Approximation of Maximum Power Transfer Point of an Electrical Circuit using Bracketing methods and Golden Section Search

[1]Kodathala Sai Varun
[1]Department of Electronics and Communication Engineering,
GITAM School of Technology,
GITAM Deemed to be University – Bengaluru, Karnataka.

Abstract – In this paper, the maximum power transfer point of an any equivalent electrical network is calculated from the theory of maximum power transfer and one dimensional golden section algorithm to minimize the single variable function. The paper is composed of two techniques one employing the root approximation using bracketing and open methods, and other involving the optimization algorithm called golden section search (GSS).

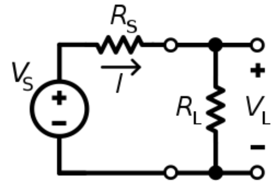
Keywords – Golden Section, bisection, false position, Newton Iterative

I. INTRODUCTION

The analysis of any electrical network refers to estimation of maximum power drawing capability under certain conditions. It is required to estimate the condition at which the maximum power is drawn by the load impedance in order to acquire maximum efficiency of electrical system. The condition need to be precised to single variable in order to maintain the condition, therefore the theory of Thevenin circuit is employed here.

Thevenin theorem states that any electrical network can be converted to a circuit with voltage source with an equivalent resistance in series. The load resistance is connected to end terminals of converted circuit, usually it would be a variable resistor whose value is altered to obtain the maximum power from the source. Therefore the objective of the work is to determine the value of variable resistor which drives maximum power.

The maximum power transfer theorem states that load resistance should possess the value of Thevenin equivalent resistance in order to obtain maximum power. The solution of equation is given according to first derivate principle. The analysis can be inferred as follows:



From the hypothesis we obtain

$$R_L = R_s$$

We can rearrange the equation as follows:

$$R_L - R_s = 0$$

The above equation can be evaluated using bracketing and open methods which are bisection and newton iterative methods.

Bisection Method:

The formal and simplest method to determine the root of an equation is bisection method. It could be used only if the function is continuous and the range of equation is known. Here the range of equation corresponds to the sign change in the value of function. That is if the range is given as $[a, b]$, then the function follows the following property- $sign(f(a)) = -sign(f(b))$. If $f(a)$ is positive then $f(b)$ is negative and vice versa.
Then the approximation of first iteration is given by the mean of the maximum and minimum values, therefore the root is given for first iteration is $r = \frac{a+b}{2}$. Then the $f(r)$ is computed and

corresponding range is computed. That is if the function lie in the range $\left[a, \frac{a+b}{2}\right]$ the range is modified from $[a, b]$ to $\left[a, \frac{a+b}{2}\right]$ and if the function lie in the range $\left[\frac{a+b}{2}, b\right]$ then the range is modified from $[a, b]$ to $\left[\frac{a+b}{2}, b\right]$. For the second iteration again the mean of the maximum and minimum is taken and computed for its range and the procedure is repeated until the accurate result is achieved [9].

Advantage of the Bisection Method

The convergence of root is certain though it is slow process because the method is implied by halving the length of bracketing interval containing the root.

Disadvantage of the Bisection Method

Though the convergence of this method is guaranteed, its rate of convergence is slow and it is difficult to extend the use for system of equations.

Newton Raphson Method

Newton – Raphson method is also known as fixed – point iteration method. This method requires that the function be differentiable. If the function f(x) is differentiable on the domain of function, and r_0 is the initial guess, then the first approximation r_1 is obtained by intersection of tangent at $(r_0, f(r_0))$ with the x-axis, and is defined by [10]

$$r_1 = r_0 - \frac{f(r_0)}{f'(r_0)}$$

And the successive approximations are

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}$$

Advantage of Newton – Raphson Method

The convergence rate is linear and this method is very fast compared to the bisection, false position and secant method.

Disadvantage of Newton – Raphson Method

The only drawback of Newton Raphson method is it fails if the derivative, $f'(x)$ is near zero at some iteration.

The error which is used as an parameter in this paper is formulated as follows

$$\in_r = \frac{(present\ root - past\ root)}{present\ root} * 100\ \%$$

The approach to determine the maximum condition is evaluating the power function to calculate the inverse of minimum point using golden section algorithm.

The equation is given as follows:

$$P = I^2 * R_L$$

Where $I = \frac{V_s}{R_s + R_L}$

Therefore, $P = \left(\frac{V_s}{R_s + R_L}\right) * \left(\frac{V_s}{R_s + R_L}\right) * R_L$

The above equation is function of R_L therefore it determines R_L for maximum power condition.

The golden section is a method of searching for a minimum (searching for a maximum) of a unimodal function. Unlike finding a zero, where two function evaluations with opposite sign are sufficient to bracket a root, when searching for a minimum, three values are necessary. The golden-section search is an efficient way to progressively reduce the interval locating the minimum. The key is to observe that regardless of how many points have been evaluated, the minimum lies within the interval defined by the two points adjacent to the point with the least value so far evaluated.

II. METHODOLOGY

1. Bisection Method:

Theorem 1. *Let f(x) be a continuous function and (a,b) be a sufficiently small interval such that f(a)f(b)<0. Then the approximation of a root of f(x) can be determined using the formula given by*

$$r_n = \frac{a+b}{2}$$

If $f(r_n) < 0$ then $a = r_n$ and
If $f(r_n) > 0$ then $b = r_n$. The procedure is iterated till root is approximated to near root.

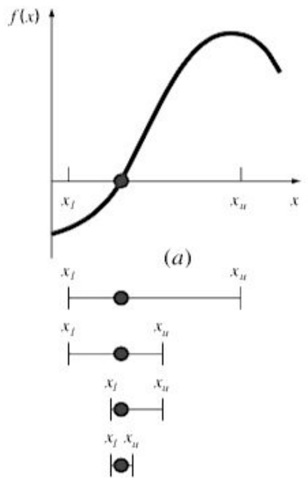


Figure 1: Bisection Method abstraction

Illustration 2.1. consider an equation

$$x^6 - x - 1 = 0$$

This equation has two real roots -0.7780895987 and 1.134724138. The following Table 1 shows the iterations with initial approximations $a = 1$ and $b = 1.5$.

Table 1. Approximation using Bisection Method

Iterations	Root
1	1.25000000000000
2	1.12500000000000
3	1.18750000000000
4	1.15625000000000
5	1.14062500000000
6	1.13281250000000
7	1.13671875000000
8	1.13476562500000
9	1.13378906250000
10	1.13427734375000
11	1.13452148437500
12	1.13464355468750
13	1.13470458984375
14	1.13473510742188

One can observe from Table 1 that the bisection method gives approximate root up to 4 decimal points after 14 iterations.

2. Newton Iterative Method

Theorem 2. *Let f(x) be a continuous function and (a,b) be a sufficiently small interval such that f(a)f(b) < 0 and the f'(x) exists for the range. Then the approximation of a root of f(x) can be determined using the iterative formula given by*

$$r_0 = a$$
$$r_1 = b$$

$$r_2 = r_1 - \frac{f(r_1)}{f'(r_1)}$$

$$r_n = r_{n-1} - \frac{f(r_{n-1})}{f'(r_{n-1})}$$

Consider the same illustration given by 2.1. The following Table 2 shows the iterations.

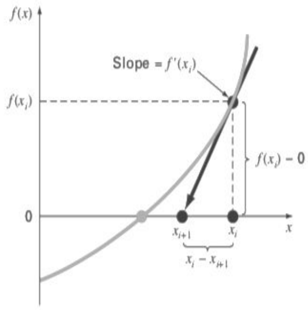


Figure 2: Newton Iterative Method abstraction

Table 2. Approximation using Newton Raphson Method

Iterations	Root
1	1.68062827225131
2	1.43073898823906
3	1.25497095610944
4	1.16153843277331
5	1.13635327417051
6	1.13473052834363
7	1.13472413850022
8	1.13472413840152
9	1.13472413840152
10	1.13472413840152
11	1.13472413840152
12	1.13472413840152
13	1.13472413840152
14	1.13472413840152

One can observe from Table 2 that the Newton Raphson method gives approximate root up to 4 decimal points after 6 iterations.

3. Golden Section Search

The method starts with two initial guesses, x_l and x_u, that bracket one local extremum of $f(x)$. Next, two interior points x_1 and x_2 are chosen according to the golden ratio,

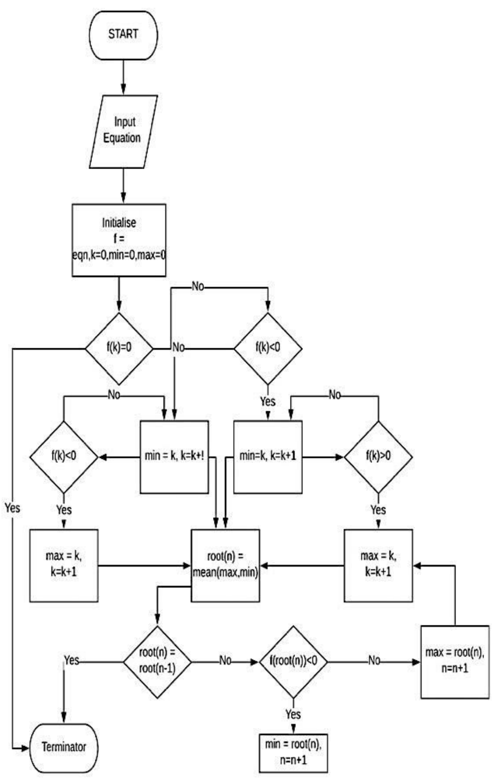
$$x_1 = x_l + d$$
$$x_2 = x_u - d$$

The function is evaluated at these two interior points. Two results can occur:

1. If, as is the case in Fig. 13.4, $f(x_1)$. $f(x_2)$, then the domain of x to the left of x_2, from x_l to x_2, can be eliminated because it does not contain the maximum. For this case, x_2 becomes the new x_l for the next round.
2. If $f(x_2)$. $f(x_1)$, then the domain of x to the right of x_1, from x_1 to x_u would have been eliminated. In this case, x_1 becomes the new x_u for the next round.

III. IMPLEMENTATION USING MATLAB

Flow Chart for Bisection Method:

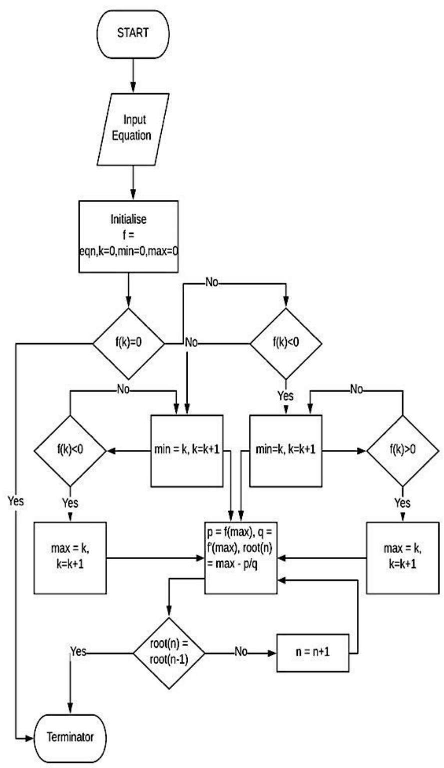


Algorithm for Bisection:

Algorithm 1: Bisection Method

1. Input: f, [a,b], iterations
2. Output: root r
3. //Initialize
4. k= 0
5. repeat
6.   //Computing the mid point
7. $r_k = \frac{a+b}{2}$
8.   //Compute the function value of root
9.   if $f(r_k) > 0$
10.   b = r_k
11.   endif
12.   if $f(r_k) < 0$
13.   a = r_k
14.   endif
15.   iterationCount = k
16. $r = r_k$
17.   k = k+1
18. until k > iterations

Flow chart of Newton Iterative Method:



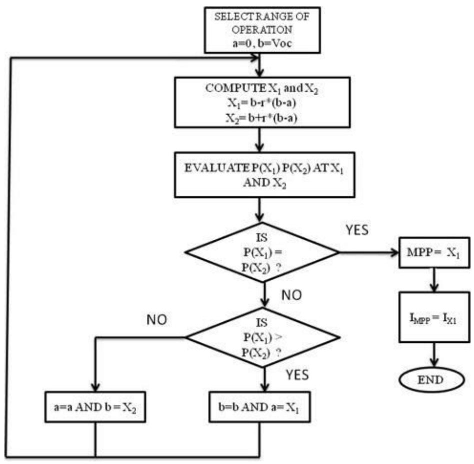
Algorithm for Newton Raphson Method:

Algorithm 2: Newton Raphson Method

---

1. Input: f, [a,b], iterations
2. Output: root r
3. //Initialize
4. k= 0
5. repeat
6. //Computing the Tangent Line
7. $r_k = r_{k-1} - \dfrac{f(r_{k-1})}{f'(r_{k-1})}$
8. iterationCount = k
9. $r = r_k$
10. k = k+1
11. until k > iterations

Flow chart for Golden Section Search:



Algorithm for Golden Section Search:

1. Specify the function to be minimized, f(x), the interval to be searched as $\{X_1,X_4\}$, and their functional values F_1 and F_4.
2. Calculate an interior point and its functional value F_2. The two interval lengths are in the ratio *c:r* or *r:c* where $r=1-\varphi$ and $c=1-r$, with φ being the golden ratio.
3. Using the triplet, determine if convergence criteria are fulfilled. If they are, estimate the X at the minimum from that triplet and return.
4. From the triplet, calculate the other interior point and its functional value. The three intervals will be in the ratio *c:cr:c*.
5. The three points for the next iteration will be the one where F is a minimum, and the two points closest to it in X.
6. Go to step 3

Program for Root Approximation:

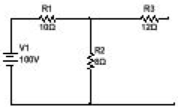
```
function
[rb,vb,rn,vn,vf,rf,nb,nf,nn,vnb,vn
f,vnn] =
max_power_nm(R1,R2,R3,Vin)
    f1 = @(Rth) Rth -
(R1*R3)/(R1+R3) - R2;
    f2 = @(Vth) Vth -
(Vin*R3)/(R1+R3);
    %%Bisection
    [rb,nb] =
bisection_method_deci(f1);
    [vb,vnb] =
bisection_method_deci(f2);
    %%False Position
    [rf,nf] = false_position(f1);
    [vf,vnf] = false_position(f2);
    %%Newton Raphson
    [rn,nn] =
newton_iterative(f1);
    [vn,vnn] =
newton_iterative(f2);
end
```

Program for Golden Section Search:

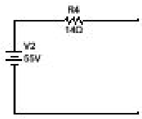
```
function [r,n] =
golden_section_user(eqn,a,b)
f = eqn;
GR = 0.618;
n=1;
x = 0:(b+a)
plot(x,f(x));
hold on;
while(1)
    d = GR*(b-a);
    x1 = a+d;
    x2 = b-d;
    if(f(x1)>f(x2))
        b=x1;
    else
        a=x2;
    end
    root(n) = (a+b)/2;
    stem(root(n),f(root(n)));
    if(n>1)
        er(n) = abs((root(n) -
root(n-1))/root(n))*100;
        if(er(n)<0.0001)
            r = root(n);
            break;
        end
    end
    n = n+1;
end
```

IV. RESULTS AND DISCUSSIONS

Consider the electrical network as follows:



The equivalent Thevenin circuit is given by



The equations are resulted as follows:

$$V_{th} = V_{in} * \frac{R_3}{R_3+R_1}$$

$$R_{th} = \frac{R_1 * R_3}{R_1+R_3} + R_2$$

$$R_{th} - \frac{R_1 * R_3}{R_1+R_3} + R_2 = 0$$

The value of R_{th} can be estimated using bisection, false position and newton iterative methods. Results are furnished as follows:

Method	Bisection	False Position	Newton Iterative
Iterations	17	2	2

$$V_{th} = 55\ V$$

$$R_{th} = 14\ \Omega$$

The solution of problem using golden section search optimization and results are furnished as follows:

$$R_{th} = 13.9999\ \Omega \text{ and } Iterations = 27$$

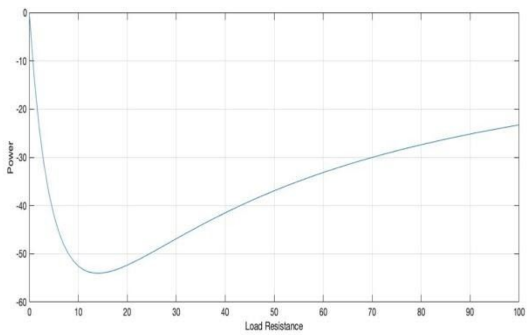


Figure 4: Inversion Graph

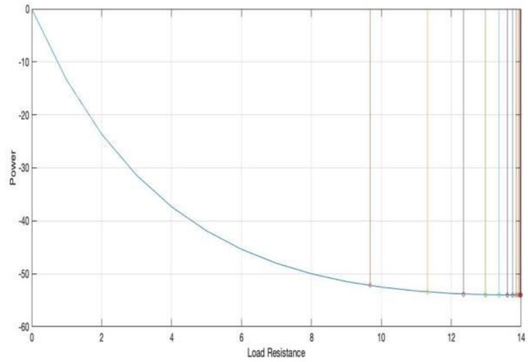


Figure 5: Golden Search Optimization

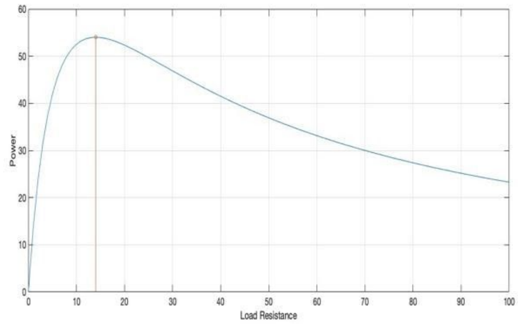


Figure 3: Result from GSS

V. CONCLUSION

The methods used to derive the maximum power transfer point are described and compared for further analysis. The one method among two methods define different bracketing and open methods, and comparison among all these methods prove that newton iterative method is more convenient than bisection method in terms of number of iterations. The golden section search optimization is quicker compared to other root approximation techniques.

VI. REFERENCES

[1] Jaime Gutierrez, Josef Schicho, Martin Weimann, " *Computer Algebra and Polynomials* ", Applications of algebra and number theory, Springer, Cham, 2015.

[2] Richard W. Hamming. 1973. "*Numerical Methods for Scientists and Engineers*". McGraw-Hill, Inc., USA.

[3] Datta, B.N. *Lecture Notes on Numerical Solution of Root Finding Problems*. 2012. Available online: **www.math.niu.edu/~dattab** (accessed on 15 January 2019).

[4] Calhoun, D. Available online: **https://math.boisestate.edu/~calhoun/teac**

hing/matlab-tutorials/lab_16/html/lab_16.html (
accessed on 13 June 2019).
[5] Thinzar, C.; Aye, N. *Detection the storm movement by sub pixel registration approach of Newton Raphson method*. *Int. J. E Educ. E Bus. E Manag. E Learn.* **2014**, *4*, 28–31. [Google Scholar]
[6] Ali, A.J.M. *The application of numerical approximation methods upon digital images*. *Am. J. Signal Process.* **2017**, *7*, 39–43. [Google Scholar] [CrossRef]
[7] Bruck, H.A.; McNeill, S.R.; Sutton, M.A.; Peters, W.H. *Digital image correlation using Newton-Raphson method of partial differential correction*. *Exp. Mech.* **1989**, *29*, 261–267. [Google Scholar] [CrossRef]
[8] Cofaru, C.; Philips, W.; van Paepegem, W. *Pixel-level robust digital image correlation*. *Opt. Express* **2013**, *21*, 29979–29999. [Google Scholar] [CrossRef] [PubMed]
[9] Claudio Gutierreza, FlavioGutierrezb, Maria-Cecilia Rivaraa, *Complexity of the bisection method*, Theoretical Computer Science Volume 382, Issue 2, 31 August 2007, Pages 131-138
[10] Diachao Sheng, Scott W. Sloan, Andrew J. Abbo, *An Automatic Newton-Raphson Scheme*, International Journal of Geomechanics, Vol. 2, Issue 4 (October 2002)