A Family Third-order Methods with One Parameter for Solving Systems of Nonlinear Equations and BVP-ODEs

Zhong-Li LIU\textsuperscript{a,*}, Hong ZHANG\textsuperscript{b}

College of Biochemical Engineering, Beijing Union University, Beijing 100023, China

\textsuperscript{a}shtzhongli@buu.edu.cn, \textsuperscript{b}hgtzhanghong@buu.edu.cn

*Corresponding author

Keywords: Systems of Nonlinear Equations, Newton’s Iterative Methods, Third-order Convergence, BVP-ODEs.

Abstract. Efficient numerical solutions for systems of nonlinear equations have always appealed greatly to people in scientific computation and engineering fields. And how to make the two iterative approximate values as precise as possible is an important issue in actual computation. In this paper, we use the linear-combination method to construct a family third-order scheme with one parameter for solving systems of nonlinear equations. Their cubic convergence and corresponding error equation are proved theoretically, and numerical examples are demonstrated as well to show the efficiency and feasibility of the suggested iterative methods.

Introduction

For a system of nonlinear equations as follows:

\[ F(x) = (f_1(x), f_2(x), \ldots, f_n(x)) = 0 \]  

where \( x = (x_1, x_2, \ldots, x_n) \) and \( F: D \subseteq \mathbb{R}^n \to \mathbb{R}^n \) is a given nonlinear vector functions. Constructing an efficiently iterative method to approximate the root of equations (1) is a typical and important issue in scientific computation and engineering fields. The Newton’s method (see [1-2]) is one of the widely used methods for solving nonlinear equations by iteration as follows (M2):

\[ x^{(n+1)} = x^{(n)} - F'(x^{(n)})^{-1}F(x^{(n)}), \quad n = 0, 1, 2, \ldots \]  

where \( F'(x) \) is the inverse of first Fréchet derivative \( F'(x) \). It converges quadratically when an initial guess value \( x^{(0)} \) is close to the roots \( \xi \) of the system of nonlinear equations.

In order to improve the order of convergence, a few two-step variants of Newton’s methods with cubic convergence have been proposed in some literature [3-9] and references therein. S. Weerakoon and T. Fernando [3] using the Newton theorem \( f(x) = f(x_0) + \int_{x_0}^{x} f'(t) \, dt \), proposed the Newton-type method with third-order convergence for one equation and \( p \)-dimension equations in the following (M3):

\[ x^{(n+1)} = x^{(n)} - 2(F'(x^{(n)}) + F'(y^{(n)})^{-1})F(x^{(n)}), \quad n = 0, 1, 2, \ldots \]
where \( y^{(n)} = x^{(n)} - F'(x^{(n)})^{-1} F(x^{(n)}) \) is Newton’s method. M. Frontini and E. Sormani [4] presented third-order midpoint-methods using numerical quadrature formula as follows (M4):

\[
x^{(n+1)} = x^{(n)} - [F'(x^{(n)})^{-1} F(x^{(n)}) + F'(y^{(n)})]^{-1} F(x^{(n)})
\]

(4)

M. Darvishi and A. Barati [5] received a third-order iterative method based on Adomian decomposition method to the systems of nonlinear equations (M5):

\[
x^{(n+1)} = x^{(n)} - [F'(x^{(n)})^{-1} F(x^{(n)}) + F(y^{(n)})]^{-1} F(x^{(n)})
\]

(5)

In recent years, M. A. Noor and M. W. Wasteem [6] used two-point Newton-Cotes formulae to develop the following cubic convergence methods (M6):

\[
x^{(n+1)} = x^{(n)} - 4[F'(x^{(n)}) + 3F'(x^{(n)} + 2y^{(n)}) / 3]^{-1} F(x^{(n)})
\]

(6)

In this paper, by using the double-linear combination method for variables and derivative functions respectively, we suggest a family scheme containing one parameter with third-order convergence as follows (M7):

\[
x^{(n+1)} = x^{(n)} - [(1 - \beta) F'(x^{(n)}) + \beta F'((2\beta - 1)x^{(n)} + y^{(n)})]^{-1} F(x^{(n)})
\]

(7)

where the parameter \( \beta \in \mathbb{R}, \beta \neq 0 \) and \( y^{(n)} = x^{(n)} - F'(x^{(n)})^{-1} F(x^{(n)}) \), \( n = 0, 1, 2, \ldots \), which requires the evaluations of one function and two first derivative per iteration.

This paper is outlined as follows. A family of third-order convergence methods is developed and its corresponding error equations are proved theoretically in section 2, and the numerical examples using various similar methods for solving systems of nonlinear equations and nonlinear ODEs are compared to show the consistent convergence behavior in section 3. Conclusions are made in section 4.

A Family of Methods with One Parameter and Its Convergence

Now we use the double-linear combination method respectively for the four variables \( x^{(n)}, y^{(n)}, F'(x) \) and \( F'(y) \) based on Newton’s method to construct a family of two-step methods:

\[
\begin{align*}
y^{(n)} &= x^{(n)} - F'(x^{(n)})^{-1} F(x^{(n)}) \\
x^{(n+1)} &= x^{(n)} - [\alpha F'(x^{(n)}) + \beta F'(ax^{(n)} + by^{(n)})]^{-1} F(x^{(n)})
\end{align*}
\]

(8)

where \( \alpha, \beta, a, \) and \( b \) are four parameters, and \( a + b = 1 \).

In order to fulfill the following convergence theorem for scheme (7), we recall the lemma of Taylor’s expansion on vector functions (see [1]).

**Lemma.** Let \( F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \) be \( k \)-time Fréchet differentiable in a convex set \( D \) and for any \( x, h \in D \), the following expression holds:

\[
F(x + h) = F(x) + F'(x)h + \frac{1}{2} F''(x)h^2 + \frac{1}{3} F'''(x)h^3 + \cdots + \frac{1}{(p-1)!} F^{(p-1)}(x)h^{p-1} + R_p
\]

(9)

where \( \|R_p\| \leq \frac{1}{p!} \sup_{l \in [a, b]} \|F^{(p)}(x + lh)\| \|h\|^p \).

The convergence theorem of scheme (7) is described and proved in what follows.

**Theorem.** Let the vector function \( F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n \) be \( k \)-time Fréchet differentiable in a convex set \( D \subseteq \mathbb{R}^n \) containing a root \( \xi \) of \( F(x) \) and the initial value \( x^{(0)} \) be close to \( \xi \). Supposing \( F'(x) \) is
continuous and nonsingular at $\xi$, then, the iterative sequence $\{x^{(n)}\}_{n \geq 0}$ from the two-step scheme with one parameter (7) is cubically convergent, and the corresponding error equation is

$$e^{(n+1)} = [C_3^2 + 3(1 - \beta + (1 - 2\beta)^2 / 4\beta - 1/3)C_4]e^{(n)} + \mathcal{O}(e^{(n)})^4$$

(10)

where $\beta \neq 0$, $e^{(n)} = x^{(n)} - \xi$ and $C_i = \frac{1}{k!}F'(\xi)^{-1}F^{(i)}(\xi) \in L^r(R^r, R^r)$.

**Proof.** By Taylor’s expansion (9) of $F(x)$ at the point $x^{(n)}$, and noting $F(\xi) = 0$, $e^{(n)} = x^{(n)} - \xi$, we have

$$F(x^{(n)}) = F'(\xi)(x^{(n)} - \xi) + \frac{1}{2!}F''(\xi)(x^{(n)} - \xi)^2 + \frac{1}{3!}F'''(\xi)(x^{(n)} - \xi)^3 + \mathcal{O}(x^{(n)} - \xi)^4$$

(11)

and

$$F'(x^{(n)}) = F'(\xi)[I + 2C_2e^{(n)} + 3C_3e^{(n)} + \mathcal{O}(e^{(n)})^3]$$

(12)

Then

$$F'(x^{(n)})^{-1}F(x^{(n)}) = e^{(n)} - C_2(e^{(n)})^2 + (2C_2^2 - 2C_3)(e^{(n)})^3 + \mathcal{O}(e^{(n)})^4$$

(13)

Therefore,

$$F'(x^{(n)})^{-1}F'(x^{(n)}) = \mathcal{O}(e^{(n)})^3$$

(14)

From the first step of (8) and (14), we have

$$y^{(n)} = \xi + C_2(e^{(n)})^2 + 2(C_3 - C_2^2)(e^{(n)})^3 + \mathcal{O}(e^{(n)})^4$$

and

$$ax^{(n)} + by^{(n)} = \xi + ae^{(n)} + (1 - a)C_2(e^{(n)})^2 + 2(1 - a)(C_3 - C_2^2)(e^{(n)})^3.$$

Using the Taylor’s expansion again, we get

$$F'(ax^{(n)} + by^{(n)}) = F'(\xi)[I + 2aC_2e^{(n)} + (2 - 2a)C_2^2 + 3a^2C_3](e^{(n)})^2 + \mathcal{O}(e^{(n)})^3].$$

Therefore,

$$\alpha F'(x^{(n)}) + \beta F'(ax^{(n)} + by^{(n)}) =$$

$$F'(\xi)[(\alpha + \beta)I + (2\alpha + 2a\beta)C_2e^{(n)} + (2 - 2a)\beta C_2^2 + (3\alpha + 3a^2\beta)C_3](e^{(n)})^2 + \mathcal{O}(e^{(n)})^3]$$

(15)

From the second step of (8) and (15), we obtain

$$[(\alpha + \beta)I + (2\alpha + 2a\beta)C_2e^{(n)} + \mathcal{O}(e^{(n)})^2]e^{(n+1)} =$$

$$(\alpha + \beta - 1)e^{(n)} + (2\alpha + 2a\beta - 1)C_2(e^{(n)})^2 + ((2 - 2a)\beta C_2^2 + (3\alpha + 3a^2\beta - 1)C_3)(e^{(n)})^3 + \mathcal{O}(e^{(n)})^4$$

(16)

Furthermore, we obtain the error equation:

$$e^{(n+1)} = \frac{1}{\alpha + \beta}[(\alpha + \beta - 1)e^{(n)} + (2\alpha + 2a\beta - 1)C_2(e^{(n)})^2 + ((2 - 2a)\beta C_2^2 + (3\alpha + 3a^2\beta - 1)C_3)(e^{(n)})^3] + \mathcal{O}(e^{(n)})^4.$$
\[
\begin{align*}
\alpha + \beta - 1 &= 0 \\
2\alpha + 2\alpha \beta - 1 &= 0
\end{align*}
\]

From the equations (17), we have \(\alpha = 1 - \beta, a = (2\beta - 1)/(2\beta), b = 1/(2\beta)\), and get the corresponding error equation (11) when \(\beta \neq 0\) as follows:

\[
e^{n+1} = [C_1 + 3(1 - \beta + (1 - 2\beta^2)/4\beta - 3/3)C_0]e^n + O(e^n)^3.
\]

This shows that the two-step scheme (7) with one parameter is third-order convergent.

**Remark 1:** The one-parameter scheme (7) can give rise to a good many new third-order convergence iterative formulae. Specifically, when \(\beta = 1/2\), then the scheme (7) becomes the method (3); when \(\beta = 1\), then the scheme (7) becomes the method (4); when \(\beta = 3/4\), then the scheme (7) becomes the method (6), and so on.

**Numerical examples**

In this section, some examples are considered to illustrate the convergence behavior of the proposed method with one parameter \(M7\), and to compare the existing methods that have been mentioned in the previous section: \(M2, M3, M4, M5, M6\).

**Example 1.** Consider the following system of three equations:

\[
\begin{align*}
3x_1 - \cos(x_2x_3) - 0.5 &= 0 \\
x_2^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 &= 0 \\
e^{-x_2x_3} + 20x_3 + (10\pi - 3)/3 &= 0
\end{align*}
\]

(18)

where \(x^{(0)} = (0.1, 0.1, -0.1)^T\) is an initial guess value, and the precise solutions \(\xi = (0.5, 0, -\pi/6)^T\). We can obtain the approximation numerical solutions

\[
\xi^e = (0.50000000000\cdots, 0.00000000000\cdots, -0.52359877559829\cdots)^T,
\]

using all the different methods that have been mentioned above. The numerical results are listed in table 1 for the equations (18).

**Table 1. Numerical Results Comparison of Different Methods for the Equations (18).**

| Method | \(\beta\) | 1 | 2 | 3 | 4 | 5 |
|--------|--------|---|---|---|---|---|
| \(M2\) | \(\|\xi - \xi^e\|\) | 1.957e-2 | 1.589e-3 | 1.244e-5 | 7.760e-10 | 3.016e-18 |
| \(M3\) | \(\|\xi - \xi^e\|\) | 3.458e-1 | 2.588e-2 | 2.012e-4 | 1.254e-8 | 4.874e-17 |
| \(\beta = 1/2\) | \(\|\xi - \xi^e\|\) | 6.322e-3 | 5.898e-6 | 4.858e-16 | 2.876e-42 | 5.686e-124 |
| \(M4\) | \(\|\xi - \xi^e\|\) | 1.055e-1 | 9.352e-5 | 7.853e-14 | 4.689e-41 | 9.614e-123 |
| \(\beta = 1\) | \(\|\xi - \xi^e\|\) | 5.987e-3 | 4.912e-6 | 2.979e-15 | 6.356e-43 | 7.332e-126 |
| \(M5\) | \(\|\xi - \xi^e\|\) | 9.956e-2 | 7.945e-5 | 4.815e-14 | 1.072e-41 | 1.814e-124 |
| \(M6\) | \(\|\xi - \xi^e\|\) | 8.798e-3 | 2.820e-5 | 1.124e-12 | 7.131e-35 | 1.819e-101 |
| \(\beta = 3/4\) | \(\|\xi - \xi^e\|\) | 1.483e-1 | 4.558e-4 | 1.817e-11 | 1.152e-33 | 2.941e-100 |
| \(M7\) | \(\|\xi - \xi^e\|\) | 6.105e-3 | 5.202e-6 | 3.530e-15 | 1.103e-42 | 3.373e-125 |
| \(\beta = 1/2\) | \(\|\xi - \xi^e\|\) | 6.105e-3 | 5.202e-6 | 3.530e-15 | 1.103e-42 | 3.373e-125 |
| \(\beta = 3/4\) | \(\|\xi - \xi^e\|\) | 1.015e-1 | 8.408e-5 | 5.706e-14 | 1.784e-41 | 5.451e-124 |

**Example 2.** Consider the system of five equations with five variables:
where \( x^{(0)} = (1.2, 1.2, 1.2, 1.2, 1.2)^T \) is an initial guess value, and \( \xi = (1, 1, 1, 1)^T \) is the precise value of the solutions. We can obtain the approximation numerical solutions 
\[
\xi_1 = (1.00000000 , 1.00000000 , 1.00000000 ,1.00000000 ,1.00000000 )^T,
\]
using the different methods mentioned above. Table 2 lists the numerical comparison results of the different methods for the equations (19).

**Table 2. Numerical Comparison Results of the Different Methods for the Equations (19).**

| Method | \( k \) | 1   | 2   | 3   | 4   | 5   |
|--------|--------|-----|-----|-----|-----|-----|
| M2     | \( ||x - \xi|| \) | 3.342e-1 | 1.173e-2 | 6.765e-4 | 5.928e-8 | 1.406e-14 |
| M3: \( \beta = 1/2 \) | \( ||x - \xi|| \) | 6.416e-1 | 1.542e-1 | 9.202e-4 | 8.268e-7 | 1.900e-14 |
| M4: \( \beta = 1 \) | \( ||x - \xi|| \) | 2.546e-1 | 1.987e-1 | 1.669e-9 | 9.285e-28 | 1.708e-82 |
| M5     | \( ||x - \xi|| \) | 1.085e-1 | 2.172e-4 | 2.979e-11 | 1.394e-33 | 2.902e-99 |
| M6: \( \beta = 3/4 \) | \( ||x - \xi|| \) | 2.475e-2 | 2.382e-1 | 5.289e-5 | 1.445e-4 | 8.908e-15 |
| M7: \( \beta = -1/4 \) | \( ||x - \xi|| \) | 2.475e-2 | 2.382e-1 | 5.289e-5 | 1.445e-4 | 8.908e-15 |

Table 1 and Table 2 show the results by various methods respectively. These two tables display the iteration number \( k \), the error 2-norm of numerical solution \( x \) and precise solution \( \xi \), the 2-norm of the difference between \( F(x) \) and \( F(\xi) \) per iterations. From the numerical computation results, we can observe that the present schemes are in concordance with the theoretical analysis, as well as the other methods mentioned in this paper.

**Example 3.** Consider solving the following nonlinear boundary-value problem of ODEs:

\[
\begin{aligned}
y'(x) &= -\frac{1+2y}{2(y-a)}, \\
y(0) &= a, \quad y(1) = b, \quad a > b.
\end{aligned}
\]

We may discretize the above nonlinear boundary-value problem of ordinary differential equations (20) with the finite difference method. Partitioning the interval \([0,1]\):

\[
x_0 = 0 < x_1 < x_2 < \cdots < x_{n+1} < x_n = 1, \quad x_{i+1} = x_i + h, \quad h = 1/n,
\]

Let \( y_0 = y(x_0) = a \), \( y_1 = y(x_1), \ldots, y_{n+1} = y(x_{n+1}) \), \( y_i = y(x_i), \ldots, y_n = y(x_n) = b \). By using the numerical differential formula for second derivative \( y''_k = \frac{y_{k+2} - 2y_{k+1} + y_{k+1}}{h^2}, \quad (k = 1,2,\ldots,n-1) \), and first derivative \( y'_k = \frac{y_{k+1} - y_{k}}{h} \), \( (k = 1,2,\ldots,n) \), we take \( a = -1, b = -2 \), \( n = 10 \) herein, and obtain the system of nonlinear equations with nine variables:

\[
\begin{aligned}
-3y'_1 + 2y_1y_1 + 2(-1 - 2y_1 + y_1) + 1 + h^2 &= 0, \\
y'_{k+1} - 3y'_1 + 2y_1y_{k+1} + 2(y_{k+1} - 2y_1 + y_{k+1}) + h^2 &= 0, \quad k = 2,3,\ldots,8, \\
y'_{k+1} - 3y'_1 - 4y_0 + 2(y_0 - 2y_0 - 2) + h^2 &= 0
\end{aligned}
\]

(21)
where \( y^{(0)} = [-1, -1.2, -1.3, -1.4, -1.5, -1.6, -1.7, -1.8, -1.9]^T \) is the initial value. We obtain the approximate numerical solutions of this problem:
\[
\xi^* = (-0.9856842066381060246..., -1.3277909209782349538..., -1.47612014505493811723..., -1.5908427689409433521..., -1.685965186436336383..., -1.76720331746441407362..., -1.837623172844393236..., -1.8991136214716080859..., -1.9529403694589526283)^T.
\]

Unlike the above two examples (18) and (19), the precise solution \( \xi \) of (21) is unknown. So, we need to calculate the computational order of convergence (COC) using the following formula
\[
COC = \frac{\log\|F(x^{(k+1)})\|/\|F(x^{(k)})\|}{\log\|F(x^{(k+2)})\|/\|F(x^{(k+1)})\|},
\]
to demonstrate the theoretical order of convergence (see[10]), where \( \| \cdot \| \) is 2-norm. In this case, we use the last three approximations in the iterative process to calculate COC per time.

Now, we only take an arbitrarily \( \beta = -2 \) in the M(7) as an example, and the results of convergence order for the system of nonlinear equations (21) are shown in Table 3.

| Method | \( k \) | 1  | 2  | 3  | 4  | 5  | 6  |
|--------|--------|----|----|----|----|----|----|
| M7     | \( \beta = -2 \) | 3.0426 | 2.9835 | 3.0099 | 2.9999 | 2.9999 | - |
|        | \( \|F(x^{(k)})\| \) | 8.1213 | 1.301e-5 | 4.065e-14 | 1.711e-39 | 7.541e-116 | 6.484e-345 |

From the above table 3, the numerical results are agree with the theoretical analysis. The proposed method can also be used to effectively solve the boundary-value problem of ODEs.

Conclusions

In this paper, we construct a family scheme with one parameter for the systems of nonlinear equations by using the linear-combination. The suggested scheme with one parameter can give rise to a good many third-order convergent iterative formulae. According to the theoretical analysis and numerical computation examples, the family scheme constructed in this paper is efficient and feasible to solve the systems of nonlinear equations and discretized two-point boundary-value problems of nonlinear ordinary differential equations.

Acknowledgement

This research was supported by the Science & Technology Program of Beijing Municipal Commission of Education (No. KM201511417012).

References

[1] J.M. Ortega, W.G. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[2] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[3] S. Weerakoon, T. G. I. Fernando, A variant of Newton’s method with accelerated third-order convergence, Appl. Math. Lett.13 (2000) 87-93.
[4] M. Frontini, E. Sormani, Third-order methods from quadrature formulae for solving systems of nonlinear equations, Appl.Math. Comput. 149(2004) 771-782.

[5] M.T. Darvish, A. Barati, A third-order Newton-type method to solve systems of nonlinear equations, Appl.Math. Comput. 187(2007) 630-635.

[6] M.A. Noor, M. Wasteem, Some iterative methods for solving a system of nonlinear equations, J. Computers and Mathematics with Applications, 57(2009) 101-106.

[7] H.H.H. Homeier, A modified Newton method with cubic convergence: the multivariable case, J. Comput. Appl. Math, 169(2004) 161-169.

[8] M.T. Darvish, A. Barati, A fourth-order method from quadrature formulae to solve systems of nonlinear equations, Appl.Math. Comput.188(2007) 257-261.

[9] Hafiz M A, Bahgat M.S. M., An efficient two-step iterative method for solving system of nonlinear equations. Journal of Mathematics Research, 4(2012) 28-34.

[10] L.O. Jay, A note on Q-order of convergence, BIT 41(2001) 422-429.