ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Special Issue , August 2019

International Conference on Recent Advances in Science, Engineering, Technology and Management at Sree Vahini Institute of Science and Technology-Tiruvuru, Krishna Dist, A.P

A novel Algorithm of adapted Bisection Method for Nonlinear Equation

D.SAIPRATAPKUMAR,P.ASHOK,CH.NARASIMHARAO

ABSTRAT: The aim of this research is to proposed a new computational algo-rithm of modified bisection method for nonlinear equation. In numerical results, we performed the algorithm on GNU Octavce[1] and compared the error and number of iterations with bisection and Newton methods.

KEYWORDS: Nonlinear equation, Bisection method, Newton method

I. INTRODUCTION

The problem of finding an approximation to the root of an nonlinear equation can be found in many fields of sciences and engineering ([2], [3], [4]). A root-finding algorithm is a numerical method, or algorithm, for finding a value x such that $f(x) = 0$, for a given function f. Such an x is called a root of the function f.

The simplest root-finding algorithm is *the bisection method*. It works when f is a continuous function and it requires previous knowledge of two initial guesses, b and c, such that $f(b)$ and $f(c)$ have opposite signs, then find the midpoint of $[b, c]$, and then decide whether the root lies on $[b, (b + c)/2]$ or $[(b+c)/2, c]$. Repeat until the interval is reach to small. By the intermediate value theorem ([2], [3], [4],[6]) implies that a number $x$exists in (b, c) with $f(x) = 0$. Although the bisection method is reliable, but it converges slowly, gaining one bit of accuracy with each iteration.

Theorem 1.1. *(Intermediate Value Theorem): Given a continuous real-valued function $f(x)$ defined on an interval $[b, c]$, then if y is a point between the values of $f(b)$ and $f(c)$, then there exists a point r such that $y = f(r)$.*

The *Newton method* is much more e cient than the bisection method. Furthermore, the tangent line often shoots wildly and might occasionally be trapped in a loop ([2], [3], [4],[8]). Newton's method may not converge if started too far away from a root. However, when it does converge, it is faster than the bisection method, and is usually quadratic.

However, Newton's Method fails to produce a solution, if there is no solu-tion to be found or the initial solution is not good enough for the method initial solutions will lead to the exact numerical solution. But some initial solutions can make the method diverges ([2], [4], [7], [8]).

Hence, in this paper, we construct and propose a new algorithm of mod-ified bisection algorithm for solve nonlinear equation. In numerical results, we tested the algorithm and compared the number of iterative and numerical error with the bisection and Newton methods.

II. MAIN RESULTS

Let f be a continuous function and defined on $[b, c]$ which $f(b) \cdot f(c) < 0$. Firstly, we set $b_1 = b$ and $c_1 = c$. For an integer $k \geq 1$, By the bisection method, we have $d_k = (b_k + c_k)/2$. Next, we consider a new subinterval (b_k, c_k) by

$$(b_k^*, c_k^*) = (b_k, d_k) \quad \text{if} \quad f(b_k)f(d_k) < 0 \qquad - (1)$$
$$= (d_k, c_k) \quad \text{if} \quad f(d_k)f(c_k) < 0$$

Then, we can find the equation of straight line from the points $(a_k^*, f(a_k^*))$ and$(b_K^*, f(b_K^*))$k

$$Y = mx + c \quad - (2) \quad \text{where} \quad m = f(c_k^*) - f(b_k^*)/c_k^* - b_k^*$$

And $c = f(c_k^*) - mc_k^*$ or $c = f(b_k^*) - mb_k^*$

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 6, Special Issue , August 2019
International Conference on Recent Advances in Science, Engineering, Technology and Management at Sree Vahini Institute of Science and Technology-Tiruvuru, Krishna Dist, A.P

Hence, the *x*-intercept of the straight line is at a point $x_k = -d/m$

$$X_k = c_k^* - f(c_k^*)\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\} \quad \text{or} \quad b_k^*-f(b_k^*)\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\}$$

Finally, we choose the new subinterval for the next iteration as follows
Finally,

$$(b_{k+1,}c_{k+1}) = \{ \ (b_k^*,x_k) \text{ if } f(b_k^*)f(x_k)<0 \ ,(x_k,c_k^*) \text{ if } f(x_k)f(c_k^*)<0 \qquad - (3)$$

The process is continued until the interval is su ciently small or the approxi-mate solution is su ciently close to the exact solution.

Therefore, we can state the algorithm for finding a solution of nonlinear equation $f(x) = 0$ on an interval $[b, c]$ as follows:

adapted Bisection Algorithm(ABA) :

Step 1: Set $k = 1$, tolerance(TOL)$\approx 0.1 \times 10^{-7}$ and $b_k = b$, $c_k = c$;
where $f(b)f(c) < 0$.

Step 2: Compute $d = (b_k + c_k)/2$.

Step 3: Compute for a subinterval (b_k^*, c_k^*) by

$$(b_k^*, c_k^*) = \{ \ (b_{k,}d_k) \text{ if } f(b_k)f(d_k)<0 \ ,(d_k,c_k) \text{ if } f(d_k)f(c_k) <0$$

Step 4: Compute for $x_{k=}$ -e/m where $m= f(c_k^*)-f(b_k^*)/c_k^*-b_k^*$ and
$C=f(c_k^*)-mc_k^*$ or $c=f(b_k^*)-mb_k^*$

Step 5: IF $|f(x_k)| < TOL$, then STOP program ,else

$$(b_{k+1,}c_{k+1}) = \{ \ (b_k^*,x_k) \text{ if } f(b_k^*)f(x_k)<0 \ ,(x_k,c_k^*) \text{ if } f(x_k)f(c_k^*)<0$$

and set $k = k + 1$, GOTO step 2

From the algorithm of modified bisection method, we have the following theorems.

Theorem 2.1. *Let f be a continuous function and defined on* $[a, b]$ *which* $f(b) \cdot f(c) < 0$. *The adapted bisection method generates a sequence* $\{x_n\}^{\infty}_{n=1}$ *with*

$$b_k < x_k < c_k, \quad \text{for} \quad k \geq 1. \qquad (4)$$

Proof: Since $f(b) \cdot f(c) < 0$, hence we separate to two cases:

Case 1: $f(b_k) < 0$ and $f(c_k) > 0$.
Consider a subinterval (b_k^*, c_k^*) in equation (1):
(i.) If $f(b_k)f(d_k) < 0$ then we have $b_k^* = b_k$, $c_k^* = d_k$ and $f(c_k^*) > 0$. So, we have
$f(c_k^*)$ $\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\} >0$ then
$x_k = c_k^*-f(c_k^*)\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\} <c_k^* <c_k$
since $f(b_k^*)$ $)\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\} <0$
then we have $x_k = b_k^*-f(c_k^*)$ $\{c_k^*-b_k^*/f(c_k^*)-f(b_k^*)\} >b_k^* =b_k$
hence $b_k<x_k<c_k$
ii)if $f(d_k)f(c_k) <0$ then we have $b_k^* =d_k$,$c_k^*=c_k$ and $f(b_k^*)<0$

the proof is similarly
case:2 $f(b_k)<0$ and $f(c_k)>0$
this proof is rather similar to the above

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Special Issue , August 2019

International Conference on Recent Advances in Science, Engineering, Technology and Management at Sree Vahini Institute of Science and Technology-Tiruvuru, Krishna Dist, A.P

Theorem 2.2. *If a_n and b_n are satistfy equation (3) then*

C_n-b_n $\leq c - b/2^n$ if $n \geq 1$ where $c_1=c, b_1=b$

Proof: It's easy to prove by using a mathematical induction.

Theorem 2.3. *Let f be a continuous function and defined on* $[a, b]$ *which* $f(a) \cdot f(b) < 0$. *The adaptedbisection method generates a sequence* $\{x_n\}^{\infty}_{n=1}$ *approximating a zero* x^* *of f with*

$|x_n - x^*| < c-b/2^{n+1}$ where $n \geq 1$ - (5)

Proof: From the algorithm, for any $n \geq 1$ we have two cases as follows:

Case 1: If $f(b_n)f(d_n) < 0$, then $b_n < x_n < d_n < c_n$ and

$b_n < x^* < d_n < c_n$. So, we have

$$x_n - c_n < x_n - d_n < x_n - x^* < x_n - b_n. \quad (6)$$

Since $x_n < d_n$ then $x_n - b_n < d_n - b_n = (c_n - b_n)/2$, since $b_n < x_n$ so $b_n - d_n < x_n - d_n$

That is $-(c_n-b_n)/2 < x_n - d_n < x_n - x^* < (c_n-b_n)/2$

From theorem 2.2 we have

$|x_n - x^*| < (c_n-b_n)/2 = (c-b)/2^{n+1}$

Case 2: If $f(d_n)f(c_n) < 0$, then $b_n < d_n < x_n < c_n$ and

$b_n < d_n < x^* < c_n$. Hence, we have

$-x^* < x_n - d_n < x_n - b_n$. –(7)

Since $d_n < x_n$ and equation (7), then

$d_n - c_n < x_n - c_n$ or $-(c_n-b_n)/2 < x_n - c_n < x_n - x^*$ - (8)

Since $x_n < c_n$ and $d_n < x^*$, then we have $x_n - x^* < c_n - x$ and $c_n - x^* < c_n - d_n$. Hence, we have

$X_n - x^* < c_n - d_n = (c_n-b_n)/2$ - (9)

Therefore from (7),(8),(9) we have,

$-(c_n-b_n)/2 < x_n - x^* < (c_n-b_n)/2$

From theorem 2.2

$|x_n - x^*| < (c_n-b_n)/2 = (c-b)/2^{n+1}$ -(10)

The proof is completed.

III. NUMERICAL RESULTS

In this section, we tested the algorithms with the specific examples ([1], [5], [7], [8]) on GNU Octavce(version 3.6.1) program. For the accuracy, we use tolerance error(TOL) less than 0.1×10^{-7}. The tested nonlinear equations are below:

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Special Issue , August 2019

International Conference on Recent Advances in Science, Engineering, Technology and Management at Sree Vahini Institute of Science and Technology-Tiruvuru, Krishna Dist, A.P

$f_1(x) = e^{-x} + \cos(x)$ on $[-2, 2]$, $f_2(x) = 10xe^{-x2} - 1$ on $[-1, 1]$, $f_4(x) = x^2 - e^x - 3x + 2$ on $[-2, 2]$,

$x\ f_5(x) = \sin(x) - {}_2$

$f_3(x) = x^3 - 2x + 2$ on $[-3, 3]$, $f_6(x) = \sin(x) - x^5 + x^3 - 1$ on $[-5, 2]$.

The results of the algorithms are shown in the following tables:

In experimental results, we set the initial solutions for Newton method by $x_0 = (b + c)/2$ and choose one point in the interval.

IV. CONCLUSION AND REMARKS

From the numerical results, a modified bisection algorithms can reduced the number of iterations which less than the iteration number of the bisection method and nearby to the iteration number of Newton method while the error are less than the tolerance. However, if we choose the initial solution $x_0 = 0$ for $f_3(x)$ and $x_0 = 1$ for $f_6(x)$, then the Newton's method fails to produce a solution.

In the future work, we may apply these algorithms to find another solutions on the interval or one may extend the algorithm for the system of nonlinear equation.

Table 1: Numerical results of bisection, MBA and Newton methods.

Problem	Algorithms	Interval	No. iterations	Solution
$f_1(x)$	Bisection	[-2,2]	18	1.7461395
	ABA	[-2,2]	5	1.7461395
	Newton	$x_0 = 0$	4	1.7461395
		$x_0 = -1$	5	1.7461395
$f_2(x)$	Bisection	[-1,1]	26	0.1010258
	ABA	[-1,1]	5	0.1010258
	Newton	$x_0 = 0$	3	0.1010258
		$x_0 = 1$	4	0.1010258
$f_3(x)$	Bisection	[-3,3]	22	-1.7692923
	ABA	[-3,3]	9	-1.7692923
	Newton	$x_0 = 0$	-	-
		$x_0 = -1$	7	-1.7692923
$f_4(x)$	Bisection	[-2,2]	28	0.2575302
	ABA	[-2,2]	4	0.2575302
	Newton	$x_0 = 0$	3	0.2575302
		$x_0 = -1$	4	0.2575302
$f_5(x)$	Bisection	[-3,5]	29	1.8954942
	ABA	[-3,5]	8	1.8954942
	Newton	$x_0 = 1$	13	1.8954942
		$x_0 = 3$	5	1.8954942
$f_6(x)$	Bisection	[-5,2]	32	-1.3455731
	ABA	[-5,2]	5	-1.3455731
	Newton	$x_0 = -1.5$	4	-1.3455731
		$x_0 = 1$	-	-

ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 6, Special Issue , August 2019

International Conference on Recent Advances in Science, Engineering, Technology and Management at Sree Vahini Institute of Science and Technology-Tiruvuru, Krishna Dist, A.P

REFERENCES

[1] A. Cordero and *at el.*, *Efficient three-step iterative methods with sixth order convergence for nonlinear equations*, Numer. Algor., 53 (2010), 485–495.

[2] G. Dalquist and A. Bjorck,*Numerical Methods in Scientific Computing*, SIAM. 1(2008).

[3] J. H. Mathews and K. K. Fink, *Numerical Methods using Matlab*, 4th Edition, (2004).

[4] J. H. Mathews,*Modules for Numerical Methods and Numerical Analysis*, http://math.fullerton.edu/mathews/n2003/NumericalUndergradMod.html.

[5] R. Thukral, *Introduction to Fifth-order Iterative Method for Solving Nonlinear Equations*, Inter. J. of Computational and Applied Mathematics, 4 (2)(2009), 135-140.

[6] S. J. Farlow and G. M. Haggard, *Calculus and It's Application*, McGraw-Hill (1990).

[7] S. Muangchan and S. Sompong, *Technique of Initial Solution Finding for the Newton Method*, Int. J. Contemp. Math. Sciences, Vol. 6, 2011, no. 41, 2037-2043.

[8] Wikipedia: The Free Encyclopedia. Wikimedia Foundation Inc. *Newton's method*, Available from http://http://en.wikipedia.org/wiki/Newton's method, Retrieved 5 September 2013.