NEW ITERATIVE METHODS BASED ON SPLINE FUNCTIONS FOR SOLVING NONLINEAR EQUATIONS

(communicated by martin hermann)

HADI TAGHVAFARD

Abstract. Two new iterative methods for solving nonlinear equations are presented using a new quadrature rule based on spline functions. Analysis of convergence shows that these methods have third-order convergence. Their practical utility is demonstrated by numerical examples to show that these methods are more efficient than that of Newton's.

1. Introduction

Solving nonlinear equations is one of the most predominant problems in numerical analysis. Newton's method is the most popular one in solving such equations. Some historical points on this method can be found in [14].

Recently, some methods have been proposed and analyzed for solving nonlinear equations [1, 2, 4, 6, 7, 8]. These methods have been suggested by using quadrature formulas, decomposition and Taylor's series [3, 8, 9, 13]. As we know, quadrature rules play an important and significant role in the evaluation of integrals. One of the most well-known iterative methods is Newton's classical method which has a quadratic convergence rate. Some authors have derived new iterative methods which are more efficient than that of Newton's [5, 9, 12, 13].

This paper is organized as follows. Section 2 provides some preliminaries which are needed. Section 3 is devoted to suggest two iterative methods by using a new quadrature rule based on spline functions. These are implicit-type methods. To implement these methods, we use Newton's and Halley's method as a predictor and then use these new methods as a corrector. The resultant methods can be considered as two-step iterative methods. In section 4, it will be shown that these two-step iterative methods are of third-order convergence. In section 5, a comparison between these methods with that of Newton's is made. Several examples are given to illustrate the efficiencies and advantages of these methods. Finally, section 6 will close the paper.

2. Preliminaries

We use the following definition [13]:

Definition 1. *Let $\alpha \in \mathbb{R}$ and $x_n \in \mathbb{R}$, $n = 0, 1, 2, \ldots$. Then the sequence x_n is said to be convergence to α if*

$$\lim_{n \to \infty} |x_n - \alpha| = 0.$$

If there exists a constant $c > 0$, an integer $n_0 \geqq 0$ and $p \geqq 0$ such that for all $n > n_0$, we have

$$|x_{n+1} - \alpha| \leqq c|x_n - \alpha|^p,$$

then x_n is said to be convergence to α with convergence order at least p. If $p = 2$ or $p = 3$, the convergence is said to be quadratic or cubic, respectively.

Notation 1. *The notation $e_n = x_n - \alpha$ is the error in the nth iteration. The equation*

$$e_{n+1} = ce_n^p + O(e_n^{p+1}),$$

is called the error equation. By substituting $e_n = x_n - \alpha$ for all n in any iterative method and simplifying, we obtain the error equation for that method. The value of p obtained is called the order of this method.

We consider the problem of numerical determine a real root α of nonlinear equation

$$f(x) = 0, \quad f : D \subset \mathbb{R} \to \mathbb{R}. \tag{2.1}$$

The known numerical method for solving equation (2.1) is the classical Newton's method given by

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \ldots$$

where x_0 is an initial approximation sufficiently close to α. The convergence order of the classical Newton's method is quadratic for simple roots [3].

If the second order derivative $f(x_n)$ is available, the third-order convergence rate can be achieved by using the Halley's method [5]. Its iterative formula is

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)}, \quad n = 1, 2, 3, \ldots .$$

3. Iterative method

For several reasons, we know that cubic spline functions are popular spline functions. They are smooth functions with which to fit data, and when used for interpolation, they do not have the oscillatory behavior that is the characteristic of high-degree polynomial interpolation.

We have derived a new quadrature rule which is based on spline interpolation [11]. In this section, we review it briefly and then present the iterative methods. Let us suppose $\Delta = \{x_i | \ i = 0, 1, 2\}$ be a uniform partition of the interval $[a, b]$ by knots $a = x_0 < \frac{a+b}{2} = x_1 < b = x_2$ and $h = x_1 - x_0 = x_2 - x_1$ and $y_i = f(x_i)$; then the cubic spline function $S_\Delta(x)$ which interpolates the values of the function f at the knots $x_0, x_1, x_2 \in \Delta$ and satisfies $S_\Delta''(a) = S_\Delta''(b) = 0$ is readily characterized by their moments, and these moments of interpolating cubic spline function can

be calculated as the solution of a system of linear equations. We can obtain the following representation of the cubic spline function in terms of its moments [10]:

$$S_\Delta(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3, \tag{3.1}$$

for $x \in [x_i, x_{i+1}], i = 0, 1$, where

$$\alpha_i = y_i, \qquad \beta_i = \frac{y_{i+1} - y_i}{h} - \frac{2M_i - M_{i+1}}{6}h, \qquad \gamma_i = \frac{M_{i+1} - M_i}{6h},$$

$$M_0 = M_2 = 0, \qquad M_1 = \frac{3}{2h^2}[f(a) - 2f(\frac{a+b}{2}) + f(b)].$$

Now from (3.1) we obtain

$$\int_a^b S_\Delta(x)dx = \int_a^{\frac{a+b}{2}} S_\Delta(x)dx + \int_{\frac{a+b}{2}}^b S_\Delta(x)dx$$

$$= \frac{h}{2}\left[f(a) + 2f(\frac{a+b}{2}) + f(b)\right] - \frac{h^3}{24}[M_0 + 2M_1 + M_2]$$

$$= \frac{h}{8}\left[3f(a) + 10f(\frac{a+b}{2}) + 3f(b)\right].$$

and thus

$$\int_a^b f(t)dt \simeq \frac{b-a}{16}\left\{3f(a) + 10f\left(\frac{a+b}{2}\right) + 3f(b)\right\}. \tag{3.2}$$

We use the above quadrature rule to approximate integrals and use it to obtain an iterative method. In order to do this, let $\alpha \in D$ be a simple zero of sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$ for an open interval D and x_0 is sufficiently close to α. To derive the iterative method, we consider the computation of the indefinite integral on an interval of integration arising from Newton's theorem

$$f(x) = f(x_n) + \int_{x_n}^x f'(t)dt. \tag{3.3}$$

Using the quadrature rule (3.2) to approximate the right integral of (3.3),

$$\int_{x_n}^x f'(t)dt = \frac{x - x_n}{16}\left\{3f'(x_n) + 10f'\left(\frac{x_n + x}{2}\right) + 3f'(x)\right\}, \tag{3.4}$$

and looking for $f(x) = 0$. From (3.3) and (3.4) we obtain

$$x = x_n - \frac{16f(x_n)}{3f'(x_n) + 10f'\left(\frac{x+x_n}{2}\right) + 3f'(x)}. \tag{3.5}$$

This fixed point formulation enables us to suggest the following implicit iterative method

$$x_{n+1} = x_n - \frac{16f(x_n)}{3f'(x_n) + 10f'\left(\frac{x_n + x_{n+1}}{2}\right) + 3f'(x_{n+1})}, \tag{3.6}$$

which requires the $(n+1)$th iterate x_{n+1} to calculate the $(n+1)$th iterate itself. To obtain the explicit form, we make use of the Newton's iterative step to compute the $(n+1)$th iterate x_{n+1} on the right-hand side of (3.6), namely replacing $f(\frac{x_n + x_{n+1}}{2})$ with $f(\frac{x_n + y_n}{2})$, where $y_n = x_n - \frac{f(x_n)}{f'(x_n)}$ is the Newton iterate and obtain the following explicit method:

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

$$x_{n+1} = x_n - \frac{16 f(x_n)}{3f'(x_n) + 10 f'(\frac{x_n + y_n}{2}) + 3f'(y_n)} \ , \quad n = 0, 1, 2, \ldots$$

where

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)}.$$

We can replace x_{n+1} in the right hand-side of (3.6) by Halley's method. Therefore we suggest the following algorithm:

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

$$x_{n+1} = x_n - \frac{16 f(x_n)}{3f'(x_n) + 10 f'\left(\frac{x_n + y_n}{2}\right) + 3f'(y_n)} \ , \quad n = 0, 1, 2, \ldots \tag{3.7}$$

where

$$y_n = x_n - \frac{2 f(x_n) f'(x_n)}{2(f'(x_n))^2 - f(x_n) f''(x_n)}. \tag{3.8}$$

4. ANALYSIS OF CONVERGENCE

In this section, we prove the convergence of Algorithm 2. In a similar way, one can prove the convergence of Algorithm 1.

Theorem 4.1. *Let $\alpha \in D$ be a simple zero of sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$ for an open interval D. If x_0 is sufficiently close to α, then the two-step iterative method defined by Algorithm 2 converges cubically to α in a neighborhood of α and it satisfies the error equation*

$$e_{n+1} = \left(\frac{-7}{16} c_2 \right) e_n^3 + O(e_n^4),$$

where $c_k = \frac{1}{k!} \frac{f^k(\alpha)}{f'(\alpha)}$, $k = 1, 2, 3, \ldots$ and $e_n = x_n - \alpha$.

Proof. Let α be a simple zero of f, i.e. $f(\alpha) = 0$ and $f'(\alpha) \neq 0$. Since f is sufficiently differentiable, by using Taylor's expansion of $f(x_n)$, $f'(x_n)$ and $f''(x_n)$ about α, we obtain

$$\begin{aligned}
f(x_n) &= f'(\alpha)(x_n - \alpha) + \frac{1}{2!} f''(\alpha)(x_n - \alpha)^2 + \frac{1}{3!} f'''(\alpha)(x_n - \alpha)^3 \\
&\quad + \frac{1}{4!} f^{(4)}(\alpha)(x_n - \alpha)^4 + \frac{1}{5!} f^{(5)}(\alpha)(x_n - \alpha)^5 + O((x_n - \alpha)^6) \\
&= f'(\alpha) \left[e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + O(e_n^6) \right], \tag{4.1}
\end{aligned}$$

$$f'(x_n) = f'(\alpha) \left[1 + 2c_2 e_n + 3c_3 e_n^2 + 4c_4 e_n^3 + 5c_5 e_n^4 + O(e_n^5) \right], \tag{4.2}$$

$$f''(x_n) = f'(\alpha) \left[2c_2 + 6c_3 e_n + 12c_4 e_n^2 + 20c_5 e_n^3 + O(e_n^4) \right], \tag{4.3}$$

where $c_k = \frac{1}{k!} \frac{f^k(\alpha)}{f'(\alpha)}$, $k = 1, 2, 3, \ldots$ and $e_n = x_n - \alpha$.

Form (4.1), (4.2) and (4.3) one obtains

$$f(x_n) f'(x_n) = f'^2(\alpha)[e_n + 3c_2 e_n^2 + (4c_3 + 2c_2^2)e_n^3 + O(e_n^4)] \tag{4.4}$$

$$f'^2(x_n) = f'^2(\alpha)[1 + 4c_2 e_n + (6c_3 + 4c_2^2)e_n^2 + (12c_2 c_3 + 8c_4)e_n^3 + O(e_n^4)] \tag{4.5}$$

$$f(x_n) f''(x_n) = f'^2(\alpha)[2c_2 e_n + (6c_3 + 2c_2^2)e_n^2 + (8c_2 c_3 + 12c_4)e_n^3 + O(e_n^4)] \tag{4.6}$$

Substituting (4.4), (4.5) and (4.6) into (3.8), we get

$$
\begin{aligned}
y_n &= x_n - \frac{2f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)} \\
&= \alpha + (c_2^2 - c_3)e_n^3 + O(e_n^4)
\end{aligned}
\tag{4.7}
$$

Now, from (4.7) we have

$$
\begin{aligned}
\frac{x_n + y_n}{2} &= x_n - \frac{f(x_n)f'(x_n)}{2(f'(x_n))^2 - f(x_n)f''(x_n)} \\
&= \alpha + \frac{1}{2}e_n + \frac{1}{2}(c_2^2 - c_3)e_n^3 + O(e_n^4)
\end{aligned}
\tag{4.8}
$$

Substituting (4.7) and (4.8) into the Taylor's expansion of $f(y_n)$, $f'(y_n)$ and $f''(\frac{x_n+y_n}{2})$ about α and get

$$
\begin{aligned}
f(y_n) &= f'(\alpha)(y_n - \alpha) + \frac{1}{2!}f''(\alpha)(y_n - \alpha)^2 + \frac{1}{3!}f'''(\alpha)(y_n - \alpha)^3 \\
&\quad + \frac{1}{4!}f^{(4)}(\alpha)(y_n - \alpha)^4 + O((y_n - \alpha)^5) \\
&= f'(\alpha)\left[(c_2^2 - c_3)e_n^3 + (-3c_2^3 + 6c_2c_3 - 3c_4)e_n^4 + O(e_n^5)\right],
\end{aligned}
\tag{4.9}
$$

$$
\begin{aligned}
f'(y_n) &= f'(\alpha)[1 + c_2(y_n - \alpha) + c_3(y_n - \alpha)^2 + c_4(y_n - \alpha)^3 + O((y_n - \alpha)^4)] \\
&= f'(\alpha)[1 + (2c_2^3 - 2c_2c_3)e_n^3 + O(e_n^4)],
\end{aligned}
\tag{4.10}
$$

$$
f'(\frac{x_n + y_n}{2}) = f'(\alpha)[1 + c_2 e_n + (c_2^3 - c_2c_3)e_n^3 + O(e_n^4)]
\tag{4.11}
$$

By substituting (4.2), (4.10) and (4.11) into (3.7), we get

$$
\begin{aligned}
x_{n+1} &= x_n - \frac{16f(x_n)}{3f'(x_n) + 10f'\left(\frac{x_n+y_n}{2}\right) + 3f'(y_n)} \\
&= \alpha + \left(\frac{-7}{16}c_2\right)e_n^3 + O(e_n^4).
\end{aligned}
$$

and thus

$$
e_{n+1} = \left(\frac{-7}{16}c_2\right)e_n^3 + O(e_n^4).
$$

This means that the method defined by algorithm 2 is cubically convergent.∎

Theorem 4.2. *Let $\alpha \in D$ be a simple zero of sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$ for an open interval D. If x_0 is sufficiently close to α, then the two-step iterative method defined by Algorithm 1 converges cubically to α in a neighborhood of α and it satisfies the error equation*

$$
e_{n+1} = \left(\frac{3}{8}c_2 + \frac{5}{8}c_2^2 - \frac{1}{32}c_3\right)e_n^3 + O(e_n^4),
$$

where $c_n = \frac{1}{n!}\frac{f^n(\alpha)}{f'(\alpha)}$, $n = 1, 2, 3, ...$ and $e_n = x_n - \alpha$.

  Proof. Similar to the proof of Theorem 4.1.∎

5. Numerical examples

In this section, we employ the methods obtained in this paper to solve some nonlinear equations and compare them with the Newton's method (NM). We use the stopping criteria $|x_{n+1} - x_n| < \epsilon$ and $|f(x_{n+1})| < \epsilon$, where $\epsilon = 10^{-14}$, for computer programs. All programs are written in Matlab.

In Table 1, the number of iterations (NI) and function evaluations (NFE) is given so that the stopping criterion is satisfied. In table 2, the comparison of the number of operations needed for obtaining solutions for examples 1 and 2 is given.

We use the following test functions and display the approximate zeros x^* found up to the 14th decimal place.

Example1 $f_1(x) = x^3 - x + 3,$ $x^* = -1.671699881657161$
Example2 $f_2(x) = x^3 + 4x^2 - 10,$ $x^* = 1.36523001341410$
Example3 $f_3(x) = -\cos(x) - x,$ $x^* = -0.73908513321516$
Example4 $f_4(x) = xe^{x^2} - \sin^2(x) + 3\cos(x) + 5,$ $x^* = -1.207647827130919$

TABLE 1.   Comparison of number of iterations and function evaluations of different iterative methods.

		NI			NFE		
$f(x)$	x_0	NM	Alg.1	Alg.2	NM	Alg.1	Alg.2
$f_1(x)$	5	41	7	6	82	28	30
$f_2(x)$	- 0.3	53	4	27	106	16	135
$f_3(x)$	$\pi/4$	70	4	4	140	16	20
$f_4(x)$	1.2	622	7	10	1044	28	50

TABLE 2.   Comparison of number of operations of different iterative methods.

		+ or −			× or ÷		
$f(x)$	x_0	NM	Alg.1	Alg.2	NM	Alg.1	Alg.2
$f_1(x)$	5	164	63	60	164	56	54
$f_2(x)$	- 0.3	212	36	297	265	32	216

In Tables 1 and 2, it is shown that in some cases, algorithms 1 and 2 are better than the Newton's method for solving nonlinear equations. It is observe that in some cases the new methods require less iteration and function evaluation than that of Newton's. Moreover, as you see in Table 2, the new methods require less number of operations than Newton's method.

6. Conclusion

We derived two iterative methods based on spline functions for solving nonlinear equations. Convergence proof is presented in details for algorithm 2. In Theorem 4.1 and 4.2, we proved that the order of convergence of these methods is three. Analysis of efficiency showed that these methods are preferable to the well-known Newton's method. From numerical examples, we showed that these methods have great practical utilities.

Acknowledgement

The author would like to thank the referees for their constructive suggestions and criticisms that led to an improved presentation.

References

[1] S. Abbasbandy, *Improving Newton?-Raphson method for nonlinear equations by modified Adomian decomposition method*, Appl. Math. Comput. **145**, (2003), 887?-893.

[2] S. Amat, S. Busquier, J.M. Gutierrez, *Geometric constructions of iterative functions to solve nonlinear equations*, J. Comput. Appl. Math. **157**, (2003) 197?-205.

[3] K. E. Atkinson, *An introduction to numerical analysis*, 2nd ed., John Wiley & Sons, New York, 1987.

[4] E. Babolian, J. Biazar, *Solution of nonlinear equations by adomian decomposition method*, Appl. Math. Comput. **132**, (2002) 167?-172.

[5] H. Bateman, *Halley's method for solving equations*, The American Mathematical Monthly, **Vol. 45, No. 1**, (1938), 11–17.

[6] F. Costabile, M.I. Gualtieri, S.S. Capizzano, *An iterative method for the solutions of nonlinear equations*, Calcolo **30**, (1999) 17?-34.

[7] M. Frontini, *Hermite interpolation and a new iterative method for the computation of the roots of non-linear equations*, Calcolo **40**, (2003) 109?-119.

[8] M. Frontini, E. Sormani, *Third-order methods from quadrature formulae for solving systems of nonlinear equations*, Appl. Math. Comput. **149**, (2004), 771?-782.

[9] A.Y. Ozban, *Some new variants of Newton's method*, Appl. Math. Lett. **17**, (2004), 677?-682.

[10] J. Stoer, R. Bulrisch, *Introduction to Numerical Analysis*, Second edition, Springer–Verlag, 1993.

[11] H. Taghvafard, *A New Quadrature Rule Derived from Spline Interpolation with Error Analysis*, Int. J. Comput. Math. Sci., **5**(3), (2011), 119?-124.

[12] J. Traub, *Iterative methods for the solution of equations*, Prentice?Hall, Englewood Cliffs, New Jersey, 1964.

[13] S. Weerakoon, T. G. I. Fernando, *A Variant of Newton's Method with Accelerated Third-Order Convergence*, Appl. Math. Lett., **13**, (2000), 87?-93.

[14] T. Yamamoto, *Historical development in convergence analysis for Newton's and Newton-like methods*, J. Comput. Appl. Math., **124**, (2000), 1–23.

Hadi Taghvafard

Department of Mathematics, Shiraz University, Shiraz, Iran.

E-mail address: `taghvafard@gmail.com`