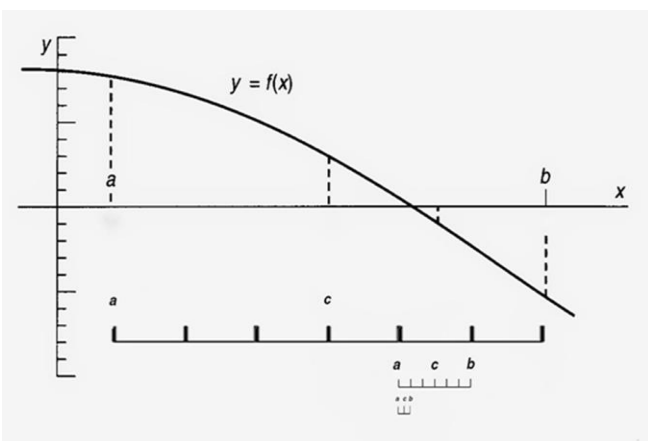


Fig. 3 The schematic representation of sixth section method

METHODOLOGY

A computer code programming is constructed by using Maple 12 software based on the Bisection method. The Maple 12 software is chosen because the programming code is relatively simple and easy to apply. This method is initialized with the limits of function on the interval $[a, b]$ where the function is defined. The function $f(x)$ is assumed to be continuous on the interval $[a, b]$ is chosen such that $f(a)f(b) < 0$. Thus, it is divided into two and more intervals. At each interval, the midpoint of the interval $c = \frac{a+b}{2}$ is computed and $f(c)$ is evaluated. Lastly, the process is repeated until a desired accuracy is achieved. Table 1 is the algorithm for Bisection method.

Table 1: Algorithm of Bisection method

No.	Step
1	Identify two numbers a and b at which f has different signs.
2	Define midpoint, $c = \frac{a+b}{2}$ and $f(c)$.
3	Determine if root exists <ol style="list-style-type: none"> i. $f(a)f(c) < 0$ then $r \in (a, c)$ or ii. $f(b)f(c) < 0$ then $r \in (c, b)$
4	Choose the desire interval either (i) or (ii)
5	Repeated until desired iteration/ accuracy

Next, Table 2 and Table 3 are the calculation of n -th section method of fourth section method and sixth section method, respectively.

Table 2: Algorithm of Fourth section method

No.	Step
1	Identify two numbers a and b at which f has different signs.
2	Define midpoint, $c = \frac{a+b}{n}$ where $n = 4$ and $f(c)$.
3	Determine if <ol style="list-style-type: none"> i. $f(a)f(b) < 0$ then $r \in (a, b)$ ii. $f(b)f(c) < 0$ then $r \in (b, c)$ iii. $f(c)f(d) < 0$ then $r \in (c, d)$
4	Choose the desire interval either (i) or (ii) or (iii)
5	Repeated until desired iteration/ accuracy

Table 3: Algorithm of Sixth section method

No.	Step
1	Identify two numbers a and b at which f has different signs.
2	Define midpoint, $c = \frac{a+b}{n}$ where $n = 6$ and $f(c)$.
3	Determine if <ol style="list-style-type: none"> i. $f(a)f(b) < 0$ then $r \in (a, b)$ ii. $f(b)f(c) < 0$ then $r \in (b, c)$ iii. $f(c)f(d) < 0$ then $r \in (c, d)$... $f(n)f(n+1) < 0$ then $r \in (n, n+1)$
4	Choose the desire interval either (i) or (ii) or (iii) or infinity
5	Repeated until desired iteration/ accuracy

RESULTS AND DISCUSSION

The results are presented for several common functions such as cubic polynomial, logarithm, exponential and trigonometric. This function has been tested through Maple to check for the number of iterations and CPU times based on Bisection method, Fourth Section and Sixth Section. Numerical result based on the test run of both methods using the standard test functions is listed below.

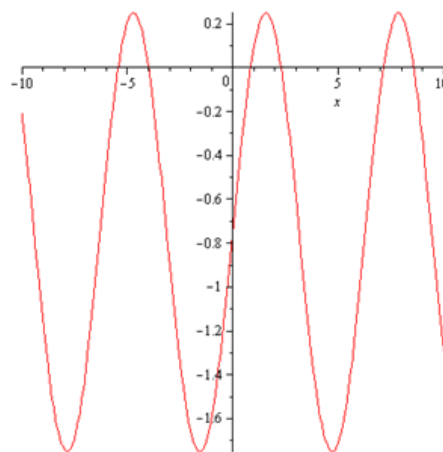


Fig. 4 Graph function of $f(x) = \sin(x) - 0.75$

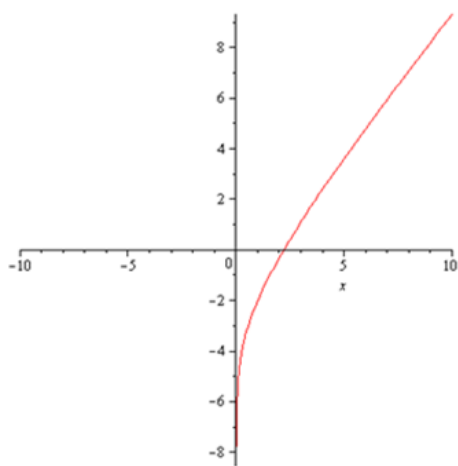


Fig. 5 Graph function of $f(x) = x + \ln(x) - 3$

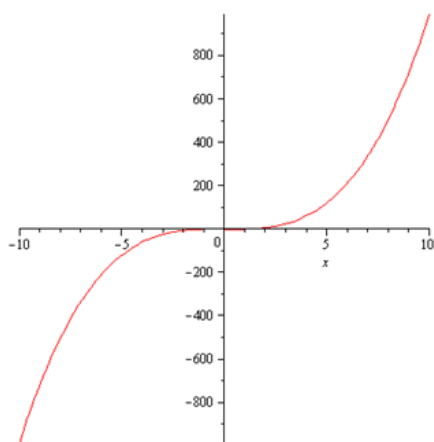


Fig. 6 Graph function of $f(x) = x^3 - x - 1$

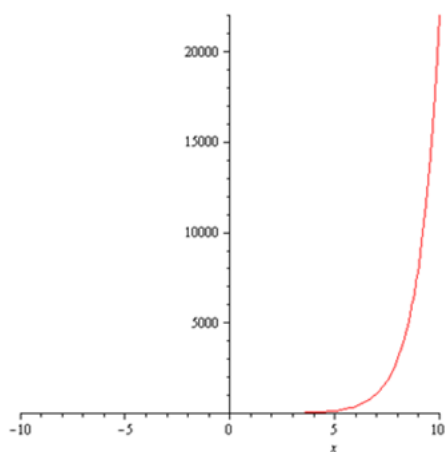


Fig. 7 Graph function of $f(x) = e^x - x - 1$

Table 5: Number of iterations for $f(x) = x + \ln(x) - 3$

	3 decimal	4 decimal	5 decimal
Bisection	10	12	17
Fourth Section	5	7	9
Sixth Section	3	5	6

Table 6: Number of iterations for $f(x) = x^3 - x - 1$

	3 decimal	4 decimal	5 decimal
Bisection	10	13	17
Fourth Section	5	8	8
Sixth Section	4	5	5

Table 7: Number of iterations for $f(x) = e^x + x - 1$

	3 decimal	4 decimal	5 decimal
Bisection	7	10	13
Fourth Section	5	7	7
Sixth Section	4	5	7

Based on the table 4, 5, 6 and 7, it shows that the number of iteration is reduced when the number of interval increased. Therefore, the sixth and fourth section possess lower number of iteration compared to Bisection.

Table 8: CPU times for $f(x) = \sin(x) - 0.75$

	CPU times		
Bisection	0.34375	0.390625	0.4375
Fourth Section	0.640625	0.65625	0.65625
Sixth Section	1.328125	1.28125	1.359375

Table 9: CPU times for $f(x) = x + \ln(x) - 3$

	CPU times		
Bisection	0.421875	0.437500	0.468750
Fourth Section	0.687500	0.640625	0.703125
Sixth Section	1.328125	1.281250	1.625000

Table 10: CPU times for $f(x) = x^3 - x - 1$

	CPU times		
Bisection	0.468750	0.390625	0.406250
Fourth Section	0.640625	0.656250	0.656250
Sixth Section	1.328125	1.421875	1.421875

Table 11: CPU times for $f(x) = e^x + x - 1$

	CPU times		
Bisection	0.328125	0.359375	0.296875
Fourth Section	0.625000	0.578125	0.578125
Sixth Section	1.265625	1.234375	1.234375

However, in term of CPU time, the CPU time increased significantly when the interval increased. Therefore, sixth and fourth section methods possessed higher CPU time compared to Bisection. The significant increased of CPU time is due to the division of interval and determination of $f(a)f(b) < 0$ in each interval.

Table 4, Table 5, Table 5, and Table 7 are the numerical results for number of iterations and Table 8, Table 9, Table 10 and Table 11 are the CPU times. The decimal places indicate the tolerance of the roots for the current iteration and previous iterations.

Table 4: Number of iterations for $f(x) = \sin(x) - 0.75$

	3 decimal	4 decimal	5 decimal
Bisection	5	9	12
Fourth Section	2	5	6
Sixth Section	2	6	5

CONCLUSION

In this research, a new modification of Bisection has been proposed. This modification is based on even number of interval. The new method is named as the Fourth section and Sixth section method. If the number of iterations reduced, then the CPU times for Fourth section and Sixth section are increased. Based on this research, if a researcher is interested only with the method which possesses lower number of iteration without concerning the CPU time, it is suggested that the researcher use the Sixth section method as it converge faster than Bisection and Fourth section method. On the other hand, if the researcher is only interested on the CPU time without concerning the number of iterations, it is suggested to use the Bisection method as it possessed lower CPU time.

ACKNOWLEDGEMENT

This research is conducted under the fundamental research grant scheme (FRGS) grant number : 600-RMI/FRGS 5/3 (64/2015).

REFERENCES

- Burden, R. L. and J. D. Faires (1993). *Numerical analysis (5th edition)*. Boston: PWS Kent Pub. Co. Print.
- Doron L. (2010). *Introduction to numerical analysis*. Departments of Mathematics and Center of Scientific Computation and Mathematical Modeling (CSCAMM) University of Maryland. Retrieved from <http://www2.math.umd.edu/~dlevy/books/na.pdf>
- Mathews, J. H. and K. D. Fink (2004). *Numerical methods using MATLAB (4th edition)*. Upper Saddle River, N.J.: Pearson. Print.
- Park, B. and S. Hitotumatu (1987). A study on New Muller's method. *Publications of the Research Institute for Mathematical Sciences*. 23(4), 667–672. Retrieved from: https://www.emis-ph.org/journals/show_pdf.php?issn=0034-5318&vol=23&iss=4&rank=5
- Solanki, C., Thapliyal, P. and Tomar, K. (2014). Role of bisection method, *International Journal of Computer Applications Technology and Research*, 3(8), 533–535.
- Yakoubsohn, J. (2005). Numerical analysis of a bisection-exclusion method to find zeros of univariate analytic functions, *Journal of Complexity*, 21(5), 652–690.