A Family of Iterative Methods for Simultaneous Computing of All Zeros of Algebraic Equation

Gyurhan H. Nedzhibov and Milko G. Petkov
Laboratory of Mathematical Modelling,
Faculty of Mathematics and Informatics,
Shumen University, Shumen 9712, Bulgaria
e-mail: gyurhan@shu-bg.net

Abstract

This study deals with a family of multi-point iterative methods of arbitrary order of convergence for simultaneous computing of all roots of an algebraic equation. These methods are analogues of the well known method for computing of a single root of a nonlinear equation. Some known methods for simultaneous finding of all roots of algebraic equations are special cases of the family considered. Numerical examples are provided.

Keywords: Simultaneous iterative methods, Multi-point iterative methods, Order of convergence, Algebraic equation.

1. Introduction

In [1] J. F. Traub pointed out the following family of iterative methods for computing of a root of a nonlinear equation of the form $f(x)=0$:

$$x_{k+1}=\lambda_j(x_k), \tag{1.1}$$

where $k=0,1,2,...$

and

$$\lambda_j(x)=\lambda_{j-1}(x)-\frac{f(\lambda_{j-1}(x))}{f'(x)},\quad j=2,3,..,p;\ \lambda_1(x)=x. \tag{1.2}$$

To each j the function $\lambda_j(x)$ is a member of this family with order of convergence j (see [2]).

Some special cases:

1) For $j=2$ we receive the iteration function

$$\lambda_2(x)=x-u(x), \tag{1.3}$$

where $u(x)=\dfrac{f(x)}{f'(x)}$,

It is the iteration function of the well known *Newton's method* with order of convergence 2;

2) For $j=3$ the corresponding iteration function is

$$\lambda_3(x)=x-u(x)-\frac{f(x-u(x))}{f'(x)}, \tag{1.4}$$

where $u(x)=\dfrac{f(x)}{f'(x)}$,

with order of convergence 3 (see [4]);

3) For $j=4$ the iteration function is

$$\lambda_4(x)=x-u(x)-\frac{f(x-u(x))}{f'(x)}-\frac{f\left(x-u(x)-\dfrac{f(x-u(x))}{f'(x)}\right)}{f'(x)}, \tag{1.5}$$

with order of convergence 4.

In general, in [1] it is proven the following

Theorem 1. *If the iteration function* $\varphi(x)$ *has an order of convergence* p*, then the iterative function*

$$\psi(x) = \varphi(x) - \frac{f(\varphi(x))}{f'(x)}$$

has an order of convergence (p+1).

It is clear that the family (1.2) can be considered as a generalization of the *Newton's method.* Our purpose further is to propose and investigate some analogues of (1.2) for simultaneous computing of all the roots of ana algebraic equation.

2. A family of iterative formulae for simultaneous root-finding of algebraic equations

Let us suppose that $f(x)$ is an algebraic polynomial with simple real or complex roots $\xi_1, \xi_2, .., \xi_n$ with the corresponding approximations $x_1, x_2, .., x_н$. Then the well known iterative procedure of *Weierstrass-Dochev-Kerner* (WDK-formula) for simultaneous computing of the all roots ξ_i is

$$\hat{x}_i = x_i - W_i \ , \quad i = 1,2,... \tag{2.1}$$

where

$$W_i = \frac{f(x_i)}{\prod_{s \neq i}(x_i - x_s)} \ , \quad x_i \neq x_s \text{ for } i \neq s \ . \tag{2.2}$$

Dochev was the first to prove that this iterative procedure has the order of convergence 2 (see [2,3]). The iterative procedure (2.1)-(2.2) may be consider as a modification of *Newton's method* if $f'(x_i)$ is replaced by $\prod_{j \neq i}(x_i - x_j)$.

Now we will propose an analogue of iteration process (1.2) for simultaneous computing of all roots ξ_i for which (2.1)-(2.2) is a special case. For this end let $x = (x_1, x_2, .., x_н)$ and

$$\lambda_{j,i}(x) = \lambda_{j-1,i}(x) - \frac{f(\lambda_{j-1,i}(x))}{\prod_{s \neq i}^{n}(x_i - x_s)} , \tag{2.3}$$

where $i = 1,..,n$; $j = 2,3,4,...$ and $\lambda_{1,i}(x) = x_i$.

Then the following basic theorem holds

Theorem 2. *If* $x_1^0, x_2^0, .., x_n^0$ *are sufficiently good initial approximations corresponding to the roots* $\xi_1, \xi_2, .., \xi_n$*, then the iteration process*

$$x_i^{k+1} = \lambda_{j,i}(x^k), \qquad k = 0,1,2,... \tag{2.4}$$

converges to ξ_i *for all* $i = 1,2,..,n$ *with order of convergence* j.

3. Proof of the basic theorem

We prove the Theorem 2. by induction with respect to j and with fixed i. Let us denote $\varepsilon_i = x_i^k - \xi_i$.

1) For $j = 2$ we receive the process (2.1)-(2.2) which as it is well known has the order of convergence 2.

2) Let now assume that the corresponding iteration formula $x_i^{k+1} = \lambda_{m,i}(x^k)$ for $j = m$ has order of convergence *m*, i.e. $\lambda_{m,i}(x) - \xi_i = O(\varepsilon_i^m)$ for $i = 1,..,n$.
3) Further, we will prove that the iteration function $x_i^{k+1} = \lambda_{m+1,i}(x^k)$ has order of convergence (*m*+1).

If we denote $\varepsilon = \max_i |\varepsilon_i|$ then from assumption 2) follows that

$$\lambda_{m,i}(x) - \xi_i = O(\varepsilon^m). \tag{3.1}$$

Further for simplicity of notification we will use of $x_i = x_i^k$ for $i = 1,..,n$ and $k = 1,2,...$.
Let us consider

$$\lambda_{m+1,i}(x) - \xi_i = \lambda_{m,i}(x) - \frac{f(\lambda_{m,i}(x))}{\prod_{s\neq i}^n (x_i - x_s)} - \xi_i = \lambda_{m,i}(x) - \xi_i - \frac{\prod_{s=1}^n (\lambda_{m,i}(x) - \xi_s)}{\prod_{s\neq i}^n (x_i - x_s)}$$

$$= \lambda_{m,i}(x) - \xi_i - (\lambda_{m,i}(x) - \xi_i)\prod_{s\neq i}^n \frac{\lambda_{m,i}(x) - \xi_s}{x_i - x_s}$$

$$= (\lambda_{m,i}(x) - \xi_i)\left(1 - \prod_{s\neq i}^n \frac{\lambda_{m,i}(x) - \xi_s}{x_i - x_s}\right),$$

i.e. we have

$$\lambda_{m+1,i}(x) - \xi_i = (\lambda_{m,i}(x) - \xi_i)\left(1 - \prod_{s\neq i}^n \frac{\lambda_{m,i}(x) - \xi_s}{x_i - x_s}\right). \tag{3.2}$$

We put $K_m(x_i) = x_i - \lambda_{m,i}(x)$,(for $m = 1,2,...$). After substituting in (3.2), we obtain

$$\lambda_{m+1,i}(x) - \xi_i = (\lambda_{m,i}(x) - \xi_i)\left(1 - \prod_{s\neq i}^n \frac{x_i - K_m(x_i) - \xi_s}{x_i - x_s}\right)$$

$$= (\lambda_{m,i}(x) - \xi_i)\left(1 - \prod_{s\neq i}^n \frac{x_i - (K_m(x_i) + \xi_s)}{x_i - x_s}\right)$$

$$= (\lambda_{m,i}(x) - \xi_i)\sum_{j\neq i}^n \frac{x_i - (K_m(x_i) + \xi_j)}{x_i - x_s}\prod_{s\neq i}^{j-1} \frac{x_i - (K_m(x_i) + \xi_s)}{x_i - x_s},$$

and we find

$$\lambda_{m+1,i}(x) - \xi_i = (\lambda_{m,i}(x) - \xi_i)\sum_{j\neq i}^n \frac{\varepsilon_j - K_m(x_i)}{x_i - x_s}\prod_{s\neq i}^{j-1} \frac{x_i - (K_m(x_i) + \xi_s)}{x_i - x_s}.$$

From the expression of $K_m(x_i)$ it is easy to verify that $K_m(x_i) = O(\varepsilon_i)$. Therefore we have

$$\lambda_{m+1,i}(x) - \xi_i = (\lambda_{m,i}(x) - \xi_i)\sum_{j\neq i}^n \frac{\varepsilon_j - O(\varepsilon_i)}{x_i - x_s}\prod_{s\neq i}^{j-1} \frac{x_i - (K_m(x_i) + \xi_s)}{x_i - x_s}. \tag{3.3}$$

For the addends participating in the sum of expression (3.3) the following performance holds $\frac{\varepsilon_j - O(\varepsilon_i)}{x_i - x_s} = O(\varepsilon)$, as $x_i \neq x_s$ for $i \neq s$. Using the equation (3.1) from (3.3) we obtain

$$\lambda_{m+1,i}(x) - \xi_i = O(\varepsilon^m)O(\varepsilon)\prod_{s\neq i}^{j-1} \frac{x_i - (K_m(x_i) + \xi_s)}{x_i - x_s}.$$

Finally we receive the expression

$$\lambda_{m+1,i}(x)-\xi_i = O(\varepsilon^{m+1}),$$

which proves the theorem

4. Some special cases of the proposed family

1) For $j=2$ we get the formula (2.1)-(2.2) corresponding to the formula (1.3) of second order.
2) For $j=3$ we have

$$\hat{x}_i = x_i - W_i - \frac{f(x_i - W_i)}{\prod_{s\neq i}^{n}(x_i - x_s)}, \ i=1,..,n \tag{4.1}$$

corresponding to (1.4).

3) For $j=4$ we have

$$\hat{x}_i = x_i - W_i - \frac{f(x_i - W_i)}{\prod_{s\neq i}^{n}(x_i - x_s)} - \frac{f\left(x_i - W_i - \dfrac{f(x_i - W_i)}{\prod_{s\neq i}^{n}(x_i - x_s)}\right)}{\prod_{s\neq i}^{n}(x_i - x_s)}, \tag{4.2}$$

where $W_i = \dfrac{f(x_i)}{\prod_{s\neq i}^{n}(x_i - x_s)}$, $i=1,..,n$, corresponding to formula (1.5).

By this way from the family (2.3) we can generate recurrently simultaneous formulas corresponding to each one formula from the family (1.2).

5. Numerical examples.

We have done numerical experiments with two examples of algebraic polynomial. All programs are realized in MATLAB's environment. We compare the observed iterative formulas on the following criterions: *number of iteration steps* and *absolute error.*

We use the following stopping criterions for the computer programs:

1) $|x_i^{k+1} - x_i^k| < \sqrt{\varepsilon}$;

2) $|f(x_i^k)| < \sqrt{\varepsilon}$, $(i=1,..,n)$,

where $\varepsilon = 2.22e-16$ is a constant of MATLAB.

With purpose of the comparison we have used in the experiments the well known Ehrlich-Aberth method (see [2]):

$$\hat{x}_i = x_i - \frac{f(x_i)}{f'(x_i) - f(x_i)\sum_{j\neq i}^{n}\dfrac{1}{x_i - x_j}}, \tag{5.1}$$

which is of third order of convergence.

We introduce the notations: x_i^0 - the initial points; *iter* – number of iterative steps; *err* – the absolute error computed by formula $\max|x_i^k - \xi_i|$; WDK – formula (2.1)-(2.2); EA – formula (5.1).

Example 1. Consider the algebraic polynomial $f(x)=x(x+1)(x+5)(x-3)$, with roots $\xi_1=-5, \xi_2=-1, \xi_3=0, \xi_4=3$. The results are included in the following table:

Table 1.

$x^0 = (-5.7,-1.6,-0.5,2.4)$				
	WDK	EA	(4.1)	(4.2)
iter	6	4	4	3
err	3.5e-14	4.2e-24	1.8e-27	1.2e-14
$x^0 = (-5.7,-0.6,0.5,3.7)$				
	WDK	EA	(4.1)	(4.2)
iter	4	4	3	3
err	5.0e-11	1.6e-13	6.9e-28	2.4e-24

Example 2. Consider the algebraic polynomial $f(x) = (x+1)(x-1)(x-3)(x-4)(x-7)$, with roots $\xi_1 = -1, \xi_2 = 1, \xi_3 = 3, \xi_4 = 4, \xi_5 = 7$. The results are included in the table below :

Table 2.

$x^0 = (-1.5,0.4,3.5,4.7,6.6)$				
	WDK	EA	(4.1)	(4.2)
iter	6	4	4	3
err	3.1e-15	0	0	0
$x^0 = (-1.3,0.6,3.6,4.4,6.7)$				
	WDK	EA	(4.1)	(4.2)
iter	5	4	4	3
err	4.2e-07	0	0	0

6. Notes and comments.

- The proposed family (2.3) generates iterative methods for simultaneous determination of all roots of an algebraic polynomial with arbitrary order of convergence.
- In particular, the formula of p'th order obtained by (2.3) requires computation of $p-1$ number of values of the function *f* and only one time the value of product $\prod_{s \neq i}^{n} (x_i - x_s)$.
- The higher order of convergence is reached at the expense of computation of just one more value of the function. Thus the efficiency of the corresponding method increases.
- Unlike all other known third or higher order simultaneous methods, it is not required to compute the first or higher derivatives of the function to carry out the iterations.
- The procedure of recurrently generation of the formulae of family (2.3) permits an easy realization of this methods on computer in program environment.

References

[1] Traub, J. F.: *Iterative methods for the Solution of Equations*, Prentice Hall, Englewood Cliffs, New Jersey, (1964);

[2] Petkov, M., Kjurkchiev, N.: *Numerical Methods for Solving Nonlinear equations*, University of Sofia, Sofia, (2000) (in Bulgarian);

[3] Sendov, B., Popov, V.: *Numerical Methods Part I*, Nauka i Izkustvo, Sofia, (in Bulgarian);

[4] Nedzhibov, G. H., Hasanov, V. I., Petkov, M. G.: On Some Families of Multi-point Iterative methods for Solving Nonlinear Equations, Submited, (2003);

[5] Nedzhibov, G. H., Petkov, M. G.: On a Family of Iterative methods for Simultaneous Extraction of All Roots of Algebraic Polynomial, Submitted, (2003).