Open Journal of
Mathematical Sciences

PSR Press

Article

Modified Abbasbandy's method free from second derivative for solving nonlinear equations

Sahar Saba[1], Amir Naseem[2,*] and Muhammad Irfan Saleem[3]

[1] Barani Institute of Sciences, Sahiwal, Pakistan.; saharsaba826@gmail.com
[2] Department of Mathematics, University of Management and Technology, Lahore 54000, Pakistan.
[3] Department of Mathematics, Lahore Leads University, Lahore 54000, Pakistan.; m.irfan_saleem@live.com
* Correspondence: amir14514573@yahoo.com

Received: 12 December 2018; Accepted: 15 March 2019; Published: 23 April 2019.

Abstract: The boundary value problems in Kinetic theory of gases, elasticity and other applied areas are mostly reduced in solving single variable nonlinear equations. Hence, the problem of approximating a solution of the nonlinear equations is important. The numerical methods for finding roots of such equations are called iterative methods. There are two type of iterative methods in literature: involving higher derivatives and free from higher derivatives. The methods which do not require higher derivatives have less order of convergence and the methods having high convergence order require higher derivatives. The aim of present report is to develop an iterative method having high order of convergence but not involving higher derivatives. We propose three new methods to solve nonlinear equations and solve text examples to check validity and efficiency of our iterative methods.

Keywords: Nonlinear equations, Newton's method, Halley's method, Abbasbandy's method.

MSC: 65H05, 65D32.

---

1. Introduction

One of the complex problem in science and specially in mathematics is to solve the non-linear equation

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

The solution of such type of equations cannot be find directly except in special cases. Therefore most of the methods for solving such type of equations are iterative methods. In iterative methods, we start with an initial guess x_0 which is improved step by step by means of iterations. In recent years, several iterative methods have been developed by using the different techniques namely: Taylor's series expansion, Adomian decomposition, Quadrature formulae etc. Some basic iterative methods are given in literature [1–3], and the references therein.

Considering (1) and assume that α is a simple zero of (1) and γ is an initial guess sufficiently close to α then by using the Taylor's series around γ for (1), we have

$$f(\gamma) + (x-\gamma)f'(\gamma) + \frac{1}{2!}(x-\gamma)^2 f''(\gamma) + \ldots = 0 \tag{2}$$

If $f'(\gamma) \neq 0$, we can evaluate the above expression as follow's:

$$f(x_k) + (x - x_k)f'(x_k) = 0.$$

If we choose x_{k+1} the root of equation, then we have

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}. \tag{3}$$

This is so-called the Newton's method (NM) [4], for root-finding of nonlinear functions and converges quadratically. From (2), we obtain

$$x_{k+1} = x_k - \frac{2f(x_k)f'(x_k)}{2f'^2(x_k) - f(x_k)f''(x_k)}. \tag{4}$$

This is so-called the Halley's method (HM) [5], for root-finding of nonlinear functions and converges cubically.

Simplification of (2) yields another method:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f^2(x_k)f''(x_k)}{2f'^3(x_k)}. \tag{5}$$

This is known as Househölder's method [6], for solving nonlinear equations and converges cubically.

In this paper, we modified the Abbasbandy's method [7] by making it three step iterative method by taking Newton's method as a pre-predictor and predictor step and Abbasbandy's method as a corrector step. We proved that this new modified methods have twelve, twelve and ten order of convergence and most efficient than existing methods. Some examples are given to show the performance of this method with other famous methods.

2. Iterative methods

Let $f : X \to R$, $X \subset R$ is a scalar function, then by using Taylor expansion, expanding $f(x)$ about the point x_k, we obtain the Abbasbandy's method as

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} - \frac{f^2(x_k)f''(x_k)}{2f'^3(x_k)} - \frac{f^3(x_k)f'''(x_k)}{6f'^4(x_k)}.$$

Algorithm 1. *For a given x_0, compute the approximate solution x_{n+1} by the following three step iterative scheme:*

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, ..., \\ w_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'(w_n)} - \frac{f^2(w_n)f''(w_n)}{2f'^3(w_n)} - \frac{f^3(w_n)f'''(w_n)}{6f'^4(w_n)}. \end{aligned}$$

By following the finite difference scheme, we develop the following algorithms:

Algorithm 2. *For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:*

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, ..., \\ w_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'(w_n)} - \frac{f^2(w_n)f''(w_n)}{2f'^3(w_n)} + \frac{f^3(w_n)f'(y_n)[f''(w_n) - f''(y_n)]}{6f(y_n)f'^4(w_n)}. \end{aligned}$$

Algorithm 3. *For a given x_0, compute the approximate solution x_{n+1} by the following iterative schemes:*

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, ..., \\ w_n &= y_n - \frac{f(y_n)}{f'(y_n)} \\ x_{n+1} &= w_n - \frac{f(w_n)}{f'(w_n)} - \frac{f'(y_n)f^2(z_n)}{2f'^3(w_n)}\left[\frac{f'(y_n) - f'(w_n)}{f(y_n)}(1 - \frac{f'(y_n)f(w_n)}{3f(y_n)f'(w_n)}) + \frac{f'(x_n)f(w_n)(f'(x_n) - f'(y_n))}{3f(x_n)f(y_n)f'(w_n)}\right] \end{aligned}$$

3. Convergence Analysis

In this section, we prove the convergence of our purposed iterative methods.

Theorem 4. *Suppose that α is a root of the equation $f(x)=0$. If $f(x)$ is sufficiently smooth in the neighborhood of α, then the convergence order of Algorithm (1), Algorithm (2) and Algorithm (3) is at least twelve, twelve and ten respectively.*

Proof. To prove the convergence, suppose that α is a root of the equation $f(x)=0$ and e_n be the error at nth iteration, then $e_n = x_n - \alpha$ and by using Taylor series expansion, we have

$$f(x_n) = f'(\alpha)e_n + \frac{1}{2!}f''(\alpha)e_n^2 + \frac{1}{3!}f'''(\alpha)e_n^3 + \frac{1}{4!}f^{(iv)}(\alpha)e_n^4 + \frac{1}{5!}f^{(v)}(\alpha)e_n^5 + \frac{1}{6!}f^{(vi)}(\alpha)e_n^6 + \ldots$$

$$f(x) = f'(\alpha)[e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + \ldots] \tag{6}$$

$$f'(x_n) = f'(\alpha)[1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + \ldots] \tag{7}$$

where

$$c_n = \frac{1}{n!}\frac{f^{(n)}(\alpha)}{f'(\alpha)}.$$

With the help of Equation (6) and Equation (7), we get

$$\begin{aligned} y_n &= f'(\alpha)[\alpha + c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (3c_4 - 7c_2c_3 + 4c_2^3)e_n^4 + (-6c_3^2 + 20c_3c_2^2 - 10c_2c_4 + 4c_5 - 8c_2^4)e_n^5 \\ &+(-17c_4c_3 + 28c_4c_2^2 - 13c_2c_5 + 5c_6 + 33c_2c_3^2 - 52c_3c_2^3 + 16c_2^5)e_n^6 + \ldots] \end{aligned} \tag{8}$$

$$\begin{aligned} f(y_n) &= f'(\alpha)[c_2e_n^2 + (2c_3 - 2c_2^2)e_n^3 + (5c_2^3 - 7c_2c_3 + 3c_4)e_n^4 + (24c_3c_2^2 - 12c_2^4 - 10c_2c_4 + 4c_5 - 6c_3^2)e_n^5 \\ &+(-73c_3c_2^3 + 34c_4c_2^2 + 28c_2^5 + 37c_2c_3^2 - 17c_4c_3 - 13c_2c_5 + 5c_6)e_n^6 + \ldots] \end{aligned} \tag{9}$$

$$\begin{aligned} f'(y_n) &= f'(\alpha)[1 + 2c_2^2e_n^2 + (4c_2c_3 - 4c_2^3)e_n^3 + (6c_2c_4 - 11c_3c_2^2 + 8c_2^4)e_n^4 + 28c_3c_2^3 - 20c_4c_2^2 + 8c_2c_5 - 16c_2^5)e_n^5 \\ &+(-16c_4c_2c_3 - 68c_3c_2^4 + 12c_3^3 + 60c_4c_2^3 - 26c_5c_2^2 + 10c_2c_6 + 32c_2^6)e_n^6 + \ldots] \end{aligned} \tag{10}$$

$$\begin{aligned} f''(y_n) &= f'(\alpha)[2c_2 + 6c_2c_3e_n^2 + (12c_3^2 - 12c_3c_2^2)e_n^3 + (-42c_2c_3^2 + 18c_4c_3 + 24c_3c_2^3 + 12c_4c_2^2)e_n^4 + (-12c_2c_4c_3 \\ &+24c_5c_3 - 36c_3^3 + 120c_3^2c_2^2 - 48c_3c_2^4 - 48c_4c_2^3)e_n^5 + (-78c_3c_2c_5 + 30c_3c_6 - 54c_4c_3^2 - 96c_3c_4c_2^2 \\ &+198c_2c_3^3 - 312c_3^2c_2^3 + 96c_3c_2^5 + 72c_2c_4^2 + 144c_4c_2^4 + 20c_5c_2^3)e_n^6 + \ldots] \end{aligned} \tag{11}$$

With the help of Equations (8), (9), (10) and (11), we get

$$w_n = f'(\alpha)[\alpha + c_2^3e_n^4 + (4c_3c_2^2 - 4c_2^4)e_n^5 + (-20c_3c_2^3 + 6c_4c_2^2 + 10c_2^5 + 4c_2c_3^2)e_n^6 + \ldots] \tag{12}$$

$$f(w_n) = f'(\alpha)[c_2^3e_n^4 + (4c_3c_2^2 - 4c_2^4)e_n^5 + (-20c_3c_2^3 + 6c_4c_2^2 + 10c_2^5 + 4c_2c_3^2)e_n^6 + \ldots] \tag{13}$$

$$f'(w_n) = f'(\alpha)[1 + 2c_2^4e_n^4 + (8c_3c_2^3 - 8c_2^5)e_n^5 + (-40c_3c_2^4 + 12c_4c_2^3 + 20c_2^6 + 8c_3^2c_2^2)e_n^6 + \ldots] \tag{14}$$

$$f''(w_n) = f'(\alpha)[2c_2 + 6c_3c_2^3e_n^4 + (24c_3^2c_2^2 - 24c_3c_2^4)e_n^5 + (-120c_3^2c_2^3 + 36c_3c_4c_2^2 + 60c_3c_2^5 + 24c_2c_3^3)e_n^6 + \ldots] \tag{15}$$

$$\begin{aligned} f'''(w_n) &= f'(\alpha)[6c_3 + 24c_4c_2^3e_n^4 + (96c_3c_4c_2^2 - 96c_4c_2^4)e_n^5 + (-480c_3c_4c_2^3 + 144c_4^2c_2^2 \\ &+240c_4c_2^5 + 96c_4c_2c_3^2)e_n^6 + \ldots] \end{aligned} \tag{16}$$

Using Equations (12), (13), (14), (15) and (16) in Algorithm (1), (2) and (3), we get
$x_{n+1} = \alpha + (2c_2^{11} - 2c_3c_2^9)e_n^{12} + O(e_n^{13})$,
$x_{n+1} = \alpha + 2c_2^{11}e_n^{12} + O(e_n^{13})$,
and $x_{n+1} = \alpha - \frac{3c_3c_2^7}{2}e_n^{10} + O(e_n^{11})$.
Which implies that

$$e_{n+1} = (2c_2^{11} - 2c_3c_2^9)e_n^{12} + O(e_n^{13}) \tag{17}$$

$$e_{n+1} = 2c_2^{11}e_n^{12} + O(e_n^{13}) \tag{18}$$

$$e_{n+1} = -\frac{3c_3c_2^7}{2}e_n^{10} + O(e_n^{11}) \tag{19}$$

Equations (17), (18) and (19) shows that the Algorithms (1), (2) and (3) have convergence of order twelve, twelve and ten respectively. □

4. Applications

In this section we solved some nonlinear functions to illustrate the efficiency of our developed algorithms. We compare our developed methods with Newton'smethod (NM), Halley's method (HM and Abbasbanday's method (AM).

Example 1. In this example we solved $f(x) = x^3 + 4x^2 - 25$ by taking $x_0 = -0.8$. It can be observed from Table 1 that NM takes 35 iterations, HM takes 36 iterations, AM takes 13 iterations and our Algorithms (1), (2) and (3) takes 12, 5 and 5 iterations respectively to reach at root of $f(x) = x^3 + 4x^2 - 25$.

Table 1. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	35	70	$1.105260e-24$	
HM	36	108	$2.995246e-17$	1.365230013414096845760806828980
AM	13	52	$6.423767e-20$	
Algorithm 1	12	72	$2.738493e-48$	
Algorithm 2	5	25	$2.812883e-25$	
Algorithm 3	5	20	$3.108248e-83$	

Example 2. In this example we solved $f(x) = x^3 + x^2 - 2$ by taking $x_0 = -0.1$. It can be observed from Table 2 that NM takes 13 iterations, HM takes 17 iterations, AM takes 19 iterations and our Algorithms (1), (2) and (3) takes 5, 4 and 5 iterations respectively to reach at root of $f(x) = x^3 + x^2 - 2$.

Table 2. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	13	26	$2.203086e-19$	
HM	17	51	$4.338982e-22$	1.000000000000000000000000000000
AM	19	76	$2.239715e-27$	
Algorithm 1	5	30	$2.338056e-31$	
Algorithm 2	4	20	$5.192250e-45$	
Algorithm 3	5	20	$6.607058e-83$	

Example 3. In this example we solved $f(x) = e^{(x^2+7x-30)} - 1$ by taking $x_0 = 4.5$. It can be observed from Table 3 that NM takes 27 iterations, HM takes 14 iterations, AM takes 16 iterations and our Algorithms (1), (2) and (3) takes 8, 7 and 7 iterations respectively to reach at root of $f(x) = e^{(x^2+7x-30)} - 1$.

Table 3. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	27	54	$6.454129e-23$	
HM	14	42	$1.217550e-25$	3.000000000000000000000000000000
AM	16	64	$1.136732e-17$	
Algorithm 1	8	48	$1.261140e-22$	
Algorithm 2	7	35	$6.546702e-15$	
Algorithm 3	7	28	$9.047215e-71$	

Example 4. In this example we solved $f(x) = x^2 - e^x - 3x + 2$ by taking $x_0 = 3.5$. It can be observed from Table 4 that NM takes 6 iterations, HM takes 5 iterations, AM takes 5 iterations and our Algorithms (1), (2) and (3) takes 2, 3 and 3 iterations respectively to reach at root of $f(x) = x^2 - e^x - 3x + 2$.

Table 4. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	6	12	$4.925534e-15$	
HM	5	15	$1.463064e-40$	0.257530285439860760455367304937
AM	5	20	$1.120893e-28$	
Algorithm 1	2	12	$8.978612e-19$	
Algorithm 2	3	15	$0.000000e+00$	
Algorithm 3	3	12	$4.980111e-66$	

Example 5. In this example we solved $f(x) = xe^{x^2} - sin^2x + 3cosx + 5$ by taking $x_0 = 1.1$. It can be observed from Table 5 that NM takes 45 iterations, HM takes 44 iterations, AM takes 50 iterations and our Algorithms (1), (2) and (3) takes 14, 12 and 12 iterations respectively to reach at root of $f(x) = xe^{x^2} - sin^2x + 3cosx + 5$.

Table 5. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	45	90	$1.268546e-15$	
HM	44	132	$1.169824e-26$	$-1.207647827130918927009416758360$
AM	50	200	$2.868208e-29$	
Algorithm 1	14	84	$1.935782e-64$	
Algorithm 2	12	60	$4.515078e-97$	
Algorithm 3	12	48	$4.515078e-97$	

Example 6. In this example we solved $f(x) = x^2 + sin(\frac{x}{5}) - \frac{1}{4}$ by taking $x_0 = 2.2$. It can be observed from Table 6 that NM takes 7 iterations, HM takes 5 iterations, AM takes 7 iterations and our Algorithms (1), (2) and (3) takes 2, 2 and 2 iterations respectively to reach at root of $f(x) = x^2 + sin(\frac{x}{5}) - \frac{1}{4}$.

Table 6. Comparison of NM, HM, AM and Algorithms (1), (2) and (3) .

Method	N	N_f	$\lvert f(x_{n+1})\rvert$	x_{n+1}
NM	7	14	$7.777907e-23$	
HM	5	15	$1.210132e-42$	0.409992017989137131621258376499
AM	7	28	$2.132547e-32$	
Algorithm 1	2	12	$5.800844e-23$	
Algorithm 2	2	10	$5.897018e-23$	
Algorithm 3	2	8	$4.106937e-22$	

5. Conclusions

Three new algorithms for solving nonlinear functions has been established. We can conclude that the efficiency indexes of algorithms (1), (2) and (3) are 1.5131, 1.6438, and 1.7783 respectively. The convergence orders of algorithms (1), (2) and (3) are twelve, twelve and ten respectively. By solving some examples, the performance of our developed algorithms is discussed. Our developed algorithms are performing well in comparison to Newton's method, Halley's method and Abbasbanday's method.

Author Contributions: All authors contributed equally to the writing of this paper.

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] Daftardar-Gejji, V., & Jafari, H. (2006). An iterative method for solving nonlinear functional equations. *Journal of Mathematical Analysis and Applications*, *316*(2), 753-763.

[2] Abdou, M. A., & Soliman, A. A. (2005). Variational iteration method for solving Burger's and coupled Burger's equations. *Journal of Computational and Applied Mathematics*, *181*(2), 245-251.

[3] Amat, S., Busquier, S., & Gutiérrez, J. M. (2003). Geometric constructions of iterative functions to solve nonlinear equations. *Journal of Computational and Applied Mathematics*, *157*(1), 197-205.

[4] Dembo, R. S., Eisenstat, S. C., & Steihaug, T. (1982). Inexact newton methods. *SIAM Journal on Numerical analysis, 19*(2), 400-408.

[5] Scavo, T. R., & Thoo, J. B. (1995). On the geometry of Halley's method. *The American Mathematical Monthly, 102*(5), 417-426.

[6] Noor, K. I., Noor, M. A., & Momani, S. (2007). Modified Householder iterative method for nonlinear equations. *Applied mathematics and computation*, *190*(2), 1534-1539.

[7] Abbasbandy, S. (2003). Improving NewtonŰRaphson method for nonlinear equations by modified Adomian decomposition method. *Applied Mathematics and Computation*, *145*(2-3), 887-893.