Three-step iterative weight function scheme with memory for solving nonlinear problems

Alicia Cordero [1] | **Neus Garrido**[2] | **Juan R. Torregrosa**[*1]
| **Paula Triguero-Navarro**[1]

[1]Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, València, Spain

[2]Escuela Superior de Ingeniería y Tecnología, Universidad Internacional de La Rioja, Logroño, Spain

Correspondence
Email: jrtorre@upv.es

Funding information

In this manuscript, we present a parametric family of derivative-free 3-steps iterative methods with a weight function for solving nonlinear equations. We study various ways of introducing memory to this parametric family in order to increase the order of convergence without new functional evaluations. We also performed numerical experiments to compare the iterative methods from different points of view.

KEYWORDS
Iterative method; memory methods; nonlinear equation; divided differences

1 | INTRODUCTION

In most problems appearing in Applied Mathematics, nonlinear equations of the form $f(x) = 0$ must be solved, where $f : D \subseteq \mathbb{R} \to \mathbb{R}$. Sometimes it is not possible or is very complicated to solve these equations exactly. This is the reason why methods that obtain an approximation to the zero α of function f have arisen. Iterative schemes aim us to obtain a sequence of iterations, which under certain conditions, converges to this solution.

These methods are categorised in different ways, depending on whether they use derivatives to obtain the approximations, if they are methods with memory, that is, whether they use only one previous iteration or if they use more than one previous iteration to obtain the next one, whether they have an auxiliary function or weight function in the iterative expression, etc.

In this work, we present an extension of the derivative-free two-step parametric family with a weight function,

obtained in [9], which has the following expression:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f[w_k, x_k]}, & \text{where} \quad w_k = x_k + \beta f(x_k) \quad \forall \beta \in \mathbb{R} \backslash \{0\}, \\[3mm] x_{k+1} = y_k - H(\mu_k) \dfrac{f(y_k)}{f[y_k, x_k]}, & \text{where} \quad \mu_k = \dfrac{f(y_k)}{f(w_k)}, \ k = 0, 1, \dots \end{cases} \tag{1}$$

By adding one more step with a similar feature, we define the following parametric family of iterative methods:

$$\begin{cases} y_k = x_k - \dfrac{f(x_k)}{f[w_k, x_k]}, & \text{where } w_k = x_k + \beta f(x_k) \\[3mm] z_k = y_k - H(\mu_k) \dfrac{f(y_k)}{f[y_k, x_k]}, & \text{where } \mu_k = \dfrac{f(y_k)}{f(w_k)}, \\[3mm] x_{k+1} = z_k - G(v_k) \dfrac{f(z_k)}{f[z_k, y_k]}, & \text{where } v_k = \dfrac{f(z_k)}{f(y_k)}, \end{cases} \tag{2}$$

being $f[a, b] = \frac{f(a) - f(b)}{a - b}$ with $a \neq b$, $a, b \in \mathbb{R}$, being $H(t)$ and $G(t)$ real functions. We denote this parametric family by $M_{6,\beta}$, as it is further proven that it has order of convergence 6 for any real value of $\beta \neq 0$.

Usually, methods with higher order of convergence tend to converge in fewer iterations. For this reason we want to increase the order, but without adding functional evaluations, as that would increase the computational cost. For this reason, we introduce memory to class $M_{6,\beta}$ by replacing the parameter with an expression that depends on the previous iterations and their functional evaluations.

The development of iterative schemes with memory is quite recent, far from classical Secant method

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_{k-1}, x_k]}, \quad \text{given} \quad x_0, x_1, k \geq 1.$$

whose order of convergence is $p = \frac{1+\sqrt{5}}{2} \approx 1.618$. In 1964, Traub in [26] developed the scheme

$$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{\Gamma_k}, \\ \Gamma_k &= f[x_k + \gamma_k f(x_k), x_k], \\ \gamma_k &= -\frac{1}{\Gamma_{k-1}}, \text{ given } x_0, \gamma_0. \end{aligned}$$

that is a version of known Steffensen's method including memory, by using an accelerating parameter by first time. The order of convergence of this procedure was $p = 1 + \sqrt{2} \approx 2.414$. In 2011, Grau-Sánchez et al. in [11]designed a generalized Secant method, with order of convergence $p = \frac{1+\sqrt{1+4n}}{2}$ consisting in n steps, defined by

$$x_k^{(n)} = x_k^{(n-1)} - [x_{k-1}, x_k; F]^{-1} F(x_k^{(n-1)}), \qquad k > 1,$$

where $x_k^{(0)} = x_k$ and, in the last step, the last computed term is $x_{k+1} = x_k^{(n)}$. Also more general high-order methods have been generated in the last decades, as those proposed by Neta in [17], by using inverse interpolation, with order $p \approx 10.815$ by means of several accelerating parameters. Also, in [21, 22], by using Newton's interpolation approach in the replacement of the accelerating parameters, Petkovic et al. constructed, among others, a general scheme with one accelerating parameter, with high order of convergence. However, these high increasing of the order of convergence usually involves a greater computational cost. This is the reason why a balanced relation between order

of convergence and computational efficiency is needed.

To prove the order of the introduced memory methods we use the Ortega-Rheinboldt theorem, which can be found in [19]:

Theorem 1 *Let ϕ be an iterative method with memory that generates a sequence $\{x_k\}$ of approximations to the root α, and let this sequence converges to α. If there exist a nonzero constant η and positive numbers t_i, $i = 0, \ldots, m$ such that the inequality*

$$|e_{k+1}| \leq \eta \prod_{i=0}^{m} |e_{k-i}|^{t_i},$$

holds, then the R-order of convergence of the iterative method ϕ is at least p, where p is the unique positive root of the equation

$$p^{m+1} - \sum_{i=0}^{m} t_i p^{m-i} = 0.$$

Along this manuscript, the convergence analysis of class $M_{6,\beta}$ is made in Section 1, meanwhile the memorizing process and the demonstration of its order of convergence is made in Section 2. Finally, in Section 3 we conclude with some numerical experiments where the solutions obtained by the different methods with and without memory are compared. Also, the approximations obtained and the number of iterations necessary to obtain them are analyzed for the sake of comparison, as well as the wideness of the set of converging initial guesses and the total efficiency index of the methods applied to specific problems.

2 | CONVERGENCE ANALYSIS

Let $f : D \subset \mathbb{R} \to \mathbb{R}$ be a sufficiently differentiable function in an open set D that contains a root α of $f(x) = 0$. Let us consider the expression

$$f[x + h, x] = \int_0^1 f'(x + th) \, dt, \tag{3}$$

obtained by Genochi-Hermite in [19]. Using the Taylor's expansion $f'(x + th)$ around x and integrating, we obtain the following development

$$f[x + h, x] = f'(x) + \frac{1}{2} f''(x) h + \frac{1}{6} f'''(x) h^2 + O(h^3), \tag{4}$$

which we use to prove that the order of convergence of methods $M_{6,\beta}$, defined in (2), is 6 for any $\beta \in \mathbb{R} \setminus \{0\}$.

Theorem 2 *Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood D of α, such that $f(\alpha) = 0$. We assume that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfy that $H(0) = 1$, $H'(0) = 1$ and $|H''(0)| < \infty$, and let $G(t)$ be a real function that satisfy that $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking an initial approximation x_0 close enough to α, the sequence of iterated $\{x_k\}$ generated by the proposed method (2) converge to α with order 6, and its error equation*

is:

$$e_{k+1} = \frac{-C_2}{4}\left(1 + \beta f'(\alpha)\right)\left(-2C_3(1 + \beta f'(\alpha)) + C_2^2(6 + 4\beta f'(\alpha) - H_2)\right) \tag{5}$$

$$\cdot \left(-2C_3(1 + \beta f'(\alpha))G_1 + C_2^2(-2 + 6G_1 + 2\beta f'(\alpha)(-1 + 2G_1) - G_1 H_2)\right)e_k^6 + O(e_k^7). \tag{6}$$

where $C_j = \frac{1}{j}\frac{f^{(j)}(\alpha)}{f'(\alpha)}$ for $j = 2, 3, \ldots$, $e_k = x_k - \alpha$, $H_2 = H''(0)$ and $G_1 = G''(0)$.

Proof As we proved for the parametric family of the two first steps of (1), (see [9]), error equation e_z is:

$$e_z = \frac{1}{2}C_2(1 + \beta f'(\alpha))(-2C_3(1 + \beta f'(\alpha)) + C_2^2(6 + 4\beta f'(\alpha) - H_2))e_k^4 + O(e_k^5), \tag{7}$$

being $e_z = z_k - \alpha$. Let us therefore calculate e_{k+1}.

Let us now consider the Taylor expansion of $f(y_k)$ around α:

$$f(y_k) = f'(\alpha)\left(e_y + C_2 e_y^2 + C_3 e_y^3 + C_4 e_y^4 + C_5 e_y^5 + O(e_y^6)\right), \tag{8}$$

being $e_y = y_k - \alpha$.

Let us consider the Taylor expansion of $f(z_k)$ around α, in the same way that is done for $f(y_k)$, being $e_z = z_k - \alpha$.
Now we calculate $f[z_k, y_k]$ and $v_k = \frac{f(z_k)}{f(y_k)}$ using the above equations:

$$f[z_k, y_k] = \frac{f(z_k) - f(y_k)}{z_k - y_k}$$

$$= f'(\alpha)(1 + C_2^2(1 + \beta f'(\alpha))e_k^2$$

$$+ C_2 f'(\alpha)\left((-C_2^2 + C_3)(2 + 2\beta f'(\alpha) + \beta^2 f'(\alpha)^2) + C_3 \beta f'(\alpha)\right)e_k^3 + O(e_k^4).$$

On the other hand,

$$\frac{f(z_k)}{f(y_k)} = -C_3(1 + \beta f'(\alpha)) + C_2^2(3 + 2\beta f'(\alpha) - \frac{H_2}{2}))e_k^2 + O(e_k^3).$$

It follows that

$$G(v_k) = 1 + G_1 v_k + O(v_k^2)$$

$$= 1 + G_1(-C_3(1 + \beta f'(\alpha)) + C_2^2(3 + 2\beta f'(\alpha) - \frac{H_2}{2}))e_k^2 + O(e_k^3).$$

Using the above expressions, we calculate $e_{k+1} = e_z - G(v_k)\frac{f(z_k)}{f[z_k, y_k]}$. It follows that

$$e_{k+1} = \frac{-C_2}{4}\left(1 + \beta f'(\alpha)\right)\left(-2C_3(1 + \beta f'(\alpha)) + C_2^2(6 + 4\beta f'(\alpha) - H_2)\right) \tag{9}$$

$$\cdot \left(-2C_3(1 + \beta f'(\alpha))G_1 + C_2^2(-2 + 6G_1 + 2\beta f'(\alpha)(-1 + 2G_1) - G_1 H_2)\right)e_k^6 + O(e_k^7). \tag{10}$$

Thus it is proved that method (2) has order 6 under these conditions.

In particular, if $H_2 = H''(0) = 2$, then

$$e_{k+1} = C_2(2C_2^2 - C_3)(C_2^2(1 - 2G_1) + C_3G_1)(1 + \beta f'(\alpha))^3 e_k^6 + O(e_k^7). \tag{11}$$

$\square$

3 | INTRODUCING MEMORY

From the error equation, we note that if $\beta = -\frac{1}{f'(\alpha)}$, then the order increase at least one unit. Since the value of α is unknown, we approximate the value of $f'(\alpha)$ in order to increase the order of the iterative scheme. In this way, we obtain a method with memory. We then approximate the parameter to obtain a new method with memory. If we take

$$\beta_k = -\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})},$$

and replace it in (2), we obtain a method with memory, denoted by M_6N_1.

Theorem 3 *Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood D of α such that $f(\alpha) = 0$. We assume that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfy $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$ and let $G(t)$ be a real function satisfying $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking a initial approximation x_0 close enough to α, the sequence of iterates $\{x_k\}$ generated by the method M_6N_1 converges to α with order $p = 3 + 2\sqrt{3} \approx 6.4641$.*

Proof From the error equation (11)

$$e_{k+1} \sim (1 + \beta_k f'(\alpha))^3 e_k^6.$$

Using Taylor's series developments of $f(x_k)$ and $f(x_{k-1})$ and by the Genochi-Hermite formula it follows that

$$1 + \beta_k f'(\alpha) \sim C_2 e_{k-1}.$$

The error equation (11) and the above relations give:

$$e_{k+1} \sim (C_2 e_{k-1})^3 e_k^6 \sim e_{k-1}^3 e_k^6. \tag{12}$$

Suppose that the R-order of the methods is at least p. Therefore,

$$e_{k+1} \sim D_{k,p} e_k^p,$$

where $D_{k,p}$ tends to the asymptotic error constant, D_p, when $k \to \infty$.

Assuming that the R-order of the method is at least p yields the following relation

$$e_{k+1} \sim D_{k,p} e_k^p \sim D_{k,p}(D_{k-1,p} e_{k-1}^p)^p \sim e_{k-1}^{p^2}. \tag{13}$$

In the same way as the relation (12) is obtained, it follows that

$$e_{k+1} \sim e_{k-1}^3 (D_{k-1,p} e_{k-1}^p)^6 = D_{k-1,p}^6 e_{k-1}^{6p+3}.$$ (14)

Then by equating the exponents of e_{k-1} of (13) and (14), it follows that

$$p^2 = 6p + 3,$$

whose only positive solution is the order of convergence of the $M_6 N_1$ method (see Theorem 1), being $p \approx 6.4641$. □

Instead of using the divided differences above, we can use either ones $f[x_k, y_{k-1}]$ or $f[x_k, z_{k-1}]$.

By replacing the parameter for the approximations in (2), we obtain two methods with memory, which we denote by $M_6 N_{1Y}$ and $M_6 N_{1Z}$, respectively.

In the same way that it has been proved that $M_6 N_1$ has order $3+2\sqrt{3}$, we can prove that, under the same conditions as the previous result, $M_6 N_{1Y}$ has order 7 and that $M_6 N_{1Z}$ has order 8, that is, we can prove the following Theorem 4 in a similar way as Theorem 3.

Theorem 4 *Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood of α, such that $f(\alpha) = 0$. Suppose that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfies $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$, and let $G(t)$ be a real function satisfying $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking an initial point x_0, close enough to α, the sequence of iterates generated by $M_6 N_{1Y}$ converge to α with order 7 and the sequence of iterates generated by $M_6 N_{1Z}$ converge to α with order 8.*

Another way to approximate the derivative of a function is to use the Kurchatov divided difference instead of the usual divided difference. For this reason, we approximate the parameter using a Kurchatov's divided difference. We replace the parameter in (2) by

$$\beta_k = -\frac{1}{f[2x_k - x_{k-1}, x_{k-1}]},$$

and we obtain a memory method, denoted by $M_6 K_1$.

Theorem 5 *Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood of α, such that $f(\alpha) = 0$. Suppose that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfies $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$, and let $G(t)$ be a real function satisfying $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking an initial point x_0, close enough to α, sequence of iterates, $\{x_k\}$, generated by $M_6 K_1$ converge to α with order $p = 3 + \sqrt{15} \approx 6.873$.*

Proof From error equation (11)

$$e_{k+1} \sim (1 + \beta_k f'(\alpha))^3 e_k^6.$$

Using Taylor's expansion of $f(x_k)$ and $f(x_{k-1})$ and by the Genochi-Hermite formula, we obtain

$$1 + \beta_k f'(\alpha) \sim e_{k-1}^2.$$

From the error equation (11) and the last relation, it follows that

$$e_{k+1} \sim (e_{k-1}^2)^3 e_k^6 \sim e_{k-1}^6 e_k^6. \tag{15}$$

By assuming that the R-order of the method is at least p, we obtain relation (13). In the same way that the relationship (15) is obtained, we obtain that

$$e_{k+1} \sim (e_{k-1}^2)^3 (D_{k-1,p} e_{k-1}^p)^6 = D_{k-1,p}^6 e_{k-1}^{6p+6}. \tag{16}$$

Then, by equating exponents of e_{k-1} in (13) and in (16), it follows that

$$p^2 = 6p + 6,$$

whose only positive solution is the order of convergence of the $M_6 K_1$ method, where $p = 3 + \sqrt{15} \approx 6.873$. □

We can also apply Kurchatov's divided differences using y_{k-1} and z_{k-1} instead of x_{k-1}. By replacing the parameter by these approximations in (2), we obtain two methods with memory, which we denote by $M_6 K_Y$ and by $M_6 K_Z$, respectively.

In the same way that it has been proved that $M_6 K$ has order $3 + \sqrt{15}$, we can prove that, under the same conditions as the previous theorem, $M_6 K_Y$ has order 8 and that $M_6 K_Z$ has order 9, that is, we can prove Theorem 6 in a similar way to that of the Theorem 5.

Theorem 6 *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood of α, denoted by $D \subset \mathbb{R}$, such that $f(\alpha) = 0$. Suppose that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfies $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$, and let $G(t)$ be a real function that satisfies $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking an initial point x_0, close enough to α, the sequence of iterates by $M_6 K_Y$ converge to α with order 8, and the sequence of iterates by $M_6 K_Z$ converge to α with order 9.*

In the methods with memory discussed above, we have used divided difference operators to approximate the value of $f'(\alpha)$, but there are more ways to approximate this value. One of them is to use interpolating polynomials. In this case we use a Newton interpolating polynomial of degree two at the nodes x_k, x_{k-1} and y_{k-1}, which is defined as follows:

$$N_2(t) = f(x_k) + f[x_k, x_{k-1}](t - x_k) + f[x_k, x_{k-1}, y_{k-1}](t - x_k)(t - x_{k-1}).$$

Then, $N_2'(x_k)$ is an approximation of $f'(\alpha)$, so that, taking

$$\beta_k = -\frac{1}{N_2'(x_k)},$$

and replace this in (2), we obtain a method with memory, denoted by $M_6 N_2$.

Theorem 7 *Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in an neighbourhood of α which we denote by $D \subset \mathbb{R}$ such that $f(\alpha) = 0$. We assume that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfy that $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$ and let $G(t)$ be a real function that satisfy that $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking*

a initial approximation x_0 close enough to α, he sequence of iterates $\{x_k\}$ generated by the method $M_6 N_2$ converges to α with order $p = \frac{1}{2}(7 + \sqrt{61}) \approx 7.4051$.

Proof From the error equation (11)

$$e_{k+1} \sim (1 + \beta_k f'(\alpha))^3 e_k^6.$$

Using Taylor's series developments of $f(x_k)$, $f(x_{k-1})$ and $f(y_{k-1})$, it follows that

$$N_2'(x_k) = f'(\alpha) + 2C_2 f'(\alpha) e_k + C_3 f'(\alpha) e_k e_{y,k-1} + C_3 f'(\alpha)(e_k - e_{y,k-1}) e_{k-1}$$
$$+ O(e_{k-1}^2) + O(e_k^2) + O(e_{y,k-1}^2) + O_3(e_{y,k-1}, e_k, e_{k-1}).$$

This means that $1 + \beta_k f'(\alpha)$ behaves as e_k, as $e_k e_{y,k-1}$, as $e_{k-1} e_k$ or as $e_{k-1} e_{y,k-1}$.

It is obvious that $e_k e_{y,k-1}$ tends faster to zero than e_k when $k \to \infty$, and that $e_{k-1} e_k$ tends to zero faster than $e_{k-1} e_{y,k-1}$. For this reason you have to look at if e_k converges faster to zero or does it $e_{k-1} e_{y,k-1}$.

Suppose the R-order of the method is at least p. Consider the sequence $\{y_k\}$ generated by the first step of the method, and suppose that it converges to R-order at least p_1. Therefore, it is satisfied

$$e_{k+1} \sim D_{k,p} e_k^p \qquad \text{and} \qquad e_{y,k} \sim D_{k,p_1} e_k^{p_1},$$

where $D_{k,p}$ tends to the asymptotic error constant, D_p, and where D_{k,p_1} tends to the asymptotic error constant, D_{p_1}, when $k \to \infty$.

Therefore

$$\frac{e_k}{e_{k-1} e_{y,k-1}} \sim \frac{D_{k-1,p} e_{k-1}^p}{D_{k-1,p_1} e_{k-1} e_{k-1}^{p_1}}.$$

Then, if $p > p_1 + 1$ it follows that

$$1 + \beta_k f'(\alpha) \sim e_{k-1} e_{y,k-1}.$$

The error equation(11) and the above relation gives:

$$e_{k+1} \sim (e_{k-1} e_{y,k-1})^3 e_k^6 \sim e_{k-1}^3 e_{y,k-1}^3 e_k^6. \tag{17}$$

Assuming that the R-order of the method is at least p we obtain the relation (13), and assuming that the sequence $\{y_k\}$ converges with R-order of at least p_1, we obtain the relation

$$e_{y,k+1} \sim D_{k,p_1} e_k^{p_1} \sim D_{k,p_1} (D_{k-1,p} e_{k-1}^p)^{p_1} \sim e_{k-1}^{p p_1}. \tag{18}$$

In the same way that the relation (17) is obtained, we obtain that

$$e_{k+1} \sim e_{k-1}^3 (D_{k-1,p_1} e_{k-1}^{p_1})^3 (D_{k-1,p} e_{k-1}^p)^6 = D_{k-1,p_1}^3 D_{k-1,p}^6 e_{k-1}^{3p_1} e_{k-1}^{6p+3}. \tag{19}$$

Moreover, we know that

$$e_{k,y} \sim (1 + \beta_k f'(\alpha)) e_k^2 \sim e_{k-1} e_{y,k-1} e_k^2 \sim e_{k-1} (D_{k-1,p_1} e_{k-1}^{p_1})(D_{k-1,p} e_{k-1}^{p})^2 \sim e_{k-1}^{2p+1+p_1}. \tag{20}$$

Then equating the exponents of e_{k-1} of (13) and (19), and equating those of (18) and (20), we get that

$$p^2 = 6p + 3 + 3p_1,$$
$$pp_1 = 2p + 1 + p_1,$$

whose only positive solution is the order of convergence of the method $M_6 N_2$ see Theorem 1, being $p \approx 7.4051$ and $p_1 \approx 2.4684$.

□

Other Newton polynomials interpolation can be used of degree 2 or 3:

- $N_{2Z}(t) = f(x_k) + f[x_k, x_{k-1}](t - x_k) + f[x_k, x_{k-1}, z_{k-1}](t - x_k)(t - x_{k-1}).$
- $N_3(t) = N_2(t) + f[x_k, x_{k-1}, y_{k-1}, w_{k-1}](t - x_k)(t - x_{k-1})(t - y_{k-1}).$
- $N_{3Z}(t) = N_{2Z}(t) + f[x_k, x_{k-1}, z_{k-1}, w_{k-1}](t - x_k)(t - x_{k-1})(t - z_{k-1}).$

Then, by choosing β_k as $-\frac{1}{N'_{2Z}(x_k)}$ and substituting this into (2) we get a new method with memory, which we denote by $M_6 N_{2Z}$. In the same way, with polynomials N_3 and N_{3Z}, we obtain methods with memory, denoted by $M_6 N_3$ and $M_6 N_{3Z}$, respectively.

In the same way that the order of convergence of $M_6 N_2$ is demonstrated, we can prove the following result where the orders of convergence obtained for the other methods with memory obtained by using interpolating polynomials are given.

Theorem 8 *Let $f : D \subset \mathbb{R} \longrightarrow \mathbb{R}$ be a sufficiently differentiable function in a neighbourhood of α such that $f(\alpha) = 0$. We assume that $f'(\alpha) \neq 0$. Let $H(t)$ be a real function that satisfy that $H(0) = 1$, $H'(0) = 1$, $H''(0) = 2$ and $|H'''(0)| < \infty$ and let $G(t)$ be a real function satisfying $G(0) = 1$ and $|G'(0)| < \infty$. Then, taking a initial approximation x_0 close enough to α,*

- *the sequence of iterates generated by $M_6 N_{2Z}$ converges to α with order $4 + \sqrt{19} \approx 8.3589$.*
- *the sequence of iterates generated by $M_6 N_3$ converges to α with order 8.*
- *the sequence of iterates generated by $M_6 N_{3Z}$ converge to α with 9.*

4 | NUMERICAL EXPERIMENTS

In this section, we solve some nonlinear equations to compare the parametric family with the methods with memory designed previously, and with new schemes obtained in [6] and in [12], with orders 4 and 6, and denoted by CC and GS, respectively.

For the computational calculations, we use Matlab R2020b with variable precision arithmetic of 1000 digits, and initial estimation x_0 and as stoping criterion the distance between consecutive iterations less than or equal to 10^{-100}. The numerical results we are going to compare the methods in the different examples are:

- the approximation obtained,
- the norm of the equation evaluated in that approximation,
- the norm of the distance between the last two approximations,
- the number of iterations necessary to satisfy the required tolerance,
- the computational time and the approximate computational convergence order (ACOC), defined by Cordero and Torregrosa in [10], which has the following expression

$$p \approx ACOC = \frac{\ln(|x_{k+1} - x_k|/|x_k - x_{k-1}|)}{\ln(|x_k - x_{k-1}|/|x_{k-1} - x_{k-2}|)}.$$

The test functions used are:

- Equation $(x - 1)^3 - 1 = 0$, which has a real root at 2, and we take as initial estimates for all methods $x_1 = 1.5$, $x_0 = 0$, $w_0 = 0.5$, $y_0 = 1.1$ and $z_0 = 1.3$.
- Equation $\arctan(x) = 0$, which has a real root at 0, and we take as initial estimates for all methods $x_1 = 0.75$, $x_0 = 1.5$, $w_0 = 1.25$, $y_0 = 1$ and $z_0 = 0.8$.
- Equation $\cos(x) - x = 0$, which has a real root at 0.73908513, and we take as initial estimates for all methods $x_1 = 2$, $x_0 = 2.4$, $w_0 = 2.3$, $y_0 = 2.2$ and $z_0 = 2.1$.

Let us look at the results obtained for the equation $(x - 1)^3 - 1 = 0$ in Table 1. As can be seen in the ACOC column, the theoretical order of convergence coincides with the ACOC in all the methods except in $M_6 N_3$ in which the order increases by one unit.

For the computational time, the M_6 and $M_6 N_1$ methods stand out at least, although they are not the methods that obtain the best approximation, as can be seen in the second and third columns. The conclusion of this numerical experiment is that the most recommended methods are $M_6 N_3$ and $M_6 N_{3Z}$, as they have the highest ACOC, obtain the best approximation and are among the methods that perform the fewest iterations.

In Figure 1 we draw the dynamical planes obtained for the nonlinear equation $(x - 1)^3 - 1 = 0$ by all the methods we have discussed above.

To generate the dynamical planes, we choose a initial point and what we do is apply our methods taking these point as the initial estimation. To choose the point we have generate a mesh of 400×400 points of the set $[-0.5, 2.5] \times [-1.5, 1.5]$, and then each point of the mesh corresponds to a complex point, $z = \text{Re}(z) + i\text{Im}(z)$, which we will take as a starting point.

We have also defined that the maximum number of iterations that each initial estimate must do is 80, and that we determine that the initial point converges to one of the solutions if the distance to that solution is less than 10^{-3}. We paint in green the initial points that converge to the root $\frac{1 - \sqrt{3}i}{2}$, in orange the initial points that converge to the root 2, in purple the initial points that converge to the root $\frac{1 + \sqrt{3}i}{2}$, and in black the initial points that do not converge to any root in less than 80 iterations.

In these dynamical planes, Figure 1, we can observe a priori what will happen to the initial points. In this case, we can see that although M_6 and GS have the same order, the convergence zones of the M_6 method are simpler and larger than those of the GS method, which has a larger black zone.

On the other hand, in the dynamical planes it can be seen that the methods with memory in which the iteration z_{k-1} intervenes, have a smaller black area than the rest of the methods with memory.

TABLE 1 Results for the equation $(x-1)^3 - 1 = 0$.

Method	$|x^{(k+1)} - x^{(k)}|$	$|f(x^{(k)})|$	Iteration	ACOC	Time
$M_{6,\beta=-0.1}$	$1.28999{\times}10^{-225}$	$1.2288{\times}10^{-1348}$	5	6	0.3016
$M_6 N_1$	$1.07719{\times}10^{-218}$	$4.0404{\times}10^{-1409}$	5	6.47927	0.3063
$M_6 K_1$	$3.78521{\times}10^{-242}$	$1.62794{\times}10^{-1660}$	5	6.85364	0.5469
$M_6 N_{1y}$	$8.79934{\times}10^{-361}$	$1.22538{\times}10^{-2520}$	6	7.0162	0.6266
$M_6 K_2$	$1.38495{\times}10^{-584}$	$1.35351{\times}10^{-4672}$	5	7.98531	0.5766
$M_6 N_{1z}$	$7.76951{\times}10^{-186}$	$1.65981{\times}10^{-1480}$	5	7.98009	0.5500
$M_6 K_3$	$8.21326{\times}10^{-537}$	$2.26765{\times}10^{-4824}$	5	8.99996	0.5609
$M_6 N_2$	$2.44389{\times}10^{-303}$	$2.62354{\times}10^{-2242}$	6	7.36596	0.4422
$M_6 N_{2Z}$	$2.2108{\times}10^{-192}$	$1.54585{\times}10^{-1602}$	5	8.53031	0.3656
$M_6 N_3$	$9.73317{\times}10^{-479}$	$1.04526{\times}10^{-4301}$	5	9.0	0.4281
$M_6 N_{3Z}$	$9.73317{\times}10^{-479}$	$1.04526{\times}10^{-4301}$	5	9.0	0.3984
CC	$1.8372{\times}10^{-266}$	$1.7135{\times}10^{-1062}$	21	3.999	0.4501
GS	$1.5075{\times}10^{-455}$	$1.0952{\times}10^{-2728}$	5	6.0	0.3204

(a) $M_{6,\beta=0.1}$

(b) $M_6 N_1$

(c) $M_6 N_{1Y}$

(d) $M_6 N_{1Z}$

(e) $M_6 K$

(f) $M_6 K_y$

(g) $M_6 K_z$

(h) $M_6 N_2$

(i) $M_6 N_{2Z}$

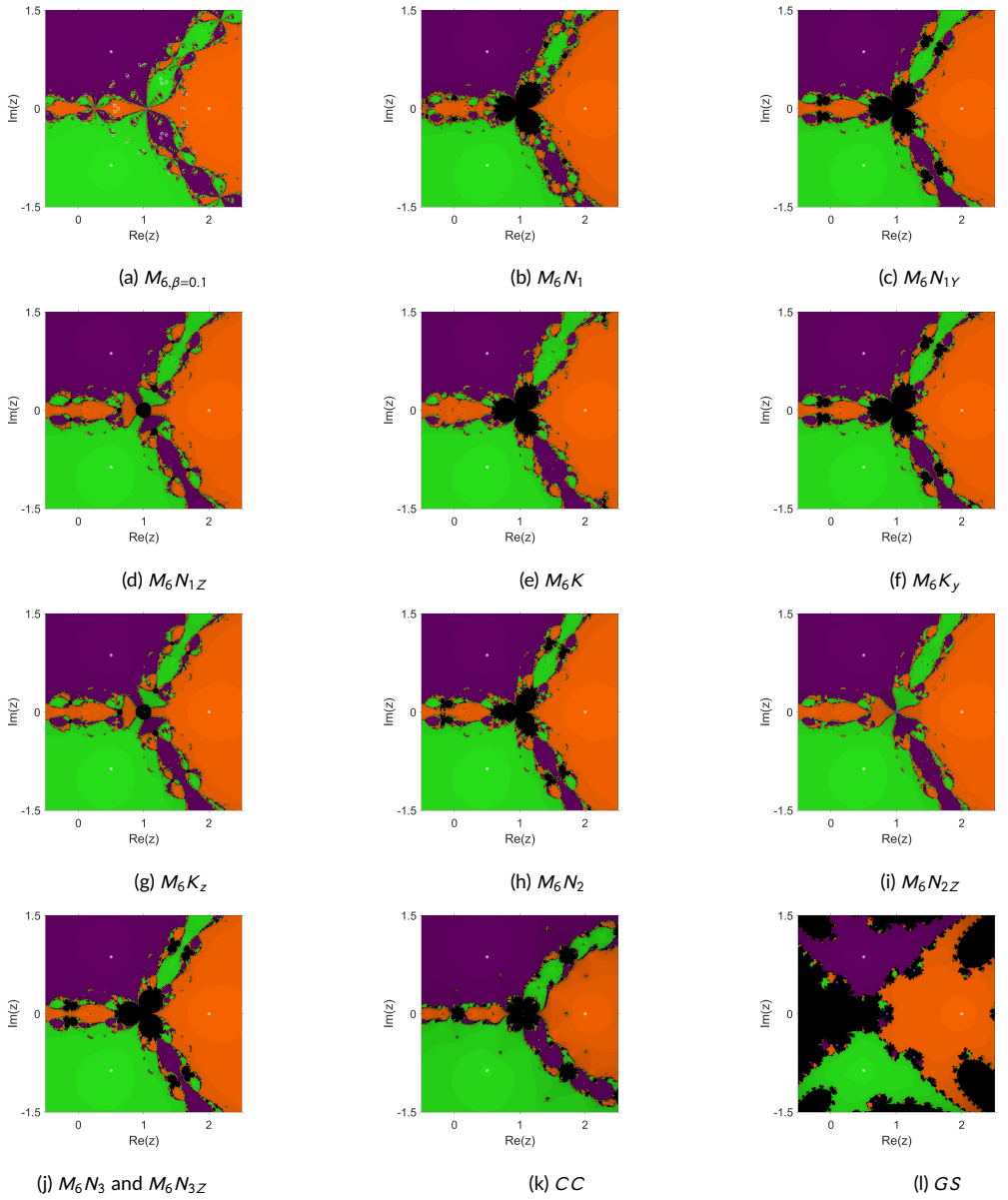(j) $M_6 N_3$ and $M_6 N_{3Z}$

(k) CC

(l) GS

FIGURE 1   Dynamical planes generated by all the methods on equation $(x-1)^3 - 1 = 0$

Let us now see the results obtained for the equation $\arctan(x) = 0$ in the Table 2. In this case all methods converge to the solution although there are differences between them. We can see that the method without memory increases its order by one unit according to the ACOC for this numerical example, and that the methods with memory increase it by more than one unit, but not more than two, obtaining that the method with the highest ACOC is the $M_6 N_{3Z}$ method. If we look at the computational time, we can see that there is practically no difference between the methods. In this case all the methods perform the same number of iterations, and if we look at the columns 2 and 3 we can see that the method that gives a better approximation is the method $M_6 N_{3Z}$. The conclusion of this numerical experiment is that it is advisable in this case to use the $M_6 N_{3Z}$ method.

TABLE 2 Results for the equation $\arctan(x) = 0$.

Method	$|x^{(k+1)} - x^{(k)}|$	$|f(x^{(k)})|$	Iteration	ACOC	Time
M_6	9.63399×10^{-303}	$2.88138 \times 10^{-2122}$	4	7	0.3594
$M_6 N_1$	2.40435×10^{-233}	$6.78309 \times 10^{-1810}$	4	7.81347	0.3203
$M_6 K_1$	7.26769×10^{-255}	$5.77411 \times 10^{-1977}$	4	7.82898	0.3422
$M_6 N_{1y}$	1.10005×10^{-357}	$4.06924 \times 10^{-3367}$	4	9.36642	0.2656
$M_6 K_2$	1.54723×10^{-383}	$1.25996 \times 10^{-3610}$	4	9.39834	0.3141
$M_6 N_{1z}$	3.96266×10^{-481}	$2.54191 \times 10^{-4789}$	4	11.1413	0.2766
$M_6 K_3$	2.60934×10^{-544}	$1.81813 \times 10^{-4890}$	4	11.1632	0.3187
$M_6 N_2$	2.76956×10^{-308}	$2.72641 \times 10^{-2642}$	4	8.5653	0.3531
$M_6 N_{2Z}$	7.34124×10^{-375}	8.011×10^{-2937}	4	9.40829	0.3312
$M_6 N_3$	1.13781×10^{-375}	7.5513×10^{-2939}	4	9.3993	0.4094
$M_6 N_{3Z}$	2.53673×10^{-448}	$7.51537 \times 10^{-4609}$	4	10.2653	0.3438
CC	3.0773×10^{-1394}	7.7912×10^{-694}	5	5.0	0.3107
GS	1.9359×10^{-126}	4.5284×10^{-881}	4	7.0	0.3825

In Figure 2 we draw the dynamical planes obtained for the studied nonlinear equation $\arctan(x) = 0$ by all the methods we have discussed above.

To generate the dynamical planes, we choose a initial point and what we do is apply our methods taking these point as the initial estimation. To choose the point we have generate a mesh of 400×400 points of the set $[-0.5, 2.5] \times [-1.5, 1.5]$, and then each point of the mesh corresponds to a complex point, $z = \text{Re}(z) + i\text{Im}(z)$, which we will take as a starting point.

We have also defined that the maximum number of iterations that each initial estimate must do is 80, and that we determine that the initial point converges to one of the solutions if the distance to that solution is less than 10^{-3}. We paint in orange the initial points that converge to the root 0 and in black the initial points that do not converge to 0 in less than 80 iterations.

In these dynamical planes, Figure 2, we can see a priori what will happen with the initial points. In this case we can see that the GS method has a larger orange zone, so it has more points that converge to 0 in less than 80 iterations, something that did not happen in the previous example in which it was the method that obtained more points that did not converge than the rest. We can see that there is no great difference between the M_6 family and the methods

(a) $M_{6,\beta=0.1}$

(b) $M_6 N_1$

(c) $M_6 N_{1Y}$

(d) $M_6 N_{1Z}$

(e) $M_6 K$

(f) $M_6 K_y$

(g) $M_6 K_z$

(h) $M_6 N_2$

(i) $M_6 N_{2Z}$

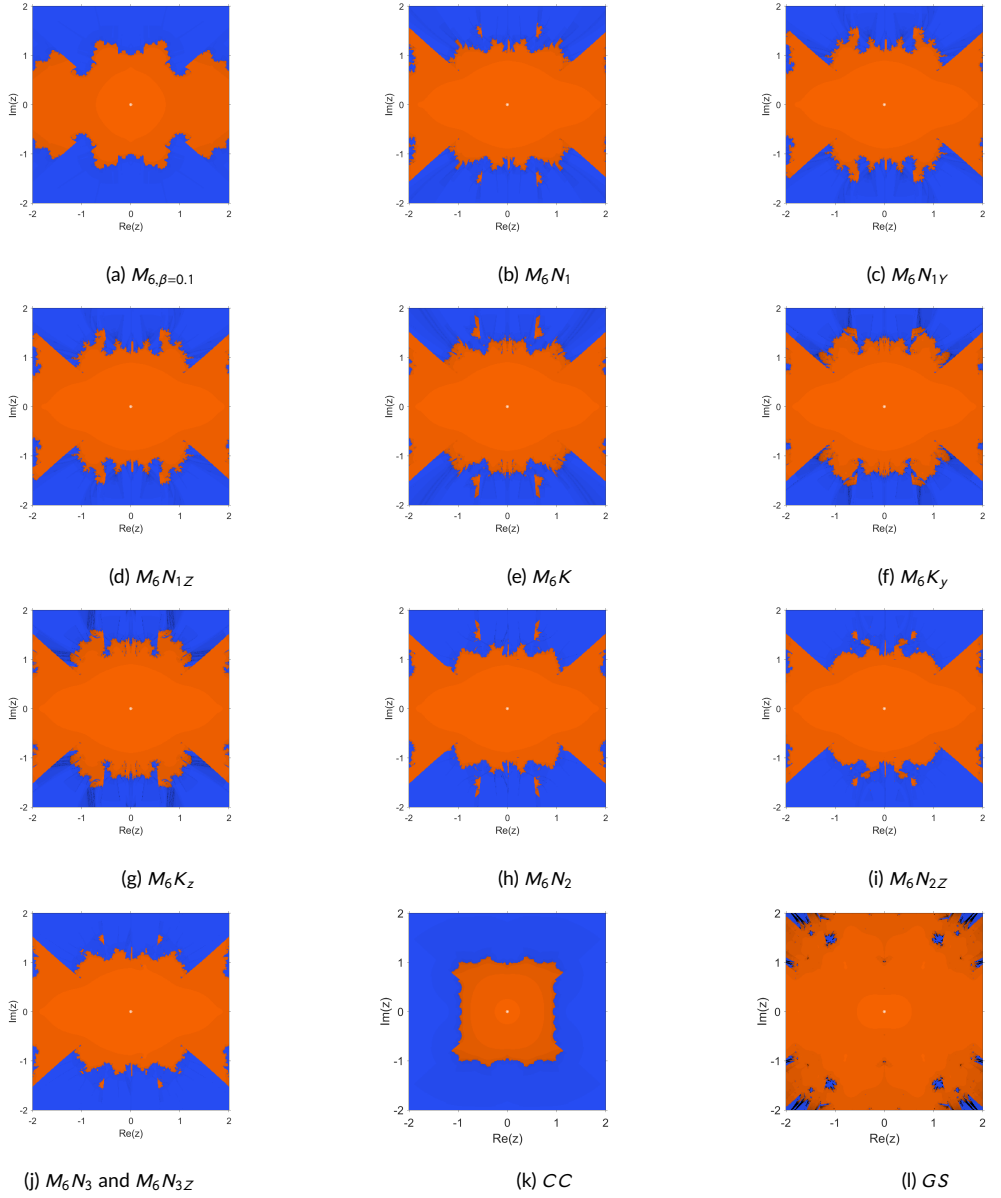(j) $M_6 N_3$ and $M_6 N_{3Z}$

(k) CC

(l) GS

FIGURE 2   Dynamical planes generated by all the methods on equation $\arctan(x) = 0$

with memory obtained from it, we can see that there is no great difference, although it is true that the methods with memory have a slightly larger orange area.

As the last part of the numerical experiments, we compare the efficiency of the discussed iterative methods.

We do this with the efficiency index in one iteration as well as with the total efficiency index, where the number of iterations that our method will perform is also important, given that the more iterations we evaluate, the more evaluations, and therefore the higher the computational cost. We compare the efficiency indexes of the methods with a real example, which is equation $\cos(x) - x = 0$.

Thus, in Table 3 we calculate the efficiency index for this numerical experiment by calculating $p^{\frac{1}{d}}$ where p is the approximate computational order of convergence, or ACOC, and where d is the number of evaluations per iteration.

We also compute the total efficiency index, which is calculated in the same way as the efficiency index but taking into account the number of iterations required to verify the convergence criteria, i.e, $p^{\frac{1}{d \cdot k}}$, where k is the number of iterations needed.

TABLE 3 Tabla órdenes e índices de eficiencia

Method	ACOC	Number evaluations	Efficiency index	Number iterations	Total Efficiency index
$M_{6,\beta=-0.1}$	6	4	1.5651	6	1.0775
$M_6 N_1$	6.4968	5	1.4539	4	1.0981
$M_6 K_1$	7.1016	6	1.3864	4	1.0851
$M_6 N_{1Y}$	7.0064	5	1.4760	4	1.1022
$M_6 K_Y$	7.9888	6	1.4139	4	1.0904
$M_6 N_{1Z}$	8.0229	5	1.5166	4	1.1097
$M_6 K_Z$	9.0	6	1.4422	4	1.0959
$M_6 N_2$	7.4167	6	1.3965	4	1.0871
$M_6 N_{2Z}$	8.3741	6	1.4250	4	1.0926
$M_6 N_3$	7.9697	7	1.3452	4	1.0769
$M_6 N_{3Z}$	8.9911	7	1.3685	4	1.0816
CC	4.0	3	1.5874	5	1.0968
GS	6.0	5	1.4309	5	1.0743

In Figures 3 and 4 we show the efficiency indices appearing in Table 3.

As can be seen in Table 3, all methods with memory perform 4 iterations while the method M_6 performs 6, and the CC and GS methods perform 5 iterations. As can be seen in Figures 3 and 4, the method with the highest efficiency index is the CC method and the M_6 method, followed by the $M_6 N_{1Z}$ method, but when we calculate the total efficiency index things change quite a lot since the method with the highest values is the $M_6 N_{1Z}$ method, and one of the methods with the lowest value is M_6, since it needs to perform two more iterations than the rest of the methods, which means that the total efficiency index is lower.
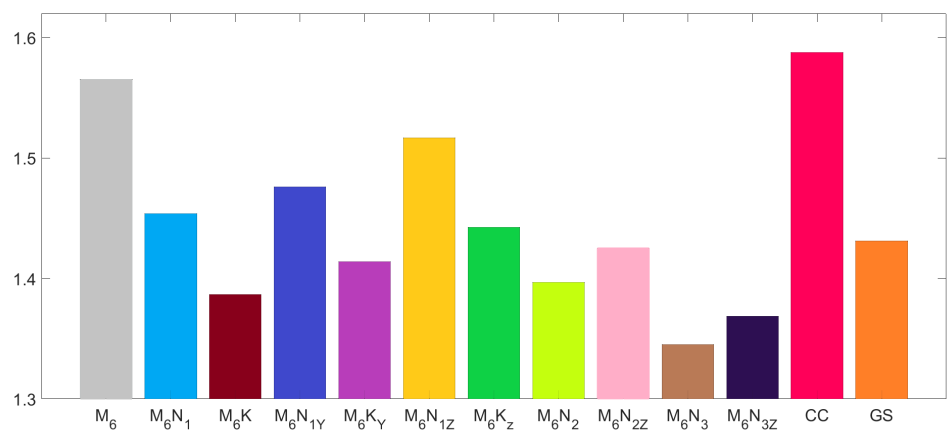
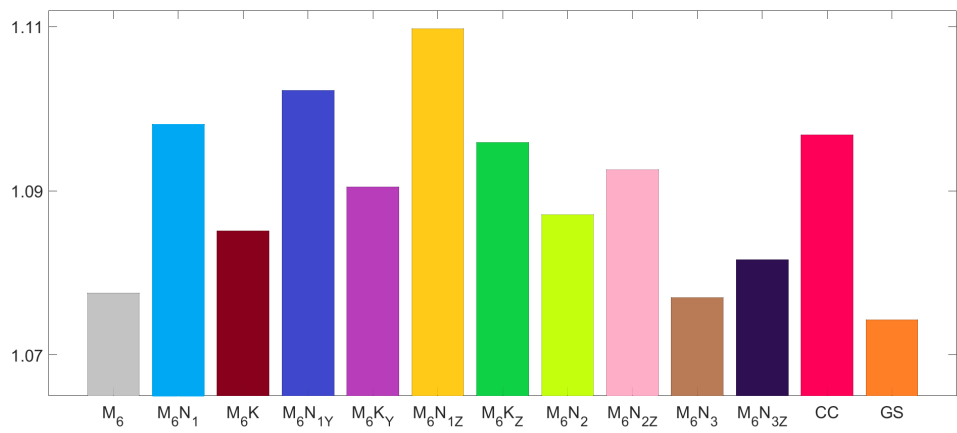**FIGURE 3** Efficiency index comparative

FIGURE 4 Total efficiency index comparative

5 | CONCLUSIONS

In this manuscript, we have designed a parametric family of 3-steps derivative-free iterative methods with convergence order 6.

We have introduced memory to this parametric family, through various divided difference operators and Newton interpolating polynomials, in order to increase the order of convergence without adding new functional evaluations. Thus, we have managed to increase the order by 3 units, in other words, from a method of order 6 we have managed to obtain a method with memory of order 9.

Furthermore, in the numerical experiments, for different problems we compare the approximations obtained by each method, the convergence zones to the roots and the efficiency of the methods according to the number of evaluations they make in each iteration and the number of iterations they make to verify the convergence conditions.

Conflicts of interest: This work does not have any conflicts of interest.

References

[1]   S. Amat and S. Busquier. *Advances in Iterative Methods for NONlinear Equations*. SEMA SIMAI Springer Series 10; Springer, Cham, 2016.

[2]   R. Behl, A.S. Alshomrani, and C. Chun. " A general class of optimal eighth-order derivative free methods for nonlinear equations." In: *J Math Chem.* 58 (2020), pp. 854–867. doi: 10.1007/s10910-020-01115-4.

[3]   F. I. Chicharro, A. Cordero, J. R. Torregrosa, and M. Penkova. "King-Type Derivative-Free Iterative Families: Real and Memory Dynamics". In: *Complexity* 2017 (Oct. 2017), pp. 1–15. doi: 10.1155/2017/2713145.

[4]   F.I. Chicharro, A. Cordero, N. Garrido, and J.R. Torregrosa. "Impact on Stability by the Use of Memory in Traub-Type Schemes." In: *Mathematics* 8.274 (2020). doi: 10.3390/math8020274.

[5]   F.I. Chicharro, A. Cordero, N. Garrido, and J.R. Torregrosa. "Wide stability in a new family of optimal fourth-order iterative methods." In: *Comput. Math. Methods* (2019). doi: 10.1002/cmm4.1023.

[6]   F.I. Chicharro, A. Cordero, J.R. Torregrosa, and M Penkova. "King-Type Derivative-Free Iterative Families: Real and Memory Dynamics." In: *Complexity* (2017), pp. 1–15. doi: 10.1155/2017/2713145.

[7]   C. Chun and B. Neta. "An Efficient Derivative-Free Method for the Solution of Systems of Equations." In: *Numerical Functional Analysis and Optimization* 42.7 (2021), pp. 834–848. doi: 10.1080/01630563.2021.1931313.

[8]   C. Chun and B. Neta. "Comparative study of eighth-order methods for finding simple roots of nonlinear equations." In: *Numer Algor* 74 (2017), pp. 1169–1201. doi: 10.1007/s11075-016-0191-y.

[9]   A. Cordero, N. Garrido, J.R. Torregrosa, and P. Triguero-Navarro. "Memory in the iterative processes for nonlinear problems". In: *Math. Methods Appl. Sci.* submitted ().

[10]   A. Cordero and J. R. Torregrosa. "Variants of Newton's method using fifth-order quadrature formulas." In: *Appl. Math. Comput.* 190 (2007), pp. 686–698. doi: 10.1016/j.amc.2007.01.062.

[11] M. Grau-Sánchez, A. Grau, and M. Noguera. "Frozen divided difference scheme for solving systems of nonlinear equations." In: *J. Comput. Appl. Math.* 235 (2011), pp. 1739–1743.

[12] M. Grau-Sánchez, Á. Grau, and M. Noguera. "Frozen divided difference scheme for solving systems of nonlinear equations." In: *J. Comput. Appl. Math.* 235 (2011), pp. 1739–1743. doi: 10.1155/2017/2713145.

[13] P. Jarratt. "Some efficient fourth-order multipoint methods for solving equations." In: *BIT 9* (1969), pp. 119–124. doi: 10.1007/BF01933248.

[14] R.F. King. "A family of fourth order methods for nonlinear equations." In: *SIAM J. Numer. Anal.* 10 (1973), pp. 876–879. doi: 10.1007/BF01933248.

[15] S. Kumar, D. Kumar, J. Sharma, and I. Argyros. "An efficient class of fourth-order derivative-free method for multiple-roots." In: *International Journal of Nonlinear Sciences and Numerical Simulation* (2021). doi: 10.1515/ijnsns-2020-0161.

[16] H. T Kung and J. F. Traub. "Optimal order of one-point and multi-point iteration." In: *Journal of the Association for Computing Machinery* 21 (1974), pp. 643–651. doi: 10.1145/321850.321860.

[17] B. Neta. "A new family of high order methods for solving equations." In: *Int. J. Comput. Math.* 14 (1983), pp. 191–195.

[18] B. Neta and A. N. Johnson. "High-order nonlinear solver for multiple roots". In: *Computers & Mathematics with Applications* 55.9 (2008), pp. 2012–2017. doi: 10.1016/j.camwa.2007.09.001.

[19] J.M. Ortega and W.C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, 1970.

[20] M. S. Petković, B. Neta, L. D. Petković, and J. Džunić. *Multipoint methods for solving nonlinear equations*. Elsevier, 2013.

[21] M.S. Petković, J. Džunić, and L.D. Petković. "A family of two-point methods with memory for solving nonlinear equations." In: *Appl. Anal. Discrete Math.* 5 (2011), pp. 298–317.

[22] M.S. Petković, B. Neta, L.D. Petković, and J. Džunić. *Multipoint Methods for Solving Nonlinear Equations*. Academic Press, Elsevier., 2013.

[23] R.C. Robinson. *An Introduction to Dynamical Systems: Continuous and Discrete*. American Mathematical Society, Providence, 2012.

[24] D. Sharma, I. K. Argyros, S. K. Parhi, and S. K. Sunanda. "Local Convergence and Dynamical Analysis of a Third and Fourth Order Class of Equation Solvers." In: *Fractal and Fractional.* 5(2):27 (2021). doi: 10.3390/fractalfract5020027.

[25] J.F. Steffensen. "Remarks on iteration." In: *Skand. Aktuarietidskr.* 16 (1933), pp. 64–72.

[26] J.F. Traub. *Iterative Methods for the Solution of Equations*. Prentice-Hall, 1964.