

Article

Two Iterative Methods with Memory Constructed by the Method of Inverse Interpolation and Their Dynamics

Xiaofeng Wang *  and Mingming Zhu

School of Mathematical Sciences, Bohai University, Jinzhou 121000, China; mmzbhu@163.com

* Correspondence: w200888w@163.com; Tel.: +86-150-4169-9258

Received: 11 June 2020; Accepted: 1 July 2020; Published: 3 July 2020



Abstract: In this paper, we obtain two iterative methods with memory by using inverse interpolation. Firstly, using three function evaluations, we present a two-step iterative method with memory, which has the convergence order 4.5616. Secondly, a three-step iterative method of order 10.1311 is obtained, which requires four function evaluations per iteration. Herzberger's matrix method is used to prove the convergence order of new methods. Finally, numerical comparisons are made with some known methods by using the basins of attraction and through numerical computations to demonstrate the efficiency and the performance of the presented methods.

Keywords: multipoint iterative methods; with memory; nonlinear equations; inverse interpolation; root-finding

MSC: 65H05; 65B99

1. Introduction

Solving nonlinear equations is one of the most important problems in scientific computation. Since 1960's, many multipoint iterative methods have been proposed for solving nonlinear equations of the form $f(x) = 0$. Inverse interpolation method and self-accelerating parameter method are two effective ways to construct iterative method with memory. An iterative method uses some self-accelerating parameters, which is called the self-accelerating type iterative method with memory. The self-accelerating parameters is a variable parameter, which can be constructed by Newton interpolation or Hermite interpolation. Many efficient self-accelerating type iterative methods have been presented in recent years, see [1–11]. Džunić et al. [1], Soleymani et al. [2] and Sharma et al. [3] have proposed some derivative free iterative methods with one self-accelerating parameter for solving nonlinear equations. We [4–6] have obtained some Newton type iterative methods with memory using one simple self-accelerating parameter, which is constructed by the iterative sequences. By increasing the numbers of the self-accelerating parameters, Cordero et al. [7], Lotfi et al. [8] and Zafar et al. [9] have obtained some iterative method with high efficiency. Chicharro et al. [10] have analyzed the stability of iterative method with memory by dynamical theory. Narang et al. [11] have presented a class of Steffensen type method with memory for solving nonlinear systems. However, the self-accelerating parameter will be very complex if it is constructed by high order interpolation polynomial. In order to save computing time, we should construct the self-accelerating parameter with simple structure. If an iterative method with memory is constructed by using inverse interpolation polynomial, we will call it inverse interpolation method with memory. Neta [12] has derived a very fast inverse interpolation iterative method with memory, which is given by

$$\begin{cases} y_k = N(x_k) + [f(y_{k-1})\phi(z_{k-1}) - f(z_{k-1})\phi(y_{k-1})] \frac{f(x_k)^2}{f(y_{k-1}) - f(z_{k-1})}, \\ z_k = N(x_k) + [f(y_k)\phi(z_{k-1}) - f(z_{k-1})\phi(y_k)] \frac{f(x_k)^2}{f(y_k) - f(z_{k-1})}, \\ x_{k+1} = N(x_k) + [f(y_k)\phi(z_k) - f(z_k)\phi(y_k)] \frac{f(x_k)^2}{f(y_k) - f(z_k)}, \end{cases} \quad (1)$$

where

$$N(x_k) = x_k - \frac{f(x_k)}{f'(x_k)}, \quad (2)$$

$$\phi(t) = \frac{1}{f(t) - f(x_k)} \left(\frac{t - x_k}{f(t) - f(x_k)} - \frac{1}{f'(x_k)} \right). \quad (3)$$

Method (1) is denoted by Neta’s method (NETM) in this paper. Petković and Neta [13] have obtained that the convergence order of NETM is at least 10.1311. Inspired by Neta’s idea, Petković et al. [14] have presented the following two-step iterative method with memory

$$\begin{cases} y_k = N(x_k) + f(x_k)^2\phi(y_{k-1}), \\ x_{k+1} = N(x_k) + f(x_k)^2\phi(y_k), \end{cases} \quad (4)$$

where $N(x_k)$ and $\phi(t)$ are defined by (2) and (3), respectively. The convergence order of method (4) is 4.562. Method (4) is denoted by Petković’s method (PETM) in this paper.

In this paper, two new iterative methods with memory are proposed for solving nonlinear equations, which are constructed by using inverse interpolation method. We construct a two-step iterative method of order 4.5616 by using three function evaluations. In order to further improve the convergence order, a three-step iterative method with convergence order 10.1311 is obtained, which requires four function evaluations. Herzberger’s matrix method is used to prove the order of convergence of new methods. Finally, numerical experiments are employed to support the theory developed in this work. The basins of attraction of existing methods and our methods are presented and compared to illustrate their performance.

2. Two Inverse Interpolation Iterative Methods with Memory

Using inverse interpolation rational polynomial, we construct a two-step iterative method with memory. Let

$$R(f(x)) = \frac{a_0 + a_1(f(x) - f(x_k))}{1 + a_2(f(x) - f(x_k))}, \quad (5)$$

be a rational polynomial satisfying

$$x_k = R(f(x_k)), \quad (6)$$

$$\frac{1}{f'(x_k)} = R'(f(x_k)), \quad (7)$$

$$y_{k-1} = R(f(y_{k-1})). \quad (8)$$

From (6)–(8), we get

$$a_0 = x_k, \quad (9)$$

and the following system

$$\begin{cases} a_1 = \frac{1}{f'(x_k)} + a_2 x_k, \\ y_{k-1} = \frac{x_k + a_1(f(y_{k-1}) - f(x_k))}{1 + a_2(f(y_{k-1}) - f(x_k))}. \end{cases} \tag{10}$$

Solving system (10), we obtain

$$a_1 = \frac{1}{f'(x_k)} + \varphi(y_{k-1})x_k, \tag{11}$$

$$a_2 = \varphi(y_{k-1}), \tag{12}$$

$$\varphi(t) = \frac{1}{t - x_k} \left(\frac{1}{f'(x_k)} - \frac{t - x_k}{f(t) - f(x_k)} \right), \tag{13}$$

where $\varphi(t)$ is a rational function.

From (5), (9), (11) and (12), we get

$$y_k = R(0) = \frac{x_k - a_1 f(x_k)}{1 - a_2 f(x_k)} = x_k - \frac{f(x_k)}{f'(x_k)(1 - \varphi(y_{k-1})f(x_k))} \tag{14}$$

In the next step, x_{k+1} can be obtained by carrying out the same calculation as y_k but y_{k-1} should be instead by y_k . We get

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)(1 - \varphi(y_k)f(x_k))}. \tag{15}$$

Together with (14) and (15), we obtain a new two-step method with memory as follows:

$$\begin{cases} y_k = x_k - \frac{f(x_k)}{f'(x_k)(1 - \varphi(y_{k-1})f(x_k))}, \\ x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)(1 - \varphi(y_k)f(x_k))}. \end{cases} \tag{16}$$

where $\varphi(t)$ is defined by (13).

We will use Herzberger’s matrix method [15] to determine the convergence order of Method (16).

Theorem 1. *Let $\alpha \in I$ be a simple zero of a sufficiently differentiable function $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval I . If an initial approximation x_0 is sufficiently close to a simple zero α of f , then the order of convergence of the two-step method (16) with memory is at least 4.5616.*

Proof. The lower bound of order of a single step s-point method $x_k = G(x_{k-1}, x_{k-2}, \dots, x_{k-s})$ is the spectral radius of a matrix $M^{(s)} = (m_{ij})$, associated to this method, with elements

$m_{1,j}$ = amount of information required at point x_{k-j} , ($j = 1, 2, \dots, s$),

$m_{i,i-1} = 1$ ($i = 2, 3, \dots, s$),

$m_{i,j} = 0$ otherwise.

The lower bound of order of an s-step method $G = G_1 \circ G_2 \circ \dots \circ G_s$ is the spectral radius of the product of matrices $M^{(s)} = M_1 \cdot M_2 \cdot \dots \cdot M_s$. According to Method (16), we get the following matrices.

$$x_{k+1} = G_1(y_k, x_k) \rightarrow M_1 = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix},$$

$$y_k = G_2(x_k, y_{k-1}) \rightarrow M_2 = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}.$$

The matrix $M^{(2)}$ corresponding to Method (15) is

$$M^{(2)} = \begin{bmatrix} 1 & 2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 4 & 1 \\ 2 & 1 \end{bmatrix}.$$

The characteristic polynomial of matrix $M^{(2)}$ is

$$P_2(\lambda) = \lambda^2 - 5\lambda + 2$$

and the eigenvalues of matrix $M^{(2)}$ are $\{4.5616, 0.4384\}$. Since the spectral radius of matrix $M^{(2)}$ is $\rho(M^{(2)}) \approx 4.5616$. We conclude that the convergence order of Method (16) with memory is at least 4.5616. \square

In order to improve the computational efficiency of iterative method, we construct a three-step iterative method by using inverse interpolation rational polynomial. Let

$$R(f(x)) = \frac{a_0 + a_1(f(x) - f(x_k)) + a_2(f(x) - f(x_k))^2}{1 + a_3(f(x) - f(x_k))}, \tag{17}$$

be a rational polynomial satisfying

$$x_k = R(f(x_k)), \tag{18}$$

$$\frac{1}{f'(x_k)} = R'(f(x_k)), \tag{19}$$

$$y_{k-1} = R(f(y_{k-1})). \tag{20}$$

$$z_{k-1} = R(f(z_{k-1})). \tag{21}$$

From (18)–(21), we get

$$a_0 = x_k, \tag{22}$$

and the following system

$$\begin{cases} a_1 + a_3x_k = \frac{1}{f'(x_k)}, \\ a_3y_{k-1} - a_1 - a_2(f(y_{k-1}) - f(x_k)) = \frac{x_k - y_{k-1}}{f(y_{k-1}) - f(x_k)}, \\ a_3z_{k-1} - a_1 - a_2(f(z_{k-1}) - f(x_k)) = \frac{x_k - z_{k-1}}{f(z_{k-1}) - f(x_k)}. \end{cases} \tag{23}$$

Solving system (23), we obtain

$$a_1 = \frac{1}{f'(x_k)} + \varphi(y_{k-1})x_k, \tag{24}$$

$$a_2 = \frac{\varphi(y_{k-1}) - \varphi(z_{k-1})}{f[z_{k-1}, x_k] - f[y_{k-1}, x_k]}, \tag{25}$$

$$a_3 = \frac{\varphi(z_{k-1})f[y_{k-1}, x_k] - \varphi(y_{k-1})f[z_{k-1}, x_k]}{f[y_{k-1}, x_k] - f[z_{k-1}, x_k]}, \tag{26}$$

where $\varphi(t)$ is defined by (13).

From (16), (21) and (23)–(25), we have

$$y_k = R(0) = \frac{x_k - a_1f(x_k) + a_2f(x_k)^2}{1 - a_3f(x_k)}$$

$$= x_k - \frac{f(x_k)[1 - M_{k,1}(\varphi(y_{k-1}) - \varphi(z_{k-1}))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,1}(\varphi(y_{k-1})f[z_{k-1}, x_k] - \varphi(z_{k-1})f[y_{k-1}, x_k])f(x_k)\}}, \tag{27}$$

where $M_{k,1} = \frac{1}{f[z_{k-1}, x_k] - f[y_{k-1}, x_k]}$.

In the next step, z_k can be obtained by carrying out the same calculation as y_k but y_{k-1} should be instead by y_k . We get

$$z_k = x_k - \frac{f(x_k)[1 - M_{k,2}(\varphi(y_k) - \varphi(z_{k-1}))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,2}(\varphi(y_k)f[z_{k-1}, x_k] - \varphi(z_{k-1})f[y_k, x_k])f(x_k)\}}, \tag{28}$$

where $M_{k,2} = \frac{1}{f[z_{k-1}, x_k] - f[y_k, x_k]}$. x_{k+1} can be obtained by carrying out the same calculation as z_k but z_{k-1} should be instead by z_k .

$$x_{k+1} = x_k - \frac{f(x_k)[1 - M_{k,3}(\varphi(y_k) - \varphi(z_k))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,3}(\varphi(y_k)f[z_k, x_k] - \varphi(z_k)f[y_k, x_k])f(x_k)\}}, \tag{29}$$

where $M_{k,3} = \frac{1}{f[z_k, x_k] - f[y_k, x_k]}$.

Together with (27)–(29), we obtain a new three-step method with memory as follows:

$$\begin{cases} y_k = x_k - \frac{f(x_k)[1 - M_{k,1}(\varphi(y_{k-1}) - \varphi(z_{k-1}))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,1}(\varphi(y_{k-1})f[z_{k-1}, x_k] - \varphi(z_{k-1})f[y_{k-1}, x_k])f(x_k)\}}, \\ z_k = x_k - \frac{f(x_k)[1 - M_{k,2}(\varphi(y_k) - \varphi(z_{k-1}))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,2}(\varphi(y_k)f[z_{k-1}, x_k] - \varphi(z_{k-1})f[y_k, x_k])f(x_k)\}}, \\ x_{k+1} = x_k - \frac{f(x_k)[1 - M_{k,3}(\varphi(y_k) - \varphi(z_k))f(x_k)f'(x_k)]}{f'(x_k)\{1 - M_{k,3}(\varphi(y_k)f[z_k, x_k] - \varphi(z_k)f[y_k, x_k])f(x_k)\}}, \end{cases} \tag{30}$$

where $M_{k,1} = \frac{1}{f[z_{k-1}, x_k] - f[y_{k-1}, x_k]}$, $M_{k,2} = \frac{1}{f[z_{k-1}, x_k] - f[y_k, x_k]}$ and $M_{k,3} = \frac{1}{f[z_k, x_k] - f[y_k, x_k]}$.

Now, we give the order of convergence of Method (30) by the following theorem.

Theorem 2. Let $\alpha \in I$ be a simple zero of a sufficiently differentiable function $f : I \subset \mathbb{R} \rightarrow \mathbb{R}$ for an open interval I . If an initial approximation x_0 is sufficiently close to a simple zero α of f , then the order of convergence of the three-step Method (30) with memory is at least 10.1311.

Proof. Using Herzberger’s matrix method [15], we obtain the following matrices of Method (30)

$$x_{k+1} = G_1(z_k, y_k, x_k) \rightarrow M_1 = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$z_k = G_2(y_k, x_k, z_{k-1}) \rightarrow M_2 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

$$y_k = G_2(x_k, z_{k-1}, y_{k-1}) \rightarrow M_3 = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

Matrix $M^{(3)}$ corresponding to three-step Method (30) is

$$M^{(3)} = M_1 M_2 M_3 = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 8 & 3 & 2 \\ 4 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}.$$

The characteristic polynomial of matrix $M^{(3)}$ is

$$P_3(\lambda) = \lambda^3 - 11\lambda^2 + 9\lambda - 2$$

and the eigenvalues of matrix $M^{(3)}$ are $\{10.1311, 0.4344 \pm 0.0932i\}$. Since the spectral radius of matrix $M^{(3)}$ is $\rho(M^{(3)}) \approx 10.1311$. We conclude that the convergence order of method with memory (30) is at least 10.1311. \square

3. Numerical Results

Our methods (16) and (30) were compared with Neta’s method (NETM), Petković’s method (PETM) and Wang’s method (WANM) for solving some nonlinear equations. Numerical computations were performed in the Matlab 7 computer algebra system. The computer specifications are Microsoft Windows 7 Intel(R), Core(TM) i3-2350M CPU, 1.79 GHz with 2GB of RAM.

Wang’s method [8]

$$\begin{cases} z_n = x_n - \frac{f(x_n)}{f'(x_n)}, \\ y_n = z_n - \lambda_n(z_n - x_n)^2, \\ x_{n+1} = y_n - \frac{f(y_n)}{2f[x_n, y_n] - f'(x_n)}, \end{cases} \tag{31}$$

where $\lambda_n = \frac{1}{x_{n-1} - z_{n-1}} \left[\frac{x_n - z_{n-1}}{z_{n-1} - x_{n-1}} + \frac{2(z_n - x_n)}{x_n - y_{n-1}} \right]$.

The following test functions were used in numerical experiments.

$$f_1(x) = xe^{x^2} - \sin^2(x) + 3 \cos(x) + 5, a \approx -1.2076478271309189, x_0 = -1.5,$$

$$f_2(x) = x^5 + x^4 + 4x^2 - 15, a \approx 1.3474280989683050, x_0 = 1.1,$$

$$f_3(x) = \arcsin(x^2 - 1) - 0.5x + 1, a \approx 0.59481096839836918, x_0 = 1.1.$$

$$f_4(x) = \ln(x^2 - 2x + 2) + \exp(x^2 - 4x + 4) \sin(x - 1), a = 1, x_0 = 0.54$$

$$f_5(x) = \sin(x) - x/3, a \approx 2.2788626600758283, x_0 = 3.27$$

$$f_6(x) = 10xe^{-x^2} - 1, a \approx 1.6796306104284499, x_0 = 2.1$$

The absolute errors $|x_k - a|$ in the first four iterations are given in Table 1, where a is the exact root computed with 3600 significant digits. For Methods (1), (4), (16) and (30), the initial approximation y_{-1} is calculated by $y_{-1} = N(x_0)$, z_{-1} is calculated by $z_{-1} = y_{-1} + |f(x_0)|/10$. The approximated computational order of convergence (ACOC) is defined by [16]:

$$\rho \approx \frac{\ln(|x_{n+1} - x_n| / |x_n - x_{n-1}|)}{\ln(|x_n - x_{n-1}| / |x_{n-1} - x_{n-2}|)}. \tag{32}$$

Table 1. Numerical results for $f_i(x) (i = 1, \dots, 6)$.

Methods	$f_i(x)$	$ x_1 - a $	$ x_2 - a $	$ x_3 - a $	$ x_4 - a $	ρ
PETM	$f_1(x)$	0.35559×10^{-2}	0.19719×10^{-10}	0.59271×10^{-48}	0.47227×10^{-219}	4.5599492
	$f_2(x)$	0.15137×10^{-2}	0.14866×10^{-12}	0.51501×10^{-58}	0.26586×10^{-265}	4.5613777
	$f_3(x)$	0.14922×10^{-4}	0.65850×10^{-24}	0.57645×10^{-112}	0.15218×10^{-513}	4.5603961
	$f_4(x)$	0.47316×10^{-1}	0.40385×10^{-5}	0.11174×10^{-23}	0.26640×10^{-108}	4.5598985
	$f_5(x)$	0.13949×10^{-2}	0.61756×10^{-14}	0.18890×10^{-65}	0.25756×10^{-300}	4.5592146
	$f_6(x)$	0.11377	0.51251×10^{-4}	0.59924×10^{-19}	0.51705×10^{-87}	4.5582387
WANM	$f_1(x)$	0.51808×10^{-2}	0.86196×10^{-8}	0.38505×10^{-33}	$0.408023 \times 10^{-142}$	4.2988150
	$f_2(x)$	0.25772×10^{-2}	0.18965×10^{-11}	0.23539×10^{-49}	$0.224057 \times 10^{-212}$	4.3006584
	$f_3(x)$	0.11168×10^{-1}	0.13757×10^{-8}	0.13978×10^{-38}	$0.108856 \times 10^{-167}$	4.3046156
	$f_4(x)$	0.55431×10^{-1}	0.58447×10^{-3}	0.89039×10^{-13}	0.477788×10^{-54}	4.2038911
	$f_5(x)$	0.96475×10^{-2}	0.22227×10^{-10}	0.50290×10^{-47}	$0.226046 \times 10^{-204}$	4.2937795
	$f_6(x)$	0.57236×10^{-1}	0.69530×10^{-5}	0.34120×10^{-22}	0.116978×10^{-96}	4.3020510
Method (16)	$f_1(x)$	0.22190×10^{-3}	0.81848×10^{-18}	0.18945×10^{-83}	0.92583×10^{-383}	4.5601986
	$f_2(x)$	0.80426×10^{-4}	0.26205×10^{-19}	0.94196×10^{-90}	0.53248×10^{-411}	4.5603052
	$f_3(x)$	0.89331×10^{-3}	0.21584×10^{-15}	0.10233×10^{-72}	0.42013×10^{-334}	4.5598019
	$f_4(x)$	0.11157×10^{-1}	0.39468×10^{-11}	0.16859×10^{-53}	0.14820×10^{-246}	4.5564931
	$f_5(x)$	0.13173×10^{-2}	0.19405×10^{-14}	0.41159×10^{-68}	0.81326×10^{-313}	4.5591294
	$f_6(x)$	0.92941×10^{-2}	0.29837×10^{-9}	0.23171×10^{-43}	0.63284×10^{-199}	4.5606727
NETM	$f_1(x)$	0.99121×10^{-4}	0.28313×10^{-38}	0.31361×10^{-388}		10.130669
	$f_2(x)$	0.18992×10^{-5}	0.51355×10^{-59}	0.52899×10^{-579}		10.122309
	$f_3(x)$	0.23910×10^{-10}	0.60431×10^{-113}	$0.32556 \times 10^{-1151}$		10.119841
	$f_4(x)$	0.20425×10^{-4}	0.23991×10^{-48}	0.21325×10^{-492}		10.108125
	$f_5(x)$	0.30800×10^{-8}	0.68005×10^{-89}	0.63841×10^{-905}		10.117379
	$f_6(x)$	0.31973×10^{-2}	0.12293×10^{-24}	0.50418×10^{-251}		10.099743
Method (30)	$f_1(x)$	0.35789×10^{-10}	0.93271×10^{-112}	$0.61422 \times 10^{-1140}$		10.121489
	$f_2(x)$	0.50281×10^{-9}	0.54597×10^{-96}	0.14347×10^{-976}		10.125776
	$f_3(x)$	0.26423×10^{-10}	0.10036×10^{-111}	$0.36975 \times 10^{-1138}$		10.120580
	$f_4(x)$	0.11409×10^{-9}	0.10448×10^{-99}	$0.76373 \times 10^{-1012}$		10.130546
	$f_5(x)$	0.15455×10^{-6}	0.10783×10^{-75}	0.75250×10^{-775}		10.109797
	$f_6(x)$	0.19512×10^{-7}	0.14705×10^{-81}	0.15699×10^{-831}		10.117955

Table 1 shows that numerical results are in concordance with the theory developed in this paper. The computing accuracy of our Method (16) is better than that of NETM for solving nonlinear equations $f_i (i = 1, 2, 4, 5, 6)$. The computing accuracy of our Method (30) is better than that of method PETM for solving nonlinear equations $f_i (i = 1, 2, 4, 6)$.

4. Dynamical Analysis

The dynamical properties of the rational function give us important information about numerical features of the iterative method as its stability and reliability. In this section, we compare our Methods (16) and (30) to Newton’s method (2), NETM (1), PETM (4) and WANM (30) by using the basins of attraction for three complex polynomials $f(z) = z^k - 1, k = 2, 3, 4, 5, 6$. To generate the basins of attraction for the zeros of a polynomial and an iterative method we take a grid of 300×300 points in a

rectangle $D = [-3.0, 3.0] \times [-3.0, 3.0] \subset \mathbb{C}$ and we use these points as z_0 . If the sequence generated by iterative method reaches a zero z^* of the polynomial with a tolerance $|z_k - z^*| < 10^{-5}$ and a maximum of 25 iterations, we decide that z_0 is in the basin of attraction of the zero and we paint this point in a blue color for this root. In the same basin of attraction, the number of iterations needed to achieve the solution is showed in darker or brighter colors (the less iterations, the brighter color). Black color denotes lack of convergence to any of the roots (with the maximum of iterations established) or convergence to the infinity. The parameters used in iterative Method (30) is $\lambda_0 = 0.001$. All the figures are created by the Mathematica 4 computer algebra system. The computer specifications are Microsoft Windows 7 Intel(R), Core(TM) i3-2350M CPU, 1.79 GHz with 2GB of RAM.

Figure 1 show that our Method (16) and Newton’s method are global convergence for solving complex polynomial $f(z) = z^2 - 1$. Compared to the other method, our Method (16) has the least diverging points in Figure 2. Figure 3 show that the convergence speed of our method (30) is faster than that of the other methods. Figures 1–5 show that the stability of Method (16) is better than the other methods and method WANM has the worst stability. The basins of attraction for our Method (16) with memory are larger than the other methods. On the whole, our Methods (16) and (30) are better than the other methods in this paper.

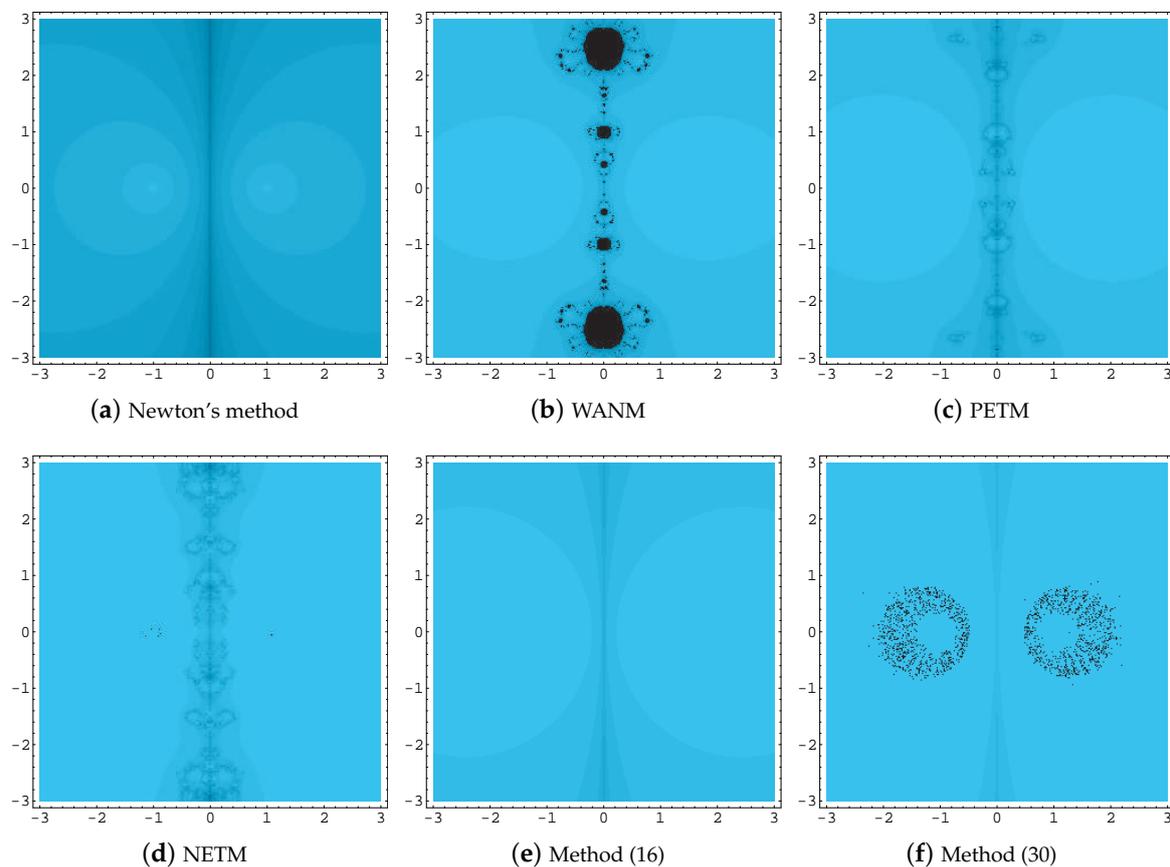


Figure 1. The results are for the polynomial $z^2 - 1$.

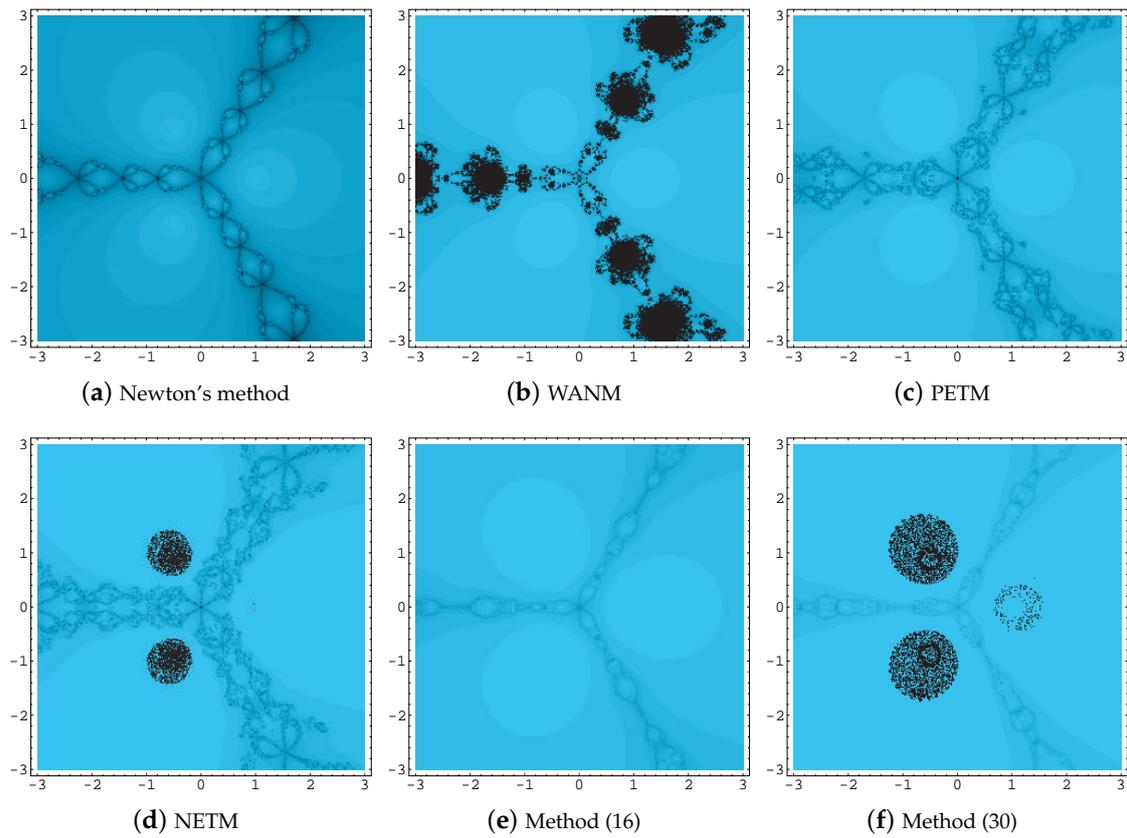


Figure 2. The results are for the polynomial $z^3 - 1$.

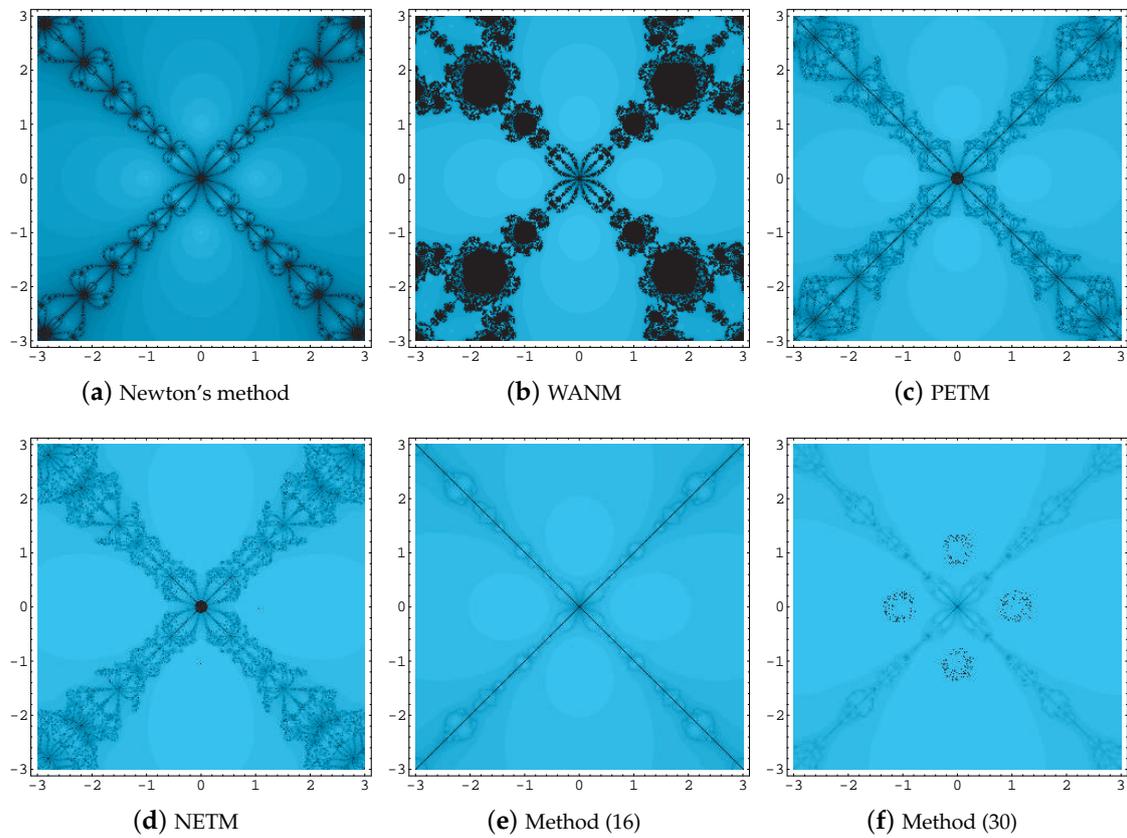


Figure 3. The results are for the polynomial $z^4 - 1$.

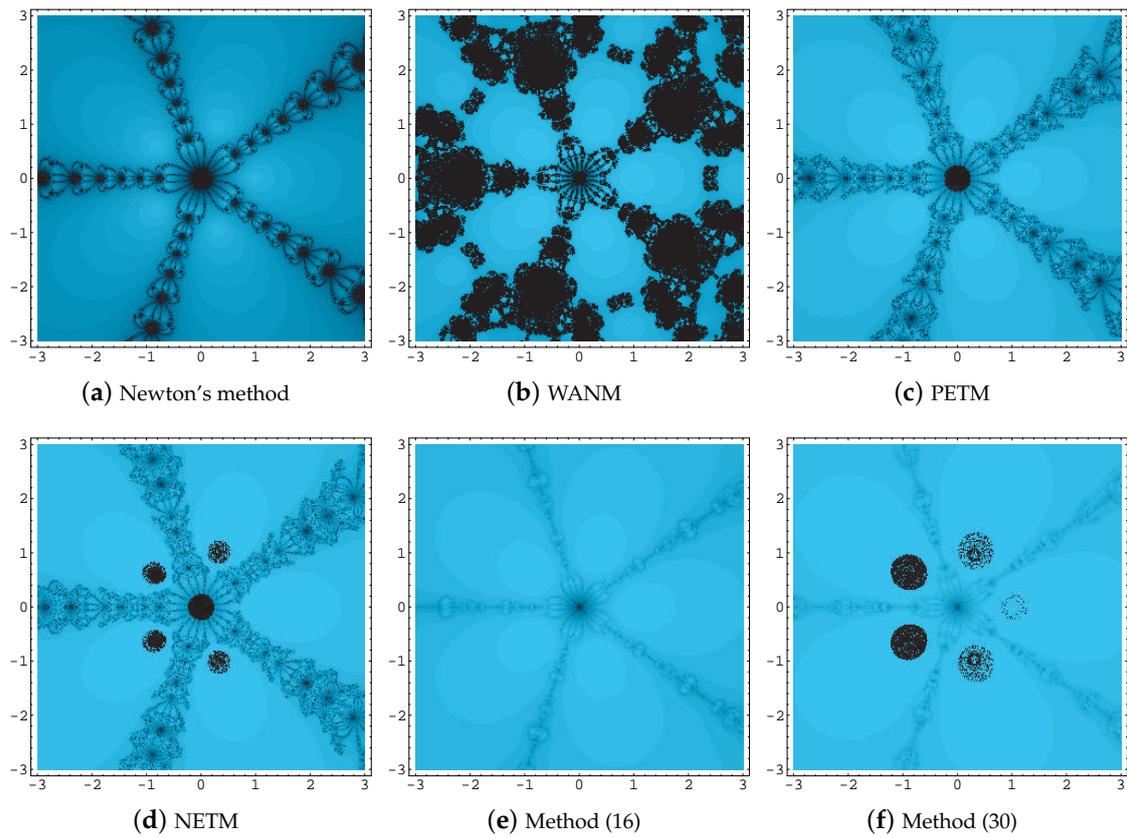


Figure 4. The results are for the polynomial $z^5 - 1$.

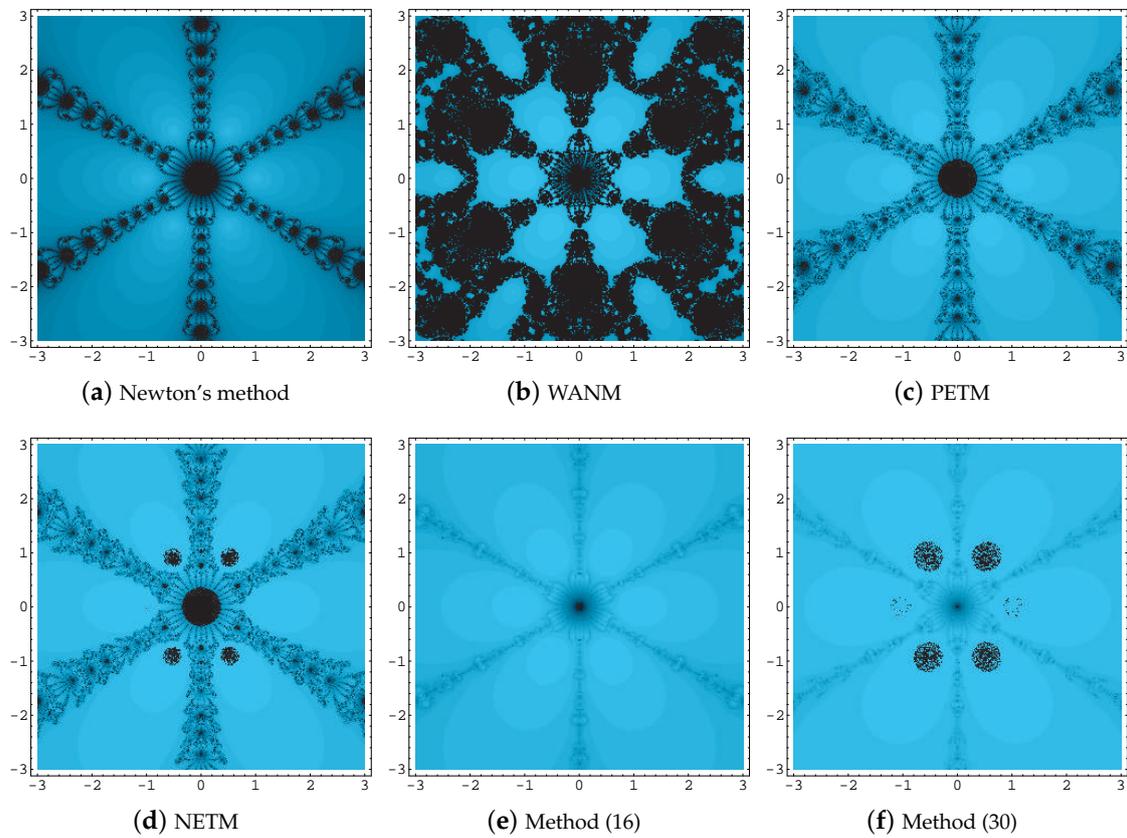


Figure 5. The results are for the polynomial $z^6 - 1$.

5. Conclusions

In this paper, two new iterative methods with memory are proposed for solving nonlinear equations, which are constructed by inverse interpolation method. We first propose a two-step iterative method with convergence order 4.5616, which requires three function evaluations. By increasing a function evaluation, a three-step method is obtained, which has the convergence order 10.1311. New methods are compared in performance with the existing methods by numerical examples. Numerical examples confirm the theoretical results. The basins of attraction of existing methods and our methods are presented and compared to illustrate their performance. The basin of attraction for our Method (16) is better than any of the other methods.

Author Contributions: Methodology, X.W. and M.Z.; writing—original draft preparation, X.W.; writing—review and editing, X.W. and M.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 61976027), Natural Science Foundation of Liaoning Province of China (No. 20180551262), Educational Commission Foundation of Liaoning Province of China (No. LJ2019010) and University-Industry Collaborative Education Program (Nos. 201901077017, 201902014012, 201902184038).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Džunić, J.; Petković, M.S.; Petković, L.D. Three-point methods with and without memory for solving nonlinear equations. *Appl. Math. Comput.* **2012**, *218*, 4917–4927. [[CrossRef](#)]
2. Soleymani, F.; Lotfi, T.; Tavakoli, E.; Haghani, F.K. Several iterative methods with memory using- accelerators. *Appl. Math. Comput.* **2015**, *254*, 452–458. [[CrossRef](#)]
3. Sharma, J.R.; Guha, R.K.; Gupta, P. Some efficient derivative free methods with memory for solving nonlinear equations. *Appl. Math. Comput.* **2012**, *219*, 699–707. [[CrossRef](#)]
4. Wang, X.; Tao, Y. A new Newton method with memory for solving nonlinear equations. *Mathematics* **2020**, *8*, 108. [[CrossRef](#)]
5. Wang, X.; Fan, Q. A modified Ren's method with memory using a simple self-accelerating parameter. *Mathematics* **2020**, *8*, 540. [[CrossRef](#)]
6. Wang, X. A family of Newton-type iterative methods using some special self-accelerating parameters. *Int. J. Comput. Math.* **2018**, *95*, 2112–2127. [[CrossRef](#)]
7. Cordero, A.; Lotfi, T.; Bakhtiari, P.; Torregrosa, J.R. An efficient two-parametric family with memory for nonlinear equations. *Numer. Algor.* **2015**, *68*, 323–335. [[CrossRef](#)]
8. Lotfi, T.; Assari, P. New three- and four-parametric iterative with memory methods with efficiency index near 2. *Appl. Math. Comput.* **2015**, *270*, 1004–1010. [[CrossRef](#)]
9. Zafar, F.; Cordero, A.; Torregrosa, J.R.; Rafi, A. A class of four parametric with- and without memory root finding methods. *Comp. Math. Methods* **2019**, *1*, 1–14. [[CrossRef](#)]
10. Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Dynamics of iterative families with memory based on weight functions procedure. *J. Comput. Appl. Math.* **2019**, *354*, 286–298. [[CrossRef](#)]
11. Narang, M.; Bhatia, S.; Alshomrani, A.S. General efficient class of steffensen type methods with memory for solving systems of nonlinear equations. *J. Comput. Appl. Math.* **2019**, *352*, 23–39. [[CrossRef](#)]
12. Neta, B. A New Family of Higher Order Methods for Solving Equations. *Int. J. Comput. Math.* **1983**, *14*, 191–195. [[CrossRef](#)]
13. Petković, M.S.; Neta, B.; Petković, L.D.; Džunić, J. Multipoint methods for solving nonlinear equations: A survey. *Appl. Math. Comput.* **2014**, *226*, 635–660.
14. Petković, M.S.; Džunić, J.; Neta, B. Interpolatory multipoint methods with memory for solving nonlinear equations. *Appl. Math. Comput.* **2011**, *218*, 2533–2541. [[CrossRef](#)]

15. Herzberger, J. Über Matrixdarstellungen für iterationverfahren bei nichtlinearen Gleichungen. *Computing* **1974**, *12*, 215–222. [[CrossRef](#)]
16. Grau-Sánchez, M.; Noguera, M.; Grau, À.; Herrero, J.R. On new computational local orders of convergence. *Appl. Math. Lett.* **2012**, *25*, 2023–2030.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).