

An algorithm for Complex Solution of Non-Linear Equations in One Variable Using Fixed Point Iteration Method

Aniruddha Lomate¹, Omkar Joshi², Santosh Joshi³

¹U.G. Student, ³Assistant Professor, Department of Mechanical Engineering,

²U.G. Student, Department of Production Engineering,

Vishwakarma Institute of Technology, Pune, Maharashtra, India.

Abstract: An algorithm of fixed point iteration method to find complex roots of non-linear equations in one variable is presented. The methods of functional iterations can be applied to find the complex roots with simple modifications. The main objective while deploying the new algorithm is to obtain complex root with help of even root of a negative value of $h(x)$. The algorithm makes sure to enter the complex plane and hence, eliminating the calculations in real domain if any positive value of $h(x)$ is considered. The algorithm provides only iterative method avoiding the pre-calculations of recurrence relations in Bairstow's method and gives both real as well as complex roots with appropriate choice of initial approximation. The numerical results are demonstrated after satisfying certain termination criterion of performing number of iterations. The proposed method of successive approximation employs a single real initial approximation unlike the case of Bairstow's method where two initial guesses as the values of r and s are considered.

Keywords: Non-linear equations in variable, Functional iteration, Complex roots, Numerical analysis, Successive approximation, Single initial approximation.

I. INTRODUCTION

Developments of numerical methods to find the complex roots of algebraic polynomials are important. The numerical methods such as, Bisection method, Regula-Falsi method, Newton-Raphson method and Secant method are helpful to locate the real roots of non-linear equations with real coefficients. Every numerical method has its own advantages and disadvantages. Bairstow's method involves more amount of pre-calculations of solving recurrence relations simultaneously in the loop. Moreover, there is need of actual long division when the polynomials with degree greater 3 are dealt. This long division process and division by zero while solving recurrence relations simultaneously can be avoided in the proposed algorithm of successive approximation.

A fundamental principle of numerical analysis and computer science is iteration. In this paper, an algorithm to locate the complex roots of non-linear equations using fixed point iteration method is presented. The successive approximation for locating the roots of an equation essentially needs an initial approximation. Then a sequence of values P_k is obtained using a specific iterative rule. Ideally, this initial approximation is to be chosen wisely in the vicinity of root of the equation, otherwise, the iterates may diverge. But, in the case of complex root finding using the successive approximation, the idea of location of root is of second the importance as the initial approximation is not necessarily close to one of the roots.

In our case, the sequence has pattern,

$$P_{k+1} = g(P_k)$$

A fixed point of a function $g(x)$ is a real number P such that $P = g(P)$. The iteration $P_{n+1} = g(P_n)$ for $n = 0; 1; 2 \dots$ is called fixed-point iteration.

II. A NEW ALGORITHM TO FIND THE COMPLEX ROOTS OF NON LINEAR EQUATIONS USING SUCCESSIVE APPROXIMATION

We write a polynomial in the form,

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0.$$

in which the coefficients a_k are real numbers and x is a variable. If $a_n \neq 0$, then p has degree n .

A. Check for Complex Solution of Given Non-Linear Equation

- 1) **Theorem 1:** Descartes' rule of signs: It states that if the terms of a single variable polynomial with real coefficients are ordered by descending variable exponent, then the number of positive roots of the polynomial is either equal to the number of sign differences between consecutive non-zero coefficients, or is less than it by an even number.
- 2) **Corollary 1:** As a corollary of the rule, the number of negative roots is the number of sign changes after multiplying the coefficients of odd-power terms by negative one (- 1), or fewer than it by an even number.
- 3) **Corollary 2:** Any n^{th} degree polynomial has exactly n roots in the complex plane, if counted according to multiplicity. So if $f(x)$ is a polynomial which does not have a root at zero then the minimum number of non-real roots is equal to $(n - p + q)$, where p denotes the maximum number of positive roots, q denotes the maximum number of negative roots and n denotes the degree of the equation.

Thus, Descartes rule of signs is helpful to determine if a non-linear equation has complex roots. In case, the non-linear equation has only real roots, the real roots are to be calculated using fixed point iteration method along with suitable initial guess. The other case focuses on the non-real roots of the equation. Some rearrangement of terms of equation (1) is to be carried out so that complex root is obtained after substitution of initial approximation. The equation (1) is to be rearranged in such a way that,

$$a_n x^n = -(a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + \dots + \dots + a_2 x^2 + a_1 x + a_0) \dots (2)$$

if degree n is even.

$$a_{n-1} x^{n-1} = -(a_n x^n + a_{n-2} x^{n-2} + \dots + \dots + \dots + a_2 x^2 + a_1 x + a_0) \dots (3)$$

if degree n is odd. The equation (2) or equation (3) is rewritten in the form of fixed iteration i.e. $x = g(x)$.

$$x = g(x) = [h(x)]^{1/k} \dots \dots (4)$$

where, $k = n$ when degree n is even, or $k = n - 1$ when degree n is odd. i.e. $k = 2,4,6..(i.e. \text{even})$.

B. Criterion of Initial Approximation and Proposed Fixed Point Iteration Method

The selection of initial guess plays crucial role in finding the complex root of the equation. The initial approximation (x_0) is to be chosen in such a way that the value of $h(x)$ is negative. Generally, the initial approximation is chosen close to the one of roots which lies on real line, but in our case, the location of the root is not necessarily required to decide the initial guess. There exists some $h(x)$ which can never be negative for any value of x as initial approximation. In such cases proposed method fails to produce complex roots initially. Hence, the tentative check for the value of $h(x)$ to be negative should be carried out before proceeding further. The check needs basic inputs as the degree of $h(x)$ and the nature and number of roots of $h(x)$. The polynomial $h(x)$ with odd degree would always be having atleast single negative value of $h(x)$. But, $h(x)$ with even degree should possess atleast two real roots in order to have negative value of $h(x)$.

Since, the algorithm deals with complex calculations, it becomes cumbersome to decide complex initial guess. An important factor to be noticed while deploying the new algorithm is that the initial approximation is real though the complex roots are to be located. In some cases like constant coefficient of $h(x)$ is negative, deciding initial approximation becomes easy part. Clearly, the initial approximation in such case is zero. The plot of function $h(x)$ using *MATLAB* helps to choose the initial approximation. The algorithm involves plotting the graph of $h(x)$ in a specific interval. The even-root of a negative number leads to a complex number.

$$x_0 = g(x_0) = z_0 \rightarrow j_0 \dots (5)$$

where, z_j is corresponding approximated complex solution at j^{th} iteration. The algorithm iterates to give corresponding approximated complex solution till certain termination criterion is satisfied.

$$z_b = p + iq \dots \dots (6)$$

where b is performed iteration number.

- 1) **Theorem 2:** If p is a polynomial whose coefficient are all real, and if w is non real root of p , then \bar{w} is also a root and $(x - w)(x - \bar{w})$ is a real quadratic factor of p .

Thus, $z_k = p - iq$ is another complex root.

The algorithm further suggests to repeat the process considering the next even powered term. i.e.

$$a_{n-2} x^{n-2} = -(a_n x^n + a_{n-1} x^{n-1} + a_{n-3} x^{n-3} + \dots + \dots + \dots + a_2 x^2 + a_1 x + a_0)$$

if the degree n is even.

$$a_{n-3} x^{n-3} = -(a_n x^n + a_{n-2} x^{n-2} + a_{n-1} x^{n-1} + \dots + \dots + \dots + a_2 x^2 + a_1 x + a_0)$$

if the degree n is odd. Different complex root ($z_c = r + is$) may be obtained with suitable initial approximation at required c^{th} iteration. ($z_c = r - is$) is another complex root at the given iteration.

III. MATLAB CODE

In this section, we will consider important points included in the MATLAB code which localize the complex roots conveniently with the help of successive approximation method. As earlier discussed, the main check to decide initial approximation is that the negative value of $h(x)$. The code provides choice to plot the graph of function $h(x)$ in the interval entered by user and enables user to check various intervals leading to easier determination of initial guess. At the end, code provides all details such as iteration number at which the process is terminated, solution at the iteration, main function value and relative errors. This combined detailed data helps user regarding the further analysis of roots, errors and their behaviour in case of successive approximation.

IV. APPLICATIONS

We present some examples to illustrate the proposed method. The numerical results are summarized in the table 1.

A. Example 1. Consider the equation

$$p(x) = x^5 - 3.5x^4 + 2.75x^3 + 2.125x^2 - 3.875x + 1.25$$

Descartes' rule of signs helps to know that the $p(x)$ has at least two complex roots.

$$x = g(x) = (0.2875x^5 + 0.7578x^3 + 0.6071x^2 - 1.1071x + 0.3571)^{(1/4)}$$

$$h(x) = 0.2875x^5 + 0.7578x^3 + 0.6071x^2 - 1.1071x + 0.3571$$

Here, $k = 4$ (even).

As earlier discussed, we need initial approximation such that value of $h(x)$ is negative. Plotting function $h(x)$ in an interval of (-5) to (5) helps us to locate initial guess. Here initial guess (x_0) is taken as -1:5. The table 1 gives detailed description of numerical results performed using successive approximation.

At 47th iteration (j_{47}), $p(x_{47}) = p(1.010 + 0.4956i) = (0.0022 - 0.0176i)$:

The relative error in the real part at 47th iteration is 0.0869.

The relative error in imaginary part at 47th iteration is 0.0579.

This solution at 47th iteration is close to the root calculated by Bairstow's method i.e. $(1 + 0.499i)$

At 131th iteration (j_{131}), $p(x_{131}) = p(1.0000 + 0.5000i) = (0.00011199 - 0.000013764i)$.

The actual roots of $p(x)$ are 0.5, -1.0, 2.0, $(1 + 0.5i)$ and $(1 - 0.5i)$.

Table 1 Numerical Results of Example 1

j_k	x_k
1	0.7743 + 0.7743i
2	0.8268 + 0.7554i
3	0.8665 + 0.7362i
...
...
46	1.0109 + 0.4960i
47	1.010 + 0.4956i
...
...
130	1.0 + 0.5001i
131	1.0 + 0.5000i

V. CONCLUSION

A new algorithm of finding complex solution of non-linear equations in one variable using fixed point iteration method is studied. Some applicability conditions are taken into consideration to check if the values of $h(x)$ are negative. The new algorithm enters the



iterative part directly with a single initial guess unlike the case of Bairstow's method where number of pre-calculations are performed using recurrence relations using two initial guess as r and s .

VI. ACKNOWLEDGMENT

We thank the Department of Mechanical Engineering and Department of Production Engineering, Vishwakarma Institute of Technology, Pune for the guidance and support provided for carrying out this work.

REFERENCES

- [1]. J. Mathews, K. Fink, Numerical Methods Using MATLAB, Pearson Education., 2006.
- [2]. D. Kincaid, W. Cheney, "Numerical Analysis Mathematics of Scientific Computing," Third edition.
- [3]. S. C. Chapra, R. Canale, "Numerical Methods of Engineers," Mc Graw Hill Education., Seventh edition.
- [4]. S. Kang et al., "A New Second-Order Iteration Method for Solving Nonlinear Equations," Hindawi Publishing Corporation Abstract and Applied Analysis Volume 2013, Article ID 487062.