Applicable Analysis and Discrete Mathematics
available online at http://pefmath.etf.rs

Appl. Anal. Discrete Math. **17** (2023), 418–431.
https://doi.org/10.2298/AADM220708003D

NEW FAMILY OF ROOT-FINDING ALGORITHMS BASED ON INVERSE RATIONAL INTERPOLATION

Jovana Džunić

Inverse interpolation with rational functions is investigated for the use in iterative refinement of the root approximation. A new family of optimal methods of arbitrary large order of convergence for solving nonlinear equations is presented. Experiments are conducted to check the influence of polynomial degrees in numerator and denominator on convergence properties of the proposed methods. Wolfram Mathematica 12 software was used to carry the computation due to its capabilities of arbitrary large precision arithmetic.

1. INTRODUCTION

Rational interpolation gave rise to very powerful root-finding algorithms ([**8**], [**4**], [**6**]) based on direct interpolation. Inverse interpolation is a very natural idea in root-finding algorithms. If α is the solution to the problem $f(x) = 0$, and there exists an inverse function $\mathcal{F}(x) = f^{-1}(x)$ in some neighbourhood of α, then $\alpha = \mathcal{F}(0)$. Brent's hybrid bracketing method relies on a quadratic inverse interpolating polynomial to acquire the next bracketing option. Kung and Traub, in their monumental paper [**7**], also used inverse polynomial interpolation and Neville's algorithm [**13**] to construct the optimal $n-$point family of methods. It comes as a natural idea to combine rational interpolation and inverse interpolation for a new $n-$point family of root-finding algorithms. Experience from practical implementation suggests that rational interpolants with close degrees of numerator and denominator show good global interpolating and extrapolating properties. Since interpolatory root-finding

2020 Mathematics Subject Classification. 65H05.
Keywords and Phrases. Nonlinear equations, Multipoint iterative methods, Rational functions, Inverse interpolation.

algorithms really depend on local approximation properties, the idea is to investigate the influence of type (degree) of the rational interpolant to the performance of the induced iterative algorithm.

Since optimal multipoint methods can be pieced together of different interpolatory schemes ([**11, 12**]), we combine different rational interpolant types in the algorithm. This opens an option to explore the matching properties of sub-iterations, through computing. High order multipoint methods (several steps) are therefore used, and high precision arithmetic is employed (3000 binary digits). Comparison was conducted on multiple test functions, some of which were algebraic polynomials. Such a choice can be considered an intro to a comparative study of the explored methods through basins of attraction (as presented in [**2, 9**]), and are to be the subject of a succeeding research.

2. PRELIMINARIES

Approximation to a solution of a scalar nonlinear equation $f(x) = 0$ is explored where the sought zero α is simple. It is assumed that the close enough initial value x_0 is available. Divided differences for a function g are denoted $g[t_0, t_1, \ldots, t_s]$, where we allow some of the points t_i to coincide. This is very helpful when we wish to express interpolation conditions for the Hermitian information set

$$S = \left\{ \left(t_k, \mathcal{F}^{(j)}(t_k)\right) \mid j = 0, 1, \ldots, s_k - 1, \quad k = 0, 1, \ldots, m \right\},$$
$$s_0 + s_1 + \cdots + s_m = n + 1,$$

Interpolation conditions

$$\mathcal{F}^{(j)}(t_k) = \mathcal{G}^{(j)}(t_k), \qquad j = 0, 1, \ldots, s_k - 1, \quad k = 0, 1, \ldots, m, \tag{1}$$

are expressed in a more compact manner as

$$\mathcal{G}[t_0] = \mathcal{F}[t_0],\ \mathcal{G}[t_0, t_1] = \mathcal{F}[t_0, t_1],\ \ldots\ ,\ \mathcal{G}[t_0, t_1, \ldots, t_n] = \mathcal{F}[t_0, t_1, \ldots, t_n],$$

due to properties of divided differences, [**13**].

$$(\mathcal{F} + \mathcal{G})[t_0, t_1, \ldots, t_n] = \mathcal{F}[t_0, t_1, \ldots, t_n] + \mathcal{G}[t_0, t_1, \ldots, t_n],$$

$$(\lambda \cdot \mathcal{F})[t_0, t_1, \ldots, t_n] = \lambda \cdot \mathcal{F}[t_0, t_1, \ldots, t_n],$$

$$\lim_{t_1 \to t_0} \mathcal{F}[t_0, t_1, t_2, \ldots, t_n] = \mathcal{F}[t_0, t_0, t_2, \ldots, t_n], \tag{2}$$

$$(\mathcal{F} \cdot \mathcal{G})[t_0, t_1, \ldots, t_n] = \sum_{j=0}^{n} \mathcal{F}[t_0, t_1, \ldots, t_j] \cdot \mathcal{G}[t_j, t_{j+1}, \ldots, t_n], \tag{3}$$

$$\mathcal{F}[t_0, t_1, \ldots, t_n] = \mathcal{F}[t_{i_0}, t_{i_1}, \ldots, t_{i_n}],$$

where $(i_0, i_1, \ldots, i_n)$ is any permutation of indices $(0, 1, \ldots, n)$.

Theorem 1 (Cauchy Theorem). *Let $\mathcal{F}(t)$ and $\mathcal{G}(t)$ be sufficiently differentiable real functions and let them coincide on the Hermitian data set*

$$S=\Big\{\big(t_k,\mathcal{F}^{(j)}(t_k)\big)\mid j=0,1,\dots,s_k-1,\quad k=0,1,\dots,m\Big\},$$
$$s_0+s_1+\cdots+s_m=n+1,$$

that is, they satisfy (1). *Then,*

$$\begin{aligned}\mathcal{F}(t)-\mathcal{G}(t)&=(\mathcal{F}-\mathcal{G})[t,t_0,t_1,\dots,t_n](t-t_0)(t-t_1)\dots(t-t_n)\\&=\frac{\mathcal{F}^{(n+1)}(\zeta)-\mathcal{G}^{(n+1)}(\zeta)}{(n+1)!}(t-t_0)(t-t_1)\dots(t-t_n),\end{aligned}$$

for some ζ in the interior of $I(t,t_0,\dots,t_n)$, the smallest segment that contains all $t,t_0,\dots,t_n$.

We use a class of functions $\mathcal{S}$ defined as in [**7**]:

$\mathcal{S}=\{f|f$ is a real analytic function defined on an open interval $D_f\subset\mathbb{R}$ which contains a simple zero α_f of f and f' does not vanish on $D_f\}$.

Definition 2. *Let φ be an iteration function and α_f denote a zero of the function f. If there exists an $r=r(\varphi)\in\mathbb{N}$ such that for any $f\in\mathcal{S}$,*

$$\lim_{x\to\alpha_f}\frac{\varphi(f)(x)-\alpha_f}{(x-\alpha_f)^r}=A$$

exists for a constant A which does not vanish at least for one $f\in\mathcal{S}$, then φ is of order of convergence r and an asymptotic error constant $A=AEC(\varphi,f)$.

Kung-Traub's conjecture: Multipoint iterative methods without memory, costing n function evaluations per iteration, have order of convergence at most 2^{n-1}.

Such multipoint methods are called optimal.

3. NEW FAMILY OF METHODS

The interpolatory multipoint iteration function $x_{k+1}=\varphi(x_k),\ k\in\mathbb{N}$ is investigated in the form

$$\text{(4)}\qquad\begin{cases}y_0=x_k,\quad y_1=y_0+\gamma f(y_0),\\[4pt] y_2=y_0-\dfrac{f(y_0)}{f[y_1,y_0]},\\ y_j=\Psi_j(y_0,y_1,...,y_{j-1}),\quad j=3,4,\dots,n,\\[4pt] x_{k+1}=y_n.\end{cases}$$

The value $\gamma = 0$ for the real parameter is allowed. In this manner both derivative free methods and Newton based iterations are explored simultaneously, [**4, 12**]. The approximation y_2 in (4) defines Newton's step for $\gamma = 0$ and Steffensen's iteration [**14**] when $\gamma \neq 0$. This allows that the two sampling procedures are simultaneously considered:

1^o for $\gamma = 0$ information set is

$$\begin{aligned} S &= \big\{ \big(y_0, f(y_0)\big), \big(y_0, f'(y_0)\big), \big(y_2, f(y_2)\big), \dots, \big(y_{n-1}, f(y_{n-1})\big) \big\} \\ &\equiv \big\{ \big(y_0, f(y_0)\big), \big(y_1, f'(y_1)\big), \dots, \big(y_{n-1}, f(y_{n-1})\big) \big\},\ y_0 = y_1; \end{aligned}$$

2^o for $\gamma \neq 0$ information set is

$$S = \big\{ \big(y_0, f(y_0)\big), \big(y_1, f(y_1)\big), \big(y_2, f(y_2)\big), \dots, \big(y_{n-1}, f(y_{n-1})\big) \big\}.$$

Both sampling types are known to provide optimal order of convergence for multipoint methods [**7, 11, 12**]. Approximations $y_j,\ j = 3, 4, \dots, n$ will be the result of inverse rational interpolation based on the available information. Iterative scheme (4) uses n function evaluations per iteration. It is aimed at optimal with the order of convergence $r = 2^{n-1}$.

Data set on function f in some neighborhood of the simple zero α can be transformed into an information set on its inverse function $\mathcal{F} \equiv f^{-1}$. Each pair $\big(y_j, f(y_j)\big)$ is information of the type $\big(\mathcal{F}(f(y_j)), f(y_j)\big)$. Also, by differentiating the key relation

$$y = f\big(f^{-1}(y)\big) = f\big(\mathcal{F}(y)\big),$$

we gain information on $\mathcal{F}'(y)$

$$f'\big(\mathcal{F}(y)\big)\,\mathcal{F}'(y) = 1 \quad \Longrightarrow \quad \mathcal{F}'(y) = \frac{1}{f'\big(\mathcal{F}(y)\big)} = \big(f'\big(\mathcal{F}(y)\big)\big)^{-1}.$$

Corollary 3. *Let $R(t) = R(t; t_0, t_1, \dots, t_n)$ be a rational function that coincides with $\mathcal{F} \equiv f^{-1}$ in the following manner:*

$$(\mathcal{F} - R)\big[f(t_0), \dots, f(t_j)\big] = 0, \quad j = 0, 1, \dots, n.$$

Points t_j are some approximations to the zero α of the function f, not necessarily distinct. Then,

$$\mathcal{F}(0) - R(0) = \alpha - R(0) = \frac{(\mathcal{F} - R)^{(n+1)}(\zeta)}{(n+1)!} \prod_{j=0}^{n} (0 - f(t_j)) = \mathcal{O}\left(\prod_{k=0}^{n} (t_j - \alpha) \right). \tag{5}$$

Proof. The first part of (5) is the Cauchy theorem 1. The second part easily follows from the Taylor expansion of f,

$$f(t_j) = \underbrace{f(\alpha)}_{0} + f'(\alpha)(t_j - \alpha) + \mathcal{O}(t_j - \alpha)^2 = \mathcal{O}(t_j - \alpha).$$

And this settles the proof. □

Corollary 4. *Multipoint iterative scheme* (4) *where steps*

$$y_j = \Psi_j(y_0, y_1, ..., y_{j-1}), \quad j = 3, 4, \ldots, n,$$

are defined by inverse rational interpolation

$$y_j = R_j(0) = R_j(0; y_0, y_1, ..., y_{j-1}),$$

is an optimal iteration.

Proof. Note that $y_0 - \alpha = \mathcal{O}(x_k - \alpha)$, and $y_1 - \alpha = \mathcal{O}(x_k - \alpha)$ and $y_2 - \alpha = \mathcal{O}(x_k - \alpha)^2$. For $n = 3$, according to (5) we have

$$\begin{aligned} y_3 - \alpha &= \mathcal{O}(y_0 - \alpha)\mathcal{O}(y_1 - \alpha)\mathcal{O}(y_2 - \alpha) \\ &= \mathcal{O}(x_k - \alpha)\mathcal{O}(x_k - \alpha)\mathcal{O}(x_k - \alpha)^2 = \mathcal{O}(x_k - \alpha)^4 = 2^{3-1}. \end{aligned}$$

Thus, from here we proceed by the induction.

$$\begin{aligned} y_n - \alpha &= \mathcal{O}(y_0 - \alpha)\mathcal{O}(y_1 - \alpha)\mathcal{O}(y_2 - \alpha)\mathcal{O}(y_3 - \alpha) \ldots \mathcal{O}(y_{n-1} - \alpha) \\ &= \mathcal{O}(x_k - \alpha)\mathcal{O}(x_k - \alpha)\mathcal{O}(x_k - \alpha)^2\mathcal{O}(x_k - \alpha)^4 \ldots \mathcal{O}(x_k - \alpha)^{2^{n-2}} \\ &= \mathcal{O}(x_k - \alpha)^{1+1+2+2^2+\cdots+2^{n-2}} = \mathcal{O}(x_k - \alpha)^{2^{n-1}}. \end{aligned}$$

This confirms the optimality of the proposed family of methods. □

Corollary 5. *Let multipoint iterative scheme* (4) *be optimal with arbitrarily defined optimal steps*

$$y_j = \Psi_j(y_0, y_1, ..., y_{j-1}), \quad y_j - \alpha = \mathcal{O}(x_k - \alpha)^{2^{j-1}} \quad j = 3, 4, \ldots, n,$$

Then, an iterative scheme

$$\begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_j = \Psi_j(y_0, y_1, ..., y_{j-1}), \quad j = 3, 4, \ldots, n, \\ x_{k+1} = y_{n+1} = R_{n+1}(0; y_0, y_1, ..., y_n), \end{cases}$$

is an optimal iteration of higher order of convergence than (4).

Remark 6. *Rational interpolation problem is not always solvable. Previous statements assume the existence of the solution to the rational interpolating problem. The procedure for calculating R_j presented in the following section will always produce a result. However, cancellation of terms in the final rational expression will lead to the drop in convergence order in the induced iteration with accordance to the lost information.*

4. OBTAINING FORMULAS FOR THE NEW FAMILY OF METHODS

We devise a procedure for determining a rational interpolant of the form

$$R_{n+1}(t) \equiv R_{a,b}(t) = \frac{P_a(t)}{Q_b(t)}\,, \quad a,b \in \mathbb{N}_0, \quad \deg(P_a) = a,\ \deg(Q_b) = b,$$
$$a + b + 1 = n + 1,$$

based on information at points $y_0, y_1, \dots, y_n$,

$$R_{a,b}[f(y_0)] = \mathcal{F}[f(y_0)], \quad R_{a,b}[f(y_0), f(y_1)] = \mathcal{F}[f(y_0), f(y_1)],\ \dots,$$
$$R_{a,b}[f(y_0), f(y_1), \dots, f(y_n)] = \mathcal{F}[f(y_0), f(y_1), \dots, f(y_n)].$$

Note that for $b = 0$ we are handling a polynomial interpolant.

Let us assume first that all y_j are distinct. For brevity we will use $f(y_j) = f_j$. Then

$$\frac{P_a(f_j)}{Q_b(f_j)} = \mathcal{F}(f_j) \quad \Longrightarrow \quad P_a(f_j) = \mathcal{F}(f_j)\, Q_b(f_j),\ j = 0, 1, \dots, n$$

(6) $$\Longleftrightarrow \quad P_a(f_j) - \mathcal{F}(f_j)\, Q_b(f_j) = 0,\ j = 0, 1, \dots, n.$$

We introduce Newton's form for the polynomials P_a and Q_b.

$$\begin{aligned} P_a(t) &= P_a[f_0] + P_a[f_0, f_1](t - f_0) + \dots \\ &\quad + P_a[f_0, \dots, f_a](t - f_0)\dots(t - f_{a-1}), \\ &= p_0 + p_1(t - f_0) + \dots + p_a(t - f_0)\dots(t - f_{a-1}), \\ \text{where } p_0 &= P_a[f_0],\ p_1 = P_a[f_0, f_1], \dots, p_a = P_a[f_0, \dots, f_a], \end{aligned}$$

is used for short notation. Also,

$$\begin{aligned} Q_b(t) &= 1 + Q_b[f_0, f_1](t - f_0) + \dots + Q_b[f_0, \dots, f_b](t - f_0)\dots(t - f_{b-1}), \\ &= 1 + q_1(t - f_0) + \dots + q_b(t - f_0)\dots(t - f_{b-1}), \\ q_1 &= Q_b[f_0, f_1], \dots, q_b = Q_b[f_0, \dots, f_b], \\ &\Longrightarrow\ Q_b[f_0] = 1 \text{ is a chosen fixed value.} \end{aligned}$$

Conditions (6) are thus rewritten as

(7) $$(P_a - Q_b\, \mathcal{F})[f_0, \dots, f_j] = 0, \quad j = 0, 1, \dots, n.$$

Based on Leibniz formula (3), having in mind $Q_b[f_0] = 1$, the system (7) becomes

(8) $$\begin{cases} p_0 = \mathcal{F}[f_0] = \mathcal{F}(f(y_0)) = y_0, \\ p_1 - q_1\mathcal{F}[f_1] = \mathcal{F}[f_0, f_1], \\ p_2 - q_1\mathcal{F}[f_1, f_2] - q_2\mathcal{F}[f_2] = \mathcal{F}[f_0, f_1, f_2], \\ \qquad\vdots \\ p_n - q_1\mathcal{F}[f_1, \dots, f_n] - q_2\mathcal{F}[f_2, \dots, f_n] - \dots - q_n\mathcal{F}[f_n] = \mathcal{F}[f_0, \dots, f_n]. \end{cases}$$

Obviously, for $P_a(t)$ and $Q_b(t)$ it is valid

$$p_j = P_a[f_0,\dots,f_j] = \frac{P_a^{(j)}(\zeta)}{j!} = 0, \forall j > a,$$
$$q_j = Q_b[f_0,\dots,f_j] = \frac{Q_b^{(j)}(\zeta)}{j!} = 0, \forall j > b.$$

These features provide us with the block triangular system of equations (8), making it easier to solve. If, for example $a \geq b$, matrix form of (8) reads

$$Av = u, \quad v = \begin{bmatrix} p_0 & p_1 & \dots & p_a & q_1 & \dots & q_b \end{bmatrix}^T,$$
$$u = \begin{bmatrix} \mathcal{F}[f_0] & \mathcal{F}[f_0,f_1] & \dots & \mathcal{F}[f_0,f_1,\dots,f_n] \end{bmatrix}^T,$$
$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & -\mathcal{F}[f_1] & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & -\mathcal{F}[f_1,f_2] & -\mathcal{F}[f_2] & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -\mathcal{F}[f_1,\dots,f_a] & -\mathcal{F}[f_2,\dots,f_a] & \dots & -\mathcal{F}[f_b,\dots,f_a] \\ 0 & 0 & 0 & \dots & 0 & -\mathcal{F}[f_1,\dots,f_{a+1}] & -\mathcal{F}[f_2,\dots,f_{a+1}] & \dots & -\mathcal{F}[f_b,\dots,f_{a+1}] \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & -\mathcal{F}[f_1,\dots,f_n] & -\mathcal{F}[f_2,\dots,f_n] & \dots & -\mathcal{F}[f_b,\dots,f_n] \end{bmatrix}.$$

In the case when $a < b$ matrix of the system $Av = u$ takes the form

$$A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & -\mathcal{F}[f_1] & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & -\mathcal{F}[f_1,f_2] & -\mathcal{F}[f_2] & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 & -\mathcal{F}[f_1,\dots,f_a] & -\mathcal{F}[f_2,\dots,f_a] & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & -\mathcal{F}[f_1,\dots,f_n] & -\mathcal{F}[f_2,\dots,f_n] & \dots & -\mathcal{F}[f_b,\dots,f_n] \end{bmatrix}.$$

Once the system is solved we can seek the next approximation $y_{n+1} = R_{a,b}(0)$ using Horner-like form for polynomials P_a and Q_b.

$$P_a(t) = p_0 + (t - f_0)\Bigg(p_1 + (t - f_1)\Bigg(p_2 + \dots$$
$$+ (t - f_{a-2})\big(p_{a-1} + p_a(t - f_{a-1})\big)\dots\Bigg)\Bigg).$$

When y_0 and y_1 coincide, by the argument of continuity (2) the above results still hold. Thus, both derivative free and Newton based iterations are constructed using (8).

Iteration scheme (4) with inverse rational interpolation does not specify the type of rational interpolant. For this reason more explicit results on the general iterative formula can not be obtained. It is expected that particular choice for a and b from one step to another can produce triangular algorithms that lead to easier implementation.

Remark 7. *When working in double precision arithmetic it is a more stable choice to use Newton's polynomial form starting from the latest approximation. However, when working in computing environment of very high precision arithmetic such approach is not very relevant. The above strategy is chosen with simplicity of notation in mind.*

5. NUMERICAL EXPERIMENTS

Based on error relation (5) we can conclude that the geometry of the rational interpolant $R_j(t; f_0, f_1, \dots, f_j)$ in each sub step has strong influence on the error relation, that is on the asymptotic error constant of the method. For this reason we here explore through examples and computing the possibility of the rational type preference for an iteration function.

We list particular members of the proposed family, both derivative free and Newton based iterations.

$$M1: \begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_3 = R_{1,1}(0; y_0, y_1, y_2), \\ x_{k+1} = y_3, \end{cases} \qquad r = 4; \quad R_{1,1}(t) = \frac{p_0 + p_1(t - f_0)}{1 + q_1(t - f_0)}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\mathcal{F}[f_1] \\ 0 & 0 & -\mathcal{F}[f_1, f_2] \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ q_1 \end{bmatrix} = \begin{bmatrix} \mathcal{F}[f_0] \\ \mathcal{F}[f_0, f_1] \\ \mathcal{F}[f_0, f_1, f_2] \end{bmatrix}.$$

$$M2: \begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_3 = R_{2,0}(0; y_0, y_1, y_2), \\ x_{k+1} = y_3, \end{cases} \qquad r = 4;$$

$$R_{2,0}(t) = p_0 + p_1(t - f_0) + p_2(t - f_0)(t - f_1)$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} \mathcal{F}[f_0] \\ \mathcal{F}[f_0, f_1] \\ \mathcal{F}[f_0, f_1, f_2] \end{bmatrix}.$$

Next, we present iterative methods of order 8.

$$M3:\ \begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_3 = R_{1,1}(0; y_0, y_1, y_2), \\ y_4 = R_{1,2}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases} = \begin{cases} y_3 = M_1(x_k), \\ y_4 = R_{1,2}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases}$$

$$R_{1,2}(t) = \frac{p_0 + p_1(t - f_0)}{1 + q_1(t - f_0) + q_2(t - f_0)(t - f_1)},$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\mathcal{F}[f_1] & 0 \\ 0 & 0 & -\mathcal{F}[f_1, f_2] & -\mathcal{F}[f_2] \\ 0 & 0 & -\mathcal{F}[f_1, f_2, f_3] & -\mathcal{F}[f_2, f_3] \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \mathcal{F}[f_0] \\ \mathcal{F}[f_0, f_1] \\ \mathcal{F}[f_0, f_1, f_2] \\ \mathcal{F}[f_0, f_1, f_2, f_3] \end{bmatrix}.$$

$$M4:\ \begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_3 = R_{1,1}(0; y_0, y_1, y_2), \\ y_4 = R_{2,1}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases} = \begin{cases} y_3 = M_1(x_k), \\ y_4 = R_{2,1}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases};$$

$$R_{2,1}(t) = \frac{p_0 + p_1(t - f_0) + p_2(t - f_0)(t - f_1)}{1 + q_1(t - f_0)},$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\mathcal{F}[f_1] \\ 0 & 0 & 1 & -\mathcal{F}[f_1, f_2] \\ 0 & 0 & 0 & -\mathcal{F}[f_1, f_2, f_3] \end{bmatrix} \begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ q_1 \end{bmatrix} = \begin{bmatrix} \mathcal{F}[f_0] \\ \mathcal{F}[f_0, f_1] \\ \mathcal{F}[f_0, f_1, f_2] \\ \mathcal{F}[f_0, f_1, f_2, f_3] \end{bmatrix}.$$

$$M5:\ \begin{cases} y_0 = x_k, \quad y_1 = y_0 + \gamma f(y_0), \\ y_2 = y_0 - \dfrac{f(y_0)}{f[y_1, y_0]}, \\ y_3 = R_{1,1}(0; y_0, y_1, y_2), \\ y_4 = R_{3,0}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases} = \begin{cases} y_3 = M_1(x_k), \\ y_4 = R_{3,0}(0; y_0, y_1, y_2, y_3), \\ x_{k+1} = y_4, \end{cases};$$

$$R_{3,0}(t) = p_0 + p_1(t - f_0) + p_2(t - f_0)(t - f_1) + p_3(t - f_0)(t - f_1)(t - f_2)$$

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\begin{bmatrix}p_0\\p_1\\p_2\\p_3\end{bmatrix}=\begin{bmatrix}\mathcal{F}[f_0]\\\mathcal{F}[f_0,f_1]\\\mathcal{F}[f_0,f_1,f_2]\\\mathcal{F}[f_0,f_1,f_2,f_3]\end{bmatrix}.$$

$$M6:\begin{cases}y_0=x_k,\quad y_1=y_0+\gamma f(y_0),\\ y_2=y_0-\dfrac{f(y_0)}{f[y_1,y_0]},\\ y_3=R_{2,0}(0;y_0,y_1,y_2),\\ y_4=R_{1,2}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases}=\begin{cases}y_3=M_2(x_k),\\ y_4=R_{1,2}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases};$$

$$M7:\begin{cases}y_0=x_k,\quad y_1=y_0+\gamma f(y_0),\\ y_2=y_0-\dfrac{f(y_0)}{f[y_1,y_0]},\\ y_3=R_{2,0}(0;y_0,y_1,y_2),\\ y_4=R_{2,1}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases}=\begin{cases}y_3=M_2(x_k),\\ y_4=R_{2,1}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases};$$

$$M8:\begin{cases}y_0=x_k,\quad y_1=y_0+\gamma f(y_0),\\ y_2=y_0-\dfrac{f(y_0)}{f[y_1,y_0]},\\ y_3=R_{2,0}(0;y_0,y_1,y_2),\\ y_4=R_{3,0}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases}=\begin{cases}y_3=M_2(x_k),\\ y_4=R_{3,0}(0;y_0,y_1,y_2,y_3),\\ x_{k+1}=y_4,\end{cases}.$$

Methods of order $r=16$, there are 16 in total

$$M9{-}M33:\begin{cases}y_4=M_3(x_k)-M_8(x_k),\\ y_5=R_{a,b}(0;y_0,y_1,y_2,y_3,y_4),\quad (a,b)\in\{(1,3),(2,2),(3,1),(4,0)\}\\ x_{k+1}=y_4,\end{cases}$$

In the process we use the following:

$$R_{1,3}(t)=\frac{p_0+p_1(t-f_0)}{1+q_1(t-f_0)+q_2(t-f_0)(t-f_1)+q_3(t-f_0)(t-f_1)(t-f_2)},$$

$$\begin{bmatrix}1&0&0&0&0\\0&1&-\mathcal{F}[f_1]&0&0\\0&0&-\mathcal{F}[f_1,f_2]&-\mathcal{F}[f_2]&0\\0&0&-\mathcal{F}[f_1,f_2,f_3]&-\mathcal{F}[f_2,f_3]&-\mathcal{F}[f_3]\\0&0&-\mathcal{F}[f_1,f_2,f_3,f_4]&-\mathcal{F}[f_2,f_3,f_4]&-\mathcal{F}[f_3,f_4]\end{bmatrix}\begin{bmatrix}p_0\\p_1\\q_1\\q_2\\q_3\end{bmatrix}=\begin{bmatrix}\mathcal{F}[f_0]\\\mathcal{F}[f_0,f_1]\\\mathcal{F}[f_0,f_1,f_2]\\\mathcal{F}[f_0,f_1,f_2,f_3]\\\mathcal{F}[f_0,f_1,f_2,f_3,f_4]\end{bmatrix}.$$

$$R_{2,2}(t)=\frac{p_0+p_1(t-f_0)+p_2(t-f_0)(t-f_1)}{1+q_1(t-f_0)+q_2(t-f_0)(t-f_1)},$$

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & -\mathcal{F}[f_1] & 0\\ 0 & 0 & 1 & -\mathcal{F}[f_1,f_2] & -\mathcal{F}[f_2]\\ 0 & 0 & 0 & -\mathcal{F}[f_1,f_2,f_3] & -\mathcal{F}[f_2,f_3]\\ 0 & 0 & 0 & -\mathcal{F}[f_1,f_2,f_3,f_4] & -\mathcal{F}[f_2,f_3,f_4]\end{bmatrix}\begin{bmatrix}p_0\\ p_1\\ p_2\\ q_1\\ q_2\end{bmatrix}=\begin{bmatrix}\mathcal{F}[f_0]\\ \mathcal{F}[f_0,f_1]\\ \mathcal{F}[f_0,f_1,f_2]\\ \mathcal{F}[f_0,f_1,f_2,f_3]\\ \mathcal{F}[f_0,f_1,f_2,f_3,f_4]\end{bmatrix}.$$

$$R_{3,1}(t)=\frac{p_0+p_1(t-f_0)+p_2(t-f_0)(t-f_1)+p_3(t-f_0)(t-f_1)(t-f_2)}{1+q_1(t-f_0)},$$

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & -\mathcal{F}[f_1]\\ 0 & 0 & 1 & 0 & -\mathcal{F}[f_1,f_2]\\ 0 & 0 & 0 & 1 & -\mathcal{F}[f_1,f_2,f_3]\\ 0 & 0 & 0 & 0 & -\mathcal{F}[f_1,f_2,f_3,f_4]\end{bmatrix}\begin{bmatrix}p_0\\ p_1\\ p_2\\ p_3\\ q_1\end{bmatrix}=\begin{bmatrix}\mathcal{F}[f_0]\\ \mathcal{F}[f_0,f_1]\\ \mathcal{F}[f_0,f_1,f_2]\\ \mathcal{F}[f_0,f_1,f_2,f_3]\\ \mathcal{F}[f_0,f_1,f_2,f_3,f_4]\end{bmatrix}.$$

$$R_{4,0}(t)=p_0+p_1(t-f_0)+p_2(t-f_0)(t-f_1)+p_3(t-f_0)(t-f_1)(t-f_2)$$
$$+p_4(t-f_0)(t-f_1)(t-f_2)(t-f_3),$$

$$\begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}p_0\\ p_1\\ p_2\\ p_3\\ p_4\end{bmatrix}=\begin{bmatrix}\mathcal{F}[f_0]\\ \mathcal{F}[f_0,f_1]\\ \mathcal{F}[f_0,f_1,f_2]\\ \mathcal{F}[f_0,f_1,f_2,f_3]\\ \mathcal{F}[f_0,f_1,f_2,f_3,f_4]\end{bmatrix}.$$

Tested methods do belong to some two and three point families of methods proposed in [**3, 10**] and [**5**], for example. $n-$point family based on inverse polynomial interpolation was first proposed in [**7**]. Methods $M1-M33$ were tested in order to look for particular signs of compatibility between types of rational interpolants. Similar tests can be conducted on higher order iterative schemes with the same question in mind.

Very fast convergence of the tested methods is the reason that $\tau=10^{-200}$ was used as the tolerance error bound throughout iterations. Maximal number of iterations was set to 20. Computational order of convergence [**11**]

$$r_C\approx\frac{\log|f(x_{k+1})/f(x_k)|}{\log|f(x_k)/f(x_{k-1})|}.$$

was calculated in each example to verify conclusions derived in theory. Convergence for all test functions was obtained, and r_C confirmed theoretical order of convergence.

Test functions used in experiments are tabled in 1. Results of the experiments are stored in the supplementary Excel file.

Table 1: Table of test functions

test function	zero	initial approximation
$f_1(t) = (t-2)(t^4+t+1)e^{-t^2-4t}$	$\alpha = 2$	$x_0 = 1.3$
$f_2(t) = e^{-t^4+t+2} - \cos(t+1) + t^7 + 1$	$\alpha = -1$	$x_0 = 0.3$
$f_3(t) = (t-1)(t^8+t^3+1)\sin(t)$	$\alpha = 1$	$x_0 = 1.7$
$f_4(t) = e^{t^2-1}\sin(t/3) + \dfrac{t\sqrt{t^4+1}}{t^2+4}$	$\alpha = 0$	$x_0 = -1.8$
$f_5(t) = t^2 - (1-t)^{25}$	$\alpha \approx 0.144$	$x_0 = 0.52$
$f_6(t) = t^2\sin(t)^2 + e^{t\cos(t)\sin(t))} - 18;$	$\alpha \approx 9.690$	$x_0 = 10$
$f_7(t) = e^{t^2-4} + \sin(t-2) - t^4 + 15$	$\alpha = 2$	$x_0 = 0.83$
$f_8(t) = \prod_{k=1}^{12}(t-k)$	$\alpha = 5$	$x_0 = 5.41$
$f_9(t) = \log(4-t^2)\sin(t) + \cos(2t) - 1$	$\alpha = 0$	$x_0 = -0.9$
$f_{10}(t) = \log(t^2+1) + e^t\sin(t)$	$\alpha = 0$	$x_0 = 0.3$
$f_{11}(t) = t^4 + \sin(\pi/t^2) - 5$	$\alpha \approx 1.414$	$x_0 = 1$
$f_{12}(t) = \dfrac{1}{t^4} - t^2 - \dfrac{1}{t} + 1$	$\alpha = 1$	$x_0 = 1.8$
$f_{13}(t) = (t+2)\log(t^{10}+t+1);$	$\alpha = -2$	$x_0 = -3$
$f_{14}(t) = e^{t^3+t\cos t-1}t + \log(t\sin t+1)$	$\alpha = 0$	$x_0 = 0.46$
$f_{15}(t) = t^5 + t^4 + \dfrac{1}{t^2+1} - \dfrac{5t^2}{2}$	$\alpha = 1$	$x_0 = 1.5$

Experiments on these test functions show several features of multipoint methods that are statistically speaking true:

- Increase of convergence order of the method does lower the number of function evaluations used.

- Some test functions work better with derivative-free methods, and some are more suitable for Newton-type methods.
- The change in degree of the polynomial in the numerator of the rational interpolant does not affect the number of iterations of the method, since this is dictated by the convergence order.
- The increase in degree of the polynomial has an effect on the approximation accuracy in the manner depicted in figure 1.

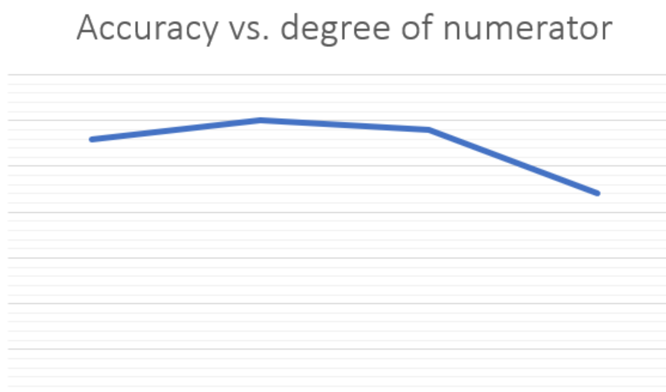

Figure 1: Influence of the rational interpolant type

6. CONCLUSIONS AND FUTURE EXPLORATION

Test functions of different kinds show slight preference to rational interpolants of degree $R_{b+1,b}$ or similar close degrees combined. Perhaps this could be explained with the linearization of functions when examined on very small domains.

Inverse rational interpolation of small degree can be explored for use in hybrid methods such as Brent's [**1**]. Such an approach may enhance the rate of convergence using less steps of the Bisection method. This new family of methods provides space to explore possible recursive triangular algorithms in implementation, with the operational complexity and stability of computation in mind.

Acknowledgements. This research was supported by the Serbian Ministry of Education, Science and Technological Development under grant 451-03-68/2022-14/ 200102.

REFERENCES

1. R. P. Brent: *Algorithms for Minimization without Derivatives*, Prentice-Hall, Inc, 1973.
2. C. Chun, B. Neta: *Comparative study of eighth-order methods for finding simple roots of nonlinear equations*, Numer. Algor. **74** (2017), 1169–1201.
3. C. Chun, Y. Ham: *A one-parameter fourth-order family of iterative methods for nonlinear equations*, Appl. Math. Comput., **189** (2007), 610–614.
4. J. Džunić, I. Damnjanović: *General approach to constructing optimal multipoint families of iterative methods using Hermite's rational interpolation*, J. of Comput. and Appl. Math., **321** (2017), 261–269.
5. J. Džunić, M. S. Petković, L. D. Petković: *A family of optimal three-point methods for solving nonlinear equations using two parametric functions*, Appl. Math. Comput., **217** (2011), 7612–7619.
6. P. Jarratt, D. Nudds: *The use of rational functions in the iterative solution of equations on a digital computer*, Comput. J., **8** (1965), 62–65.
7. H. T. Kung, J. F. Traub: *Optimal Order of One-Point and Multipoint Iteration*, Journal of the ACM, **21** (1974), 643–651.
8. A. M. Ostrowski: *Solution of Equations and Systems of Equations*, Academic Press, New York, 1960.
9. I. Petković, Dj. Herceg: *Computers in mathematical research: the study of three-point root-finding methods*, Numer. Algor. **84** (2020), 1179–1198.
10. M. S. Petković, S. Ilić, J. Džunić: *Derivative free two-point methods with and without memory for solving nonlinear equations*, Appl. Math. Comput., **217** (2010), 1887–1895.
11. M. S. Petković, B. Neta, L. D. Petković, J. Džunić: *Multipoint Methods for Solving Nonlinear Equations*, Elsevier/Academic Press, 2013.
12. M. S. Petković, B. Neta, L. D. Petković, J. Džunić: *Multipoint methods for solving nonlinear equations: a survey*, Appl. Math. Comput. **226** (2014), 635–660.
13. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery: *NUMERICAL RECIPES: The Art of Scientific Computing*, Cambridge University Press, 2007.
14. I. F. Steffensen: *Remarks on iteration*, Skand. Aktuarietidskr., **16** (1933), 64–72.

Jovana Džunić
Department of Mathematics,
Faculty of Electronic Engineering
University of Niš,
Niš, Serbia
E-mail: *jovana.dzunic@elfak.ni.ac.rs*

(Received 08. 07. 2022.)
(Revised 04. 02. 2023.)