Journal of Computational and Applied Mathematics 340 (2018) 11–42

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Comparative study of methods of various orders for finding repeated roots of nonlinear equations

Changbum Chun [a], Beny Neta [b,*]

[a] Department of Mathematics, Sungkyunkwan University, Suwon 16419, Republic of Korea
[b] Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA 93943, United States

ARTICLE INFO

Article history:
Received 11 October 2017

MSC:
65H05
65B99

Keywords:
Iterative methods
Order of convergence
Rational maps
Basin of attraction
Conjugacy classes

ABSTRACT

In this paper we are considering 20 (families of) methods for finding repeated roots of a nonlinear equation. The methods are of order up to 8. We use the idea of basin of attraction to compare the methods. We found that 4 methods performed best based on 3 quantitative criteria.

Published by Elsevier B.V.

1. Introduction

There are many iterative methods for the solution of a single nonlinear equation [1,2]. Most are for simple roots and a few are for a repeated root. Here we are only interested in methods for repeated roots. In fact, we will not discuss derivative-free methods or methods with memory.

The usual technique of comparing a new method to existing ones, is by comparing the performance on selected problems using one or two initial points or by comparing the efficiency index (see [1]). In recent work, one can find a visual comparison, by plotting the basins of attraction for the methods. The idea of using basins of attraction appeared first in Stewart [3] and followed by the works of Amat et al. [4,5], and [6], Scott et al. [7], Chicharro et al. [8], Chun et al. [9–12], Cordero et al. [13], Neta et al. [14,15], Argyros and Magreñan, [16], Magreñan, [17] and Geum et al. [18–20] and [21]. In later works [11,12,22–24], we have introduced a more quantitative comparison, by listing the average number of iterations per point, the CPU time and the number of points requiring 40 iterations. We have also discussed methods to choose the parameters appearing in the method and/or the weight function (see, e.g. [25]). The only papers comparing basins of attraction for methods to obtain multiple roots are due to Geum et al. [18,19] and [20], Neta et al. [26], Neta and Chun [27–29], and Chun and Neta [30,31].

First we list the methods we consider here with their order of convergence (p), number of function- (and derivative-) evaluations per step (ν) and efficiency (I).

(1) A method of order 1.5 for **double** roots ($p = 1.5,\ \nu = 3,\ I = 1.1447$)
(2) Modified Newton's method (also known as Schröder's method) ($p = 2,\ \nu = 2,\ I = 1.4142$)
(3) Halley or Hansen–Patrick ($p = 3,\ \nu = 3,\ I = 1.4422$)
(4) Victory–Neta ($p = 3,\ \nu = 3,\ I = 1.4422$)

* Corresponding author.
E-mail addresses: cbchun@skku.edu (C. Chun), bneta@nps.edu (B. Neta).

https://doi.org/10.1016/j.cam.2018.02.009
0377-0427/Published by Elsevier B.V.

(5) Neta (Chebyshev-based method) ($p = 3,\ \nu = 3,\ I = 1.4422$)
(6) Dong (4 methods) ($p = 3,\ \nu = 3,\ I = 1.4422$)
(7) Osada ($p = 3,\ \nu = 3,\ I = 1.4422$)
(8) Laguerre ($p = 3,\ \nu = 3,\ I = 1.4422$)

- Euler–Cauchy
- Halley
- Ostrowski
- Hansen–Patrick

(9) Chun and Neta ($p = 3,\ \nu = 3,\ I = 1.4422$)
(10) Chun–Bae–Neta ($p = 3,\ \nu = 3,\ I = 1.4422$)
(11) Li et al. (6 methods) ($p = 4,\ \nu = 3,\ I = 1.5874$)
(12) Kanwar et al. ($p = 4,\ \nu = 3,\ I = 1.5874$)
(13) Zhou et al. ($p = 4,\ \nu = 3,\ I = 1.5874$)
(14) Liu and Zhou ($p = 4,\ \nu = 3,\ I = 1.5874$)
(15) Sbibih et al. ($p = 4,\ \nu = 3,\ I = 1.5874$)
(16) Soleymani ($p = 4,\ \nu = 3,\ I = 1.5874$)
(17) Geum et al. ($p = 4,\ \nu = 3,\ I = 1.5874$).
(18) Geum et al. ($p = 6,\ \nu = 4,\ I = 1.5651$)
(19) Geum et al. ($p = 6,\ \nu = 4,\ I = 1.5651$)
(20) Geum et al. ($p = 8,\ \nu = 4,\ I = 1.6818$).

(1) A method of order 1.5 for **double** roots given by Werner [32]

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= x_n - s_n u_n, \end{aligned} \tag{1}$$

where

$$s_n = \begin{cases} \dfrac{2}{1+\sqrt{1-4r_n}} & \text{if } r_n \le \dfrac{1}{4} \\ \dfrac{1}{2r_n} & \text{otherwise.} \end{cases}$$

We always use

$$u_n = \frac{f_n}{f_n'}, \tag{2}$$

$$r_n = \frac{f(y_n)}{f_n}, \tag{3}$$

and $f_n^{(i)}$ is short for $f^{(i)}(x_n),\ i = 1, 2, \ldots$.

Remark. We will not experiment with this method, since it is of a low order and limited to the case of double roots. One can see the basins for this method for the case of $(z^2-1)^2$ in [26].

(2) The quadratically convergent modified Newton's method is (see Schröder [33] or Rall [34])

$$x_{n+1} = x_n - m u_n. \tag{4}$$

(3) The cubically convergent Halley's method [35] which is a special case of the Hansen and Patrick's method [36]

$$x_{n+1} = x_n - \frac{u_n}{\frac{m+1}{2m} - \frac{u_n f_n''}{2f_n'}}. \tag{5}$$

(4) The third order method developed by Victory and Neta [37]

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f_n'}\frac{1+Ar_n}{1+Br_n}, \end{aligned} \tag{6}$$

where

$$\begin{aligned} A &= \mu^{2m} - \mu^{m+1}, \\ B &= -\frac{\mu^m(m-2)(m-1)+1}{(m-1)^2}, \\ \mu &= \frac{m}{m-1}. \end{aligned} \tag{7}$$

(5) The third order method developed by Neta [38] and based on Chebyshev's method (see [39–41]).

$$\begin{aligned} y_n &= x_n - \alpha u_n, \\ x_{n+1} &= x_n - u_n\left[\beta + \gamma \frac{f(y_n)}{f_n}\right], \end{aligned} \tag{8}$$

where

$$\alpha = \frac{1}{2}\frac{m(m+3)}{m+1},$$

$$\beta = \frac{m^3+4m^2+9m+2}{(m+3)^2},$$

$$\gamma = \frac{2^{m+1}(m^2-1)}{(m+3)^2\left(\frac{m-1}{m+1}\right)^m}.$$

(6) The four third order methods developed by Dong [42] and [43]:

(a) Dong1

$$\begin{aligned} y_n &= x_n - \sqrt{m}\, u_n, \\ x_{n+1} &= y_n - m\left(1 - \frac{1}{\sqrt{m}}\right)^{1-m} \frac{f(y_n)}{f_n'}, \end{aligned} \tag{9}$$

(b) Dong2

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n + \frac{u_n r_n}{r_n - \left(1 - \frac{1}{m}\right)^{m-1}}, \end{aligned} \tag{10}$$

(c) Dong3

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n - \frac{f_n}{\left(\frac{m}{m-1}\right)^{m+1} f'(y_n) + \frac{m-m^2-1}{(m-1)^2} f_n'}, \end{aligned} \tag{11}$$

(d) Dong4

$$\begin{aligned} y_n &= x_n - \frac{m}{m+1} u_n, \\ x_{n+1} &= y_n - \frac{\frac{m}{m+1} f_n}{\left(1 + \frac{1}{m}\right)^m f'(y_n) - f_n'}. \end{aligned} \tag{12}$$

(7) The third order method due to Osada [44]

$$x_{n+1} = x_n - \frac{1}{2}m(m+1)u_n + \frac{1}{2}(m-1)^2\frac{f_n'}{f_n''}. \tag{13}$$

(8) Laguerre's family of methods

$$x_{n+1} = x_n - \frac{\lambda u_n}{1 + sgn(\lambda - m)\sqrt{\left(\frac{\lambda-m}{m}\right)\left[(\lambda-1) - \lambda \frac{u_n f''(x_n)}{f'(x_n)}\right]}} \tag{14}$$

where λ ($\neq 0$, m) is a real parameter. When $f(x)$ is a polynomial of degree n, this method with $\lambda = n$ is the ordinary Laguerre method for multiple roots, see Bodewig [45]. This method converges cubically. Some special cases are:

- Euler–Cauchy for $\lambda = 2m$

$$x_{n+1} = x_n - \frac{2mu_n}{1 + \sqrt{(2m-1) - 2m\frac{u_n f''(x_n)}{f'(x_n)}}}. \tag{15}$$

- Halley for $\lambda \to 0$ after rationalization

$$x_{n+1} = x_n - \frac{u_n}{\frac{m+1}{2m} - \frac{u_n f''(x_n)}{2f'(x_n)}}. \tag{16}$$

- Ostrowski for $\lambda \to \infty$

$$x_{n+1} = x_n - \frac{\sqrt{m}u_n}{\sqrt{1 - \frac{u_n f''(x_n)}{f'(x_n)}}}. \tag{17}$$

- Hansen–Patrick family [36] for $\lambda = m(1/\nu + 1)$

$$x_{n+1} = x_n - \frac{m(\nu+1)u_n}{\nu + \sqrt{\big(m(\nu+1) - \nu\big) - m(\nu+1)\frac{u_n f''(x_n)}{f'(x_n)}}}. \tag{18}$$

Petković et al. [46] have shown the equivalence between Laguerre family (14) and Hansen–Patrick family (18). When $\lambda \to m$ the method becomes second order given by (4).
Neta and Chun [27] have shown that the best method of Laguerre family is Euler–Cauchy.

(9) Chun and Neta third order [47], denoted CN3,

$$x_{n+1} = x_n - \frac{2m^2 u_n^2 f''(x_n)}{m(3-m)u_n f''(x_n) + (m-1)^2 f'(x_n)}. \tag{19}$$

(10) Chun, Bae and Neta [48]
Two new third-order families of methods for multiple roots.

(a) CBN1

$$\begin{aligned} x_{n+1} &= x_n - \frac{m[(2\theta-1)m + 3 - 2\theta]}{2}u_n + \frac{\theta(m-1)^2}{2}\frac{f'(x_n)}{f''(x_n)} \\ &\quad - \frac{(1-\theta)m^2}{2}\frac{u_n^2 f''(x_n)}{f'(x_n)}, \end{aligned} \tag{20}$$

(b) CBN2

$$\begin{aligned} y_n &= x_n - u_n, \\ x_{n+1} &= y_n + \theta\frac{u_n r_n}{r_n - (1-\frac{1}{m})^{m-1}} - (1-\theta)\frac{f(y_n)}{f'(x_n)}\frac{1 + Ar_n}{1 + Br_n}, \end{aligned} \tag{21}$$

where A and B are given by (7).

(11) The six fourth order methods developed by Li et al. [49] and based on the results of Neta and Johnson [50] and Neta [51].

(a) LCN1

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ z_n &= x_n - \frac{2m}{m+2}u_n + 2\left(\frac{m}{m+2}\right)^m v_n, \\ x_{n+1} &= x_n - \frac{f(x_n)}{a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(z_n)}, \end{aligned} \tag{22}$$

where we always use

$$v_n = \frac{f_n}{f'(y_n)}, \tag{23}$$

and

$$\begin{aligned} a_1 &= -\frac{1}{16}\frac{3m^4 + 16m^3 + 40m^2 - 176}{m(m+8)}, \\ a_2 &= \frac{1}{8}\frac{m^4 + 3m^3 + 10m^2 - 4m + 8}{(\frac{m}{m+2})^m m(m+8)}, \\ a_3 &= \frac{1}{16}\frac{m^5 + 6m^4 + 8m^3 - 16m^2 - 48m - 32}{m^2(m+8)}. \end{aligned}$$

(b) LCN2

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ z_n &= x_n - 2(\frac{m}{m+2})^m v_n, \\ x_{n+1} &= x_n - \frac{f(x_n)}{a_1 f'(x_n) + a_2 f'(y_n) + a_3 f'(z_n)}, \end{aligned} \tag{24}$$

where

$$\begin{aligned} a_1 &= \frac{1}{8}\frac{m^6 - m^5 - 14m^4 + 12m^3 + 48m^2 - 80m + 32}{m(m^3 + 2m^2 - 8m + 4)}, \\ a_2 &= -\frac{m}{16}\frac{3m^4 - 6m^3 - 20m^2 + 40m - 16}{(\frac{m}{m+2})^m (m^3 + 2m^2 - 8m + 4)}, \\ a_3 &= \frac{1}{16}\frac{m^3(m^2 - 4)}{(\frac{m}{m+2})^m (m^3 + 2m^2 - 8m + 4)}. \end{aligned}$$

(c) LCN3

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ z_n &= x_n - \frac{2m}{m+2}u_n + 2(\frac{m}{m+2})^m v_n, \\ x_{n+1} &= x_n - a_1 u_n - a_2 v_n - a_3 \frac{f(x_n)}{f'(z_n)}, \end{aligned} \tag{25}$$

where

$$\begin{aligned} a_1 &= \frac{m}{8}\frac{m^4 + 4m^3 - 8m + 48}{m^2 + 2m + 6}, \\ a_2 &= \frac{1}{4}\frac{(\frac{m}{m+2})^m m(m^3 + 12m^2 + 36m + 32)}{m^2 + 2m + 6}, \\ a_3 &= -\frac{1}{8}\frac{m^2(m^3 + 6m^2 + 12m + 8)}{m^2 + 2m + 6}. \end{aligned}$$

(d) LCN4

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ z_n &= x_n - 2(\frac{m}{m+2})^m v_n, \\ x_{n+1} &= x_n - a_1 u_n - a_2 v_n - a_3 \frac{f(x_n)}{f'(z_n)}, \end{aligned} \tag{26}$$

where

$$\begin{aligned} a_1 &= -\frac{1}{4}\frac{m(2m^4 - m^3 - 12m^2 + 20m - 8)}{m^2 - 4m + 2}, \\ a_2 &= \frac{1}{8}\frac{(\frac{m}{m+2})^m m(5m^4 + 10m^3 - 16m^2 - 24m + 16)}{m^2 - 4m + 2}, \\ a_3 &= -\frac{1}{8}\frac{m^3(m+2)^2(\frac{m}{m+2})^m}{m^2 - 4m + 2}. \end{aligned}$$

(e) LCN5

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ x_{n+1} &= x_n - a_3 v_n - \frac{u_n}{b_1 + b_2 t_n}, \end{aligned} \tag{27}$$

where

$$t_n = \frac{f'(y_n)}{f'(x_n)} \tag{28}$$

and

$$a_3 = -\frac{1}{2}\frac{(\frac{m}{m+2})^m m(m^4+4m^3-16m-16)}{m^3-4m+8},$$
$$b_1 = -\frac{(m^3-4m+8)^2}{m(m^4+4m^3-4m^2-16m+16)(m^2+2m-4)},$$
$$b_2 = \frac{m^2(m^3-4m+8)}{(\frac{m}{m+2})^m(m^4+4m^3-4m^2-16m+16)(m^2+2m-4)}.$$

(f) LCN6

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ x_{n+1} &= x_n - a_3 u_n - \frac{u_n}{b_1+b_2 t_n}, \end{aligned} \tag{29}$$

where

$$a_3 = -\frac{1}{2}m^2 + m,$$
$$b_1 = -\frac{1}{m}, \quad b_2 = \frac{1}{m(\frac{m}{m+2})^m}.$$

(12) The fourth-order family of methods by Kanwar et al. [52] is given by

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}\frac{u_n}{1-pu_n}, \\ x_{n+1} &= x_n - \frac{u_n}{1-pu_n} Q\left(\frac{t_n+h}{\tau - pu_n}\right), \end{aligned} \tag{30}$$

where Q is a real valued weight function satisfying

$$\begin{aligned} &Q(\mu) = m, \\ &Q'(\mu) = -\frac{m^3\left(\frac{m}{m+2}\right)^{-m}}{4(1+m)}, \\ &Q''(\mu) = \frac{m^4\left(\frac{m}{m+2}\right)^{-2m}}{4(m+1)^2}, \\ &|Q'''(\mu)| < \infty, \end{aligned} \tag{31}$$

and

$$\begin{aligned} &\mu = \frac{2(m+1)}{m+2}\left(\frac{m}{m+2}\right)^{m-1} = \frac{2(m+1)}{m}\left(\frac{m}{m+2}\right)^{m}, \\ &\tau = \frac{1}{m+1}, \\ &h = -\left(\frac{m}{m+2}\right)^{m}. \end{aligned} \tag{32}$$

Remark. The authors gave an erroneous value of μ which is corrected in [30].
The authors considered three members of the family. In all cases the parameter p is taken as ± 1 so that there is no subtraction in the denominator. The third member chosen by Kanwar et al. was a quadratic polynomial for Q. It will not be considered here, since Chun and Neta [29] have shown that such a choice will give inferior results.

- KBK1

$$Q(t) = \frac{A}{t} + B, \tag{33}$$

where

$$\begin{aligned} A &= m(1+m)\left(\frac{m}{m+2}\right)^m, \\ B &= -\frac{m(m-2)}{2}. \end{aligned} \tag{34}$$

- KBK2

$$Q(t) = \frac{A}{(t+C)^2} + B, \tag{35}$$

where

$$\begin{aligned} A &= \frac{27}{8}(m+1)^2\left(\frac{m}{m+2}\right)^{2m}, \\ B &= -\frac{3}{8}m^2 + m, \\ C &= \frac{m+1}{m}\left(\frac{m}{m+2}\right)^m. \end{aligned} \tag{36}$$

(13) The method presented by Zhou et al. [53]

$$\begin{aligned} y_n &= x_n - \frac{2m}{m+2}u_n, \\ x_{n+1} &= x_n - \phi(t_n)u_n, \end{aligned} \tag{37}$$

where ϕ is at least twice differentiable function satisfying the following conditions

$$\begin{aligned} &\phi(\lambda) = m, \\ &\phi'(\lambda) = -\frac{1}{4}m^3\left(\frac{m+2}{m}\right)^m, \\ &\phi''(\lambda) = \frac{1}{4}m^4\left(\frac{m+2}{m}\right)^{2m}, \end{aligned} \tag{38}$$

and $\lambda = \left(\frac{m}{m+2}\right)^{m-1}$, we will consider the following functions:

- ZCS1 [29]

$$\phi(t) = \frac{b+ct+dt^2}{1+at+gt^2}. \tag{39}$$

where

$$\begin{aligned} b &= \frac{m}{8}\left((m+2)^2\lambda ma + (m+2)\lambda^2 m^2 g + m^3 + 6m^2 + 8m + 8\right), \\ c &= -\frac{m}{4\lambda}\left((m^3+3m^2+2m-4)\lambda a + (m^2+m-2)\lambda^2 mg \right.\\ &\quad \left. + m(m+2)(m+3)\right), \\ d &= \frac{m}{8\lambda^2}\left(m^2(m+2)\lambda a + (m^3-4m+8)\lambda^2 g + m(m+2)^2\right), \end{aligned} \tag{40}$$

with $a = -4, g = 0$.

- ZCS2 [29]
  Same weight function ϕ given by (39) with $a = -6.01, g = 8.04$.
- ZCS3 [53]

$$\phi(t) = \frac{B+Ct}{1+At}, \tag{41}$$

where $A = -\left(\frac{m+2}{m}\right)^m, B = -\frac{m^2}{2}, C = \frac{1}{2}m(m-2)\left(\frac{m+2}{m}\right)^m$.

(14) There are two other optimal fourth order methods from the family developed by Liu and Zhou [54]

$$\begin{aligned} y_n &= x_n - mu_n, \\ x_{n+1} &= x_n - mH(w_n)u_n, \end{aligned} \tag{42}$$

where

$$w_n = \sqrt[(m-1)]{t_n}$$

and $H(0) = 0,\ H'(0) = 1,\ H''(0) = \frac{4m}{m-1}$.
The two members given there are

- LZ11 (Liu and Zhou [54])

$$\begin{aligned} y_n &= x_n - m u_n, \\ x_{n+1} &= y_n - m\left(w_n + \frac{2m}{m-1} w_n^2\right) u_n, \end{aligned} \tag{43}$$

- LZ12 (Liu and Zhou [54])

$$\begin{aligned} y_n &= x_n - m u_n, \\ x_{n+1} &= y_n + \frac{(m-1) m w_n}{1 - m + 2 m w_n} u_n. \end{aligned} \tag{44}$$

(15) Sbibih et al. [55] SSTZ

$$\begin{aligned} y_n &= x_n - \mu u_n, \\ x_{n+1} &= x_n - \phi(r_n) u_n, \end{aligned} \tag{45}$$

where the weight function ϕ is a complex function, and μ is a non-zero real or complex number. They have shown that the family is of order three, for $m \geq 2$, and of order four for simple roots, if the function ϕ satisfies the following conditions:

$$\begin{aligned} &\phi(t^m) = m \\ &\phi'(t^m) = \frac{1}{t^{m-1}(1-t)^2} \\ &\left|\left(\frac{1}{\phi'}\right)'(t^m)\right| < \infty \end{aligned} \tag{46}$$

where $t = 1 - \frac{\mu}{m}$.
They have also demonstrated that the following methods are special cases:

- Dong3 and Dong4 [43]
- Victory and Neta [37]
- Neta [38]
- Chun and Neta [47]
- Homeier [56]
- Geum and Kim [57]
- Kim and Geum [58].

The authors picked 4 different weight functions

- SSTZ1

$$\begin{aligned} &\phi(x) = ax + b \\ &a = \frac{1}{t^{m-1}(1-t)^2} \\ &b = m - \frac{t}{(1-t)^2} \end{aligned} \tag{47}$$

- SSTZ2

$$\begin{aligned} &\phi(x) = \frac{a}{b - x} \\ &a = m^2 t^{m-1} (1-t)^2 \\ &b = m t^{m-1} (1-t)^2 + t^m \end{aligned} \tag{48}$$

- SSTZ3

$$\begin{aligned} &\phi(x) = x^2 + ax + b \\ &a = \frac{1}{t^{m-1}(1-t)^2} - 2t^m \\ &b = m + t^{2m} - \frac{t}{(1-t)^2} \end{aligned} \tag{49}$$

- SSTZ4

$$\phi(x) = \frac{x^2+ax+b}{(1-x)^2}$$
$$a = -2t^m - 2m(1-t^m) + \frac{(1-t^m)^2}{t^{m-1}(1-t)^2} \tag{50}$$
$$b = t^{2m} + m(1-t^{2m}) - \frac{t(1-t^m)^2}{(1-t)^2}.$$

(16) Soleymani and Babajee [59], denoted SB,

$$y_n = x_n - \frac{2m}{m+2}u_n,$$
$$x_{n+1} = x_n + \frac{4md}{d(m^2+2m-4)-m^2t_n}\left[1-\frac{m^3(m-2)}{16d^2}(t_n-\frac{m+2}{m}d)^2\right]u_n \tag{51}$$

where $d = (\frac{m}{m+2})^m$.

(17) Geum et al. [20]

A fourth order family of methods

$$y_n = x_n - \gamma u_n, \ \gamma \text{ is a real number},$$
$$x_{n+1} = x_n - Q_f(s)u_n, \tag{52}$$

where $s = t_n^{1/k}$, k is integer, t_n given by (28) and $\gamma = 2m/(m+2)$; Q_f is analytic in a neighborhood of λ with λ is real number to be determined later for optimal quartic-order convergence. Since s is a one-to-k multiple-valued function, we consider its principal analytic branch [60]. Hence, it is convenient to treat s as a principal root given by $s = \exp[\frac{1}{k}\mathrm{Log}(t_n)]$, with $\mathrm{Log}(t_n) = \mathrm{Log}|t_n| + i\,\mathrm{Arg}(t_n)$ for $-\pi < \mathrm{Arg}(t_n) \le \pi$; this convention of Arg(z) for complex z agrees with that of Log[z] command of Mathematica [61] to be adopted in numerical experiments. By means of further inspection of s, we find that λ is characterized in such a way that $s = |t_n|^{1/k} \cdot \exp[\frac{i}{k}\,\mathrm{Arg}(t_n)] = \lambda + O(e_n)$.
Several possible weight functions were suggested in [20] and found that the following performed best:

- GKN2A1

$$Q_f(s) = \frac{m+a_2(s-\rho)}{1+b_2(s-\rho)} \tag{53}$$

where

$$a_2 = \frac{\mu}{4\rho}$$

$$b_2 = \frac{\delta}{2\rho}$$

with $\mu = m(2+2m-m^3),\ \delta = 1+m+m^2,\ \rho = \left(\frac{m}{m+2}\right)^{1-1/m}$

- GKN2A2

$$Q_f(s) = \frac{m+a_2(s-\rho)}{1+b_2(s-\rho)} \tag{54}$$

where a_2, and b_2 are given as in GKN2A1 and $\mu = m(8+2m-3m^2-m^3),\ \delta = (m+2)^2,\ \rho = \left(\frac{m}{m+2}\right)^{(m-1)/(m+3)}$

- GKN2C

$$Q_f(s) = \frac{m+a_3(s-\rho)^2}{1+b_3(s-\rho)} \tag{55}$$

where

$$a_3 = \frac{\tau\mu}{16\rho^2}$$

$$b_3 = \frac{\tau}{4\rho}$$

with

$$\mu = m(8+2m-3m^2-m^3),\ \tau = m(m+2)(m+3),\ \rho = \left(\frac{m}{m+2}\right)^{(m-1)/(m+3)}.$$

Table 1
The function H_f for each of the methods.

Method	H_f
Werner	s_n
Schröder	m
Halley	$\frac{1}{\frac{m+1}{2m}-\frac{1}{2}u_n\frac{f''_n}{f'_n}}$
Victory–Neta	$1+r_n\frac{1+Ar_n}{1+Br_n}$
N3	$\beta+\gamma r_n$
Dong1	$\sqrt{m}+m\left(1-\frac{1}{\sqrt{m}}\right)^{1-m}r_n$
Dong2	$1-\frac{r_n}{r_n-\left(1-\frac{1}{m}\right)^{m-1}}$
Dong3	$1+\frac{f'(x_n)}{\left(\frac{m}{m-1}\right)^{m+1}f'(y_n)+\frac{m-m^2-1}{(m-1)^2}f'(x_n)}$
Dong4	$\frac{m}{m-1}+\frac{\frac{m}{m+1}f'(x_n)}{\left(1+\frac{1}{m}\right)^{m}f'(y_n)-f'(x_n)}$
Osada	$\frac{1}{2}m(m+1)-\frac{1}{2}(m-1)^2\frac{f'(x_n)^2}{f''(x_n)f(x_n)}$
Euler–Cauchy	$\frac{2m}{1+\sqrt{(2m-1)-2m\frac{u_nf''(x_n)}{f'(x_n)}}}$
CN3	$\frac{2m^2u_nf''(x_n)}{m(3-m)u_nf''(x_n)+(m-1)^2f'(x_n)}$
CBN1	$\frac{m[(2\theta-1)m+3-2\theta]}{2}-\frac{\theta(m-1)^2}{2}\frac{f'(x_n)^2}{f(x_n)f''(x_n)}+\frac{(1-\theta)m^2}{2}u_n\frac{f''(x_n)}{f'(x_n)}$

Table 2
The function H_f for each of the methods.

Method	H_f
LCN6	$a_3+\frac{1}{b_1+b_2t_n}$
SSTZ2	$\frac{a}{b-r_n}$
SB	$1-\frac{m^3(m-2)}{16d^2}\left(t_n-\frac{m+2}{m}d\right)^2$
GKN2A1	$Q_f(s)$
GKN2A2	$Q_f(s)$
GKN2C	$Q_f(s)$
GKN4C	$m+Q_f(s,q)\frac{r_n}{t_n}$
GKN5YD	$K_f(s,q)$
WI3X	$L_f(s)+K_f(s,v)$

(18) Geum et al. sixth order [19]

A family of two-point sixth-order multiple-zero finders of modified double-Newton type

$$\begin{cases} y_n = x_n - mu_n, \\ x_{n+1} = y_n - Q_f(s,q)\cdot\dfrac{f(y_n)}{f'(y_n)}, \end{cases} \tag{56}$$

where the desired form of the weight function Q_f using only two-point functional information at x_n and y_n, with

$$s=\left(\frac{f(y_n)}{f(x_n)}\right)^{1/m}$$

and

$$q=\left(\frac{f'(y_n)}{f'(x_n)}\right)^{1/(m-1)}.$$

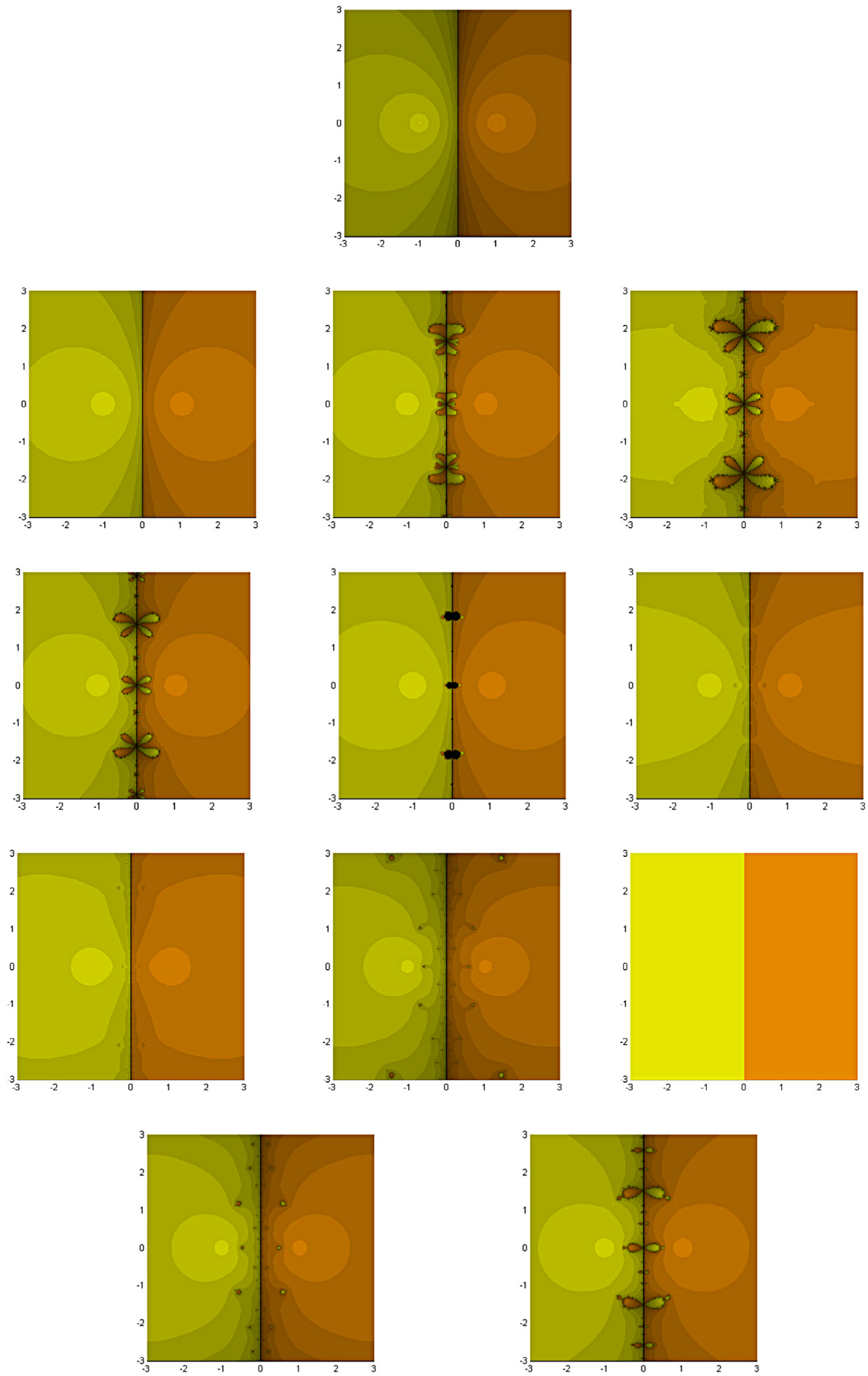

Fig. 1. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^2 - 1)^2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Extraneous fixed points for each of the methods.

Method	Extraneous fixed points	Stability
Werner	None	
Schröder	None	
Halley	None	
Victory–Neta	$\pm.304883324753541 \pm .218806866816708i$, $\pm.236865602520895 \pm .0485319817905315i$,	All repulsive
N3	$\pm.496010694841520 \pm .247226513585838i$	All repulsive
Dong1	$\pm.411795739431937 \pm .180936391794009i$	All repulsive
Dong2	0, 0, 0, 0	Parabolic
Dong3	$\pm.365828568271531, \pm.824187531341104i$	All repulsive
Dong4	$\pm.2, \pm.4472135955i$	All repulsive
Osada	$\pm.6546536707$	All repulsive
Euler–Cauchy	None	
CN3	$\pm.5773502692$	Repulsive
CBN1	$\pm.5278690810 \pm .04826983348i$	All repulsive but almost parabolic
LCN6	None	
SSTZ2	$0, 0, 0, 0, \pm1, \pm1$	All parabolic
SB	0, 0, 0, 0	All parabolic
GKN2A1	$\pm.191563 \pm .158752i$	Repulsive
GKN2A2	$\pm.202398 \pm .164549i$	Repulsive
GKN2C	$\pm.349353, \pm.675194i$	Repulsive
GKN4C	$\pm.286835 \pm .655947i, \pm.240302i, \pm.620034, \pm.650152$	Repulsive
GKN5YD	$\pm1.29099i, \pm i, \pm.57735i, \pm.377964i$	Repulsive
WI3X	$0(double), \ \pm i, \pm2.41421i, \ \pm 414214i$	Indifferent

Table 4
Average number of function evaluations per point for each example (1–9) and each of the methods.

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Ex7	Ex8	Ex9	Average
Schröder	11.65	15.21	15.21	14.62	28.30	22.22	14.81	28.30	20.37	18.97
Halley	11.63	13.31	13.31	14.71	18.57	16.04	13.18	18.57	15.74	15.01
Victory–Neta	12.41	18.27	15.99	15.11	27.69	24.79	14.35	30.30	21.69	20.07
N3	12.62	21.43	22.34	17.55	41.81	34.95	17.87	41.96	32.42	27.0
Dong1	12.94	20.45	20.44	17.67	39.31	33.37	17.10	40.41	29.60	25.7
Dong2	11.92	18.65	16.42	14.8	25.79	22.75	13.99	26.72	21.24	19.14
Dong3	11.11	15.08	11.62	12.84	16.58	13.81	11.13	16.31	14.03	13.61
Dong4	10.27	11.77	12.30	13.50	18.26	15.00	12.03	17.99	14.84	14.0
Osada	14.72	19.89	18.95	18.54	37.49	28.90	18.10	37.95	26.23	24.53
Euler–Cauchy	3.00	11.44	11.43	12.44	22.05	16.83	10.40	22.05	14.22	13.76
CN3	14.14	19.22	16.55	16.88	30.13	29.17	16.73	37.26	20.24	22.26
CBN1	13.29	18.54	18.36	18.1	37.05	29.38	16.65	37.63	26.86	23.99
LCN6	13.26	19.92	13.78	13.87	23.93	17.80	13.24	22.85	18.47	17.46
SSTZ2	11.63	15.22	14.44	14.76	23.05	19.94	15.26	24.15	18.47	17.44
SB	13.26	19.92	14.67	14.04	26.10	21.48	13.36	26.32	19.83	18.78
GKN2A1	10.24	12.46	13.83	13.89	24.29	17.91	13.08	22.90	18.57	16.35
GKN2A2	10.19	12.37	13.82	13.89	24.29	17.98	13.07	22.97	18.57	16.35
GKN2C	10.04	12.17	13.29	13.69	24.18	18.36	12.60	23.44	17.85	16.18
GKN4C	32.70	35.78	35.32	35.42	50.02	39.99	31.13	48.06	59.73	40.91
GKN5YD	15.31	16.76	27.49	24.23	41.55	26.76	20.36	33.46	35.48	26.82
WI3X	11.44	14.36	35.08	17.80	45.32	16.68	25.28	34.40	46.92	27.48

Four different families were suggested by the authors and experimented with. It was found that the best is GKN4C, where

$$Q_f(s,q) = \frac{m + a_1 s}{1 + b_1 s + b_2 s^2} \times \frac{1}{1 + c_1 q}, \tag{57}$$

where $a_1 = \frac{2m(4m^4-16m^3+31m^2-30m+13)}{(m-1)(4m^2-8m+7)}$, $b_1 = \frac{4(2m^2-4m+3)}{(m-1)(4m^2-8m+7)}$, $b_2 = -\frac{4m^2-8m+3}{4m^2-8m+7}$ and $c_1 = 2(m-1)$.

(19) Geum et al. [21]
Another family of sixth order three-point iterative methods

$$\begin{cases} y_n = x_n - m u_n, \\ z_n = x_n - m \cdot Q_f(s) u_n, \\ x_{n+1} = x_n - m \cdot K_f(s,q) u_n, \end{cases} \tag{58}$$

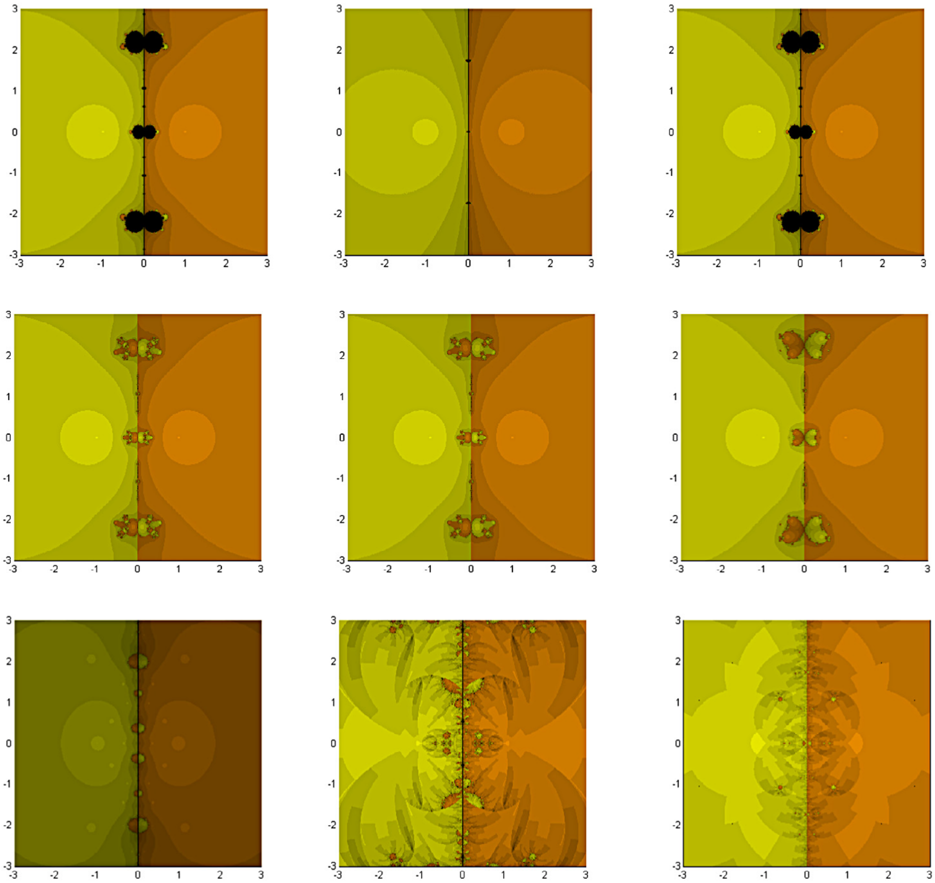

Fig. 2. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^2-1)^2$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
CPU time (in seconds) required for each example (1–9) and each of the methods.

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Ex7	Ex8	Ex9	Average
Schröder	151.48	252.85	306.87	282.13	865.09	537.69	322.77	862.42	454.57	448.43
Halley	176.92	284.34	353.00	383.79	804.44	536.46	413.37	734.09	524.13	467.84
Victory–Neta	251.26	456.80	477.19	454.95	1248.66	891.89	523.93	1409.11	802.02	723.98
N3	179.70	373.70	449.47	364.78	1274.11	846.97	447.44	1409.28	780.99	680.72
Dong1	239.95	447.79	448.46	373.36	1288.02	1001.82	502.24	1502.98	871.75	741.82
Dong2	212.36	435.79	466.71	400.77	1065.32	760.86	463.42	1146.11	678.36	625.52
Dong3	171.82	312.00	297.65	311.32	644.78	424.68	323.90	614.42	405.95	389.61
Dong4	166.53	255.97	317.80	346.21	707.31	450.48	354.78	700.32	462.26	417.96
Osada	188.70	373.98	467.52	440.58	1441.29	876.32	524.98	1354.32	775.47	715.91
Euler–Cauchy	110.84	422.97	488.28	497.10	1284.76	789.51	483.74	1222.44	678.03	664.18
CN3	242.99	485.59	563.83	592.74	1604.36	1163.50	658.07	1786.29	829.05	880.71
CBN1	269.24	542.73	728.32	697.39	2202.41	1349.16	767.65	2070.65	1224.64	1094.69
LCN6	225.19	428.52	356.59	358.72	928.92	558.40	397.55	832.61	562.60	516.57
SSTZ2	190.54	228.23	333.45	320.96	725.45	530.36	394.10	733.97	467.24	442.70
SB	240.44	448.02	399.55	388.44	1036.19	688.65	407.27	980.53	630.70	579.98
GKN2A1	170.19	520.23	1285.43	1249.01	2666.63	1753.03	1275.85	2360.92	1847.14	1458.71
GKN2A2	779.11	1002.09	1303.15	1267.24	2690.49	1793	1271.49	2355.26	1766.40	1580.91
GKN2C	779.74	1013.60	1229.63	1255.29	2551.65	1809.69	1237.96	2447.59	1636.47	1551.29
GKN4C	891.41	1147.76	3641.83	3473.80	5924.45	3585.96	2505.16	4177.14	6454.71	3533.58
GKN5YD	578.08	714.77	2928.72	2549.29	4991.94	3219.67	2342.15	3830.95	3962.43	2790.89
WI3X	528.95	767.416	1984.55	2033.89	3202.67	2066.64	1596.09	2406.81	2898.58	1942.84

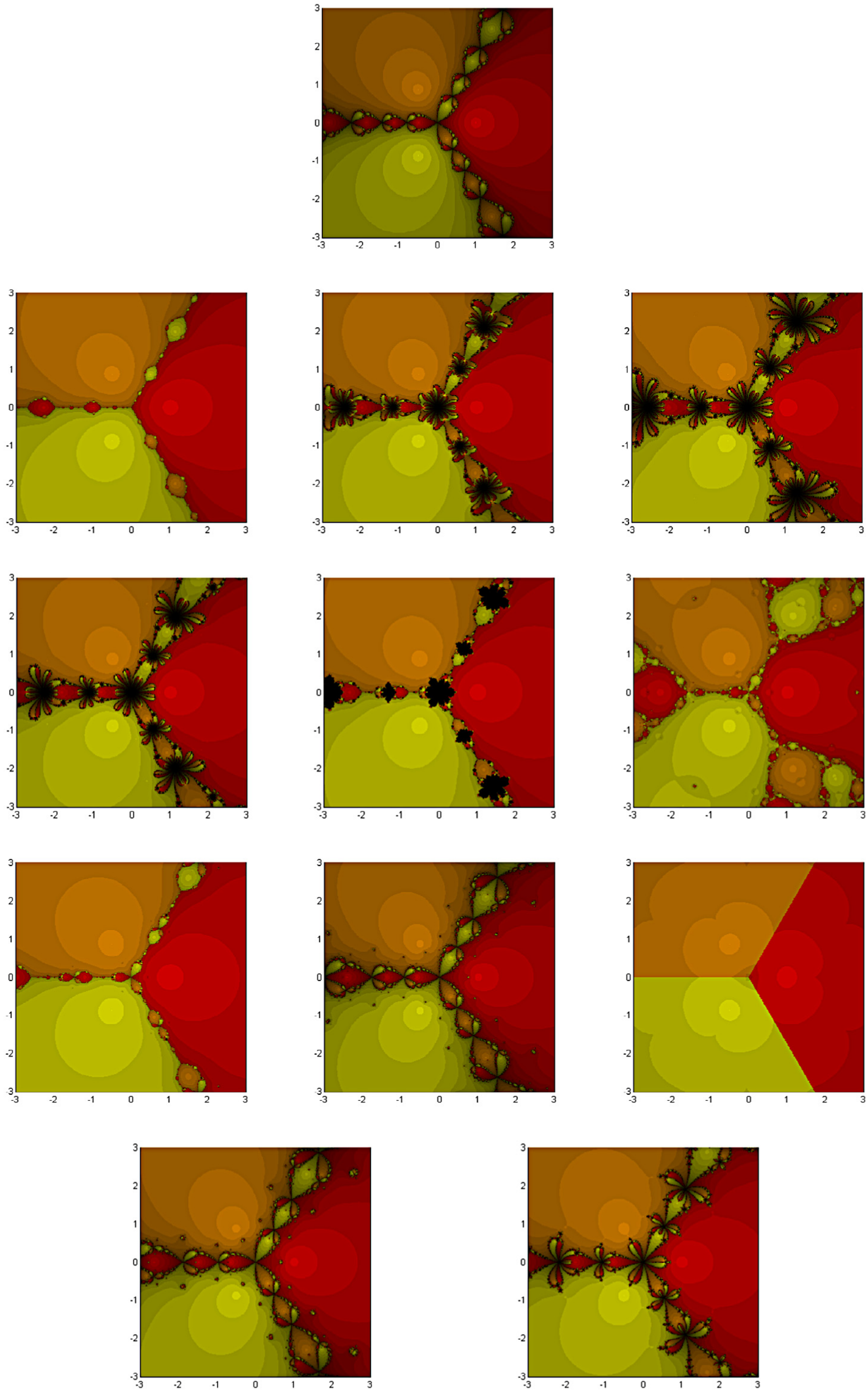

Fig. 3. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^3 - 1)^2$.

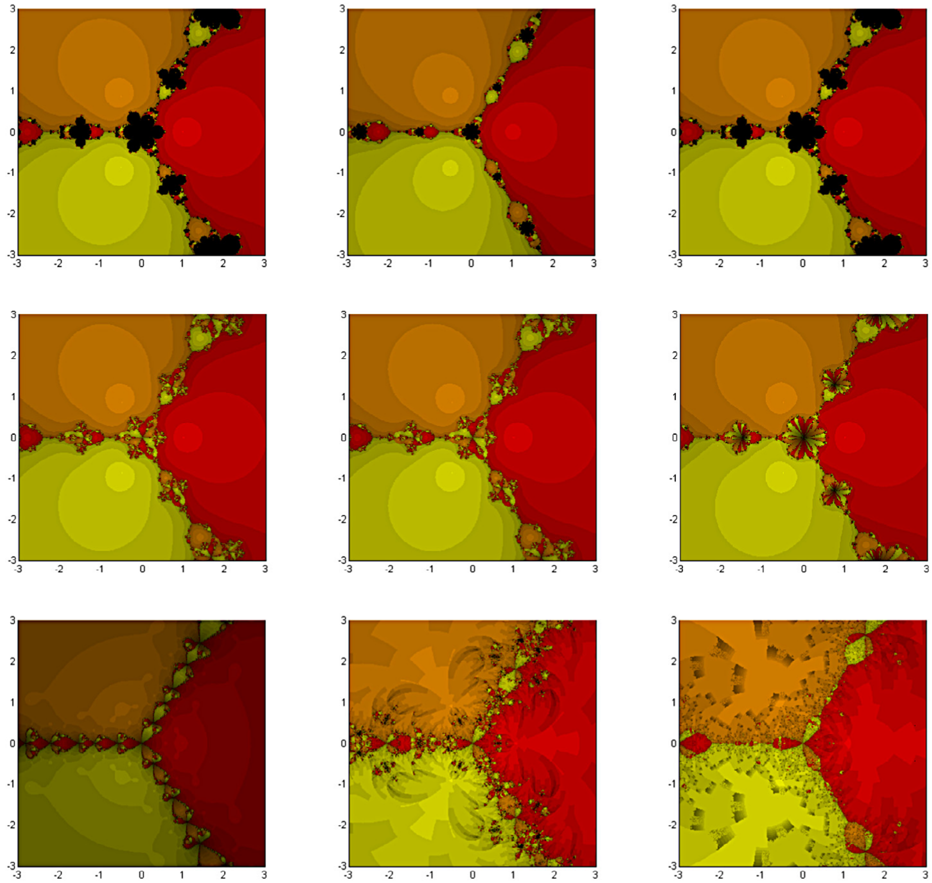

Fig. 4. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^3-1)^2$.

where

$$s = r_n^{1/m}, \tag{59}$$

$$q = \left[\frac{f(z_n)}{f(x_n)}\right]^{\frac{1}{m}}, \tag{60}$$

and where r_n is given by (3) and Q_f is analytic in a neighborhood of 0 and K_f is holomorphic [62] in a neighborhood of $(0, 0)$. Since s and υ are respectively one-to-m multiple-valued functions, we consider their principal analytic branches [60].

Several possible weight functions were suggested in [21] and it was shown that GKN5YD is best, i.e.

$$Q_f(s) = \frac{(s-2)(2s-1)}{(s-1)(5s-2)} \tag{61}$$

$$K_f(s, q) = \frac{(s-2)(2s-1)}{(5s-2)(s+q-1)}. \tag{62}$$

(20) Geum et al. [63]

This is the only known family of eighth order methods

$$\begin{cases} y_n = x_n - m \cdot \dfrac{f(x_n)}{f'(x_n)}, \\ z_n = x_n - m \cdot L_f(s) \cdot \dfrac{f(x_n)}{f'(x_n)}, \\ x_{n+1} = x_n - m \cdot [L_f(s) + K_f(s, \upsilon)] \cdot \dfrac{f(x_n)}{f'(x_n)}, \end{cases} \tag{63}$$

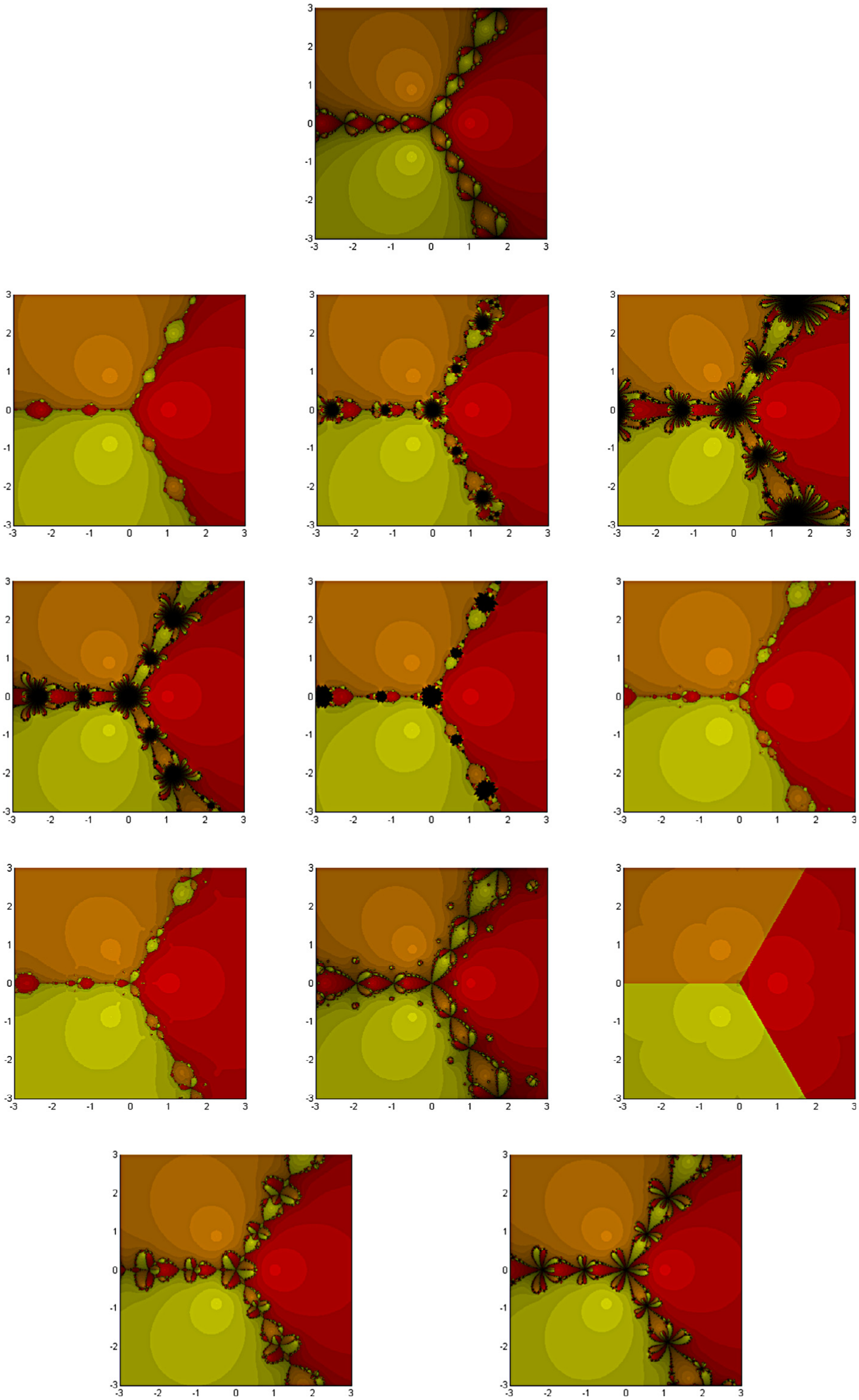

Fig. 5. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^3 - 1)^4$.

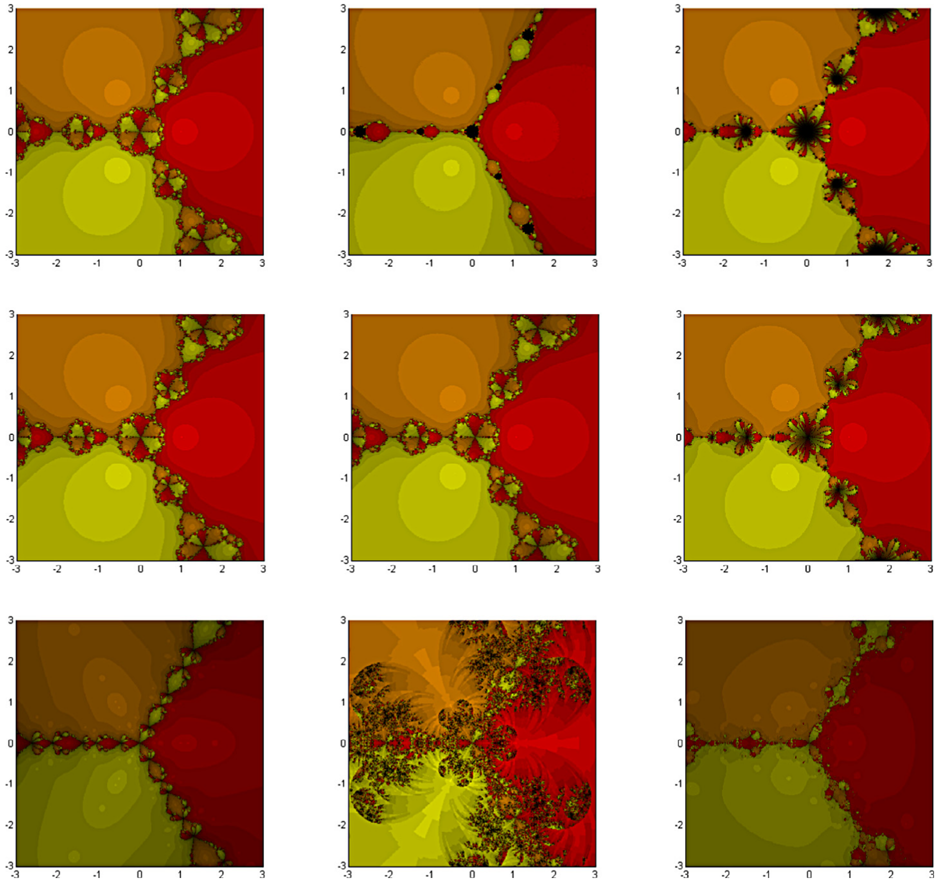

Fig. 6. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^3-1)^4$.

where

$$s=\left[\frac{f(y_n)}{f(x_n)}\right]^{\frac{1}{m}}, \tag{64}$$

$$v=\left[\frac{f(z_n)}{f(y_n)}\right]^{\frac{1}{m}}. \tag{65}$$

It was found that the best method (**WI3X**) is when

$$L_f(s)=\frac{1-s}{1-2s}$$

and

$$K_f(s,v)=-sv\frac{1-3s+s^2}{-1+5s-6s^2-s^3+(1-3s-s^2+6s^3)v}.$$

2. Extraneous fixed points

In this section, we introduce the notion of extraneous fixed points and show how to find those for any given method. It is easy to see that any method can be written as

$$x_{n+1}=x_n-H_f\frac{f_n}{f_n'} \tag{66}$$

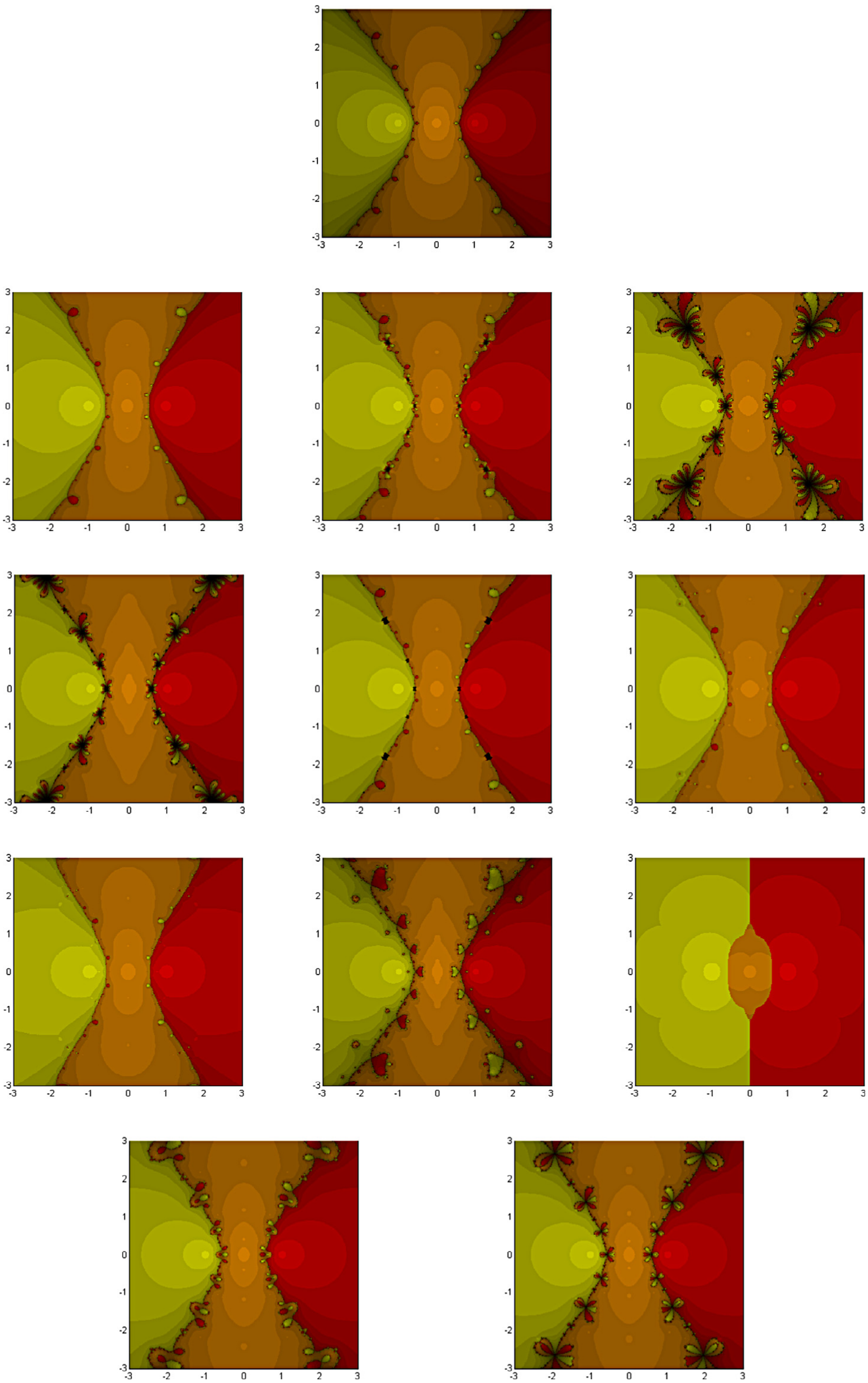

Fig. 7. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^3 - z)^4$.

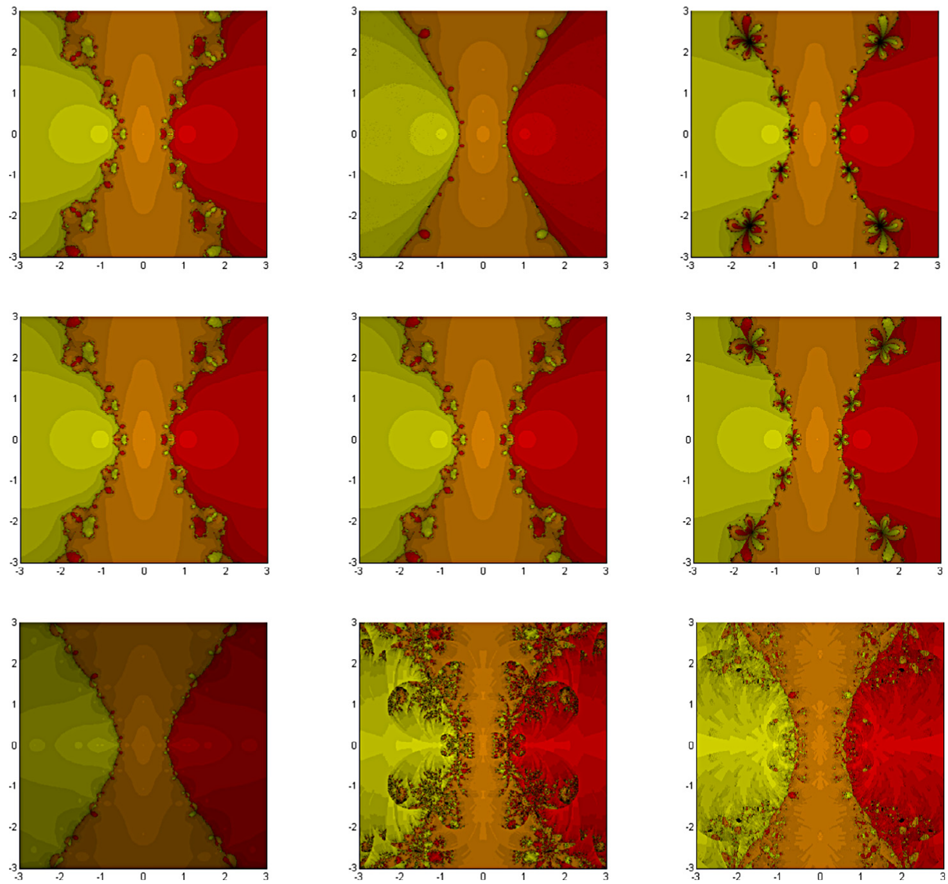

Fig. 8. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^3 - z)^4$.

where the function H_f depends on x_n and other intermediate values. In Tables 1 and 2 we list the function H_f for each of the methods discussed here (see Tables 1 and 2).

It is clear that if x_n is a zero of the function $f(x)$ then x_n is a fixed point of the iterative method (66). But even if x_n is a zero of H_f and not of $f(x)$ it is a fixed point. Those fixed points that are zeros of H_f and not of $f(x)$ are called extraneous fixed points. For example, Schröder's method does **not** have any extraneous fixed point, since $H_f = 1$. In order to find the extraneous fixed points, we substitute the quadratic polynomial $(z^2 - 1)^m$ for $f(z)$ and then find the zeros of H_f. See Table 3 for the extraneous fixed points for each method.

In our previous work, we found that methods without extraneous fixed point or those having such points on the imaginary axis perform better than others. For families of methods, we showed how to choose the parameter(s) such that the extraneous fixed points are on or close to the imaginary axis. When a method contains a weight function, we suggested a rational function as a weight function. This leading to a family of methods with at least one parameter. We also demonstrated that a polynomial weight function does not give as good results.

To choose the parameters in the methods, the following criterion can be used, which was developed in [24] and is defined below.

Let $E = \{z_1, z_2, \ldots, z_n\}$ be the set of the extraneous fixed points corresponding to the values given to the parameters. We define

$$d = \max_{z_i \in E} |Re(z_i)|. \tag{67}$$

We look for the parameters which attain the minimum of the function d given in (67).

For the method (20) the best value of $\theta = -0.2$ and for (21) the best parameter is $\theta = 1$ which is Dong2.

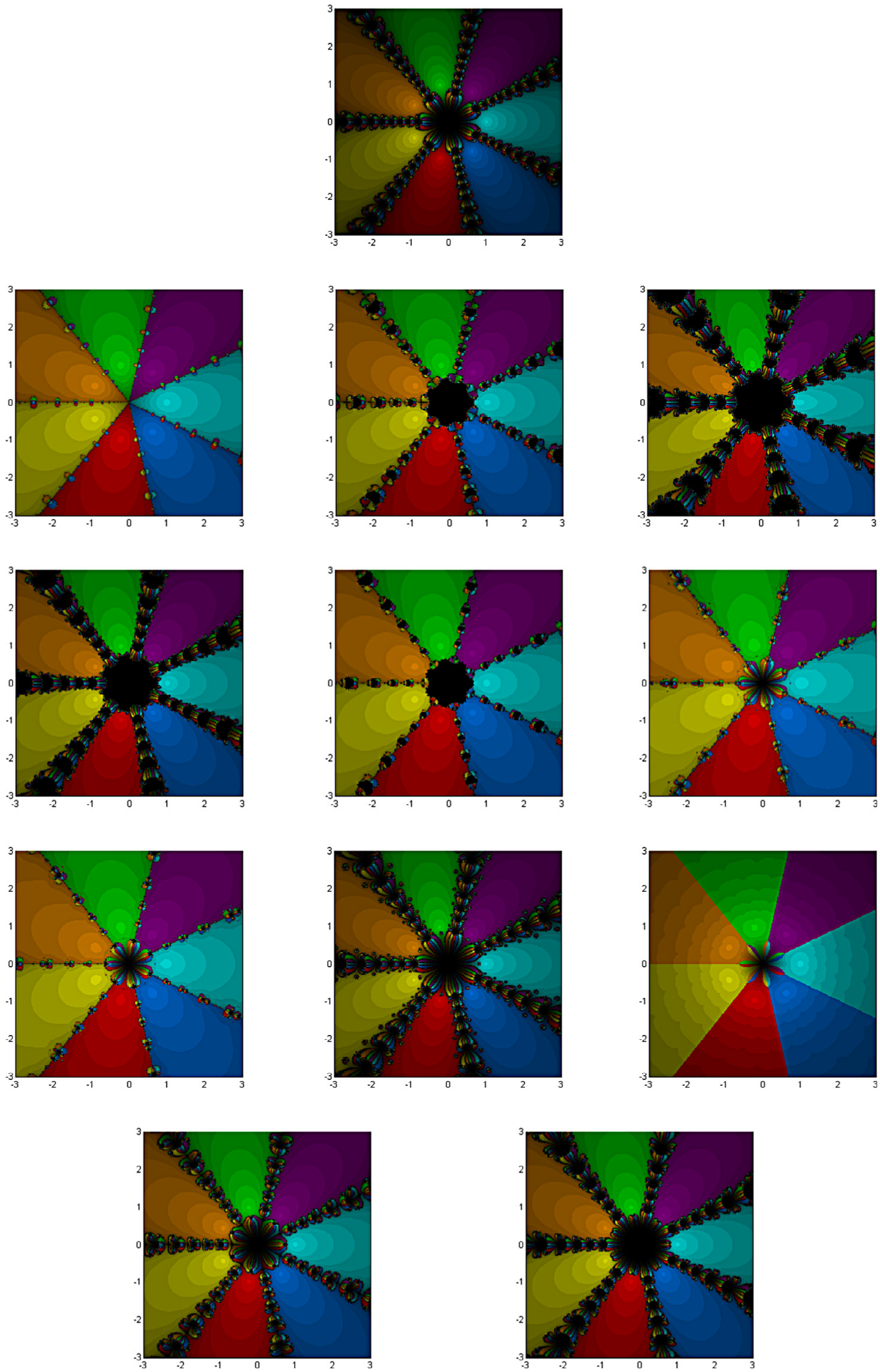

Fig. 9. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^7 - 1)^4$.

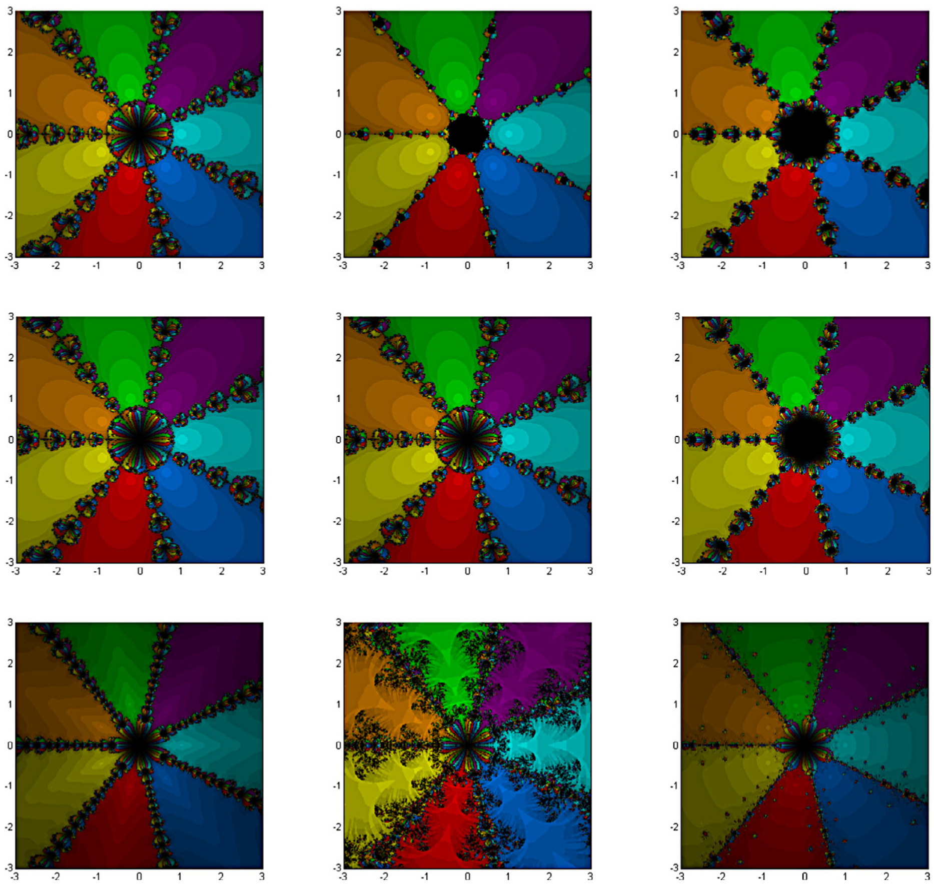

Fig. 10. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^7 - 1)^4$.

Remarks.

(1) The four methods LCN1–LCN4 [49] are not optimal as defined by Kung and Traub [64] and therefore will not be included here. Neta and Chun [28] have compared LCN5, LCN6, ZCS3, LZ11 and LZ12. They have shown that LCN6 and ZCS3 are best and therefore we will not include LCN5 and the methods developed by Liu and Zhou [54].
(2) Chun and Neta [30] found that KBK1 and KBK2 and ZCS1–ZCS3 cannot compete with LCN6 and they will not be included in the comparison.
(3) It was shown [30] that ZCS3 is just a rearrangement of LCN6 therefore ZCS3 will not be included here.
(4) Chun and Neta [31] have shown that out of the 4 members in Sbibih et al. [55], only SSTZ2 with $\mu = \frac{1}{3}$ is best. Therefore we will not use the other 3 members of that family here.

3. Numerical experiments

In this section, we detail the experiments we have used with each of the methods. All the examples have roots within a square of [−3,3] by [−3,3]. We have taken 360,000 equally spaced points in the square as initial points for the methods and we have registered the total number of iterations required to converge to a root and also to which root it converged. We have also collected the CPU time (in seconds) required to run each method on all the points using Dell Optiplex 990 desktop computer. We then computed the average number of function evaluations required per point and the number of points requiring 40 iterations.

Example 1. In our first example, we have taken the polynomial

$$p_1(z) = (z^2 - 1)^2 \tag{68}$$

whose roots $z = \pm 1$ are both real and of multiplicity $m = 2$.

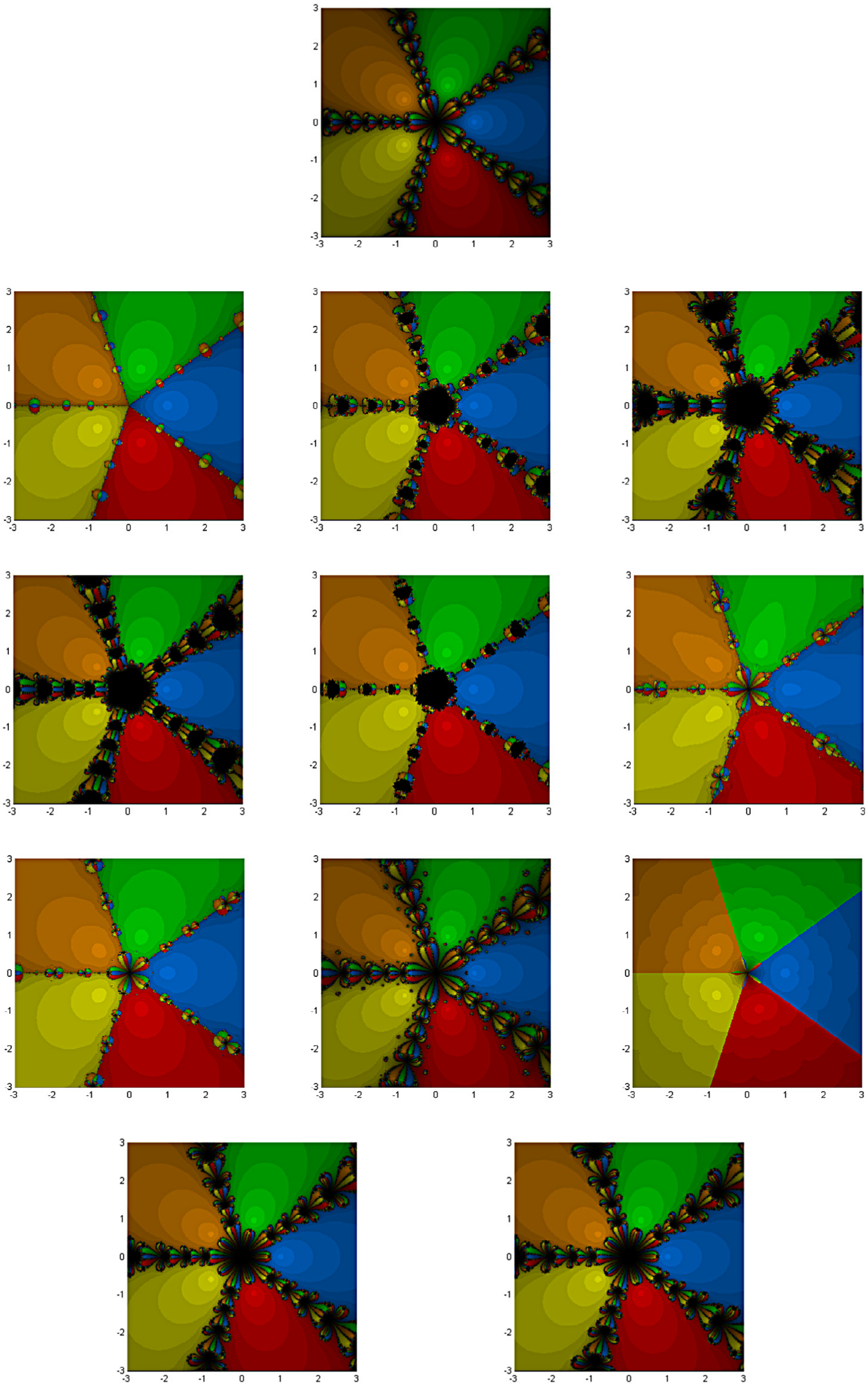

Fig. 11. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^5 - 1)^3$.

The basins for the 12 methods of order 2–3 are given in Fig. 1. Fig. 2 displays the basins for methods of order 4 and 6. The basin for each root is colored differently. The darker the shading, the higher is the number of function evaluations per

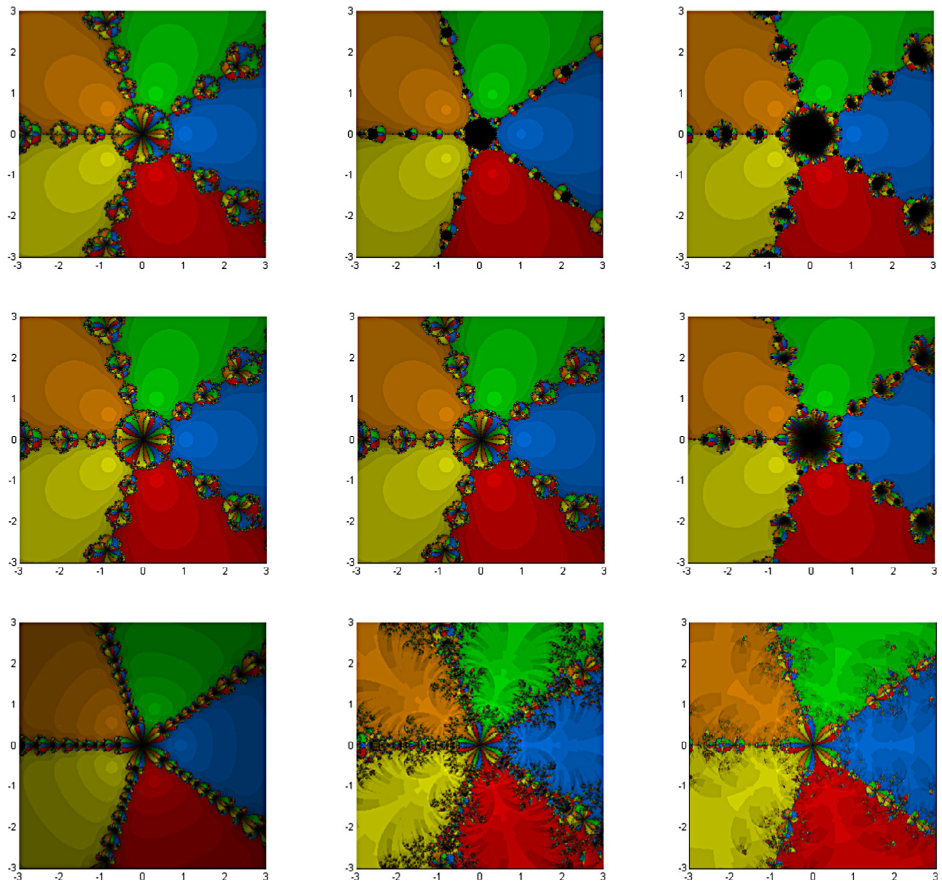

Fig. 12. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^5 - 1)^3$.

point on average. The reason we have used the number of function evaluations and not the number of iterations is because the methods require a different number of function evaluations per step. For example, Schröder's method uses 2 function evaluations per step, but Osada's method uses 3 function evaluations. The boundary between the two basins is a straight line for the following methods: Schröder (Fig. 1 top row), Halley (Fig. 1 second row left), Dong3 (Fig. 1 third row right), Dong4 (Fig. 1 fourth row left), Euler–Cauchy (Fig. 1 fourth row right) and SSTZ2 (Fig. 2 top row center). In order to have a more quantitative comparison, we have collected the number of function evaluations per point on average in Table 4, the CPU time in seconds required to get the method to run over all 601^2 initial points in the square containing the roots (Table 5) and the number of black points, i.e. those points for which the method did not converge after 40 iterations, in Table 6. The method using the lowest number of function evaluations is Euler–Cauchy (3.0) followed by GKN2C (10.04), GKN2A2 (10.19), GKN2A1 (10.24) and Dong4 (10.27), the highest is GKN4C (32.70). All other methods require between 11.11 and 15.31. The fastest methods are Euler–Cauchy (110.84 s), Schröder (151.48), Dong4 (166.53), GKN2A1 (170.19), Dong3 (171.82) and Halley (176.92). The slowest is the sixth order method GKN4C (891.41 s). It is surprising that the other sixth order method (GKN5YD) and the eighth order method (WI3X) are faster than some of the fourth order methods. The least number of black points (1) was achieved by Euler–Cauchy, GKN2A1, GKN2A2 and GKN2C. The highest number is for LCN6 and SB (10 289 points). Notice that Euler–Cauchy was best in all 3 measures for this example.

Example 2. The polynomial has the three roots of unity,

$$p_2(z) = (z^3 - 1)^2. \tag{69}$$

The basins are given in Figs. 3 and 4. Now the only one with straight line boundaries is Euler–Cauchy (Fig. 3 fourth row right). The least number of function evaluations per point on average was achieved by Euler–Cauchy, Dong4, GKN2C, GKN2A2 and GKN2A1 (in that order). The highest is GKN4C (35.78 function evaluations). The fastest methods are SSTZ2 (228.23), Schröder (252.85), Dong4 (255.97) and Halley (284.34). Euler–Cauchy is no longer among the fastest (422.97). The slowest is GKN4C (1147.76 s). The lowest number of black points (1) is for Dong3, Euler–Cauchy, GKN2A2, GKN2C and GKN4C. Five

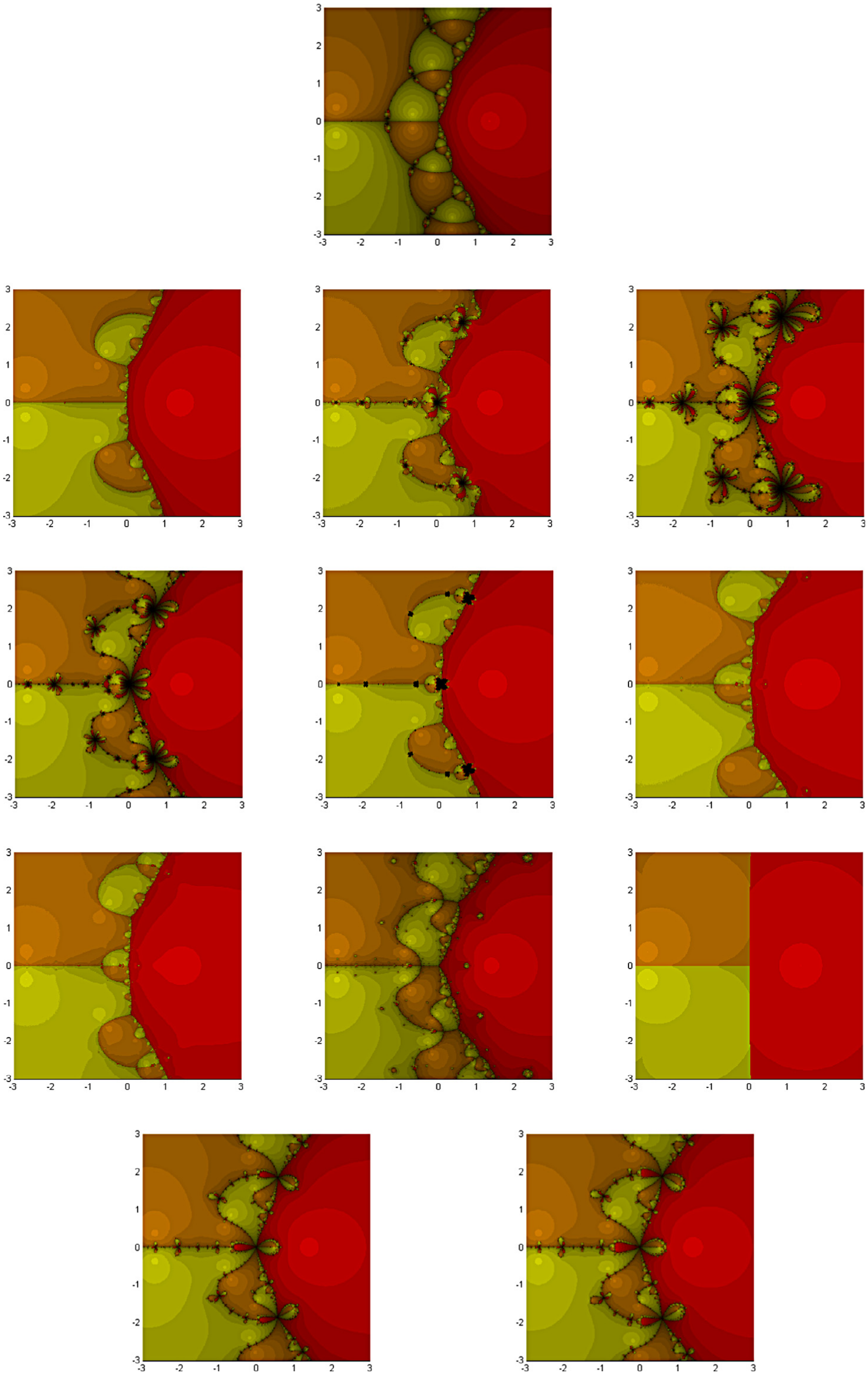

Fig. 13. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^3+4z^2-10)^3$.

other methods have less than 10 black points: Schröder (8), Halley (2), Osada (7), CN3 (5) and GKN2A1 (2). The worst are again LCN6 and SB with 26 951 points.

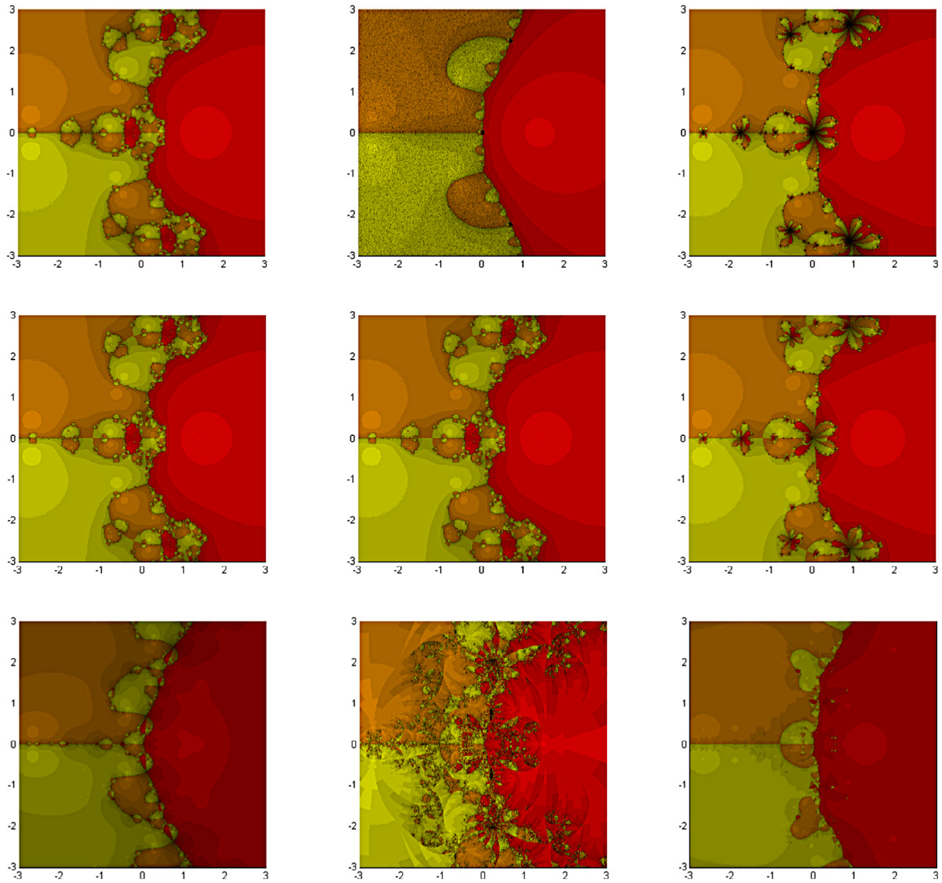

Fig. 14. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^3 + 4z^2 - 10)^3$.

Example 3. The third example is a polynomial whose roots are all of multiplicity four. The roots are the three roots of unity, i.e.

$$p_3(z) = (z^3 - 1)^4. \tag{70}$$

The basins are given in Figs. 5 and 6. The difference between this example and the previous one is the multiplicity. The best method is again Euler–Cauchy for which the boundaries are straight lines. The methods requiring the least number of function evaluations per point on average are Euler–Cauchy (11.43) followed by Dong3 (11.62) and Dong4 (12.30). The fastest methods are Dong3 (297.65), Schröder (306.87) and Dong4 (317.8). The slowest is GKN4C (3562.90 s). The least number of black points is achieved by Dong3, Osada, Euler–Cauchy, GKN2A2 and GKN4C. Three other methods have less than 10 black points, namely Halley (2), GKN2A1 (6) and Schröder (8). The highest number is for Dong2 (11 699 points).

Example 4. The fourth example is a polynomial whose roots are all of multiplicity four.

$$p_4(z) = (z^3 - z)^4. \tag{71}$$

The roots are $z = 0, \pm 1$. The basins are given in Figs. 7 and 8. This is harder even for Euler–Cauchy which shows a much smaller basin for the root in the origin. The least number of function evaluations was used by Euler–Cauchy (12.44) followed by Dong3 (12.84). The highest number (35.42) was required by GKN4C. Notice that in all these examples the sixth order method GKN5YD performed better than the other sixth order method, GKN4C. The fastest methods are Schröder, Dong3 and SSTZ2 and the slowest is as always GKN4C (3473.80 s). Twelve methods do not have black points: Schröder, Halley, Dong3, Osada, Euler–Cauchy, CN3, CBN1, LCN6, GKN2A1, GKN2A2, GKN2C and GKN4C.

Example 5. In our next example we took the polynomial

$$p_5(z) = (z^7 - 1)^4. \tag{72}$$

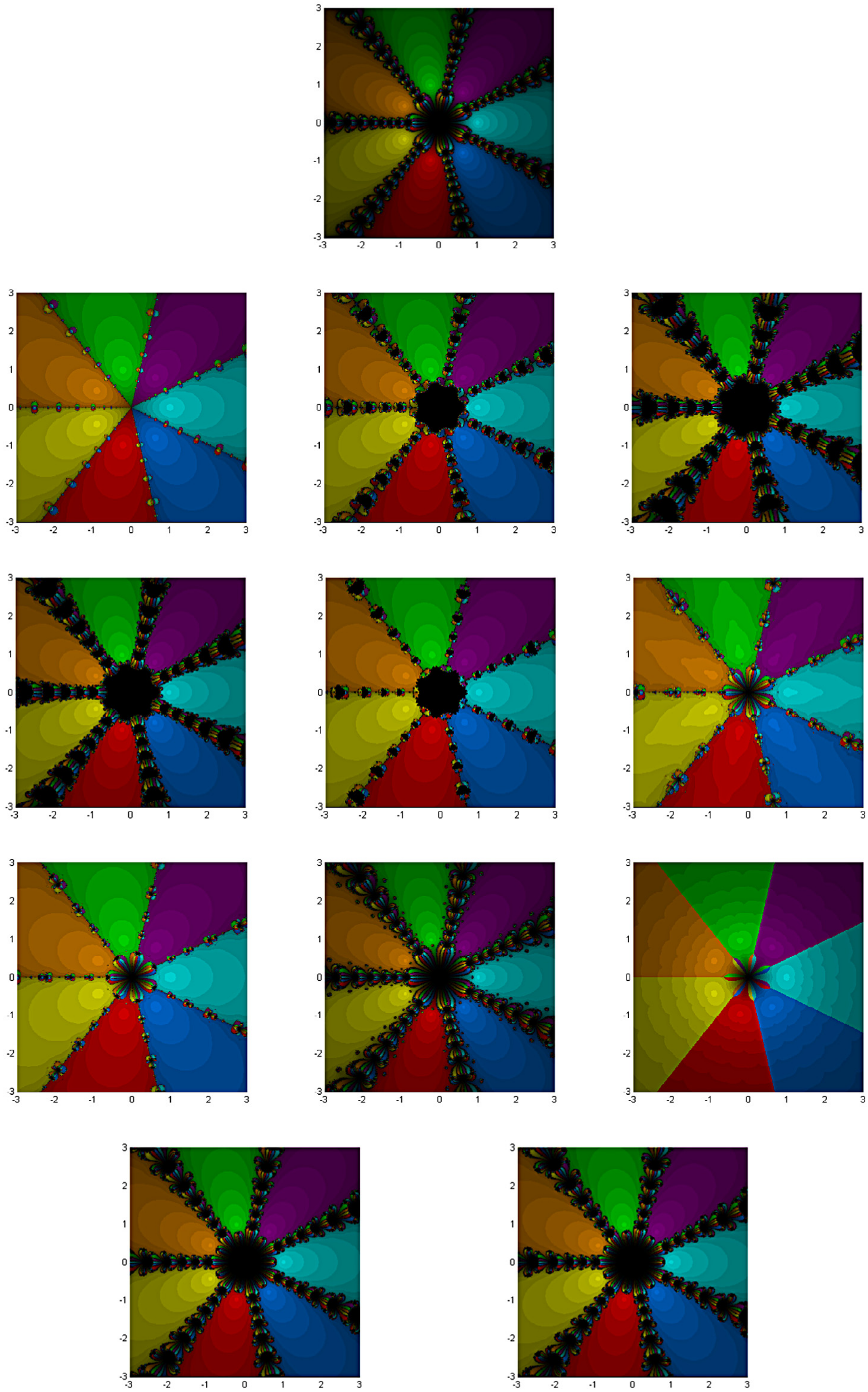

Fig. 15. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^7 - 1)^3$.

The seven roots of unity are all of multiplicity 4. The basins are plotted in Figs. 9 and 10. The best method is again Euler–Cauchy, since the boundaries are straight lines away from a neighborhood of the origin. Halley's method does not have so many black points near the origin as other schemes. the fastest method is Dong3 (644.78 s) followed by Dong4, SSTZ2, Halley

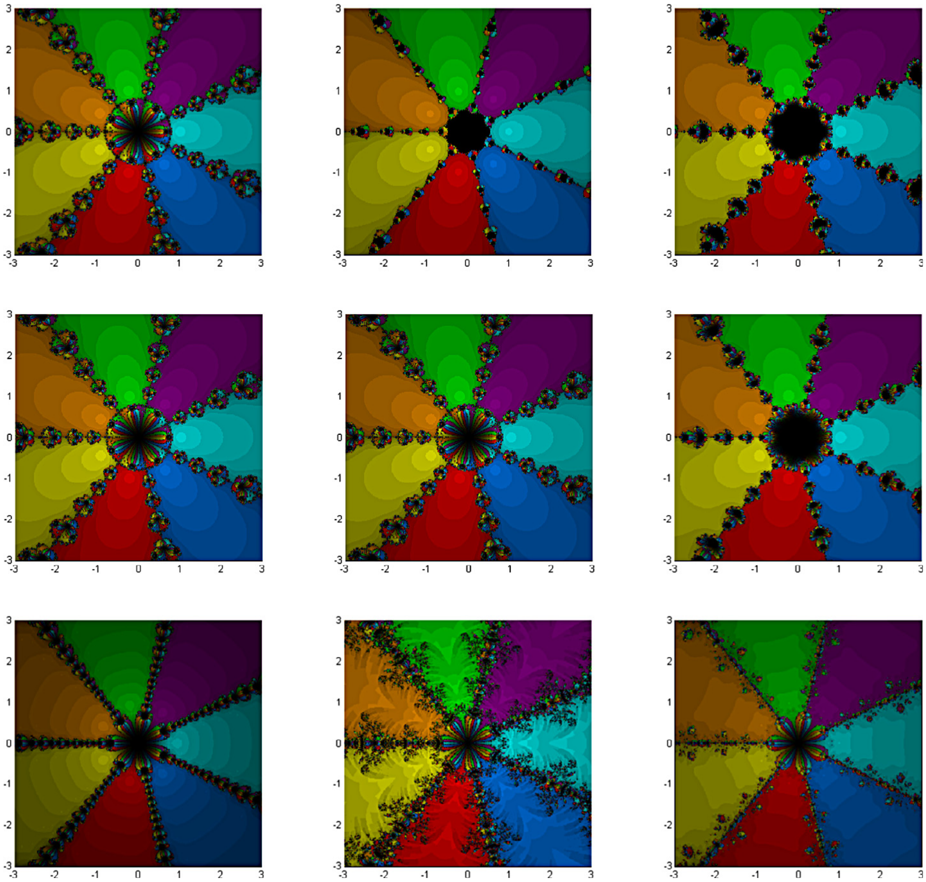

Fig. 16. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^7 - 1)^3$.

and Schröder. The least number of function evaluations is for Dong3 (16.58) and Dong4 (18.26). The least number of black point is for Halley (55) and Euler–Cauchy (69). The most number of black points is for N3 (65 295) and Dong1 (53 847).

Example 6.

$$p_6(z) = (z^5 - 1)^3. \tag{73}$$

The 5 roots of unity are all with multiplicity $m = 3$. The basins are displayed in Figs. 11 and 12. Again, the least number of function evaluations is for Dong3 followed by Dong4. In this case Euler–Cauchy comes fourth. Dong3 is the fastest followed by Dong4 and SSTZ2. In terms of black points, the best is Euler–Cauchy and WI3X (1) followed by Dong3 (3).

Example 7. Another example with 3 roots all with multiplicity 3 is:

$$p_7(z) = (z^3 + 4z^2 - 10)^3. \tag{74}$$

The basins are displayed in Figs. 13 and 14. The only method for which the boundaries are straight lines is Euler–Cauchy (Fig. 13, rightmost on the fourth row). Consulting Table 4, we find that Euler–Cauchy uses the least number of function evaluations per point on average (10.4) followed by Dong3 (11.13) and Dong4 (12.03). The worst in this sense is GKN4C (31.13). The fastest method is Schröder's method (322.77) followed by Dong3 (323.9) and the slowest is GKN4C (2505.16). The following four methods have only one black point: Euler–Cauchy, GKN2A1, GKN2A2 and GKN2C followed by GKN4C with 2 black points. All the others have at least 55 black points.

Example 8.

$$p_8(z) = (z^7 - 1)^3. \tag{75}$$

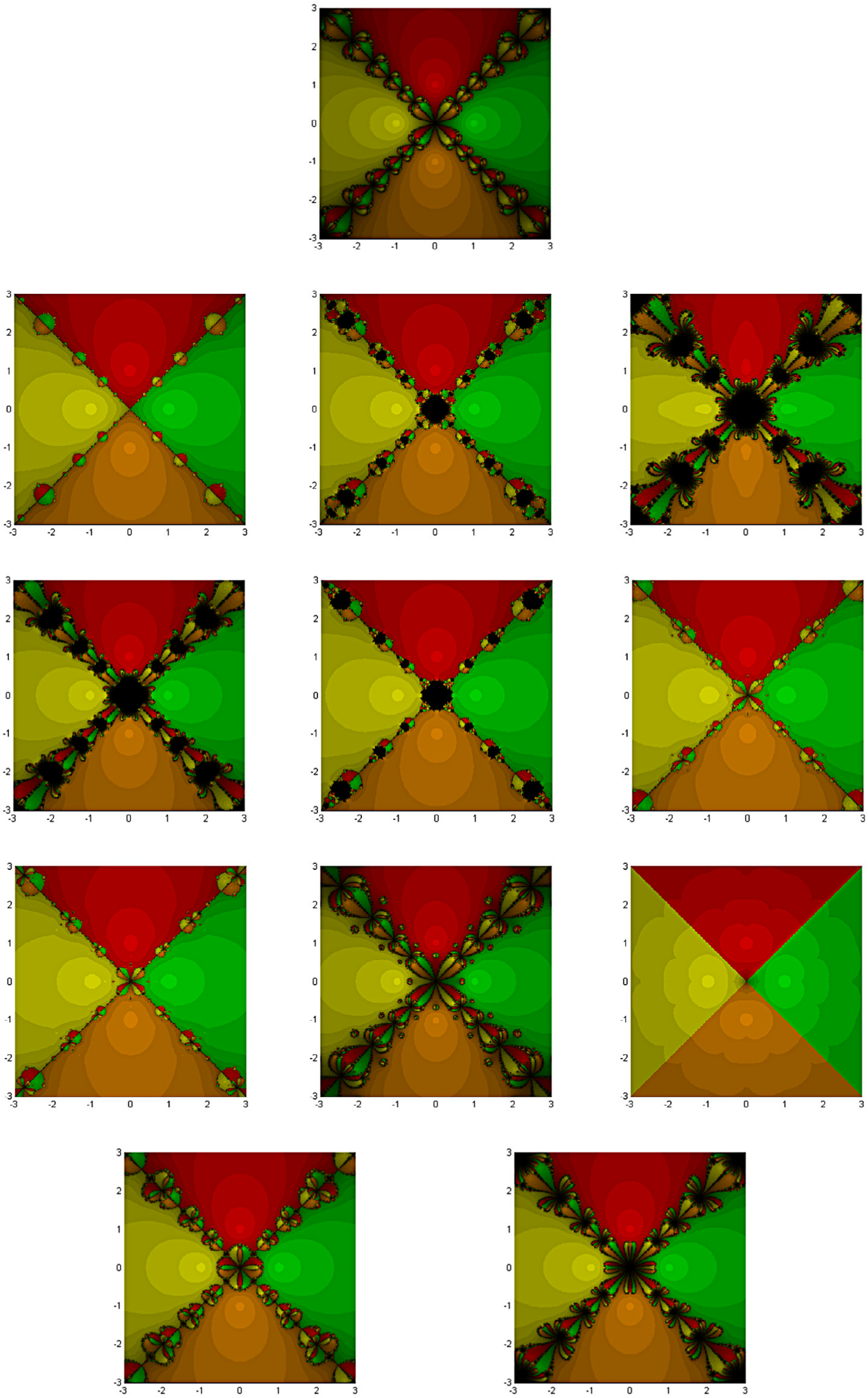

Fig. 17. The top row for Schröder's method. Second row for Halley (left), Victory–Neta (center) and N3 (right). Third row for Dong1 (left), Dong2 (center), and Dong3 (right). Fourth row for Dong4 (left), Osada (center), and Euler–Cauchy (right). Bottom row for CN3 (left) and CBN1 (right) for the roots of the polynomial $(z^4-1)^5$.

This example is similar to Example 5 except for the multiplicity. The basins are given in Figs. 15 and 16. The conclusions are identical. Therefore, we can conclude that the multiplicity does not affect the results.

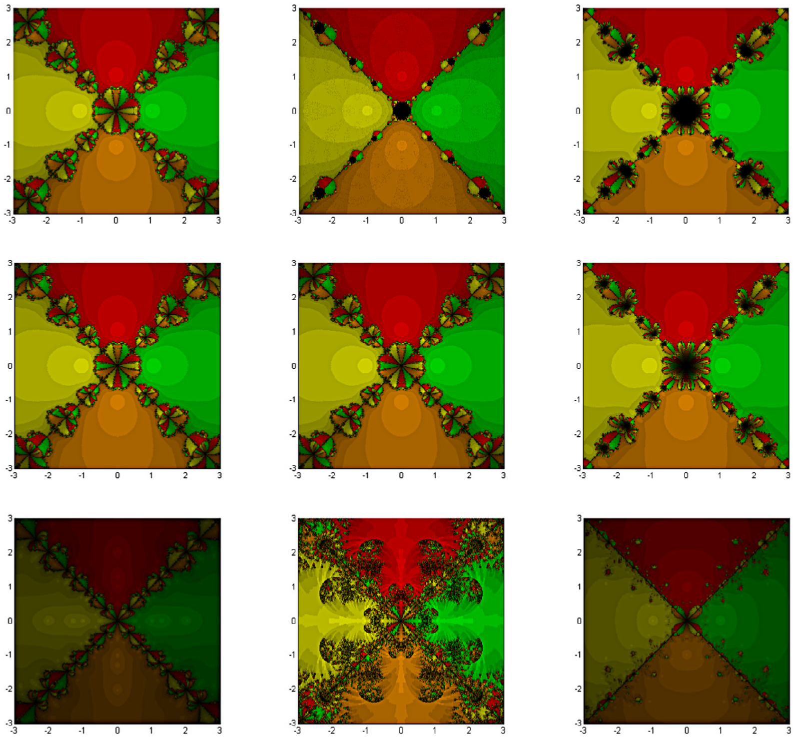

Fig. 18. The top row for LCN6 (left), SSTZ2 (center), and SB (right). Second row for GKN2A1 (left), GKN2A2 (center) and GKN2C (right). Bottom row for GKN4C (left), GKN5YD (center) and WI3X (right) for the roots of the polynomial $(z^4 - 1)^5$.

Table 6
Number of points requiring 40 iterations for each example (1–9) and each of the methods.

Method	Ex1	Ex2	Ex3	Ex4	Ex5	Ex6	Ex7	Ex8	Ex9	Average
Schröder	601	8	8	0	20 299	5158	175	20 301	2433	5443
Halley	601	2	2	0	55	20	91	54	1 201	225
Victory–Neta	603	2 125	2 771	50	24 492	19 371	135	29 705	15 029	10 476
N3	601	4 506	10 463	628	65 295	44 368	617	64 582	37 001	25 340
Dong1	601	3 922	7648	946	53 847	39 855	544	57 086	29 393	21 538
Dong2	2 729	18 953	11 699	1340	26 353	23 368	3560	29 107	21 593	15 411
Dong3	601	1	1	0	314	3	102	168	1 201	266
Dong4	603	139	105	12	2 210	1 152	105	2 324	1 697	927
Osada	601	7	1	0	16 949	3 285	93	17 726	1 793	4 495
Euler–Cauchy	1	1	1	0	69	1	1	69	1	16
CN3	601	5	22	0	7 523	10 800	72	29 221	1 241	5 498
CBN1	601	209	205	0	29 161	11 971	55	31 179	5 849	8 803
LCN6	10 289	26 951	11	0	3 158	229	93	2 957	1 225	4 990
SSTZ2	733	6 261	3 772	116	15 995	13 458	413	19 818	9 781	7 816
SB	10 289	26 951	2 612	128	26 499	16 871	154	28 233	11 833	13 730
GKN2A1	1	2	6	0	3 541	282	1	2 535	281	739
GKN2A2	1	1	1	0	3 428	315	1	2 619	225	732
GKN2C	1	1	17	0	15 179	3 667	1	14 109	1 585	3 840
GKN4C	601	1	1	0	7 595	1 128	2	7 736	817	1 987
GKN5YD	791	1 119	3 994	1702	29 887	5 563	658	14 618	10 465	7 644
WI3X	747	128	1	1024	1 729	1	61	602	1 225	613

Example 9. In our last polynomial example, we have taken a polynomial whose roots are ± 1 and $\pm i$ all of multiplicity 5

$$p_9(z) = (z^4 - 1)^5. \tag{76}$$

The basins are displayed in Figs. 17 and 18. Again, Euler–Cauchy is the only one with straight line boundaries. The least number of function evaluations per point is achieved by Dong3 (14.03), followed by Euler–Cauchy (14.22). Dong3 is also the

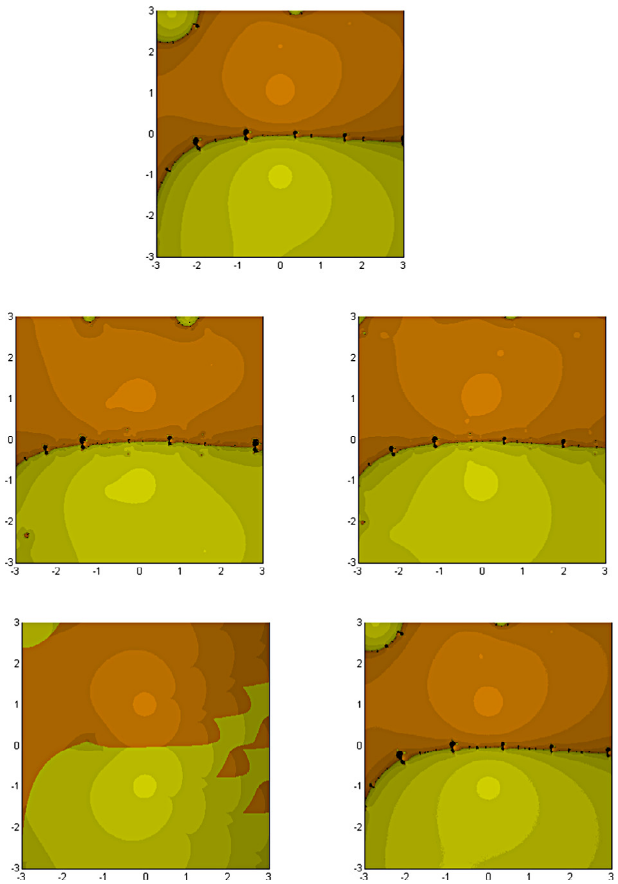

Fig. 19. The top row for Halley. The second row for Dong3 (left) and Dong4 (right). Bottom row for Euler–Cauchy (left) and SSTZ2 (right) for the roots of the function $(z-i)^3(e^{z+i}-1)^3$.

fastest (405.95 s) followed by Schröder's method with 454.57 s. Euler–Cauchy is the only method with one black point, all the other have at least 225 black points.

It is obvious from these 9 examples that Euler–Cauchy is the only one with straight line boundaries. This is important, since it says that from every point we approach the closest root. It is also the method with the least number of black points (16) when averaged across the 9 examples. Unfortunately it is not the fastest. Euler–Cauchy on average uses 664.18 s to run over all 601^2 initial points in the 6 by 6 square. The fastest is Dong3 (389.61) followed by Dong4 (417.96), SSTZ2 (442.70) and Schröder (448.43). The slowest is GKN4C with 3533.58 s. Euler–Cauchy uses 13.76 function evaluations per point with Dong3 slightly less (13.61). The worst is GKN4C (40.91). All other methods use between 14.0 and 27.48.

We now add a non-polynomial example. We ran the example on the top 3 methods for each category, namely: Halley, Dong3, Dong4, Euler–Cauchy and SSTZ2.

Example 10. The function used is

$$p_{10}(z) = (z-i)^3(e^{z+i}-1)^3 \tag{77}$$

whose roots are $\pm i$ all of multiplicity 3. The basins are displayed in Fig. 19. The best is Dong4 even though it has black points and Euler–Cauchy does not. We have collected the number of function evaluations per point, the CPU time in seconds and the number of black points in Table 7. Dong3 and Dong4 use the least number of function evaluations per point and SSTZ2 uses the most. The fastest is SSTZ2 (472.246 s) and the slowest is Euler–Cauchy (825.869 s). Euler–Cauchy is the only one with no black points followed by Dong4 (795) and Dong3 (1078).

4. Conclusions

Based on the 9 polynomial examples, we conclude that Dong3 was at the top 3 methods in the 3 categories. Euler–Cauchy and Dong4 were in the top 3 in two categories, but Euler–Cauchy is the only one that has straight line boundaries. Upon

Table 7
Results for Example 10.

Method	Number of function evaluation per point	CPU	Number of black points
Halley	12.47	608.731	1210
Dong3	10.95	549.685	1078
Dong4	11.03	573.584	795
Euler–Cauchy	12.67	825.869	0
SSTZ2	16.69	539.842	1378
WI3X	25.88	2225.07	579

considering the last example, we find that Dong3 is at the top based on the number of function evaluations per point, SSTZ2 is at the top based on the CPU time and Euler–Cauchy is at the top with no black point. Since Dong3 and Dong4 were at the top 3 in the 9 polynomial examples and in one category for the last example, we recommend them along with Euler–Cauchy (no black points) and SSTZ2 (fastest).

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Republic of Korea (NRF-2016R1D1A1A09917373).

References

[1] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1964.
[2] M.S. Petković, B. Neta, L.D. Petković, J. Džunić, Multipoint Methods for Solving Nonlinear Equations, Elsevier, 2013.
[3] B.D. Stewart, Attractor Basins of Various Root-Finding Methods (M.S. thesis), Naval Postgraduate School, Department of Applied Mathematics, Monterey, CA, 2001.
[4] S. Amat, S. Busquier, S. Plaza, Review of some iterative root-finding methods from a dynamical point of view, Scientia 10 (2004) 3–35.
[5] S. Amat, S. Busquier, S. Plaza, Dynamics of a family of third-order iterative methods that do not require using second derivatives, Appl. Math. Comput. 154 (2004) 735–746.
[6] S. Amat, S. Busquier, S. Plaza, Dynamics of the King and Jarratt iterations, Aequationes Math. 69 (2005) 212–2236.
[7] M. Scott, M.B. Neta, C. Chun, Basin attractors for various methods, Appl. Math. Comput. 218 (2011) 2584–2599.
[8] F. Chicharro, A. Cordero, J.M. Gutiérrez, J.R. Torregrosa, Complex dynamics of derivative-free methods for nonlinear equations, Appl. Math. Comput. 219 (2013) 7023–7035.
[9] C. Chun, M.Y. Lee, B. Neta, J. Džunić, On optimal fourth-order iterative methods free from second derivative and their dynamics, Appl. Math. Comput. 218 (2012) 6427–6438.
[10] C. Chun, B. Neta, Sujin Kim, On Jarratt's family of optimal fourth-order iterative methods and their dynamics, Fractals 22 (2014) 1450013. http://dx.doi.org/10.1142/S0218348X14500133. (16 pages).
[11] C. Chun, B. Neta, On the new family of optimal eighth order methods developed by Lotfi et al., Numer. Algorithms 72 (2016) 363–376.
[12] C. Chun, B. Neta, Comparison of several families of optimal eighth order methods, Appl. Math. Comput. 274 (2016) 762–773.
[13] A. Cordero, J. García-Maimó, J.R. Torregrosa, M.P. Vassileva, P. Vindel, Chaos in King's iterative family, Appl. Math. Lett. 26 (2013) 842–848.
[14] B. Neta, M. Scott, C. Chun, Basin of attractions for several methods to find simple roots of nonlinear equations, Appl. Math. Comput. 218 (2012) 10548–10556.
[15] B. Neta, C. Chun, M. Scott, Basins of attractions for optimal eighth order methods to find simple roots of nonlinear equations, Appl. Math. Comput. 227 (2014) 567–592.
[16] I.K. Argyros, A.A. Magreñan, On the convergence of an optimal fourth-order family of methods and its dynamics, Appl. Math. Comput. 252 (2015) 336–346.
[17] A.A. Magreñan, Different anomalies in a Jarratt family of iterative root-finding methods, Appl. Math. Comput. 233 (2014) 29–38.
[18] Y.H. Geum, Y.I. Kim, B. Neta, On developing a higher-order family of double-Newton methods with a bivariate weighting function, Appl. Math. Comput. 254 (2015) 277–290.
[19] Y.H. Geum, Y.I. Kim, B. Neta, A class of two-point sixth-order multiple-zero finders of modified double-Newton type and their dynamics, Appl. Math. Comput. 270 (2015) 387–400.
[20] Y.H. Geum, Y.I. Kim, B. Neta, A family of optimal quartic-order multiple-zero finders with a weight function of the principal kth root of a derivative-to-derivative ratio and their basins of attraction, Math. Comput. Simulation 136 (2017) 1–21.
[21] Y.H. Geum, Y.I. Kim, B. Neta, A sixth-order family of three-point modified Newton-like multiple-zero finders and the dynamics behind their extraneous fixed points, Appl. Math. Comput. 283 (2016) 120–140.
[22] C. Chun, B. Neta, The basins of attraction of Murakami's fifth order family of methods, Appl. Numer. Math. 110 (2016) 14–25.
[23] C. Chun, B. Neta, An analysis of a new family of eighth-order optimal methods, Appl. Math. Comput. 245 (2014) 86–107.
[24] C. Chun, B. Neta, An analysis of a King-based family of optimal eighth-order methods, Amer. J. Algorithms Comput. 2 (2015) 1–17.
[25] C. Chun, B. Neta, J. Kozdon, M. Scott, Choosing weight functions in iterative methods for simple roots, Appl. Math. Comput. 227 (2014) 788–800.
[26] B. Neta, M. Scott, C. Chun, Basin attractors for various methods for multiple roots, Appl. Math. Comput. 218 (2012) 5043–5066.
[27] B. Neta, C. Chun, On a family of Laguerre methods to find multiple roots of nonlinear equations, Appl. Math. Comput. 219 (2013) 10987–11004.
[28] B. Neta, C. Chun, Basins of attraction for several optimal fourth order methods for multiple roots, Math. Comput. Simulation 103 (2014) 39–59.
[29] B. Neta, C. Chun, Basins of attraction for Zhou-Chen-Song fourth order family of methods for multiple roots, Math. Comput. Simulation 109 (2015) 74–91.
[30] C. Chun, B. Neta, Comparing the basins of attraction for Kanwar-Bhatia-Kansal family to the best fourth order method, Appl. Math. Comput. 266 (2015) 277–292.
[31] C. Chun, B. Neta, Basin of attraction for several third order methods to find multiple roots of nonlinear equations, Appl. Math. Comput. 268 (2015) 129–137.

[32] W. Werner, Iterationsverfahren höherer ordnung zur lösung nicht linearer gleichungen, ZAMM Z. Angew. Math. Mech. 61 (1981) T322–T324.
[33] E. Schröder, Über unendlich viele algorithmen zur auflösung der gleichungen, Math. Ann. 2 (1870) 317–365.
[34] L.B. Rall, Convergence of the Newton process to multiple solutions, Numer. Math. 9 (1966) 23–37.
[35] E. Halley, A new, exact and easy method of finding the roots of equations generally and that without any previous reduction, Philos. Trans. R. Soc. Lond. 18 (1694) 136–148.
[36] E. Hansen, M. Patrick, A family of root finding methods, Numer. Math. 27 (1977) 257–269.
[37] H.D. Victory, B. Neta, A higher order method for multiple zeros of nonlinear functions, Int. J. Comput. Math. 12 (1983) 329–335.
[38] B. Neta, New third order nonlinear solvers for multiple roots, Appl. Math. Comput. 202 (2008) 162–170.
[39] V.F. Candela, A. Marquina, Recurrence relations for rational cubic methods II: The Chebyshev method, Computing 45 (1990) 355–367.
[40] D.J. Hofsommer, Note on the computation of the zeros of functions satisfying a second order differential equation, Math. Tables Other Aids Comput. 12 (1958) 58–60.
[41] D.B. Popovski, A family of one point iteration formulae for finding roots, Int. J. Comput. Math. 8 (1980) 85–88.
[42] C. Dong, A basic theorem of constructing an iterative formula of the higher order for computing multiple roots of an equation, Math. Numer. Sin. 11 (1982) 445–450.
[43] C. Dong, A family of multipoint iterative functions for finding multiple roots of equations, Int. J. Comput. Math. 21 (1987) 363–367.
[44] N. Osada, An optimal multiple root-finding method of order three, J. Comput. Appl. Math. 51 (1994) 131–133.
[45] E. Bodewig, Sur la méthode Laguerre pour l'approximation des racines de certaines équations algébriques et sur la critique d'Hermite, Indag. Math. 8 (1946) 570–580.
[46] L.D. Petkovi'c, M.S. Petković, D. Živković, Hansen-Patrick's family is of Laguerre's type, Novi Sad J. Math. 33 (2003) 109–115.
[47] C. Chun, B. Neta, A third-order modification of Newton's method for multiple roots, Appl. Math. Comput. 211 (2009) 474–479.
[48] C. Chun, H. Bae, B. Neta, New families of nonlinear third-order solvers for finiding multiple roots, Comput. Math. Appl. 57 (2009) 1574–1582.
[49] S.G. Li, L.Z. Cheng, B. Neta, Some fourth-order nonlinear solvers with closed formulae for multiple roots, Comput. Math. Appl. 59 (2010) 126–135.
[50] B. Neta, A.N. Johnson, High order nonlinear solver for multiple roots, Comput. Math. Appl. 55 (2008) 2012–2017.
[51] B. Neta, Extension of Murakami's High order nonlinear solver to multiple roots, Int. J. Comput. Math. 8 (2010) 1023–1031.
[52] V. Kanwar, S. Bhatia, M. Kansal, New optimal class of higher-order methods for multiple roots, permitting $f'(x_n) = 0$, Appl. Math. Comput. 222 (2013) 564–574.
[53] X. Zhou, X. Chen, Y. Song, Construction of higher order methods for multiple roots of nonlinear equations, J. Comput. Appl. Math. 235 (2011) 4199–4206.
[54] B. Liu, X. Zhou, A new family of fourth-order methods for multiple roots of nonlinear equations, Nonlinear Anal. Model. Control 18 (2) (2013) 143–152.
[55] D. Sbibih, A. Serghini, A. Tijini, A. Zidna, A general family of third order method for finding multiple roots, Appl. Math. Comput. 233 (2014) 338–350.
[56] H.H.H. Homeier, On Newton-type methods for multiple roots with cubic convergence, J. Comput. Appl. Math. 231 (2009) 249–254.
[57] Y.H. Geum, Y.I. Kim, Cubic convergence of parameter-controlled Newton-secant method for multiple zeros, J. Comput. Appl. Math. 233 (2009) 931–937.
[58] Y.I. Kim, Y.H. Geum, A cubic-order variant of Newton's method for finding multiple roots of nonlinear equations, Comput. Math. Appl. 62 (2011) 249–254.
[59] F. Soleymani, D.K.R. Babajee, Computing multiple zeros using a class of quartically convergent methods, Alexandria Eng. J. 52 (2013) 531–541.
[60] L.V. Ahlfors, Complex Analysis, McGraw-Hill Book, Inc., 1979.
[61] S. Wolfram, The Mathematica Book, fifth ed., Wolfram Media, 2003.
[62] L. Hörmander, An Introduction to Complex Analysis in Several Variables, North-Holland Pub. Co, 1973.
[63] Y.H. Geum, Y.I. Kim, B. Neta, Constructing a family of optimal eighth-order modified Newton-type multiple-zero finders along with the dynamics behind their purely imaginary extraneous fixed points, J. Comput. Appl. Math. 333 (2018) 131–156.
[64] H.T. Kung, J.F. Traub, Optimal order of one-point and multipoint iterations, J. Assoc. Comput. Mach. 21 (1974) 643–651.