An Efficient Tenth Order Iterative Method for Solving Non-linear Equations with Chemical Applications

Mani Sandeep Kumar Mylapalli*
Department of Mathematics, GITAM (Deemed to be University), Visakhapatnam, India.
E-mail: mmylapal@gitam.edu

Rajesh Kumar Palli
Research Scholar, Department of Mathematics, GITAM (Deemed to be University), Visakhapatnam, India. E-mail: rajeshkumar.viit@gmail.com

Ramadevi Sri
Department of Mathematics, Dr. L. Bullayya College Visakhapatnam, India.
E-mail: ramadevisri9090@gmail.com

Muralidhar Pamerla
Department of Chemistry, GITAM (Deemed to be University), Visakhapatnam, India.
E-mail: mpamerla@gitam.edu

Received October 07, 2020; Accepted November 16, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V18SI02/WEB18009

Abstract

The scope of this paper is to establish a new tenth order convergent method to find the root of non-linear equations. Here we presented a modification of Newton’s method with higher-order convergence and the study of convergence concluded that the order of convergence is tenth. With some numerical examples, we concluded that this algorithm is better than classical Newton's method and other methods with tenth order.

Keywords

Iterative Method, Non-linear Scalar Equation, Functional Evaluations, Convergence Analysis, Efficiency Index.

Introduction

In science and engineering, a lot of development happened in solving a non-linear scalar equation. In this proposal of finding a zero of non-linear equations, Newton's method (NR) [1] is one of the optimal second-order methods to obtain the root of a non-linear scalar equation.

$$g(t) = 0 \tag{1.1}$$

and is given by,

$$t_{n+1} = t_n - \frac{g(t_n)}{g'(t_n)} \qquad n = 0, 1, 2, \ldots \tag{1.2}$$

and NR method converges quadratically and its efficiency index is $\sqrt{2} = 1.414$.

An efficient three-step tenth order method (MA) without second derivative proposed by Hafiz [4] is given by,

$$\begin{aligned}
y_n &= t_n - \frac{g(t_n)}{g'(t_n)} \\
z_n &= y_n - \frac{2g(y_n)g'(y_n)}{2g'(z_n)^2 - g(y_n)\rho_g(t_n, y_n)} \\
&\text{where } \rho_g(t_n, y_n) = \frac{2}{y_n - t_n}\left(2g'(y_n) + g'(t_n) - 3\frac{g(y_n) - g(t_n)}{y_n - t_n}\right) \\
t_{n+1} &= z_n - \frac{g(z_n)}{g[z_n, y_n] + (z_n - y_n)g[z_n, y_n, y_n]} \\
&\text{Here } g[z_n, y_n, y_n] = \frac{g[z_n, y_n] - g'(y_n)}{z_n - y_n}
\end{aligned} \tag{1.3}$$

Solving non-linear equations using a new tenth order method (PCMH) free from second derivatives proposed by Hafiz, Salwa, Al-Goria [4] is given by

$$\begin{aligned}
y_n &= t_n - \frac{g(t_n)}{g'(t_n)} \\
z_n &= y_n - \frac{g(y_n)}{g'(y_n)} - \frac{[g(y_n)]^2 \rho_1(y_n)}{2[g'(y_n)]^3} \\
&\textit{Where } \rho_1(y_n) = g''(y_n) = \frac{2}{t_n - y_n}\left[3\,\frac{g(t_n) - g(y_n)}{t_n - y_n} - 2g'(y_n) - g'(t_n)\right] \\
t_{n+1} &= z_n - \frac{g(z_n)}{g[z_n, y_n] + (z_n - y_n)g[z_n, y_n, y_n]}
\end{aligned} \tag{1.4}$$

A quadrature based three-step tenth order iterative method (SK) proposed by Khattri [7] is given by

$$y_n = t_n - \frac{g(t_n)}{g'(t_n)}$$
$$z_n = y_n - \frac{(t_n - y_n)g(y_n)}{g(t_n) - 2g(y_n)} \quad (1.5)$$
$$t_{n+1} = z_n - \frac{g(z_n)g'(z_n)}{(g'(z_n))^2 - g(z_n)\left(\frac{g(y_n) - g(z_n) - g'(z_n)(y_n - z_n)}{(y_n - z_n)^2}\right)}$$

Tenth order iterative method (KN) for roots of non-Linear equations proposed by Nouri, Ranijbar [2] is given by

$$\nu_n = x_n - \frac{f(t_n)}{f'(t_n)}$$
$$\eta_n = \nu_n - \frac{f(\nu_n)}{f'(\nu_n)}$$
$$\lambda_n = \nu_n + \frac{f(\nu_n)}{f'(\nu_n)} \quad (1.6)$$
$$t_{n+1} = \nu_n - \frac{(\nu_n - \eta_n)\left(f^2(x_n)\right)\left(f(\eta_n) + f(\lambda_n)\right)}{f^2(\nu_n)\left(f(\lambda_n) - f(\eta_n)\right) - 4f(\nu_n)f^2(\eta_n) - 6f^3(\eta_n)}$$

In section II, we illustrated the new three-step iterative method, and section III, we proved this method is with tenth order convergence. At last in section IV, we compared our new method with other schemes with the same order of convergence using some defined examples.

Tenth Order Convergent (SKM) Method

Consider t^* is an exact root of "(1.1)" where $g(t)$ is continuous and has well defined first derivatives. Let t_n be the root of n[th] approximation of "(1.1)" and is

$$t^* = t_n + \varepsilon_n \quad (2.1)$$

Where ε_n is the error.

Thus, we get

$$g\left(t^{*}\right)=0 \tag{2.2}$$

Expanding $g\left(t^{*}\right)$ by Taylor's series about t_n, we have,

$$g\left(t^{*}\right)=g\left(t_n\right)+\left(t^{*}-t_n\right)g'\left(t_n\right)+\frac{\left(t^{*}-t_n\right)^2}{2!}g''\left(t_n\right)+\ldots$$

$$g\left(t^{*}\right)=g\left(t_n\right)+\varepsilon_n g'\left(t_n\right)+\frac{\varepsilon_n^{\ 2}}{2!}g''\left(t_n\right)+\ldots \tag{2.3}$$

By neglecting higher power ε_n, i.e. neglect terms from $\varepsilon_n{}^3$ onwards. Using "(2.2)" and "(2.3)", we have

$$\varepsilon_n^2 g''\left(t_n\right)+2\varepsilon_n g'\left(t_n\right)+2g\left(t_n\right)=0$$

$$\varepsilon_n=\left[-2g'\left(t_n\right)\pm\sqrt{4g'\left(t_n\right)-8g\left(t_n\right)g''\left(t_n\right)}\right]\div 2g''\left(t_n\right) \tag{2.4}$$

On Substituting t^{*} by t_{n+1} in "(2.1)" and from "(2.4)",

We get

$$t_{n+1}=t_n-\frac{2g\left(t_n\right)}{g'\left(t_n\right)}\left(\frac{1}{1+\sqrt{1-2\mu_n}}\right)$$

$$\text{Where, } \mu_n=\frac{g\left(t_n\right)g''\left(t_n\right)}{\left[g'\left(t_n\right)\right]^2}$$

and here

$$g''\left(t_n\right)=\frac{2}{t_{n-1}-t_n}\left[3\,\frac{g\left(t_{n-1}\right)-g\left(t_n\right)}{t_{n-1}-t_n}-2g'\left(t_n\right)-g'\left(t_{n-1}\right)\right] \tag{2.5}$$

Using divide difference formula, Newton's method "(1.2)", can be written as

$$t_{n+1} = t_n - \frac{g(t_n)}{\left(\frac{g(t_n)-g(t_{n-1})}{t_n-t_{n-1}}\right)} \tag{2.6}$$

Here we developed a new algorithm by taking "(1.2)" as the first step and "(2.5)" as second step and "(2.6)" as third step.

Algorithm

The iterative scheme is computed by x_{n+1} as,

$$\begin{aligned}
z_n &= t_n - \frac{g(t_n)}{g'(t_n)} \\
y_n &= z_n - \frac{2g(z_n)}{g'(z_n)}\left(\frac{1}{1+\sqrt{1-2\mu_n}}\right) \\
\text{where } \mu_n &= \frac{g(z_n)g''(z_n)}{\left[g'(z_n)\right]^2} \\
\text{and } g''(z_n) &= \frac{2}{t_n-z_n}\left[3\,\frac{g(t_n)-g(z_n)}{t_n-z_n} - 2g'(z_n) - g'(t_n)\right] \\
t_{n+1} &= y_n - \frac{g(y_n)(y_n-z_n)}{g(y_n)-g(z_n)}
\end{aligned} \tag{2.7}$$

The method"(2.7)" is known as tenth order convergent method (SKM), it requires three functional evaluations and two of its first derivatives.

Convergence Criteria

Thoerem: Let $t_0 \in I$ be a single zero of a sufficiently differentiable function g for an open interval I. If t_0 is in the neighborhood of t^*. Then the algorithm "(2.7)" has tenth order convergence.

Proof

Let the single zero of "(1.1)" be t^* and $t^* = t_n + \varepsilon_n$

Thus, $g\left(t^*\right) = 0$

By Taylor's series, writing $g\left(t^*\right)$ about t_n, we obtain

$$g\left(t_n\right) = g'\left(t^*\right)\left(\varepsilon_n + c_2\varepsilon_n{}^2 + c_3\varepsilon_n{}^3 + c_4\varepsilon_n{}^4 + \ldots\right) \quad (3.1)$$

$$g'\left(t_n\right) = g'\left(t^*\right)\left(1 + 2c_2\varepsilon_n + c_3\varepsilon_n{}^2 + 4c_4\varepsilon_n{}^3 + \ldots\right) \quad (3.2)$$

Dividing "(3.1)" by "(3.2)", we get

$$\frac{g(t_n)}{g'(t_n)} = \left(\varepsilon_n - c_2\varepsilon_n{}^2 - \left(2c_3 - 2c_2{}^2\right)\varepsilon_n{}^3 - \left(3c_4 - 7c_2c_3 + 4c_2^3\right)\varepsilon_n{}^4 + \ldots\right) \quad (3.3)$$

From

$$z_n = t_n - \frac{g(t_n)}{g'(t_n)},$$

we get

$$z_n = t^* + \omega_n$$

Where

$$\omega_n = c_2\varepsilon_n{}^2 + \left(2c_3 - 2c_2{}^2\right)\varepsilon_n{}^3 + \left(3c_4 - 7c_2c_3 + 4c_2^3\right)\varepsilon_n{}^4 + \ldots$$

Now

$$g(z_n) = g'\left(t^*\right)\left(c_2\varepsilon_n{}^2 + \left(2c_3 - 2c_2{}^2\right)\varepsilon_n{}^3 + \left(3c_4 - 7c_2c_3 + 5c_2^3\right)\varepsilon_n{}^4 + \ldots\right) \quad (3.4)$$

Using the definition of second derivative in the scheme,
On simplification

$$g''\left(z_n\right) = g'\left(t^*\right)\begin{pmatrix} 2c_2 + 2\left(3c_2c_3 - c_4\right)\varepsilon_n^2 \\ -4\left(3c_2^2c_3 - 3c_3^2 - c_2c_4 + c_5\right)\varepsilon_n^3 + \ldots \end{pmatrix} \quad (3.5)$$

And $$\frac{g(z_n)}{g'(z_n)} = L_1\varepsilon_n^2 + L_2\varepsilon_n^3 + L_3\varepsilon_n^4 + \ldots \quad (3.6)$$

where $L_1 = c_2$, $L_2 = \left(2c_3 - 2c_2^2\right)$, $L_3 = \left(3c_2^3 - 7c_2c_3 + 3c_4\right)$

From

$\mu_n = \frac{g(z_n)g''(z_n)}{\left[g'(z_n)\right]^2}$ as defined in the scheme, we get

$$2\mu_n = P_1\varepsilon_n^2 + P_2\varepsilon_n^3 + P_3\varepsilon_n^4 + \ldots \quad (3.7)$$

where, $P_1=4c_2^2,\ P_2=4\left(6c_2c_3^2-2c_3c_4-6c_2^3c_3+2c_2^2c_4\right),$
$P_3=2\left(-8c_2^2c_3+4c_2c_4+2c_2^4\right)$

From “(3.7)”, we get

$$\left(1+\sqrt{1-2\mu_n}\right)^{-1}=2\left(1+M_1\varepsilon_n^2+M_2\varepsilon_n^3+M_3\varepsilon_n^4+\ldots\right) \qquad (3.8)$$

where $M_1=c_2^2, M_2=6c_2c_3^2-2c_3c_4-6c_2^3c_3+2c_2^2c_4,$
$M_3=-4c_2^2c_3+2c_2c_4+6c_2^4$

From “(3.5)” and “(3.8)”, we get

$$\frac{2g(z_n)}{g'(z_n)}\left(\frac{1}{1+\sqrt{1-2\mu_n}}\right)=\begin{pmatrix}L_1\varepsilon_n^2+L_2\varepsilon_n^3+L_3\varepsilon_n^4+L_4\varepsilon_n^5\\+\left(L_1M_3+L_3M_1+L_2M_2\right)\varepsilon_n^6+o\left(\varepsilon_n^7\right)\end{pmatrix}$$

$$y_n=t^*+\left(L_1M_3+L_3M_1+L_2M_2\right)\varepsilon_n^6+o\left(\varepsilon_n^7\right) \qquad (3.9)$$

$$y_n=t^*+Y$$

Where $Y=\left(L_1M_3+L_3M_1+L_2M_2\right)\varepsilon_n^6+o\left(\varepsilon_n^7\right)$

$$g(y_n)=g'\left(t^*\right)\left(y_n+c_2{y_n}^2+c_3{y_n}^3+c_4{y_n}^4+\ldots\right) \qquad (3.10)$$

Using (3.10) and (3.4) in the third step of (2.7),i.e.

$$x_{n+1}=y_n-\frac{g(y_n)}{g(y_n)-g(z_n)}(y_n-z_n)$$

we get

$$\varepsilon_{n+1}=\left(\frac{1}{c_2^3}\right)\left(22c_2^3c_3-7c_2^2c_4-11c_2^5-4c_2c_3^2\right)\begin{pmatrix}36c_3^2c_4-84c_2c_3^3+60c_2^3c_3^2+24c_2^4c_3^2\\ -14c_2^2c_3^3+48c_3^4+8c_2c_3^2c_4-32c_3^2c_5\\ +36c_2^4c_4-84c_2^5c_3+60c_2^7+96c_2^8\\ -182c_2^6c_3+80c_2^5c_4-32c_2^4c_5-192c_2^5c_3^2\\ +384c_2^3c_3^3-96c_2c_3^4-160c_2^2c_3^2c_4+64c_2c_3^2c_5\end{pmatrix}\varepsilon^{10}+o\left(\varepsilon^{11}\right)$$

Thus, it's proved that this new scheme is tenth order convergence and its efficiency index is $\sqrt[5]{10}=1.584$.

Numerical Examples

We consider the some examples considered by VBK Vatti, MMS [8], [5] and compared our method with NR, MA, PCMH, SK, KN methods. The computations are carried out by using mp math-PYTHON and the number of iterations for these methods are obtained for comparisons such that

$$|x_{n+1}-x_n|<10^{-201} \text{ and } |g(x_{n+1})|<10^{-201}.$$

The test functions and simple zeros are given below,

$$g_1(x)=(x+2)e^{x}-1, \qquad t^*=-0.442854010023885$$

$$g_2(x)=x^2+\sin\left(\frac{x}{5}\right)-\frac{1}{4}, \qquad t^*=0.4099920179891371$$

$$g_3(x)=\cos x-x, \qquad t^*=0.7390851332151606$$

$$g_4(x)=x^3-10, \qquad t^*=2.1544346900318837$$

$$g_5(x)=e^{-x}+\cos x, \qquad t^*=1.7461395304080124$$

$$g_6(x)=e^{\sin x}-x+1, \qquad t^*=2.6306641479279036$$

A chemical equillibrium problem: Consider the equation from [6] describing the fraction of the nitrogen hydrogen feed that gets converted to ammonia (this fraction is called fractional conversion) in polynomial form as,

$$g_7(x)=x^4-7.79075x^3+2.511x-1.674,$$

$$t^*=0.2777595428417206$$

Volume from van der Waals equation: One has to find out the volume from Van der Waals' equation [6] in polynomial form as,

$$g_8(x) = 40x^3 - 95.26535116x^2 + 35.28x - 5.6998368, \qquad t^* = 1.9707842194070294$$

Table I (a): Analogy Of Efficiency

Methods	P	N	EI
NR	2	2	1.414
MA	10	5	1.584
PCMH	10	5	1.584
SK	10	5	1.584
KN	10	5	1.584
SKM	10	5	1.584

Where P is the order of convergence, N is the number of functional values per iteration and EI is the Efficiency Index.

Table I (b): Analogy of Different Methods

g	Method	x_0	*n*	*er*	*fv*	x_0	*n*	*er*	*fv*
g_1	NR	-0.7	10	6.9(201)	1.1(200)	0.2	10	2.4(201)	4.1(201)
	MA		4	2.8(201)	4.1(201)		4	2.0(201)	4.1(201)
	PCMH		4	7.3(201)	4.1(201)		4	4.8(201)	4.1(201)
	SK		DIVERGENT				6	4.4(201)	1.1(200)
	KN		4	2.4(201)	4.1(201)		4	1.6(201)	4.1(201)
	SKM		4	2.8(201)	4.1(201)		4	1.6(201)	4.1(201)
g_2	NR	0.2	10	2.0(201)	2.2(201)	1.1	10	7.7(201)	7.7(201)
	MA		4	4.1(201)	2.2(201)		4	4.1(201)	2.2(201)
	PCMH		4	4.1(201)	2.2(201)		4	4.1(201)	2.2(201)
	SK		4	2.7(200)	2.2(201)		4	2.7(200)	2.2(201)
	KN		4	2.8(201)	2.2(201)		4	2.8(201)	2.2(201)
	SKM		4	2.0(201)	2.2(201)		4	1.2(201)	2.2(201)
g_3	NR	0.1	10	1.6(201)	2.4(201)	1.2	9	1.6(201)	2.4(201)
	MA		4	8.1(201)	1.3(200)		DIVERGENT		
	PCMH		4	2.4(201)	2.4(201)		4	8.1(201)	1.3(200)
	SK		4	8.1(201)	2.4(201)		4	1.2(200)	2.4(200)
	KN		4	1.6(201)	2.4(201)		4	1.6(201)	2.4(201)
	SKM		4	8.1(201)	2.4(201)		4	1.6(201)	2.4(201)
g_4	NR	1.5	10	1.6(200)	2.0(199)	3.2	10	1.6(200)	2.0(199)
	MA		4	3.2(201)	2.0(201)		4	3.2(201)	2.0(199)
	PCMH		4	2.6(200)	2.0(199)		4	2.6(200)	2.0(198)
	SK		8	3.2(201)	2.0(199)		8	3.2(201)	2.0(199)
	KN		4	8.8(200)	1.2(198)		4	1.3(200)	2.0(199)

	SKM		4	1.6(200)	2.0(199)		4	6.5(201)	2.0(199)
g_5	NR	1.5	9	4.8(201)	6.5(201)	-1.5	9	4.8(201)	6.5(201)
	MA		4	1.6(201)	6.5(201)		4	1.6(201)	6.5(201)
	PCMH		4	1.3(200)	6.5(201)		4	1.3(200)	6.5(201)
	SK		4	3.7(200)	6.5(201)		DIVERGENT		
	KN		4	3.2(201)	6.5(201)		5	3.2(201)	6.5(201)
	SKM		4	4.8(201)	6.5(201)		4	4.8(201)	6.5(201)
g_6	NR	2.9	9	3.5(200)	8.8(200)	1.2	24	6.2(200)	1.4(199)
	MA		4	2.9(200)	8.8(200)		5	9.1(200)	8.8(200)
	PCMH		4	7.5(201)	8.8(201)		18	7.5(200)	8.8(200)
	SK		4	1.3(200)	8.8(200)		7	1.3(200)	8.8(200)
	KN		4	6.8(200)	1.4(199)		15	1.9(200)	8.8(200)
	SKM		4	2.6(200)	8.8(200)		5	4.2(201)	8.8(201)
g_7	NR	1.5	10	2.8(201)	3.4(200)	-0.3	10	2.8(201)	3.4(200)
	MA		4	9.3(201)	3.4(200)		4	9.3(201)	3.4(200)
	PCMH		4	8.1(201)	3.4(200)		4	5.3(201)	3.4(200)
	SK		5	4.1(201)	8.6(200)		4	6.9(200)	3.4(200)
	KN		4	7.3(201)	8.6(200)		4	7.3(201)	8.6(200)
	SKM		4	3.2(201)	3.4(200)		4	1.6(201)	3.4(200)
g_8	NR	1.8	25	1.6(200)	2.1(198)	3.0	27	1.6(201)	2.1(198)
	MA		9	4.8(201)	2.1(198)		10	4.8(201)	2.1(198)
	PCMH		9	3.5(201)	2.1(201)		10	3.5(200)	2.1(198)
	SK		11	6.5(200)	1.0(197)		10	6.5(200)	1.0(197)
	KN		9	1.6(200)	2.1(198)		10	8.6(200)	1.0(197)
	SKM		5	9.7(201)	2.1(198)		6	9.7(201)	2.1(198)

Where x_0 is the initial approximation, *n* is the number of iterations, *er* is the error and *fv* is the functional value.

Conclusion

Here In this scheme, we introduced a new tenth order convergent iterative method with efficiency index 1.584. Table I (a) compares the efficiency of different methods and the computational results in table I (b) show that the method SKM competes with other efficient NR, MA, PCNH, SK, KN methods. Clearly, in some examples our method SKM is superior to the other methods in terms of iterations.

References

Traub, J.F. (1977). *Iterative Methods for the Solution of Equations.* Chelsea Publishing Company, New York.

Nouri, K., Ranjbar, H., & Torkzadeh, L. (2019). Two High Order Iterative Methods for Roots of Nonlinear Equations. *Punjab University Journal of Mathematics, 51*, 47-59.

Hafiz, M.A. (2014). An Efficient Three-Step Tenth–Order Method Without Second Order Derivative. *Palestine Journal of Mathematics, 3*(2), 198-203.

Hafiz, M.A., & Al-Goria, S.M. (2013). Solving nonlinear equations using a new tenth-and seventh-order methods free from second derivative. *International Journal of Differential Equations and Applications, 12*(4), 169-183.

Palli, R.K., Mylapalli, M.S.K., Chaganti, P., & Sri, R. (2020). An Optimal Three-Step Method for solving non-linear equations. *Journal of Critical Reviews*, *7*(6), 100-103.

Said Solaiman, O., & Hashim, I. (2019). Efficacy of optimal methods for nonlinear equations with chemical engineering applications. *Mathematical Problems in Engineering*, 2019.

Khattri, S. (2013). Another note on some quadrature based three-step iterative methods for non-linear equations. *Numerical Algebra, Control & Optimization*, *3*(3), 549-555.

Vatti, V.K., Sri, R., & Mylapalli, M.K. (2017). Eighteenth order convergent method for solving non-linear equations. *Oriental Journal of Computer Science and Technology*, *10*(1), 144-150.