Solving for Roots of Nonlinear Equations by Taylor Expansion

Apichat Neamvonk

Department of Mathematics, Faculty of Science, Burapha University,
Chonburi, THAILAND 20131.
Email: apichat {at} buu.ac.th

ABSTRACT— ***This paper illustrates an iterative numerical method to find roots of nonlinear equation in a form of*** $f(x)=0$ ***by using 2nd and 3rd order Taylor expansion. The numerical results show that this iteration method is faster than Newton–Raphson, hybrid iteration and the new hybrid iteration method. Also this iteration method needs less than number of functional evaluations than the others.***

Keywords— Nonlinear equation, Taylor expansion, Newton-Raphson Method.

1. INTRODUCTION

Many problems in Science and Engineering mostly contain nonlinear equations in a form of $f(x)=0$. As some numeric problems are not possible to be solved their exact solutions by using algebraic processing numerical iterative methods, for example Newton-Raphson method (with a 1st order Taylor expansion)

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

are often used to find the approximate solution of the problems. There are many numerical methods have been developed based on improvement of quadratic convergent Newton's Method. We attempt to get a higher convergence order than that. This present work find converting root of a given nonlinear equations using a new numerical iterative scheme through Taylor's expansion. By obtaining a well efficient method, highly convergent method not only does highly convergent faster than conventional methods but also less iteration steps and number of functional evaluations than the others. Then those of recently proposed iterative methods, such as hybrid iteration method [1], new hybrid iteration method [2] and Newton's method [3] have been compared.

2. PRELIMINARY

Consider the nonlinear equation

$$f(x)=0\ , \tag{1}$$

where f is continuous function on the open interval I and defined 1st, 2nd and 3rd derivative as f', f'' and f'''. We consider the Taylor expansion of $f(x)$,

$$f(x) = f(x_i) + \frac{(x-x_i)}{1!} f'(x_i) + \frac{(x-x_i)^2}{2!} f''(x_i) + \frac{(x-x_i)^3}{3!} f'''(x_i) + \dots\ , \tag{2}$$

where x_i is the approximation to the roots of (1). Let γ is the exact root of (1), so (2) can be written by

$$f(\gamma) = f(x_i) + \frac{(\gamma-x_i)}{1!} f'(x_i) + \frac{(\gamma-x_i)^2}{2!} f''(x_i) + \frac{(\gamma-x_i)^3}{3!} f'''(x_i) + \dots . \tag{3}$$

Using (1), we get

$$0 = f(x_i) + \frac{(\gamma-x_i)}{1!} f'(x_i) + \frac{(\gamma-x_i)^2}{2!} f''(x_i) + \frac{(\gamma-x_i)^3}{3!} f'''(x_i) + \dots . \tag{4}$$

Consider three (2nd order) and four (3rd order) terms of (4), so we get (5) and (6) respectively

$$0 = f(x_i) + \frac{(x_{i+1}-x_i)}{1!} f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!} f''(x_i) + O\left[(x_{i+1}-x_i)^3\right]. \tag{5}$$

$$0 = f(x_i) + \frac{(x_{i+1}-x_i)}{1!} f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!} f''(x_i) + \frac{(x_{i+1}-x_i)^3}{3!} f'''(x_i) + O\left[(x_{i+1}-x_i)^4\right]. \quad (6)$$

At the first step we define $i = 0$. We substitute an initial guess point x_0, $f(x_0)$, $f'(x_0)$, $f''(x_0)$ and $f'''(x_0)$ into (5) and (6). Then these equations will be 2nd and 3rd order nonlinear equations, respectively. So we can solve x_1 which is a new iteration of approximation root by using MATLAB

$$x_{i+1} = \text{solve}\left(f(x_i) + \frac{(x_{i+1}-x_i)}{1!} f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!} f''(x_i) + \frac{(x_{i+1}-x_i)^3}{3!} f'''(x_i) \right). \quad (7)$$

where $i = 0,1,2,3,\ldots$. We repeat the iteration (7) and x_i will converge to root, γ, if the starting point x_0 is close enough to γ. This process has the local convergence property.

3. ALGORITHM

Step 1: Define $i = 0$, set initial guess point x_0 and tolerance $(TOL) \approx 1\times10^{-8}$.

Step 2: Compute f, f', f'' and f'''.

Step 3: Compute $x_{i+1} = \text{solve}\left(f(x_i) + \frac{(x_{i+1}-x_i)}{1!} f'(x_i) + \frac{(x_{i+1}-x_i)^2}{2!} f''(x_i) + \frac{(x_{i+1}-x_i)^3}{3!} f'''(x_i) \right)$ by using MATLAB.

Step 4: IF $|f(x_i)| < TOL$, then STOP program ELSE go to Step3.

4. NUMERICAL RESULTS

Following examples show the results obtained by present method (three and four terms) to find root of nonlinear equations. For the accuracy, we use tolerance error (*TOL*) less than 1×10^{-8}. At the same time, our results are presented and compared the efficiency and accuracy of the method by MATLAB as shown in the following tables:

Table 1: Comparison of the results obtained by various method for solving $f(x) = x^3 - e^{-x} = 0$, $x_0 = 1$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	5	0.77288295914921012	6.5×10^{-17}
Hybrid	6	0.77288295914921012	6.5×10^{-17}
New hybrid	6	0.77288295914921013	1.62630325×10^{-19}
present method (3 terms)	3	0.77288295914921012474	2.68213×10^{-19}
present method (4 terms)	2	0.77288295914921012474	2.68213×10^{-19}

Table 2: Comparison of the results obtained by various method for solving $f(x) = \sin x - 0.5x = 0$, $x_0 = 1.6$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	5	1.895494267033980	0.000000000
Hybrid	6	0.000000000000000	0.000000000
New hybrid	5	1.895494267033999	0.000000000
present method (3 terms)	3	1.89549426703398093962	6.15449×10^{-18}
present method (4 terms)	3	1.89549426703398093962	6.15449×10^{-18}

Table 3: Comparison of the results obtained by various method for solving $f(x) = x\ln x - 1.2 = 0$, $x_0 = 2$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	5	1.88808675302834340	2.2204×10^{-12}
Hybrid	10	1.88808675302834340	2.2204×10^{-12}
New hybrid	6	1.88808675302834340	2.2204×10^{-12}
present method (3 terms)	3	1.88808675302834361175	1.43895×10^{-16}
present method (4 terms)	3	1.88808675302834361175	1.43895×10^{-16}

Table 4: Comparison of the results obtained by various method for solving $f(x) = \arctan x = 0$, $x_0 = 2$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	11	Failure	-
Hybrid	6	1×10^{-18}	1×10^{-18}
New hybrid	7	1×10^{-18}	1×10^{-18}
present method (3 terms)	5	9.68188×10^{-19}	9.68188×10^{-19}
present method (4 terms)	4	9.68188×10^{-19}	9.68188×10^{-19}

Table 5: Comparison of the results obtained by various method for solving $f(x) = x^3 - 2x - 5 = 0$, $x_0 = 2$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	5	2.09455148151852778	8.8818×10^{-12}
Hybrid	9	2.09455148151852700	3.55271×10^{-11}
New hybrid	5	2.09455148154232650	8.8818×10^{-12}
present method (3 terms)	3	2.0945514815423265098l0	9.11576×10^{-16}
present method (4 terms)	2	2.0945514815423265098l0	1.43895×10^{-16}

Table 6: Comparison of the results obtained by various method for solving $f(x) = x^2 - 5 = 0$, $x_0 = 1$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	7	2.23606898849978980	8.8818×10^{-12}
Hybrid	4	-2.23606797749978980	8.8818×10^{-12}
New hybrid	6	2.23606797499789800	8.8818×10^{-12}
present method (3 terms)	2	2.236067977499789696 40	4.09803×10^{-29}
present method (4 terms)	1	2.236067977499789696 40	4.09803×10^{-29}

Table 7: Comparison of the results obtained by various method for solving $f(x) = x^3 + 4x^2 - 10 = 0$, $x_0 = 1.5$.

Method	Iteration no.	x_i	$\lvert f(x_i)\rvert$
Newton	5	1.365230013414096850	1×10^{-17}
Hybrid	6	1.365230013893928000	7.923642×10^{-9}
New hybrid	4	1.36523001344889900	5.565379×10^{-9}
present method (3 terms)	3	1.365230013414096887 91	6.96088×10^{-16}
present method (4 terms)	1	1.365230013414096887 91	6.96088×10^{-16}

Table 8: Comparison of the results obtained by different methods for solving $f(x) = x^3 - 2x - 5 = 0$ on number of functional computing.

Method	Iteration no.	no. of functional
Newton	5	10
Hybrid	9	56
New hybrid	5	15
present method (3 terms)	3	9
present method (4 terms)	2	8

Table 9: Comparison of the results obtained by different methods for solving $f(x) = x^3 + 4x^2 - 10 = 0$ on number of functional computing.

Method	Iteration no.	no. of functional
Newton	5	10
Hybrid	6	24
New hybrid	4	12
present method (3 terms)	3	9
present method (4 terms)	1	4

5. CONCLUSION

We proposed a new iteration method by 2nd and 3rd order of Taylor expansion to solve nonlinear equations $f(x)=0$. It can be seen in Table 1-9 that the proposed method can reduced the number of iterations which less than the iteration number of Newton –Raphson method, hybrid iteration and new hybrid iteration method and also less number of functional calculations while the error are less than the tolerance.

6. REFERENCES

[1] Nasr Al-Din Ide, "A new Hybrid iteration method for solving algebraic equations", Applied Mathematics and Computation, vol. 195, pp. 772-774, 2008.
[2] Amit kumar Maheshwari, "A fourth order iterative method for solving nonlinear equations", Applied Mathematics and Computation, vol. 211, pp. 383-391, 2009.
[3] Avram Sidi, "Unified treatment of Regular Falai, Newton–Raphson, Secent, and Steffensen methods for nonlinear equations", Journal of Online Mathematics and Its Applications, pp.1-13, 2006.