

# CONSTRUCTION OF ITERATIVE FAMILIES BASED ON POWER MEANS

*Thesis Submitted in partial fulfillment of the requirements for  
The award of the degree of  
Masters of Science  
In  
Mathematics and Computing*

*Submitted by*  
Manpreet Kaur  
Reg. no. 301503013

*Under  
the guidance of*  
Dr. Sanjeev Kumar



*July, 2017*  
School of Mathematics  
Thapar University  
Patiala- 147004(PUNJAB)  
INDIA

**DEDICATED**  
**TO**  
***GOD, MY PARENTS***  
**AND**  
***SUPERVISOR***

## CERTIFICATE

*I hereby certify that the work which is bring presented in the thesis entitled "CONSTRUCTION OF ITERATIVE FAMILIES BASED ON POWER MEANS" in partial fulfillment of the requirements for the award of degree of Master of Science, School of Mathematics, Thapar University, Patiala is an authentic record of my own work carried out under the supervision of Dr. Sanjeev Kumar.*

*The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.*

*Manpreet  
Kaur.*

(Manpreet Kaur)  
Reg. No. 301503013

*This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.*

*Sanjeev Kumar.*  
(Dr. Sanjeev Kumar)  
Assistant Professor  
SOM, Thapar University  
Patiala.

## ACKNOWLEDGEMENTS

*The writing of this dissertation has been one of the most significant academic challenges I have ever had to face. Without the support, patience and guidance of the following people, this study would not have been completed. It is to them that I owe my deepest gratitude.*

*First and foremost, I would like to express my special appreciation and thanks to my supervisor, **Dr. Sanjeev Kumar**, Assistant Professor, School of Mathematics, Thapar University, Patiala, you have been a tremendous mentor for me. He has supported me throughout my thesis with his patience and knowledge. I would like to thank him for the guidance, encouragement and advice he has provided my time as his student. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly and his challenges brought this work towards a completion. The door of **Dr. Sanjeev Kumar** office was always open whenever I ran into a trouble spot or had a question about my research or writing. As he also helped me get on the road to  $\text{\LaTeX}$  and provided an experienced ear for my doubts about writing a thesis. It is with his supervision that this work came into existence.*

*I would like to express my sincere thanks to **Dr. A.K. Lal**, Head SOM, Thapar University, Patiala, for providing the necessary facilities needed for this work and directly or indirectly encouraged me to do the work.*

*I would also thank all academic and administrative staff of School of Mathematics, Thapar University, Patiala.*

*Atlast, but not the least I must express my very profound gratitude to my parents and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. And finally thanks to God who made all the things possible.*

*I am very thankful to everyone who all supported me, for I have completed my thesis effectively and moreover, on time.*

(MANPREET KAUR)

## ABSTRACT

Nonlinear equations appear in many disciplines of sciences such as engineering, mathematics, robotics and computer sciences because majority of physical systems are nonlinear in nature. One of the most basic and challenging tasks is to find the solution to nonlinear equations effectively and accurately. In this study, we provide and analyze some new iterative techniques for solving non-linear equations. This thesis consists of four chapters as follows:

**CHAPTER 1:** In this chapter, some basic concepts and definitions regarding the nonlinear equations are introduced. Some fundamental concepts of calculus are also explained in this chapter. Further, a brief explanation of classical iterative methods is provided which are commonly used in scientific and engineering problems.

**CHAPTER 2:** It provides a survey of the work proposed by Lotfi et al.[1] in year 2014. Which deals with efficient iterative methods with and without memory possessing high efficiency indices.

**CHAPTER 3:** In this chapter, new optimal iterative derivative free families based on power-mean are constructed. The proposed families give several with and without memory iterative methods having higher order of convergence.

**CHAPTER 4:** Numerical comparisons of proposed methods with some other same order iterative methods have been performed in this chapter. The numerical results are illustrate the theoretical aspects using the programming package Mathematica 10.

## GLOSSARY OF SYMBOLS

|                  |   |
|------------------|---|
| $P_\beta$        | Power mean where $\beta \in \mathbb{R}$   |
| $g(x), f(x)$     | Continuous function   |
| $x_0$            | Initial guess   |
| $\xi$            | Exact root  |
| $x_{n+1}$        | Approximation of $\xi$ at $(n + 1)^{th}$ step   |
| $\epsilon_{n+1}$ | Error at $(n + 1)^{th}$ step of $x_{n+1}$   |
| $p$              | Order of convergence  |
| $E^*$            | Efficiency index  |
| $\gamma$         | Non-zero real constant  |
| $coc$            | Computation order of convergence  |
| $P_i(x)$         | Approximating $f(x)$ by Lagrange's interpolatory polynomial of $i^{th}$ degree                          |
| $N'_3(t)$        | $N_3(t; x_n, y_{n-1}, w_{n-1}, x_{n-1})$ is Newton's interpolatory polynomial of third degree           |
| $N'_4(t)$        | $N_4(t; x_n, z_{n-1}, y_{n-1}, w_{n-1}, x_{n-1})$ is Newton's interpolatory polynomial of fourth degree |
| $Q_i(t)$         | Approximating $g(x)$ by Neville's approximation interpolatory polynomial of $i^{th}$ degree.            |

# Contents

|   |            |
|---|------------|
| <b>CERTIFICATE</b>                                      | <b>i</b>   |
| <b>ACKNOWLEDGEMENTS</b>                                 | <b>ii</b>  |
| <b>ABSTRACT</b>   | <b>iii</b> |
| <b>GLOSSARY OF SYMBOLS</b>                              | <b>iv</b>  |
| <b>1 Introduction and fundamental concepts</b>          | <b>1</b>   |
| 1.1 Review of definition of various means . . . . .     | 3          |
| 1.2 Intermediate value theorem . . . . .                | 3          |
| 1.3 Order of convergence . . . . .                      | 4          |
| 1.4 Efficiency of a method . . . . .                    | 4          |
| 1.5 One point and multipoint iterative method . . . . . | 5          |
| 1.6 Iterative method with memory . . . . .              | 5          |
| 1.7 Optimal method . . . . .                            | 5          |
| 1.8 Interpolation . . . . .                             | 6          |
| 1.9 Lagrange's method . . . . .                         | 6          |
| 1.10 Bisection method . . . . .                         | 6          |
| 1.11 Regular falsi method . . . . .                     | 8          |
| 1.12 Newton-Raphson method . . . . .                    | 8          |
| 1.13 Secant method . . . . .                            | 10         |
| 1.14 Halley's method . . . . .                          | 10         |
| 1.15 Chebyshev's method . . . . .                       | 12         |
| 1.16 Family of Chebyshev-Halley type method . . . . .   | 12         |

|          |   |           |
|----------|---|-----------|
| <b>2</b> | <b>With and without memory iterative techniques proposed by Lotfi et al.[1]</b>                   | <b>14</b> |
| 2.1      | Introduction . . . . .  | 14        |
| 2.2      | Two-step method without memory . . . . .  | 14        |
| 2.3      | Three-step method without memory . . . . .  | 16        |
| 2.4      | Extension to methods with memory . . . . .  | 22        |
| <b>3</b> | <b>Construction of new iterative families based on power means</b>                                | <b>28</b> |
| 3.1      | Development of new two-step and three-step optimal iterative families without<br>memory . . . . . | 29        |
| 3.2      | Convergence analysis . . . . .  | 32        |
| 3.3      | Extension to methods with memory . . . . .  | 34        |
| <b>4</b> | <b>Numerical illustrations</b>  | <b>38</b> |
| <b>5</b> | <b>Conclusion</b>   | <b>49</b> |

# Chapter 1

## Introduction and fundamental concepts

Numerical analysis is way to find approximate solution of complex problems by following some algorithms. Numerical analysis is an area of mathematics and computer sciences which creates, analysis and implements algorithms for obtaining numerical solutions to problems involving continuous variable. Thus, it is applicable through the business, engineering, natural sciences, social science, medical science and applied sciences. Even there are numerous application in all field due to requirement precise solutions after calculations.

Numerical methods are also applicable on large systems of equations *i.e.* linear, non-linear and complicated geometries which are common in sciences and engineering which are mostly impossible to find solution analytically with standard calculus. Since the mid 20<sup>th</sup> century, the growth in power and availability of digital computers has led to an increase the use of realistic mathematical models in sciences and engineering, and numerical analysis of increasing sophisticated is needed to solve these more detailed models of the world.

Generally, such types of problems originate from real world applications of four major disciplines of engineering: civil, chemical, electrical and mechanical which can be modeled by different mathematical equations. There are some higher order algebraic equations namely, the problem of finding the roots of an auxiliary equation of higher-order homogenous differential equations with constant coefficients, the problem of finding the eigen values of a square matrix, the solution of integral and differential equations using finite difference methods etc.

can be formulated as a nonlinear equations or a system of nonlinear equations. It may be possible that these involves exponential, trigonometry and hyperbolic terms or completely be transcendental equations.

Thus, over all goal field of numerical analysis is to analyze the problem and design the techniques which gives approximate but accurate solutions of complex real life problems.

Although, there are numerous real life problems in which we require to find the root of given problem. Thus, there are two ways for finding the root of given problem:

- i) Direct method or analytical method,
- ii) Indirect method (generally called as iterative method).

In analytical or direct method, it gives an exact root of equation in finite number of steps. Also, these methods gives all the root at a time. But, mostly they fails to find the root of higher order nonlinear equation. For example, there is no direct method for finding the root of many algebraic and transcendental equations. This limitation is overtook by indirect methods. Although indirect methods are not expected to terminate in a finite number of steps, but it gives approximate solution of every linear and non-linear equations. In this, starting from an initial guess, iterative methods form successive approximations that converges to the exact solution only in the limit. Examples include Newton's method, the bisection method, and Jacobi iteration.

## 1.1 Review of definition of various means

For a given finite real number  $\beta$ , the  $\beta^{\text{th}}$ -power mean  $P_\beta$ [2] of positive scalars  $a$  and  $b$ , is defined as follows:

$$P_\beta = \left( \frac{a^\beta + b^\beta}{2} \right)^{\frac{1}{\beta}}. \quad (1.1.1)$$

It is easy to see that:

$$\text{for } \beta = -1, \quad P_{-1} = \frac{2ab}{a+b} \quad (\text{Harmonic mean}), \quad (1.1.2)$$

$$\text{for } \beta = \frac{1}{2}, \quad P_{\frac{1}{2}} = \left\{ \frac{\sqrt{a} + \sqrt{b}}{2} \right\}^2, \quad (1.1.3)$$

$$\text{for } \beta = 1, \quad P_1 = \frac{a+b}{2} \quad (\text{Arithmetic mean}), \quad (1.1.4)$$

$$\text{for } \beta = 2, \quad P_2 = \sqrt{\frac{a^2 + b^2}{2}} \quad (\text{Root mean square}), \quad (1.1.5)$$

and

$$\text{for } \beta \rightarrow 0, \quad \text{we have } P_0 = \lim_{\beta \rightarrow 0} P_\beta = \sqrt{ab} \quad (\text{Geometric mean}). \quad (1.1.6)$$

For given positive scalars  $a$  and  $b$ , some other well-known means are defined as:

$$H_oM = \frac{a + \sqrt{ab} + b}{3} \quad (\text{Heronian mean}), \quad (1.1.7)$$

$$C_oM = \frac{a^2 + b^2}{a+b} \quad (\text{Contra-harmonic mean}), \quad (1.1.8)$$

$$C_eM = \frac{2(a^2 + ab + b^2)}{3(a+b)} \quad (\text{Centroidal mean}), \quad (1.1.9)$$

and

$$L_oM = \frac{a-b}{\log(a) - \log(b)} \quad (\text{Logarithmic mean}). \quad (1.1.10)$$

## 1.2 Intermediate value theorem

If  $f(x)$  is continuous function on some interval  $[a, b]$  and  $f(a)f(b) < 0$ , then the equation  $f(x) = 0$  has at-least one real root or an odd number of real roots in the interval  $(a, b)$  [3, pp. 19].

### 1.3 Order of convergence

Let  $x_0, x_1, x_2 \dots$  be approximation sequence and it converges to the exact root  $\xi$ . Let at  $(n + 1)^{th}$  iterate the error will be  $\epsilon_{n+1}$ , *i.e.*  $\epsilon_{n+1} = \xi - x_{n+1}$  and similarly,  $\epsilon_n$  be the error at  $n^{th}$  iterate which is equal to  $\epsilon_n = \xi - x_n$ .

Thus, an iterative method is said to be of **order**  $p$ , if there exist a finite constant  $C \neq 0$  such that

$$|\epsilon_{n+1}| \leq C |\epsilon_n|^p \tag{1.3.1}$$

The constant  $C$  is called the **asymptotic error constant** and usually depends on derivative of  $f(x)$  at  $x = \xi$  [4, pp.102].

### 1.4 Efficiency of a method

The efficiency index of an iterative method in the sense of **Traub** is defined by the equation

$$E^* = p^{1/d} \tag{1.4.1}$$

where  $p$  is the order of the method and  $d$  is the total number of function and derivative evaluations at each step of the iteration [3, pp. 59].

Obviously, if the value of the index is larger, then the method is more efficient.

Table 1.1: Comparison of Efficiency Index

| Method         | $d$ | $p$  | $E^*$ |
|----------------|-----|------|-------|
| Secant         | 1   | 1.62 | 1.62  |
| Newton-Raphson | 2   | 2    | 1.41  |
| Chebyshev      | 3   | 3    | 1.44  |
| Halley         | 3   | 3    | 1.44  |
| Superhalley    | 3   | 3    | 1.44  |

## 1.5 One point and multipoint iterative method

Let  $x_{i+1}$  be determined by only new information at  $x_i$  *i.e.* no old information is reused. Therefore,  $x_{i+1}$  can be written as:

$$x_{i+1} = \phi(x_i), \quad (1.5.1)$$

and then, this  $\phi$  is called a one-point iterative function.

Let  $x_{i+1}$  be determined by new information at  $x_i, w_1(x_i), \dots, w_n(x_i), k \geq 1$  and no old information is reused. Therefore,  $x_{i+1}$  can be written as:

$$x_{i+1} = \phi(x_i, w_1(x_i), \dots, w_n(x_i)), \quad (1.5.2)$$

and then, this  $\phi$  is called a multi-point iterative function.

## 1.6 Iterative method with memory

Let  $x_{i+1}$  be determined by using new information at  $x_i$  and by reusing information at  $x_{i-1}, x_{i-2}, \dots, x_{i-n}$ , it can be written as

$$x_{i+1} = \phi(x_i; x_{i-1}, x_{i-2}, \dots, x_{i-n}), \quad (1.6.1)$$

and this  $\phi$  is called a iterative function with memory [5, pp. 9]. The semicolon in (1.6.1) separates the point at which new data are used from the points at which old data are reused.

## 1.7 Optimal method

The highest possible computational efficiency of these methods is closely connected to the hypothesis of Kung and Traub from 1974 [6]. They have conjectured that the order of convergence of any multipoint method without memory, requiring  $n + 1$  functional evaluations per iteration, that can reach the order  $2^n$ . Thus, multipoint methods with this property are usually called optimal methods.

## 1.8 Interpolation

Let us suppose that there are given  $(n + 1)$  points in interval  $[x_0, x_n]$  and let these  $(n + 1)$  points are  $x_0 < x_1 < x_2 < \dots < x_n$ . Suppose  $y_0, y_1, \dots, y_n$ , be the values of function at  $x_0 < x_1 < x_2 < \dots < x_n$  are known. Let  $x \in (x_0, x_n)$  and to find the value of function at  $x$  by using some intermediate values in interval  $(x_0, x_n)$ , then this process is called *Interpolation* [3, 4, 7].

The process is called *extrapolating* if  $x$  lies outside the interval  $(x_0, x_n)$ , and the value of function at  $x$  is find by using some intermediate values in interval  $(x_0, x_n)$ .

## 1.9 Lagrange's method

This method determining a polynomial of degree  $\leq n$  that passes through the  $(n + 1)$  distinct points. This polynomial is same as approximating function  $f$ . So, Lagrange's polynomial [4] satisfying the points  $(x_i, y_i)$  where  $0 \leq i \leq n$  is defined as,

$$P(x) = L_0(x)y_0 + L_1(x)y_1 + \dots + L_n(x)y_n \quad (1.9.1)$$

$$= \sum_{i=0}^n L_i(x)y_i, \quad (1.9.2)$$

where  $L_i(x) = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$ ,  $0 \leq i \leq n$ .

This  $L_i(x)$  function is called Lagrange's coefficients. Also, this  $L_i(x)$ 's satisfy the *Kronecker's delta* property, *i.e.*,

$$\begin{cases} L_i(x_j) = 1, & i = j \\ 0, & i \neq j. \end{cases} \quad (1.9.3)$$

## 1.10 Bisection method

Suppose  $f \in C[a, b]$ , where  $f(a)$  and  $f(b)$  having opposite sign. Thus, applying Intermediate value theorem, there exist a number  $\xi$  lies in interval  $(a, b)$  such that  $f(\xi) = 0$ . Even the

procedure will work for multiple roots in the interval  $(a, b)$ , but for sake of simplicity we assume that the root lies in this interval is unique. This method is also called as method of halving that is, repeated halving of subintervals of  $[a, b]$  and at each step, locating the half containing  $\xi$ .

To begin, set  $a_1 = a$  and  $b_1 = b$ , let  $\xi_1$  be the midpoint of  $[a, b]$ ; that is,

$$\xi_1 = \frac{a_1 + b_1}{2}.$$

- If  $f(\xi_1) = 0$ , then  $\xi = \xi_1$ , and we are done.
- If  $f(\xi_1) \neq 0$ , then  $f(\xi)$  has the same sign as either  $f(a_1)$  or  $f(b_1)$ .
  - If  $f(\xi_1)$  and  $f(a_1)$  have the same sign,  $\xi \in (\xi_1, b_1)$ . Set  $a_2 = \xi_1$  and  $b_2 = b_1$ .
  - If  $f(\xi_1)$  and  $f(a_1)$  have opposite sign,  $\xi \in (a_1, \xi_1)$ . Set  $a_2 = a_1$  and  $b_2 = \xi_1$ .

Then reapply the process to the interval  $[a_2, b_2]$ . Then the approximate sequence obtained from this procedure will always converge to the root of  $f(x)$  provided that it should be continuous in the given interval  $[a, b]$  in which the root lies.

If the tolerance is  $\epsilon$ , then the approximate number of iterations required may be determined from the relation

$$\frac{b_0 - a_0}{2^n} \leq \epsilon.$$

After taking log on both sides we get,

$$n \geq \frac{\log(b_0 - a_0) - \log\epsilon}{\log 2}.$$

Since  $n$  is integer, we take  $n$  as the next nearest integer.

The minimum number of iteration required for converging to a root in the interval  $(0, 1)$  for a given  $\epsilon$  are listed in Table 1.2.

Table 1.2: Number of iterations

|            |           |           |           |           |
|------------|-----------|-----------|-----------|-----------|
| $\epsilon$ | $10^{-2}$ | $10^{-3}$ | $10^{-4}$ | $10^{-5}$ |
| n          | 7         | 10        | 14        | 17        |

Thus, the bisection method [3, 7] requires a large number of iterations to achieve a reasonable degree of accuracy for the root.

## 1.11 Regular falsi method

Let the function be expressed as  $y = f(x)$  and we want to find the zeros of the function. Proceed the follows [3, 4, 7]:

- (i) Find  $x_1$  and  $x_2$  where  $y_1 = f(x_1)$  and  $y_2 = f(x_2)$  have opposite signs and that there is only one value of  $x$  between  $x_1$  and  $x_2$  where  $f(x)$  vanishes.
- (ii) Approximate the function between  $x_1$  and  $x_2$  by a straight line joining points  $(x_1, y_1)$  and  $(x_2, y_2)$ . The point where the straight line cuts the  $x - axis$  is an estimate of the root. The equation of the straight line is,

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1), \quad (1.11.1)$$

and the point where it intersects the  $x - axis$  is given by putting  $y = 0$  in (1.11.1), as

$$x = \frac{x_1 y_2 - x_2 y_1}{y_2 - y_1}. \quad (1.11.2)$$

Thus the value of  $x$  given by (1.11.2) is the approximate root of  $f(x) = 0$ .

- (iii) Compute  $f(x)$  and check its sign. If  $f(x)f(x_2) > 0$ , then replace  $x_1$  by  $x$  and  $y_1 = f(x)$ .
- (iv) Repeat previous two steps, until two consecutive values of  $x$  agree within desired accuracy.

## 1.12 Newton-Raphson method

Newton-Raphson method is one of the most powerful and well-known numerical methods for solving a root-finding problem. There are many ways of introducing Newton-Raphson Method[3, 4, 7]. One of them is explained in this thesis.

Let  $x_0$  be an approximate root of  $f(x)$  and let  $x_1 = x_0 + h$  be the correct root so that  $f(x_1) = 0$ . By using Taylor's series on the function  $f(x_0 + h)$ , we obtain

$$f(x_0) + hf'(x_0) + \frac{h^2}{2!}f''(x_0) + \dots = 0.$$

Neglecting the second and higher-order derivatives, we have

$$f(x_0) + hf'(x_0) = 0,$$

which gives

$$h = -\frac{f(x_0)}{f'(x_0)}.$$

So, the better approximation than  $x_0$ , is

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Therefore, successive approximations are given by  $x_2, x_3, \dots, x_{n+1}$ , where

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (1.12.1)$$

which is the *Newton-Raphson formula*.

But there are few cases in which the Newton-Raphson method is divergence as shown in Figure 1.1.

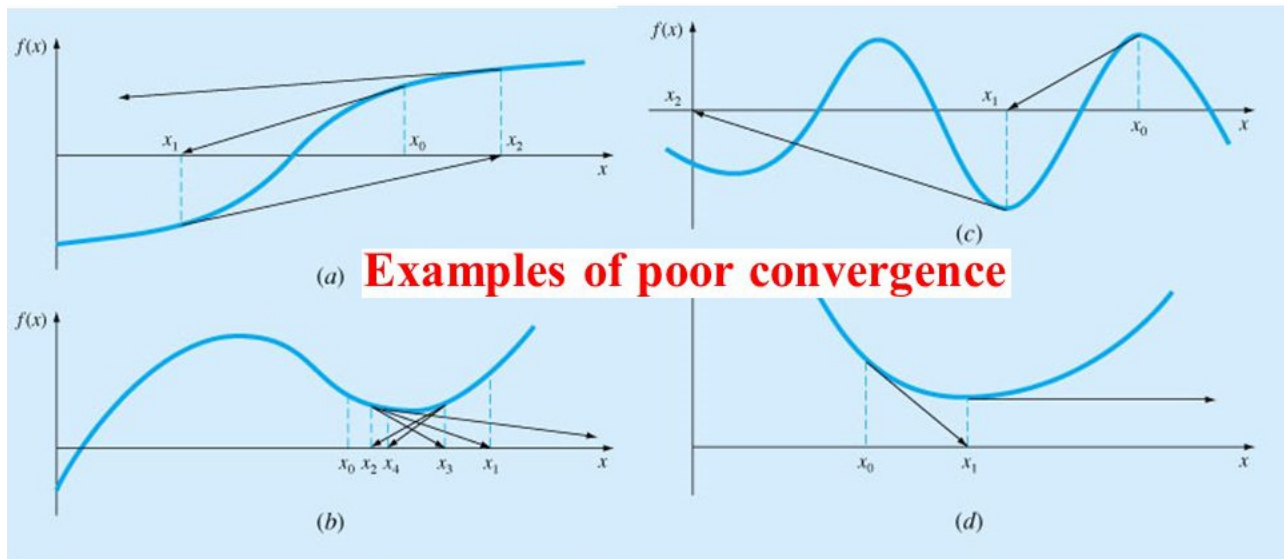


Figure 1.1: Poor convergence cases.

### Conditions for convergence of Newton-Raphson Method

Let  $f(x)$  be twice continuously differentiable on the closed finite interval  $[a, b]$  and let the following conditions be satisfied:

- i.  $f(a)f(b) < 0$
- ii.  $f'(x) \neq 0, x \in [a, b]$

iii.  $f''(x)$  is either  $\geq 0$  or  $\leq 0$  for all  $x \in [a, b]$

iv. At the endpoints  $a, b$

$$\frac{|f(a)|}{|f'(a)|} < b - a \text{ and } \frac{|f(b)|}{|f'(b)|} < b - a.$$

Then Newton-Raphson method converges to the unique solution  $\xi$  of  $f(x) = 0$  in  $[a, b]$  for any value of  $x_0 \in [a, b]$ .

## 1.13 Secant method

We have seen the Newton-Raphson method requires the evaluation of derivatives of the function and this is not always possible, particularly in the case of functions arising in the practical problem.

In the secant method[3, 4, 7], the derivative at  $f(x_i)$  is approximated by the formula:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}},$$

which can be written as

$$f'_i \approx \frac{f_i - f_{i-1}}{x_i - x_{i-1}},$$

where  $f_i = f(x_i)$ . Hence, the Newton-Raphson formula becomes

$$x_{i+1} = x_i - \frac{f_i(x_i - x_{i-1})}{f_i - f_{i-1}} = \frac{x_{i-1}f_i - x_if_{i-1}}{f_i - f_{i-1}}.$$

Note: This formula requires two initial approximations to the root.

Thus the graphical representation of iteration shown in Figure 1.2.

## 1.14 Halley's method

Let  $f(x)$  be a polynomial, the root of  $f(x)$  satisfies

$$f(x) = 0.$$

Expand this  $f(x)$  by second-order Taylor's series, and we get

$$0 \approx f(x_n) + f'(x_n)(x_{n+1} - x_n) + \frac{1}{2}f''(x_n)(x_{n+1} - x_n)^2. \quad (1.14.1)$$

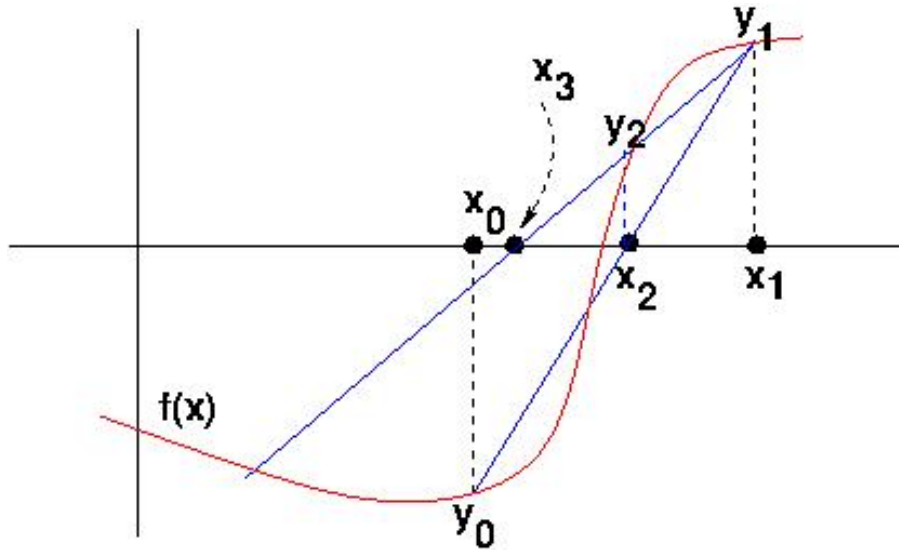


Figure 1.2: Secant method.

Rewrite (1.14.1)

$$0 = f(x_n) + (x_{n+1} - x_n) \left[ f'(x_n) + \frac{1}{2} f''(x_n)(x_{n+1} - x_n) \right],$$

implies

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

By using (1.12.1)  $x_{n+1} - x_n = -f(x_n)/f'(x_n)$  gives,

$$x_{n+1} = x_n - \frac{2f(x_n)f'(x_n)}{2[f'(x_n)]^2 - f(x_n)f''(x_n)}.$$

So, the iterative function[8] is

$$H_f(x) = x - \frac{2f(x)f'(x)}{2[f'(x)]^2 - f(x)f''(x)}. \quad (1.14.2)$$

## 1.15 Chebyshev's method

Let us assume function  $f(x)$  is a polynomial of degree two and write it as

$$f(x) = ax^2 + bx + c = 0, \quad a \neq 0, \quad (1.15.1)$$

where  $a$ ,  $b$ , and  $c$  are arbitrary parameters to be determined by prescribing three appropriate conditions on  $f(x)$  and/or its derivatives.

We determine  $a$ ,  $b$  and  $c$  in (1.15.1) using the conditions

$$\begin{cases} f(x_n) = ax_n^2 + bx_n + c \\ f'(x_n) = 2ax_n + b \\ f''(x_n) = 2a. \end{cases} \quad (1.15.2)$$

On eliminating  $a$ ,  $b$ , and  $c$  from (1.15.1) and (1.15.2), we obtain

$$f(x_n) + (x - x_n)f'(x_n) + \frac{1}{2}(x - x_n)^2 f''(x_n) = 0, \quad (1.15.3)$$

which is the Taylor series expansion of  $f(x)$  about  $x = x_n$  such that the cubic and higher powers are neglected.

Thus the next approximation is obtained from (1.15.3) and rewrite it as follows:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2}(x_{n+1} - x_n)^2 \frac{f''(x_n)}{f'(x_n)}. \quad (1.15.4)$$

By using (1.12.1)

$$x_{n+1} - x_n = -\frac{f(x_n)}{f'(x_n)}, \quad (1.15.5)$$

gives

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{1}{2} \frac{f^2(x_n)}{f'^3(x_n)} f''(x_n), \quad (1.15.6)$$

which is Chebyshev's Method[3, 4].

## 1.16 Family of Chebyshev-Halley type method

The family of Chebyshev-Halley type methods[8] is given by

$$x_{n+1} = x_n - \left[ 1 + \frac{1}{2} \left\{ \frac{L_f(x_n)}{1 - \lambda L_f(x_n)} \right\} \right] \frac{f(x_n)}{f'(x_n)}, \quad (1.16.1)$$

where  $\lambda \in \mathbb{R}$  and  $L_f(x_n) = \frac{f''(x_n)f(x_n)}{f'^2(x_n)}$ .

This family includes the

1. Classical Chebyshev's method, when  $\lambda = 0$ ,
2. Halley's method, when  $\lambda = 0.5$ , and
3. Super-Halley method, when  $\lambda = 1$ .

# Chapter 2

## With and without memory iterative techniques proposed by Lotfi et al.[1]

This chapter deals with construction of multipoint iterative methods for solving nonlinear equations.

### 2.1 Introduction

In 2014, Lotfi et al.[1] have worked on the two families of derivative-free methods without memory for approximating a simple zero of a nonlinear equation. Their schemes have an accelerator parameter with the property that it can increase the convergence rate without any new functional evaluations. In this way, they constructed a method with memory that increases considerably efficiency index from  $8^{\frac{1}{4}} \approx 1.681$  to  $12^{\frac{1}{4}} \approx 1.861$ .

### 2.2 Two-step method without memory

Let  $f$  be scalar function with domain of  $f$  is  $D$  which is subset of  $\mathbb{R}$  and mapping  $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ . Let  $\xi$  be the simple root of  $f(x)$  then  $f(\xi) = 0$  provided  $f'(\xi) \neq 0 = c_1$ .

Lofti et al.[1] proposed method without memory which is starting with two-step scheme,

by using well-known classical Newton's method (1.12.1) as follows:

$$\begin{cases} y = x - \frac{f(x)}{f'(x)}, \\ \bar{x} = y - \frac{f(y)}{f'(y)}. \end{cases} \quad (2.2.1)$$

The order of convergence of (2.2.1) scheme is four. Now, approximating the derivatives in (2.2.1) by suitable approximations as follows:

Using the point  $x$  and  $w$ , where  $w = x + \gamma f(x)$  ( $\gamma$  is a non-zero real constant), and applying the Lagrange interpolatory polynomial on these points for approximating  $f'(x)$ .

The formula

$$P_n(x) = \sum_{i=0}^n y_i l_i(x), \text{ where } y_i = f(x_i), \quad (2.2.2)$$

is called Lagrange's formula used for the interpolating polynomial.

So,

$$\begin{aligned} P_1(t) &= \sum_{i=0}^1 y_i l_i(t) \\ &= l_0(t) f(x_0) + l_1(t) f(x_1) \\ &= \left( \frac{t-w}{x-w} \right) f(x) + \left( \frac{t-x}{w-x} \right) f(w). \end{aligned} \quad (2.2.3)$$

Now derivative of  $f(x)$  is approximated by derivative of  $P_1(x)$ . Thus fixing  $f'(x) \approx P_1'(x)$ , and by setting  $t = x$ , we have

$$\begin{aligned} P_1'(x) &= \frac{1}{(x-w)} f(x) + \frac{1}{(w-x)} f(w) \\ &= \frac{f(x)}{(x-w)} - \frac{f(w)}{(x-w)} \\ &= \frac{f(w) - f(x)}{x-w} \\ &= f[x, w]. \end{aligned} \quad (2.2.4)$$

For three points  $x$ ,  $w$ , and  $y$ , the Lagrange's interpolatory polynomials can given as follows:

$$P_2(t) = y_0 l_0(t) + y_1 l_1(t) + y_2 l_2(t), \quad (2.2.5)$$

where,  $y_0 = f(x_0) = f(x)$ ,  $y_1 = f(x_1) = f(w)$ ,  $y_2 = f(x_2) = f(y)$ ,

$$\begin{aligned} l_0(t) &= \frac{(t-w)(t-y)}{(x-w)(x-y)} = \frac{t^2 - t(w+y) + wy}{(x-w)(x-y)}, \\ l_1(t) &= \frac{(t-x)(t-y)}{(w-x)(w-y)} = \frac{t^2 - t(x+y) + xy}{(w-x)(w-y)}, \\ \text{and } l_2(t) &= \frac{(t-w)(t-x)}{(y-w)(y-x)} = \frac{t^2 - t(x+w) + xw}{(y-x)(y-w)}. \end{aligned}$$

So  $P_2(t)$  becomes,

$$P_2(t) = \frac{t^2 - t(w+y) + wy}{(x-w)(x-y)} f(x) + \frac{t^2 - t(x+y) + xy}{(w-x)(w-y)} f(w) + \frac{t^2 - t(x+w) + xw}{(y-x)(y-w)} f(y). \quad (2.2.6)$$

Now  $f'(y)$  can be approximated by  $P_2'(y)$ , which is equal to

$$P_2'(y) = \frac{(2y-w-y)}{(x-w)(x-y)} f(x) + \frac{(2y-x-y)}{(w-x)(w-y)} f(w) + \frac{2y-w-x}{(y-w)(y-x)} f(y). \quad (2.2.7)$$

Substituting derivative terms  $f'(x)$  and  $f'(y)$  by its approximating function  $P_1'(x) = f[x, w]$  and  $P_2'(y)$  respectively in (2.2.1). So, the derivative-free two step iterative method is defined as follow:

$$\begin{cases} y = x - \frac{f(x)}{f[x, w]}, \\ w = x + \gamma f(x), \\ z = y - \frac{f(y)}{P_2'(y)}. \end{cases} \quad (2.2.8)$$

## 2.3 Three-step method without memory

In this section, discussing the constructed of a three-step uniparameter family of methods which is based on the method (2.2.8) discuss in previous section. Lotfi et al.[1] constructed the three-step scheme by using previous first-two steps which are given in (2.2.8), and the third step by using Newton's method (1.12.1). Thus, three-step scheme method is given as

follows:

$$\begin{cases} y = x - \frac{f(x)}{f[x, w]}, \\ w = x + \gamma f(x), \\ z = y - \frac{f(y)}{P_2'(y)}, \\ \bar{x} = z - \frac{f(z)}{f'(z)}. \end{cases} \quad (2.3.1)$$

As  $f'(y)$  is approximated by  $P_2'(y)$  in pervious section. Similarly in (2.3.1), the derivative of  $f(z)$  (*i.e.*  $f'(z)$ ) in the third step is approximated by substituting suitable approximation, so that the order of convergence is obtained as high as possible and scheme (2.3.1) become optimal.

Thus, using Lagrange's interpolating polynomial at four points  $x$ ,  $w$ ,  $y$ , and  $z$  for find the suitable approximation of  $f'(z)$ . Therefore, Lagrange's interpolating polynomial is

$$P_3(t) = y_0l_0(t) + y_1l_1(t) + y_2l_2(t) + y_3l_3(t), \quad (2.3.2)$$

where  $y_0 = f(x)$ ,  $y_1 = f(w)$ ,  $y_2 = f(y)$ ,  $y_3 = f(z)$ ,

$$\begin{aligned} l_0(t) &= \frac{(t-w)(t-y)(t-z)}{(x-w)(x-y)(x-z)} \\ &= \frac{(t^3 - t^2(w+y+z) + t(wy+yz+zw) - wyz)}{(x-w)(x-y)(x-z)}, \\ l_1(t) &= \frac{(t-x)(t-y)(t-z)}{(w-x)(w-y)(w-z)} \\ &= \frac{(t^3 - t^2(x+y+z) + t(xy+yz+zx) - xyz)}{(w-x)(w-y)(w-z)}, \\ l_2(t) &= \frac{(t-x)(t-w)(t-z)}{(y-x)(y-w)(y-z)} \\ &= \frac{(t^3 - t^2(x+w+z) + t(xw+wz+zx) - xwz)}{(y-x)(y-w)(y-z)}, \\ \text{and } l_3(t) &= \frac{(t-x)(t-w)(t-y)}{(z-x)(z-w)(z-y)} \\ &= \frac{(t^3 - t^2(x+w+y) + t(xw+wy+yx) - xwy)}{(z-x)(z-w)(z-y)}. \end{aligned}$$

So,  $P_3(t)$  becomes

$$\left\{ \begin{aligned} P_3(t) &= \frac{(t^3 - t^2(w + y + z) + t(wy + yz + zw) - wyz)}{(x - w)(x - y)(x - z)} f(x) \\ &+ \frac{(t^3 - t^2(x + y + z) + t(xy + yz + zx) - xyz)}{(w - x)(w - y)(w - z)} f(w) \\ &+ \frac{(t^3 - t^2(x + w + z) + t(xw + wz + zx) - xwz)}{(y - x)(y - w)(y - z)} f(y) \\ &+ \frac{(t^3 - t^2(x + w + y) + t(xw + wy + yx) - xwy)}{(z - x)(z - w)(z - y)} f(z). \end{aligned} \right. \quad (2.3.3)$$

Now differentiating  $P_3(t)$  and setting  $t = z$ , one can get approximation of  $f'(z)$ , denoted by  $P'_3(z)$  and obtained as follows:

$$\left\{ \begin{aligned} P'_3(z) &= \frac{(3z^2 - 2z(w + y + z) + (wy + yz + zw))}{(x - w)(x - y)(x - z)} f(x) \\ &+ \frac{(3z^2 - 2z(x + y + z) + (xy + yz + zx))}{(w - x)(w - y)(w - z)} f(w) \\ &+ \frac{(3z^2 - 2z(x + w + z) + (xw + wz + zx))}{(y - x)(y - w)(y - z)} f(y) \\ &+ \frac{(3z^2 - 2z(x + w + y) + (xw + wy + yx))}{(z - x)(z - w)(z - y)} f(z). \end{aligned} \right. \quad (2.3.4)$$

Substituting  $f'(z) \approx P'_3(z)$  in (2.3.1), three-step method becomes as:

$$\left\{ \begin{aligned} y &= x - \frac{f(x)}{f[x, w]}, \\ w &= x + \gamma f(x), \\ z &= y - \frac{f(y)}{P'_2(y)}, \\ \bar{x} &= z - \frac{f(z)}{P'_3(z)}, \end{aligned} \right. \quad (2.3.5)$$

where  $P'_2(y)$  is defined by (2.2.7) and  $P'_3(z)$  given by (2.3.4).

**Theorem 2.3.1** *If an initial approximation  $x_0$  is sufficiently close to a simple zero  $\xi$  of  $f$ , then the convergence order of the two step approach (2.2.8) is four, and its error equation is given by*

$$e_{n+1} = (1 + \gamma f'(\xi))^2 c_2 (c_2^2 - c_3) e_n^4 + \mathbf{O}(e_n^5). \quad (2.3.6)$$

**Proof**

$$fx := f'(\xi) \cdot (e + c_2 \cdot e^2 + c_3 \cdot e^3 + c_4 \cdot e^4 + c_5 \cdot e^5 + c_6 \cdot e^6 + c_7 \cdot e^7 + c_8 \cdot e^8 + c_9 \cdot e^9 + c_{10} \cdot e^{10}) :$$

$$ew := e + \gamma \cdot fx :$$

$$ew := f'(\xi) \cdot (e + c_2 \cdot e^2 + c_3 \cdot e^3 + c_4 \cdot e^4 + c_5 \cdot e^5 + c_6 \cdot e^6 + c_7 \cdot e^7 + c_8 \cdot e^8 + c_9 \cdot e^9 + c_{10} \cdot e^{10}) \gamma + e :$$

$$fw := f'(\xi) \cdot (ew + c_2 \cdot ew^2 + c_3 \cdot ew^3 + c_4 \cdot ew^4 + c_5 \cdot ew^5 + c_6 \cdot ew^6 + c_7 \cdot ew^7 + c_8 \cdot ew^8 + c_9 \cdot ew^9 + c_{10} \cdot ew^{10}) :$$

$$fw := \text{simplify}(\text{taylor}(fw, e = 0, 10)) :$$

$$fxw := \frac{fx - fw}{e - ew} : \quad fxw := \text{simplify}(\text{taylor}(fxw, e = 0, 10)) :$$

$$ey := e - \frac{fx}{fxw} :$$

$$ey := \text{simplify}(\text{taylor}(ey, e = 0, 10)) :$$

$$\text{factor}(\text{simplify}(\text{taylor}(ey, e = 0, 3)))$$

$$\text{Out}[a] : c_2 (f'(\xi) \gamma + 1) e^2 + \mathbf{O}(e^3)$$

$$fy := f'(\xi) \cdot (ey + c_2 \cdot ey^2 + c_3 \cdot ey^3 + c_4 \cdot ey^4 + c_5 \cdot ey^5 + c_6 \cdot ey^6 + c_7 \cdot ey^7 + c_8 \cdot ey^8 + c_9 \cdot ey^9 + c_{10} \cdot ey^{10}) :$$

$$fy := \text{simplify}(\text{taylor}(fy, e = 0, 10)) :$$

$$fxwy := \frac{2 \cdot ey - ew - ey}{(e - ew) \cdot (e - ey)} \cdot fx + \frac{2 \cdot ey - e - ey}{(ew - e) \cdot (ew - ey)} \cdot fw + \frac{2 \cdot ey - ew - e}{(ey - ew) \cdot (ey - e)} \cdot fy :$$

$$fxwy := \text{simplify}(\text{taylor}(fxwy, e = 0, 10)) : \quad ez := ey - \frac{fy}{fxwy} :$$

$$ez := \text{simplify}(\text{taylor}(ez, e = 0, 10)) :$$

$$\text{factor}(\text{simplify}(\text{taylor}(ez, e = 0, 5)))$$

$$\text{Out}[b] : c_2 (f'(\xi) \gamma + 1)^2 (c_2^2 - c_3) e^4 + \mathbf{O}(e^5)$$

**Theorem 2.3.2** *Let  $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$  be a scalar function which has the simple root  $\xi$  in the open interval  $D$ ; also initial approximation  $x_0$  is sufficiently close to a simple zero  $\xi$  of  $f$ . The three-step iterative method (2.3.5) has eight order and satisfies the following error equation:*

$$e_{n+1} = (1 + \gamma f'(\xi))^4 c_2^2 (c_2^2 - c_3) (c_2^3 - c_2 c_3 + c_4) e_n^8 + \mathbf{O}(e_n^8). \quad (2.3.7)$$

**Proof**

$$fx := f'(\xi) \cdot (e + c2 \cdot e^2 + c3 \cdot e^3 + c4 \cdot e^4 + c5 \cdot e^5 + c6 \cdot e^6 + c7 \cdot e^7 + c8 \cdot e^8 + c9 \cdot e^9 + c10 \cdot e^{10}) :$$

$$ew := e + \gamma \cdot fx :$$

$$ew := f'(\xi) \cdot (e + c2 \cdot e^2 + c3 \cdot e^3 + c4 \cdot e^4 + c5 \cdot e^5 + c6 \cdot e^6 + c7 \cdot e^7 + c8 \cdot e^8 + c9 \cdot e^9 + c10 \cdot e^{10}) \gamma + e :$$

$$fw := f'(\xi) \cdot (ew + c2 \cdot ew^2 + c3 \cdot ew^3 + c4 \cdot ew^4 + c5 \cdot ew^5 + c6 \cdot ew^6 + c7 \cdot ew^7 + c8 \cdot ew^8 + c9 \cdot ew^9 + c10 \cdot ew^{10}) :$$

$$fw := \text{simplify}(\text{taylor}(fw, e = 0, 10)) :$$

$$fxw := \frac{fx - fw}{e - ew} :$$

$$fxw := \text{simplify}(\text{taylor}(fxw, e = 0, 10)) :$$

$$ey := e - \frac{fx}{fxw} :$$

$$ey := \text{simplify}(\text{taylor}(ey, e = 0, 10)) :$$

$$\text{factor}(\text{simplify}(\text{taylor}(ey, e = 0, 3)))$$

$$\text{Out}[a] : c2(f'(\xi)\gamma + 1)e^2 + \mathbf{O}(e^3)$$

$$fy := f'(\xi) \cdot (ey + c2 \cdot ey^2 + c3 \cdot ey^3 + c4 \cdot ey^4 + c5 \cdot ey^5 + c6 \cdot ey^6 + c7 \cdot ey^7 + c8 \cdot ey^8 + c9 \cdot ey^9 + c10 \cdot ey^{10}) :$$

$$fy := \text{simplify}(\text{taylor}(fy, e = 0, 10)) :$$

$$fxwy := \frac{2 \cdot ey - ew - ey}{(e - ew) \cdot (e - ey)} \cdot fx + \frac{2 \cdot ey - e - ey}{(ew - e) \cdot (ew - ey)} \cdot fw + \frac{2 \cdot ey - ew - e}{(ey - ew) \cdot (ey - e)} \cdot fy :$$

$fxwy := \text{simplify}(\text{taylor}(fxwy, e = 0, 10)) :$

$$ez := ey - \frac{fy}{fxwy} :$$

$ez := \text{simplify}(\text{taylor}(ez, e = 0, 10)) :$

$\text{factor}(\text{simplify}(\text{taylor}(ez, e = 0, 5)))$

$\text{Out}[b] : c2(f'(\xi)\gamma + 1)^2(c2^2 - c3)e^4 + \mathbf{O}(e^5)$

$$fz := f'(\xi) \cdot (ez + c2 \cdot ez^2 + c3 \cdot ez^3 + c4 \cdot ez^4 + c5 \cdot ez^5 + c6 \cdot ez^6 + c7 \cdot ez^7 + c8 \cdot ez^8 + c9 \cdot ez^9 + c10 \cdot ez^{10}) :$$

$fz := \text{simplify}(\text{taylor}(fz, e = 0, 10)) :$

$$\begin{aligned} fxwyz := & \frac{ew \cdot ey + ew \cdot ez + ey \cdot ez - 2 \cdot (ew + ey + ez) \cdot ez + 3 \cdot ez \cdot ez}{(e - ew) \cdot (e - ey) \cdot (e - ez)} \cdot fx \\ & + \frac{e \cdot ey + e \cdot ez + ey \cdot ez - 2 \cdot (e + ey + ez) \cdot ez + 3 \cdot ez \cdot ez}{(ew - e) \cdot (ew - ey) \cdot (ew - ez)} \cdot fw \\ & + \frac{ew \cdot e + e \cdot ez + ew \cdot ez - 2 \cdot (e + ew + ez) \cdot ez + 3 \cdot ez \cdot ez}{(ey - e)(ey - ew)(ey - ez)} \cdot fy \\ & + \frac{ew \cdot e + e \cdot ey + ey \cdot ew - 2 \cdot (e + ew + ey) \cdot ez + 3 \cdot ez \cdot ez}{(ez - ew) \cdot (ez - e) \cdot (ez - ey)} \cdot fz : \end{aligned}$$

$fxwyz := \text{simplify}(\text{taylor}(fxwyz, e = 0, 10)) :$

$$eu := ez - \frac{fz}{fxwyz} :$$

$eu := \text{factor}(\text{simplify}(\text{taylor}(eu, e = 0, 10))) :$

$\text{factor}(\text{simplify}(\text{taylor}(eu, e = 0, 9)))$

$\text{Out}[c] : c2^2(f'(\xi)\gamma + 1)^4(c2^2 - c3)(c2^3 - c2c3 + c4)e^8 + \mathbf{O}(e^9)$

## 2.4 Extension to methods with memory

In this section, it concerned with extension of (2.2.8) and (2.3.5) methods with memory. Since the error equations contains the parameter  $\gamma$ , which can be approximated in such a way that increases the local order of convergence.

So, by setting  $\gamma = \gamma_n$  as the iteration proceeds by the formula  $\gamma_n = -1/\widehat{f}'(\xi)$  for  $n = 1, 2, \dots$ , where  $\widehat{f}'(\xi)$  is an approximation of  $f'(\xi)$ . Using the approximation

$$\gamma_n = -\frac{1}{\widehat{f}'(\xi)} = -\frac{1}{N'_3(x_n)}, \quad (2.4.1)$$

where  $N'_3(t) = N_3(t; x_n, y_{n-1}, w_{n-1}, x_{n-1})$  is Newton's interpolatory polynomial of third degree used for approximating  $f'(\xi)$  in two-step method (2.2.8), by setting through four available approximations  $x_n, y_{n-1}, w_{n-1}$ , and  $x_{n-1}$ . So,

$$\left\{ \begin{aligned} N'_3(x_n) &= \left[ \frac{d}{dt} N_3(t) \right]_{t=x_n} \\ &= \frac{d}{dt} \left[ f(x_n) + f[x_n, x_{n-1}](t - x_n) + f[x_n, x_{n-1}, y_{n-1}](t - x_n)(t - x_{n-1}) \right. \\ &\quad \left. + f[x_n, x_{n-1}, y_{n-1}, w_{n-1}](t - x_n)(t - x_{n-1})(t - y_{n-1}) \right]_{t=x_n} \\ &= \left[ f[x_n, x_{n-1}] \cdot 1 + f[x_n, x_{n-1}, y_{n-1}](2t - (x_n + x_{n-1})) + f[x_n, x_{n-1}, y_{n-1}, w_{n-1}] \right. \\ &\quad \left. \times (3t^2 - 2t(x_n + x_{n-1} + y_{n-1}) + (x_n x_{n-1} + x_{n-1} y_{n-1} + y_{n-1} x_n)) \right]_{t=x_n} \\ &= f[x_n, x_{n-1}] + f[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) + f[x_n, x_{n-1}, y_{n-1}, w_{n-1}] \\ &\quad \times (3x_n^2 - 2x_n(x_n + x_{n-1} + y_{n-1}) + (x_n x_{n-1} + x_{n-1} y_{n-1} + y_{n-1} x_n)) \\ &= f[x_n, x_{n-1}] + f[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) \\ &\quad + f[x_n, x_{n-1}, y_{n-1}, w_{n-1}](x_n^2 - x_n x_{n-1} - y_{n-1} x_n + x_{n-1} y_{n-1}) \\ &= f[x_n, x_{n-1}] + f[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) \\ &\quad + f[x_n, x_{n-1}, y_{n-1}, w_{n-1}](x_n - x_{n-1})(x_n - y_{n-1}). \end{aligned} \right. \quad (2.4.2)$$

Further the following one

$$\gamma_n = -\frac{1}{\widehat{f}'(\xi)} = -\frac{1}{N'_4(x_n)}, \quad (2.4.3)$$

where  $N'_4(t) = N_4(t; x_n, z_{n-1}, y_{n-1}, w_{n-1}, x_{n-1})$  is Newton's interpolatory polynomial of fourth degree is used for approximating  $f'(\xi)$  in three-step method (2.3.5), setting through five avail-

able approximations  $x_n$ ,  $z_{n-1}$ ,  $y_{n-1}$ ,  $w_{n-1}$ , and  $x_{n-1}$ . So,

$$\left\{ \begin{aligned}
N_4'(x_n) &= \left[ \frac{d}{dt} N_4(t) \right]_{t=x_n} \\
&= \frac{d}{dt} \left[ f(x_n) + f[x_n, z_{n-1}](t - x_n) + f[x_n, z_{n-1}, y_{n-1}](t - x_n)(t - z_{n-1}) \right. \\
&\quad + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}](t - x_n)(t - z_{n-1})(t - y_{n-1}) \\
&\quad \left. + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}](t - x_n)(t - z_{n-1})(t - y_{n-1})(t - x_{n-1}) \right]_{t=x_n} \\
&= \left[ f[x_n, z_{n-1}] \cdot 1 + f[x_n, z_{n-1}, y_{n-1}](2t - (x_n + z_{n-1})) + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}] \right. \\
&\quad \times (3t^2 - 2t(x_n + z_{n-1} + y_{n-1}) + (x_n z_{n-1} + z_{n-1} y_{n-1} + y_{n-1} x_n)) \\
&\quad + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}] \left( 4t^3 - 3t^2(x_n + z_{n-1} + y_{n-1} + x_{n-1}) \right. \\
&\quad + 2t(x_n z_{n-1} + z_{n-1} y_{n-1} + y_{n-1} x_{n-1} + x_{n-1} x_n + x_n y_{n-1} + z_{n-1} x_{n-1}) \\
&\quad \left. \left. - (x_n z_{n-1} y_{n-1} + z_{n-1} y_{n-1} x_{n-1} + y_{n-1} x_{n-1} x_n + x_{n-1} x_n z_{n-1}) \right) \right]_{t=x_n} \\
&= f[x_n, z_{n-1}] + f[x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1}) + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}] \\
&\quad \times (x_n^2 - z_{n-1} x_n + z_{n-1} y_{n-1} - y_{n-1} x_n) + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}] \\
&\quad \times \left( x_n^3 - x_n^2 z_{n-1} - x_n^2 y_{n-1} - x_n^2 x_{n-1} + x_n z_{n-1} y_{n-1} + x_n y_{n-1} x_{n-1} + x_n x_{n-1} z_{n-1} \right. \\
&\quad \left. - z_{n-1} y_{n-1} x_{n-1} \right) \\
&= f[x_n, z_{n-1}] + f[x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1}) \\
&\quad + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}](x_n - z_{n-1})(x_n - y_{n-1}) \\
&\quad + f[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}](x_n - z_{n-1})(x_n - y_{n-1})(x_n - x_{n-1}).
\end{aligned} \right. \quad (2.4.4)$$

Therefore, the two-step method and the three-step method with memory can be presented

as

$$\left\{ \begin{aligned}
y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, \\
w_n &= x_n + \gamma_n f(x_n), \\
x_{n+1} &= y_n - \frac{f(y_n)}{P_2'(y_n)},
\end{aligned} \right. \quad (2.4.5)$$

and

$$\left\{ \begin{array}{l} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, \\ w_n = x_n + \gamma_n f(x_n), \\ z_n = y_n - \frac{f(y_n)}{P_2'(y_n)}, \\ \bar{x} = x_{n+1} = z_n - \frac{f(z_n)}{P_3'(z_n)} \end{array} \right. \quad (2.4.6)$$

respectively.

**Lemma 2.4.1** *If  $\gamma_n = -1/N_3'(x_n)$ ,  $e_n = x_n - \xi$ ,  $e_{n,y} = y_n - \xi$ , and  $e_{n,w} = w_n - \xi$ , then the following relation holds:*

$$1 + \gamma_n f'(\xi) \sim c_4 e_{n-1} e_{n-1,w} e_{n-1,y} \sim e_{n-1} e_{n-1,w} e_{n-1,y}. \quad (2.4.7)$$

**Proof** Following the same terminology as in Theorem 2.3.2 and the symbolic software Mathematica 10, it would be easy to obtain (2.4.7) via writing the following code:

```
ClearAll["Global`*"]
A[t_] := InterpolatingPolynomial[{{e, fx}, {ew, fw}, {ey, fy}, {e1, fx1}}, t] // Simplify
Approximation = -1/A'[e1] // Simplify;
fx = fla * (e + c2 * e^2 + c3 * e^3 + c4 * e^4);
fw = fla * (ew + c2 * ew^2 + c3 * ew^3 + c4 * ew^4);
fy = fla * (ey + c2 * ey^2 + c3 * ey^3 + c4 * ey^4);
fx1 = fla * (e1 + c2 * e1^2 + c3 * e1^3 + c4 * e1^4);
b = Series[Approximation, {e, 0, 2}, {ew, 0, 2}, {ey, 0, 2}, {e1, 0, 2}] // Simplify;
Collect[Series[1 + b * fla, {e, 0, 1}, {ew, 0, 1}, {ey, 0, 1}, {e1, 0, 1}], e, ew, ey, e1, Simplify].
```

The proof is complete.

In order to obtain the R-order of convergence[9] of the method with memory (2.4.5), we establish the following theorem.

**Theorem 2.4.2** *If an initial approximation  $x_0$  is sufficiently close to the zero  $\xi$  of  $f(x)$  and the parameter  $\gamma_n$  in the iterative scheme(2.4.5) is recursively calculated by the forms given in (2.4.1), then the R-order of convergence for (2.4.5) is at least six.*

**Proof** Let  $\{x_n\}$  be a sequence of approximations generated by the iterative method with memory (2.4.5). If this sequence converges to the zero  $\xi$  of  $f$  with the order  $p$ , then we write

$$e_{n+1} \sim e_n^p, \quad e_n = x_n - \xi. \quad (2.4.8)$$

Thus,

$$e_{n+1} \sim (e_{n-1}^p)^p = e_{n-1}^{p^2}. \quad (2.4.9)$$

Moreover, assume that the iterative sequence  $w_n$  and  $y_n$  have the orders  $p_1$  and  $p_2$ , respectively. Then, (2.4.8) gives

$$e_{n,w} \sim e_n^{p_1} \sim (e_{n-1}^p)^{p_1} = e_{n-1}^{pp_1}, \quad (2.4.10)$$

$$e_{n,y} \sim e_n^{p_2} \sim (e_{n-1}^p)^{p_2} = e_{n-1}^{pp_2}. \quad (2.4.11)$$

Since

$$e_{n,w} \sim (1 + \gamma_n f'(\xi))e_n, \quad (2.4.12)$$

$$e_{n,y} \sim (1 + \gamma_n f'(\xi))e_n^2, \quad (2.4.13)$$

$$e_{n+1} \sim (1 + \gamma_n f'(\xi))^2 e_n^4, \quad (2.4.14)$$

using Lemma 2.4.1 and (2.4.13), induce

$$e_{n,w} \sim (1 + \gamma_n f'(\xi))e_n \sim (e_{n-1}e_{n-1,y}e_{n-1,w})e_n = e_{n-1}^{p+p_1+p_2+1}, \quad (2.4.15)$$

$$e_{n,y} \sim (1 + \gamma_n f'(\xi))e_n^2 \sim (e_{n-1}e_{n-1,y}e_{n-1,w})e_n^2 = e_{n-1}^{2p+p_1+p_2+1}, \quad (2.4.16)$$

$$e_{n+1} \sim (1 + \gamma_n f'(\xi))^2 e_n^4 \sim (e_{n-1}e_{n-1,y}e_{n-1,w})^2 e_n^4 = e_{n-1}^{4p+2p_1+2p_2+2}. \quad (2.4.17)$$

Matching the powers of  $e_{n-1}$  on the right hand sides of (2.4.10)-(2.4.15), (2.4.11)-(2.4.16), and (2.4.9)-(2.4.17), one can obtain

$$\begin{cases} pp_1 - p - p_1 - p_2 - 1 = 0, \\ pp_2 - 2p - p_1 - p_2 - 1 = 0, \\ p^2 - 4p - 2p_1 - 2p_2 - 2 = 0. \end{cases} \quad (2.4.18)$$

The nontrivial solution of this system is  $p_1 = 2$ ,  $p_2 = 3$  and  $p = 6$ . This completes the proof.

Using symbolic computations and Taylor expansions, it is easy to derive the following Lemma.

**Lemma 2.4.3** *Assuming (2.4.3) and (2.4.4), we have*

$$1 + \gamma_n f'(\xi) \sim e_{n-1} e_{n-1,w}, e_{n-1,y}, e_{n-1,z}, \quad (2.4.19)$$

where  $\gamma_n = -1/N'_4(x_n)$ ,  $e_n = x_n - \xi$ ,  $e_{n,z} = z_n - \xi$ ,  $e_{n,y} = y_n - \xi$ , and  $e_{n,w} = w_n - \xi$ .

**Proof** The proof of this lemma is similar to Lemma (2.4.1). It is hence omitted.

Similarly for the three-step method with memory (2.4.6), we have the following theorem.

**Theorem 2.4.4** *If an approximation  $x_0$  is sufficiently close to the zero  $\xi$  of  $f(x)$  and the parameter  $\gamma_n$  in the iterative scheme (2.4.6) is recursively calculated by the forms given in (2.4.3), then the order of convergence for (2.4.6) is at least twelve.*

**Proof** Let  $\{x_n\}$  be a sequence of approximations generated by the iterative method with memory (2.4.6). If this sequence converges to the zero  $\xi$  of  $f$  with the order  $p$ , then we write

$$e_{n+1} \sim e_n^p, \quad e_n = x_n - \xi. \quad (2.4.20)$$

So,

$$e_{n+1} \sim (e_{n-1}^p)^p = e_{n-1}^{p^2}. \quad (2.4.21)$$

Moreover, assume that the iterative sequences  $w_n$ ,  $y_n$ ,  $z_n$  have the order  $p_1$ ,  $p_2$  and  $p_3$ , respectively. Then, (2.4.20) gives

$$e_{n,w} \sim e_n^{p_1} \sim (e_{n-1}^p)^{p_1} = e_{n-1}^{pp_1}, \quad (2.4.22)$$

$$e_{n,y} \sim e_n^{p_2} \sim (e_{n-1}^p)^{p_2} = e_{n-1}^{pp_2}, \quad (2.4.23)$$

$$e_{n,z} \sim e_n^{p_3} \sim (e_{n-1}^p)^{p_3} = e_{n-1}^{pp_3}. \quad (2.4.24)$$

Since

$$e_{n,w} \sim (1 + \gamma_n f'(\xi)) e_n, \quad (2.4.25)$$

$$e_{n,y} \sim (1 + \gamma_n f'(\xi)) e_n^2, \quad (2.4.26)$$

$$e_{n,z} \sim (1 + \gamma_n f'(\xi))^2 e_n^4, \quad (2.4.27)$$

$$e_{n+1} \sim (1 + \gamma_n f'(\xi))^4 e_n^8, \quad (2.4.28)$$

by Lemma 2.4.3 and (2.4.26), we obtain

$$\begin{aligned} e_{n,w} &\sim (1 + \gamma_n f'(\xi))e_n \sim (e_{n-1}e_{n-1,y}e_{n-1,w}e_{n-1,z})e_n \\ &= e_{n-1}^{p+p_1+p_2+p_3+1}, \end{aligned} \quad (2.4.29)$$

$$\begin{aligned} e_{n,y} &\sim (1 + \gamma_n f'(\xi))e_n^2 \sim (e_{n-1}e_{n-1,y}e_{n-1,w}e_{n-1,z})e_n^2 \\ &= e_{n-1}^{2p+p_1+p_2+p_3+1}, \end{aligned} \quad (2.4.30)$$

$$\begin{aligned} e_{n,z} &\sim (1 + \gamma_n f'(\xi))^2 e_n^4 \sim (e_{n-1}e_{n-1,y}e_{n-1,w}e_{n-1,z})^2 e_n^4 \\ &= e_{n-1}^{4p+2p_1+2p_2+2p_3+2}, \end{aligned} \quad (2.4.31)$$

$$\begin{aligned} e_{n+1} &\sim (1 + \gamma_n f'(\xi))^4 e_n^8 \sim (e_{n-1}e_{n-1,y}e_{n-1,w}e_{n-1,z})^4 e_n^8 \\ &= e_{n-1}^{8p+4p_1+4p_2+4p_3+4}, \end{aligned} \quad (2.4.32)$$

Matching the powers of  $e_{n-1}$  on the right hand sides of (2.4.22)-(2.4.29), (2.4.23)-(2.4.30), (2.4.24)- (2.4.31), and (2.4.21)-(2.4.32), one can obtain

$$\begin{aligned} pp_1 - p - p_1 - p_2 - p_3 - 1 &= 0, \\ pp_2 - 2p - p_1 - p_2 - p_3 - 1 &= 0, \\ pp_3 - 4p - 2p_1 - 2p_2 - 2p_3 - 2 &= 0, \\ p^2 - 8p - 4p_1 - 4p_2 - 4p_3 - 4 &= 0. \end{aligned} \quad (2.4.33)$$

This system has the solutions  $p_1 = 2$ ,  $p_2 = 3$ ,  $p_3 = 6$ , and  $p = 12$ . The proof is complete.

Note that a divided difference of order  $m$ , defined recursively as

$$f[x_0, x_1, \dots, x_m] = \frac{f[x_1, \dots, x_m] - f[x_0, x_1, \dots, x_{m-1}]}{x_m - x_0}, m \geq 2. \quad (2.4.34)$$

has been used throughout this thesis.

## Chapter 3

# Construction of new iterative families based on power means

The well-known Newton's formula (1.12.1) and Modified elliptical method[10] for simple zeros are given by:

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}, \quad (3.0.1)$$

and

$$x_{n+1} = x_n - \frac{g(x_n)}{\sqrt{\{g'(x_n)\}^2 + \mu^2\{g(x_n)\}^2}} \text{ respectively, where } \mu \in \mathbb{R} - \{0\}. \quad (3.0.2)$$

Equation (3.0.2) can be rewritten as,

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \frac{1}{\left[1 + \mu^2 \left\{\frac{g(x_n)}{g'(x_n)}\right\}^2\right]^{\frac{1}{2}}}. \quad (3.0.3)$$

Since  $\left|\frac{g(x_n)}{g'(x_n)}\right| < 1$ , therefore by using binomial expansion, and neglecting higher order terms one can get,

$$\begin{aligned} x_{n+1} &= x_n - \frac{g(x_n)}{g'(x_n)} \frac{1}{\left[1 + \frac{\mu^2}{2} \left\{\frac{g(x_n)}{g'(x_n)}\right\}^2\right]}, \\ &= x_n - \frac{g(x_n)}{g'(x_n)} \left[\frac{\{g'(x_n)\}^2}{\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2}\right], \text{ where } \frac{\mu^2}{2} = \alpha \in \mathbb{R} - \{0\}. \end{aligned} \quad (3.0.4)$$

Combining (3.0.1) and (3.0.4), one can have

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \left[ \frac{\{g'(x_n)\}^2}{\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2} + 1 \right]. \quad (3.0.5)$$

Let  $a = \frac{\{g'(x_n)\}^2}{\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2}$  and  $b = 1$ . For the quantities  $a$  and  $b$  to be positive and different from zero, one can see that

$$\{g'(x_n)\}^2 < (\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2).$$

The quantity  $b = 1$  is obviously positive real number. In the case if  $x \approx x_n$  is a very good approximation to the root, then  $g(x_n)$  will sufficiently close to zero and consequently the quantity

$$\left| \frac{\{g'(x_n)\}^2}{\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2} \right|, \quad (3.0.6)$$

would be sufficiently small.

We wish to generalize the formula (3.0.5) by  $\beta^{th}$ -power mean[2]. For this we approximate the correction factor in (3.0.5) by  $\beta^{th}$ -power mean as follows:

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)} \left\{ \frac{\left( \frac{\{g'(x_n)\}^2}{\{g'(x_n)\}^2 + \alpha\{g(x_n)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}. \quad (3.0.7)$$

This is a new  $\beta^{th}$ -power mean iterative family.

### 3.1 Development of new two-step and three-step optimal iterative families without memory

#### (a) Two-step optimal family

Let us start with the following two-step scheme:

$$\left. \begin{aligned} y &= x - \frac{g(x)}{g'(x)}, \\ \bar{x} &= y - \frac{g(y)}{g'(y)} \left\{ \frac{\left( \frac{\{g'(y)\}^2}{\{g'(y)\}^2 + \alpha\{g(y)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta} \end{aligned} \right\}. \quad (3.1.1)$$

Note that we omit the iteration index  $n$  for the sake of simplicity only. The order of convergence of scheme (3.1.1) is four, but it does not contain any contribution. For contributive point of view, we approximate the derivatives involved in first and second steps by using Neville's interpolating polynomial method as follows. By using points  $x$  and  $w = x + \gamma g(x)$  ( $\gamma \in \mathbb{R} - \{0\}$ ), Neville's approximating interpolating method can be written as

$$Q_1(t) = \frac{t-w}{x-w}g(x) + \frac{t-x}{w-x}g(w). \quad (3.1.2)$$

Fix  $g'(x) \approx Q_1'(t)$ , and by setting  $t = x$ , one gets

$$g[x, w] = \frac{g(x)}{x-w} + \frac{g(w)}{w-x} = \frac{g(x) - g(w)}{x-w}. \quad (3.1.3)$$

Also, Neville's interpolating method for approximation  $g'(x)$  polynomial at points  $x$ ,  $w$ , and  $y$  as follows:

$$Q_2(t) = \frac{(t-y)(t-w)}{(x-y)(x-w)}g(x) + \frac{(t-y)(t-x)}{(w-y)(w-x)}g(w) + \frac{(t-x)(t-w)}{(y-x)(y-w)}g(y). \quad (3.1.4)$$

Fix  $g'(y) \approx g[x, w, y]$  and by setting  $t = y$ , one can have

$$\begin{aligned} g[x, w, y] &= \frac{2y - (y+w)}{(x-y)(x-w)}g(x) + \frac{2y - (x+y)}{(w-x)(w-y)}g(w) + \frac{2y - (x+w)}{(y-x)(y-w)}g(y) \\ &= \frac{(y-w)}{(x-y)(x-w)}g(x) + \frac{(y-x)}{(w-x)(w-y)}g(w) + \frac{2y-x-w}{(y-x)(y-w)}g(y). \end{aligned} \quad (3.1.5)$$

Finally, iterative-family (3.1.1) is now modified as follows:

$$z = y - \frac{g(y)}{g[x, w, y]} \left\{ \frac{\left( \frac{g[x, w, y]^2}{g[x, w, y]^2 + \alpha g(y)^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}. \quad (3.1.6)$$

This is a new fourth order derivative free optimal iterative family.

**(b) Three-step optimal family**

For three-step iterative, let us consider the following modifications:

$$\left. \begin{aligned} y &= x - \frac{g(x)}{g[x, w]}, \quad w = x + \gamma g(x), \\ z &= y - \frac{g(y)}{g[x, w, y]} \left\{ \frac{\left( \frac{\{g[x, w, y]\}^2}{\{g[x, w, y]\}^2 + \alpha \{g(y)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}, \\ \hat{x} &= z - \frac{g(z)}{g'(z)} \left\{ \frac{\left( \frac{\{g'(z)\}^2}{\{g'(z)\}^2 + \alpha \{g(z)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}. \end{aligned} \right\} \quad (3.1.7)$$

The derivative  $g'(z)$  in the third-step of (3.1.7) should be substituted by a suitable approximation in order to obtain as high as possible of convergence order and to make the scheme optimal. To provide this approximation, one can apply Neville's interpolating method for approximating  $g'(z)$  polynomial at the four points  $w, x, y$  and  $z$ ; that is

$$\left. \begin{aligned} Q_3(t) &= \frac{(t-w)(t-y)(t-z)}{(x-w)(x-y)(x-z)}g(x) + \frac{(t-x)(t-y)(t-z)}{(w-x)(w-y)(w-z)}g(w) \\ &+ \frac{(t-x)(t-w)(t-z)}{(y-x)(y-w)(y-z)}g(y) + \frac{(t-x)(t-w)(t-y)}{(z-x)(z-w)(z-y)}g(z) \end{aligned} \right\} \quad (3.1.8)$$

It is obvious that  $Q_3(z) = g(z)$ . By differentiating (3.1.8) and setting  $t = z$ , one can obtain

$$\left. \begin{aligned} g[x, w, y, z] &= \frac{3z^2 - 2(w+y+z)z + (wy+yz+wz)}{(x-w)(x-y)(x-z)}g(x) \\ &+ \frac{3z^2 - 2(x+y+z)z + (xy+yz+xz)}{(w-x)(w-y)(w-z)}g(w) \\ &+ \frac{3z^2 - 2(x+w+z)z + (xw+wz+xz)}{(y-x)(y-w)(y-z)}g(y) \\ &+ \frac{3z^2 - 2(x+w+y)z + (xw+wy+xy)}{(z-x)(z-w)(z-y)}g(z) \\ &= \frac{(z-y)(z-w)}{(x-w)(x-y)(x-z)}g(x) + \frac{(z-y)(z-x)}{(w-x)(w-y)(w-z)}g(w) \\ &+ \frac{(z-x)(z-w)}{(y-x)(y-w)(y-z)}g(y) \\ &+ \frac{3z^2 - 2(x+w+y)z + (xw+wy+xy)}{(z-x)(z-w)(z-y)}g(z). \end{aligned} \right\} \quad (3.1.9)$$

By substituting  $g'(z) \approx g[x, w, y, z]$  in (3.1.7), one can have

$$\left. \begin{aligned} y &= x - \frac{g(x)}{g[x, w]}, & w &= x + \gamma g(x), \\ z &= y - \frac{g(y)}{g[x, w, y]} \left\{ \frac{\left( \frac{\{g[x, w, y]\}^2}{\{g[x, w, y]\}^2 + \alpha \{g(y)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}, \\ \hat{x} &= z - \frac{g(z)}{g[x, w, y, z]} \left\{ \frac{\left( \frac{\{g[x, w, y, z]\}^2}{\{g[x, w, y, z]\}^2 + \alpha \{g(z)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}. \end{aligned} \right\} \quad (3.1.10)$$

This is a new eighth order derivative free optimal iterative family.

## 3.2 Convergence analysis

Using Taylor's series and symbolic computation in programming package MAPLE, one can find the convergence order and the asymptotic error constant (AEC) of methods (3.1.6) and (3.1.10). Let us define the errors  $e = x - \xi$ ,  $ew = w - \xi$ ,  $ey = y - \xi$ ,  $ez = z - \xi$ , and  $cn = g^n(\xi)/(n!g'(\xi))$ . Let us introduce the following observation used in the program  $g_{x,w} = g[x, w]$ ,  $g_{x,w,y} = g[x, w, y]$ ,  $g_{x,w,y,z} = g[x, w, y, z]$ ,  $gx = g(x)$ ,  $gw = g(w)$ ,  $gy = g(y)$  and  $gz = g(z)$ .

*Program Written in Maple.* Consider the following:

$$gx := g'(\xi) \cdot (e + c2 \cdot e^2 + c3 \cdot e^3 + c4 \cdot e^4 + c5 \cdot e^5 + c6 \cdot e^6 + c7 \cdot e^7 + c8 \cdot e^8 + c9 \cdot e^9 + c10 \cdot e^{10}) :$$

$$ew := e + \gamma \cdot gx :$$

$$ew := g'(\xi) \cdot (e + c2 \cdot e^2 + c3 \cdot e^3 + c4 \cdot e^4 + c5 \cdot e^5 + c6 \cdot e^6 + c7 \cdot e^7 + c8 \cdot e^8 + c9 \cdot e^9 + c10 \cdot e^{10})\gamma + e :$$

$$gw := g'(\xi) \cdot (ew + c2 \cdot ew^2 + c3 \cdot ew^3 + c4 \cdot ew^4 + c5 \cdot ew^5 + c6 \cdot ew^6 + c7 \cdot ew^7 + c8 \cdot ew^8 + c9 \cdot ew^9 + c10 \cdot ew^{10}) :$$

$$gw := \text{simplify}(\text{taylor}(gw, e = 0, 10)) :$$

$$g_{x,w} := \frac{gx - gw}{e - ew} :$$

$g_{x,w} := \text{simplify}(\text{taylor}(g_{x,w}, e = 0, 10)) :$

$$ey := e - \frac{gx}{g_{x,w}} \cdot \beta \sqrt{\left( \frac{\left( \frac{g_{x,w} \cdot g_{x,w}}{g_{x,w} \cdot g_{x,w} + \alpha \cdot gx \cdot gx} \right)^\beta + 1}{2} \right)} :$$

$ey := \text{simplify}(\text{taylor}(ey, e = 0, 10)) :$

$\text{factor}(\text{simplify}(\text{taylor}(ez, e = 0, 3)))$

$\text{Out}[a] : c2(g'(\xi)\gamma + 1)e^2 + \mathbf{O}(e^3)$

$$gy := g'(\xi) \cdot (ey + c2 \cdot ey^2 + c3 \cdot ey^3 + c4 \cdot ey^4 + c5 \cdot ey^5 + c6 \cdot ey^6 + c7 \cdot ey^7 + c8 \cdot ey^8 + c9 \cdot ey^9 + c10 \cdot ey^{10}) :$$

$gy := \text{simplify}(\text{taylor}(gy, e = 0, 10)) :$

$$g_{x,w,y} := \frac{ey - ew}{(e - ew) \cdot (e - ey)} \cdot gx + \frac{ey - e}{(ew - e) \cdot (ew - ey)} \cdot gw + \frac{2 \cdot ey - ew - e}{(ey - ew) \cdot (ey - e)} \cdot gy :$$

$g_{x,w,y} := \text{simplify}(\text{taylor}(g_{x,w,y}, e = 0, 10)) :$

$$ez := ey - \frac{gy}{g_{x,w,y}} \cdot \beta \sqrt{\left( \frac{\left( \frac{g_{x,w,y} \cdot g_{x,w,y}}{g_{x,w,y} \cdot g_{x,w,y} + \alpha \cdot gy \cdot gy} \right)^\beta + 1}{2} \right)} :$$

$ez := \text{simplify}(\text{taylor}(ez, e = 0, 10)) :$

$\text{factor}(\text{simplify}(\text{taylor}(ez, e = 0, 5)))$

$\text{Out}[b] : c2(g'(\xi)\gamma + 1)^2(c2^2 - c3)e^4 + \mathbf{O}(e^5)$

$$gz := g'(\xi) \cdot (ez + c2 \cdot ez^2 + c3 \cdot ez^3 + c4 \cdot ez^4 + c5 \cdot ez^5 + c6 \cdot ez^6 + c7 \cdot ez^7 + c8 \cdot ez^8 + c9 \cdot ez^9 + c10 \cdot ez^{10}) :$$

$gz := \text{simplify}(\text{taylor}(gz, e = 0, 10)) :$

$$g_{x,w,y,z} := \frac{ew \cdot ey + ew \cdot ez + ey \cdot ez - 2 \cdot (ew + ey + ez) \cdot ez + 3 \cdot ez \cdot ez}{(e - ew) \cdot (e - ey) \cdot (e - ez)} \cdot gx + \frac{e \cdot ey + e \cdot ez + ey \cdot ez - 2 \cdot (e + ey + ez) \cdot ez + 3 \cdot ez \cdot ez}{(ew - e) \cdot (ew - ey) \cdot (ew - ez)} \cdot gw + \frac{ew \cdot e + e \cdot ez + ew \cdot ez - 2 \cdot (e + ew + ez) \cdot ez + 3 \cdot ez \cdot ez}{(ey - e)(ey - ew)(ey - ez)} \cdot gy + \frac{ew \cdot e + e \cdot ey + ey \cdot ew - 2 \cdot (e + ew + ey) \cdot ez + 3 \cdot ez \cdot ez}{(ez - ew) \cdot (ez - e) \cdot (ez - ey)} \cdot gz :$$

$$\begin{aligned}
g_{x,w,y,z} &:= \text{simplify}(\text{taylor}(g_{x,w,y,z}, e = 0, 10)) : \\
eu &:= ez - \frac{gz}{g_{x,w,y,z}} \cdot \sqrt[\beta]{\left(\frac{\left(\frac{g_{x,w,y,z} \cdot g_{x,w,y,z}}{g_{x,w,y,z} \cdot g_{x,w,y,z} + \alpha \cdot gz \cdot gz}\right)^\beta + 1}{2}\right)} : \\
eu &:= \text{factor}(\text{simplify}(\text{taylor}(eu, e = 0, 10))) : \\
&\text{factor}(\text{simplify}(\text{taylor}(eu, e = 0, 9))) \\
\text{Out}[c] &: c^2(g'(\xi)\gamma + 1)^4(c^2 - c_3)(c^3 - c_2c_3 + c_4)e^8 + \mathbf{O}(e^9)
\end{aligned}$$

The outputs Out[b] and Out[c] of the above program means that the convergence order of iterative families (3.1.6) and (3.1.10) are four and eight respectively. Altogether, we can state the following theorem.

**Theorem 3.2.1** *Let  $g : D \subset \mathbb{R} \rightarrow \mathbb{R}$  be a scalar function which has the simple root  $\xi$  in the open interval  $D$ ; also initial approximation  $x_0$  is sufficiently close to a simple zero  $\xi$  of  $g$ .*

Thus new method defined by (3.1.6) is of fourth order, and satisfies error equation

$$c_2(g'(\xi)\gamma + 1)^2(c_2^2 - c_3)e^4 + \mathbf{O}(e^5) \quad (3.2.1)$$

and the method defined by (3.1.10) is eighth order and satisfies error equation

$$e_{n+1} = (1 + \gamma g'(\xi))^4 c_2^2 (c_2^2 - c_3) (c_2^3 - c_2 c_3 + c_4) e_n^8 + \mathbf{O}(e_n^9) \quad (3.2.2)$$

where  $e_n = x_n - \xi$ .

### 3.3 Extension to methods with memory

This section is concerned with extraction of efficient methods with memory from (3.1.6) and (3.1.10), one can observe from (3.2.1) and (3.2.2) that the order of convergence of the families (3.1.6) and (3.1.10) is four and eight respectively, when  $\gamma \neq -1/g'(\xi)$ . If we could provide that  $\gamma = -1/g'(\xi)$ , it can be proved that the order of the families (3.1.6) and (3.1.10) would be six and twelve, respectively.

However  $g'(\xi)$  is not available in practice and such acceleration of convergence is not possible. Instead of that, one can use an approximation  $\widehat{g}'(\xi) \approx g'(\xi)$ , calculated by available information. Various methods are available for approximating  $g'(\xi)$ , but in this work Newton's

interpolating approach is used for approximation.

We set  $\gamma = \gamma_n$  as the iteration proceeds by the formula  $\gamma_n = -1/\widehat{g}'(\xi)$  for  $n = 1, 2, \dots$ , where  $\widehat{g}'(\xi)$  is an approximation of  $g'(\xi)$ . We therefore use the approximation

$$\gamma_n = -\frac{1}{\widehat{g}'(\xi)} = -\frac{1}{N'_3(x_n)}, \quad (3.3.1)$$

for (3.1.6). Here,  $N_3(t) = N_3(t; x_n, x_{n-1}, y_{n-1}, w_{n-1})$  is Newton's interpolating polynomial of degree three, set through best four available approximations  $x_n, z_{n-1}, y_{n-1}, w_{n-1}, x_{n-1}$  and

$$\left\{ \begin{aligned} N'_3(x_n) &= \left. \frac{d}{dt} N_3(t) \right|_{t=x_n} \\ &= \left. \frac{d}{dt} \left[ g(x_n) + g[x_n, x_{n-1}](t - x_n) + g[x_n, x_{n-1}, y_{n-1}](t - x_n)(t - x_{n-1}) \right. \right. \\ &\quad \left. \left. + g[x_n, x_{n-1}, y_{n-1}, w_{n-1}](t - x_n)(t - x_{n-1})(t - y_{n-1}) \right] \right|_{t=x_n} \\ &= \left[ g[x_n, x_{n-1}] \cdot 1 + g[x_n, x_{n-1}, y_{n-1}](2t - (x_n + x_{n-1})) + g[x_n, x_{n-1}, y_{n-1}, w_{n-1}] \right. \\ &\quad \left. \times (3t^2 - 2t(x_n + x_{n-1} + y_{n-1}) + (x_n x_{n-1} + x_{n-1} y_{n-1} + y_{n-1} x_n)) \right]_{t=x_n} \\ &= g[x_n, x_{n-1}] + g[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) + g[x_n, x_{n-1}, y_{n-1}, w_{n-1}] \\ &\quad \times (3x_n^2 - 2x_n(x_n + x_{n-1} + y_{n-1}) + (x_n x_{n-1} + x_{n-1} y_{n-1} + y_{n-1} x_n)) \\ &= g[x_n, x_{n-1}] + g[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) \\ &\quad + g[x_n, x_{n-1}, y_{n-1}, w_{n-1}](x_n^2 - x_n x_{n-1} - y_{n-1} x_n + x_{n-1} y_{n-1}) \\ &= g[x_n, x_{n-1}] + g[x_n, x_{n-1}, y_{n-1}](x_n - x_{n-1}) \\ &\quad + g[x_n, x_{n-1}, y_{n-1}, w_{n-1}](x_n - x_{n-1})(x_n - y_{n-1}). \end{aligned} \right. \quad (3.3.2)$$

Hence the with memory extension of iterative scheme (3.1.6) can be presented as follows:

$$\left. \begin{aligned} y_n &= x_n - \frac{g(x_n)}{g[x_n, w_n]}, \quad w_n = x_n + \gamma_n g(x_n), \\ x_{n+1} &= y_n - \frac{g(y_n)}{g[x_n, w_n, y_n]} \left\{ \frac{\left( \frac{\{g[x_n, w_n, y_n]\}^2}{\{g[x_n, w_n, y_n]\}^2 + \alpha \{g(y_n)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}, \\ &\quad \text{where } \gamma_n = -\frac{1}{N'_3(x_n)}. \end{aligned} \right\} \quad (3.3.3)$$

This is a new sixth order iterative scheme with memory.

Further we use approximation

$$\gamma_n = -\frac{1}{\widehat{g}'(\xi)} = -\frac{1}{N_4'(x_n)}, \quad (3.3.4)$$

for (3.1.10), where  $N_4(t) = N_4(t; x_n, z_{n-1}, y_{n-1}, w_{n-1}, x_{n-1})$  is Newton's interpolating polynomial of degree four, set through best five available approximations  $x_n, z_{n-1}, y_{n-1}, w_{n-1}, x_{n-1}$  and

$$\left\{ \begin{aligned} N_4'(x_n) &= \frac{d}{dt} N_4(t) \Big|_{t=x_n} \\ &= \frac{d}{dt} \left[ g(x_n) + g[x_n, z_{n-1}](t - x_n) + g[x_n, z_{n-1}, y_{n-1}](t - x_n)(t - z_{n-1}) \right. \\ &\quad + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}](t - x_n)(t - z_{n-1})(t - y_{n-1}) \\ &\quad \left. + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}](t - x_n)(t - z_{n-1})(t - y_{n-1})(t - x_{n-1}) \right]_{t=x_n} \\ &= \left[ g[x_n, z_{n-1}] \cdot 1 + g[x_n, z_{n-1}, y_{n-1}](2t - (x_n + z_{n-1})) + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}] \right. \\ &\quad \times (3t^2 - 2t(x_n + z_{n-1} + y_{n-1}) + (x_n z_{n-1} + z_{n-1} y_{n-1} + y_{n-1} x_n)) \\ &\quad + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}] \left( 4t^3 - 3t^2(x_n + z_{n-1} + y_{n-1} + x_{n-1}) \right. \\ &\quad + 2t(x_n z_{n-1} + z_{n-1} y_{n-1} + y_{n-1} x_{n-1} + x_{n-1} x_n + x_n y_{n-1} + z_{n-1} x_{n-1}) \\ &\quad \left. \left. - (x_n z_{n-1} y_{n-1} + z_{n-1} y_{n-1} x_{n-1} + y_{n-1} x_{n-1} x_n + x_{n-1} x_n z_{n-1}) \right) \right]_{t=x_n} \\ &= g[x_n, z_{n-1}] + g[x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1}) + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}] \\ &\quad \times (x_n^2 - z_{n-1} x_n + z_{n-1} y_{n-1} - y_{n-1} x_n) + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}] \\ &\quad \times \left( x_n^3 - x_n^2 z_{n-1} - x_n^2 y_{n-1} - x_n^2 x_{n-1} + x_n z_{n-1} y_{n-1} + x_n y_{n-1} x_{n-1} + x_n x_{n-1} z_{n-1} \right. \\ &\quad \left. - z_{n-1} y_{n-1} x_{n-1} \right) \\ &= g[x_n, z_{n-1}] + g[x_n, z_{n-1}, y_{n-1}](x_n - z_{n-1}) \\ &\quad + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}](x_n - z_{n-1})(x_n - y_{n-1}) \\ &\quad + g[x_n, z_{n-1}, y_{n-1}, x_{n-1}, w_{n-1}](x_n - z_{n-1})(x_n - y_{n-1})(x_n - x_{n-1}). \end{aligned} \right. \quad (3.3.5)$$

Hence with memory extension of iterative scheme (3.1.10) can be presented as follows:

$$\left. \begin{aligned}
 y_n &= x_n - \frac{g(x_n)}{g[x_n, w_n]}, \quad w_n = x_n + \gamma_n g(x_n), \\
 z_n &= y_n - \frac{g(y_n)}{g[x_n, w_n, y_n]} \left\{ \frac{\left( \frac{\{g[x_n, w_n, y_n]\}^2}{\{g[x_n, w_n, y_n]\}^2 + \alpha \{g(y_n)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}, \\
 \widehat{x}_n = x_{n+1} &= z_n - \frac{g(z_n)}{g[x_n, w_n, y_n, z_n]} \left\{ \frac{\left( \frac{\{g[x_n, w_n, y_n, z_n]\}^2}{\{g[x_n, w_n, y_n, z_n]\}^2 + \alpha \{g(z_n)\}^2} \right)^\beta + (1)^\beta}{2} \right\}^{1/\beta}, \\
 &\text{where } \gamma_n = -\frac{1}{N'_4(x_n)}.
 \end{aligned} \right\} (3.3.6)$$

This is a new twelfth order iterative scheme with memory.

# Chapter 4

## Numerical illustrations

In this section, we test our proposed methods ((3.1.6), (3.1.10), (3.3.3) and (3.3.6)) and compared the results with already existing methods of the same order of convergence. These results are calculated by using the programming in software Mathematica 10 with 4000 arbitrary precision arithmetic. We found that the results are comparable with existing counter parts.

The errors  $|x_n - \xi|$  denote approximations to the sought zeros and  $i(-j)$  stands for  $i \times 10^{-j}$ . Moreover, the order of convergence computed as

$$coc = \frac{\log(|f(x_n)/f(x_{n-1})|)}{\log(|f(x_{n-1})/f(x_{n-2})|)}, \quad (4.0.1)$$

where *coc* stands for order of convergence.

For numerical comparison purposes, we consider the following existing methods:

**Derivative-free two-step method by Lotfi et al. (LM4)[1]:**

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - \frac{f(y_n)}{f[x_n, w_n, y_n]}, \end{cases} \quad (4.0.2)$$

where  $\gamma_n \in \mathbb{R} - \{0\}$ .

**Derivative-free two-step method by Kung and Traub (KT4)[6]:**

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - \frac{f(y_n)f(w_n)}{[f(w_n) - f(y_n)]f[x_n, y_n]}, \end{cases} \quad (4.0.3)$$

where  $\gamma_n \in \mathbb{R} - \{0\}$ .

**Derivative-free two-step method by Lotfi and Tavakoli (LT4)[11]:**

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n = x_n + \gamma_n f(x_n), \\ x_{n+1} = y_n - H(t_n, u_n) \frac{f(y_n)}{f[y_n, w_n]}, \\ t_n = \frac{f(y_n)}{f(x_n)}, & u_n = \frac{f(w_n)}{f(x_n)}, \end{cases} \quad (4.0.4)$$

where  $H(t_n, u_n) = 1 + t_n$  and  $\gamma_n \in \mathbb{R} - \{0\}$ .

**Derivative-free three-step method by Lotfi et al. (LM8)[1]:**

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n = x_n + \gamma_n f(x_n), \\ z_n = y_n - \frac{f(y_n)}{f[x_n, w_n, y_n]}, \\ x_{n+1} = z_n - \frac{f(z_n)}{f[x_n, w_n, y_n, z_n]}, \end{cases} \quad (4.0.5)$$

where  $\gamma_n \in \mathbb{R} - \{0\}$ ,  $f[x_n, w_n, y_n] = \frac{2y_n - w_n - y_n}{(x_n - w_n)(x_n - y_n)} f(x_n) + \frac{2y_n - x_n - y_n}{(w_n - x_n)(w_n - y_n)} f(w_n) + \frac{2y_n - w_n - x_n}{(y_n - w_n)(y_n - x_n)} f(y_n)$

and

$$f[x_n, w_n, y_n, z_n] = \frac{w_n y_n + w_n z_n + y_n z_n - 2(w_n + y_n + z_n)z_n + 3z_n^2}{(x_n - w_n)(x_n - y_n)(x_n - z_n)} f(x_n) + \frac{x_n y_n + x_n z_n + y_n z_n - 2(x_n + y_n + z_n)z_n + 3z_n^2}{(w_n - x_n)(w_n - y_n)(w_n - z_n)} f(w_n) + \frac{x_n w_n + x_n z_n + w_n z_n - 2(x_n + w_n + z_n)z_n + 3z_n^2}{(y_n - w_n)(y_n - x_n)(y_n - z_n)} f(y_n) + \frac{x_n w_n + x_n y_n + w_n y_n - 2(x_n + w_n + y_n)z_n + 3z_n^2}{(z_n - w_n)(z_n - x_n)(z_n - y_n)} f(z_n). \quad (4.0.6)$$

**Derivative-free three-step method by Kung-Traub (KT8)[6]:**

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n = x_n + \gamma_n f(x_n), \\ z_n = y_n - \frac{f(y_n)f(w_n)}{[f(w_n) - f(y_n)]f[x_n, y_n]}, \\ x_{n+1} = z_n - \frac{f(y_n)f(w_n)(y_n - x_n + f(x_n)/f[x_n, z_n])}{[f(y_n) - f(z_n)][f(w_n) - f(z_n)]} + \frac{f(y_n)}{f[y_n, z_n]}, \end{cases} \quad (4.0.7)$$

where  $\gamma_n \in \mathbb{R} - \{0\}$ .

**Derivative-free three-step method by Lotfi and Tavakoli(LT8)[11]:**

$$\left. \begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, & w_n &= x_n + \gamma_n f(x_n), \\ z_n &= y_n - H(t_n, u_n) \frac{f(y_n)}{f[y_n, w_n]}, \\ x_{n+1} &= z_n - G(t_n, s_n) W(v_n, s_n) \frac{f(z_n)}{f[z_n, w_n]}, \\ t_n &= \frac{f(y_n)}{f(x_n)}, & u_n &= \frac{f(w_n)}{f(x_n)}, & s_n &= \frac{f(z_n)}{f(y_n)}, & v_n &= \frac{f(z_n)}{f(x_n)}, \end{aligned} \right\} \quad (4.0.8)$$

where  $W(s_n, v_n) = 1 + s_n^2 + v_n^2$ ,  $G(t_n, s_n) = 1 + t_n + s_n + 2t_n s_n + (-1 - \phi_n)t_n^3$  and  $(\phi_n = 1/(1 + y_n f[x_n, w_n]))$  and  $H(t_n, u_n) = 1 + t_n$  are the weight functions and  $\gamma_n \in \mathbb{R} - \{0\}$ .

It is assumed that the initial estimate  $\gamma_0$  should be chosen before starting the iterative process, and also  $x_0$  is given suitably. The following test functions are used:

Table 4.1: Test Functions

| S.No. | Test Function   | Initial guess | Exact root $\xi$ |
|-------|---|---------------|------------------|
| 1     | $f_1 = \log(1 + x^2) + e^x \sin(x)$   | $x_0 = 0.03$  | 0                |
| 2     | $f_2 = x \log(1 + x \sin(x)) + e^{-1+x^2+x \cos(x)} \sin(\pi x)$                          | $x_0 = 0.01$  | 0                |
| 3     | $f_3 = (1 + x^2) \cos\left(\frac{\pi x}{2}\right) + \frac{\log(x^2 + 2x + 2)}{(1 + x^2)}$ | $x_0 = -1.1$  | -1               |
| 4     | $f_4 = e^x - 1 - \frac{x}{5}$   | $x_0 = 0.5$   | 0                |
| 5     | $f_5 = \tan^{-1}(e^{(x+2)} + 1) + \tanh(e^{-x \cos(x)}) - \sin(\pi x)$                    | $x_0 = -3.5$  | -3.38            |
| 6     | $f_6 = \frac{1}{x^4} - x^2 - \frac{1}{x} + 1,$  | $x_0 = 1.5$   | 1                |

a) Numerical comparisons of fourth order optimal iterative methods without memory

Table 4.2:  $f(x) = \log(1 + x^2) + e^x \sin(x)$ , and  $x_0 = 0.03$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.75864(-24) | 1.06174(-94) | 2.32980(-376) | 4.00000    | 0.0501    |
| $\alpha = -1, \beta = 1/2$       | 2.57748(-24) | 8.09140(-96) | 7.85845(-377) | 4.00000    | 0.0422    |
| $\alpha = 1, \beta = 1$          | 3.36976(-24) | 2.36394(-94) | 5.72515(-375) | 4.00000    | 0.0470    |
| $\alpha = 10(-2), \beta = 2$     | 2.95463(-24) | 1.39719(-94) | 6.98661(-376) | 4.00000    | 0.0453    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.95068(-24) | 1.38972(-95) | 6.83842(-376) | 4.00000    | 0.0438    |
| LM4                              | 2.95068(-24) | 1.38972(-95) | 6.83842(-376) | 4.00000    | 0.1063    |
| KT4                              | 9.88769(-23) | 3.66401(-88) | 6.90877(-350) | 4.00000    | 0.1125    |
| LT4                              | 2.67231(-22) | 2.46494(-86) | 1.78432(-342) | 4.00000    | 0.1062    |

Table 4.3:  $f(x) = x \log(1 + x \sin(x)) + e^{-1+x^2+x \cos(x)} \sin(\pi x)$ , and  $x_0 = 0.01$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 5.82338(-10) | 3.71278(-39) | 6.13472(-156) | 4.00000    | 0.1735    |
| $\alpha = -1, \beta = 1/2$       | 5.67021(-10) | 3.33729(-39) | 4.00471(-156) | 4.00000    | 0.1688    |
| $\alpha = 1, \beta = 1$          | 6.29140(-10) | 5.05810(-39) | 2.11324(-156) | 4.00000    | 0.1736    |
| $\alpha = 10(-2), \beta = 2$     | 5.98108(-10) | 4.13158(-39) | 9.40722(-156) | 4.00000    | 0.1779    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.97797(-10) | 4.12300(-39) | 9.32935(-156) | 4.00000    | 0.1702    |
| LM4                              | 5.97797(-10) | 4.12300(-39) | 9.32935(-156) | 4.00000    | 0.1734    |
| KT4                              | 2.57585(-9)  | 6.50317(-36) | 2.64205(-142) | 4.00000    | 0.1985    |
| LT4                              | 3.38307(-9)  | 2.57332(-35) | 8.61440(-140) | 4.00000    | 0.1998    |

Table 4.4:  $f(x) = (1 + x^2) \cos(\frac{\pi x}{2}) + \frac{\log(x^2 + 2x + 2)}{(1 + x^2)}$ , and  $x_0 = -1.1$

| Methods                          | $ f(x_1) $  | $ f(x_2) $   | $ f(x_3) $   | <i>coc</i> | Time Used |
|----------------------------------|-------------|--------------|--------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 6.75591(-5) | 8.45074(-20) | 2.06877(-79) | 4.00000    | 0.1218    |
| $\alpha = -1, \beta = 1/2$       | 7.62835(-5) | 1.37372(-19) | 1.44454(-78) | 4.00000    | 0.1126    |
| $\alpha = 1, \beta = 1$          | 4.50755(-5) | 1.67452(-20) | 3.18927(-82) | 4.00000    | 0.1251    |
| $\alpha = 10(-2), \beta = 2$     | 5.93223(-5) | 5.02361(-20) | 2.58345(-80) | 4.00000    | 0.1220    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.94780(-5) | 5.07656(-20) | 2.69410(-77) | 4.00000    | 0.1264    |
| LM4                              | 5.94780(-5) | 5.07656(-20) | 2.69410(-80) | 4.00000    | 0.1281    |
| KT4                              | 1.47637(-4) | 4.89518(-18) | 5.91625(-72) | 4.00000    | 0.1311    |
| LT4                              | 9.54678(-5) | 5.59675(-19) | 6.61062(-76) | 4.00000    | 0.1313    |

Table 4.5:  $f(x) = e^x - 1 - \frac{x}{5}$ , and  $x_0 = 0.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 1.12143(-21) | 1.26696(-85) | 2.06413(-341) | 4.00000    | 0.0766    |
| $\alpha = -1, \beta = 1/2$       | 1.43079(-17) | 3.35726(-69) | 1.01770(-275) | 4.00000    | 0.075     |
| $\alpha = 1, \beta = 1$          | 1.36398(-11) | 2.77279(-45) | 4.73530(-180) | 4.00000    | 0.0799    |
| $\alpha = 10(-2), \beta = 2$     | 4.09595(-14) | 2.25475(-55) | 2.07049(-220) | 4.00000    | 0.0750    |
| $\alpha = 0.1, \beta = 10(-100)$ | 3.68493(-14) | 1.47705(-55) | 3.81291(-221) | 4.00000    | 0.0750    |
| LM4                              | 3.68493(-14) | 1.47705(-55) | 3.81291(-221) | 4.00000    | 0.0812    |
| KT4                              | 5.39642(-12) | 2.13515(-46) | 5.23257(-184) | 4.00000    | 0.0892    |
| LT4                              | 1.36961(-11) | 1.24832(-44) | 8.61477(-177) | 4.00000    | 0.0893    |

Table 4.6:  $f(x) = \tan^{-1}(\exp(x+2)+1) + \tanh(\exp(-x\cos(x))) - \sin(\pi x)$ , and  $x_0 = -3.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 9.79951(-19) | 4.62340(-72) | 2.29082(-285) | 4.00000    | 0.3703    |
| $\alpha = -1, \beta = 1/2$       | 1.36293(-18) | 1.72997(-71) | 4.49049(-283) | 4.00000    | 0.3844    |
| $\alpha = 1, \beta = 1$          | 3.43655(-19) | 6.99250(-74) | 1.19860(-292) | 4.00000    | 0.3768    |
| $\alpha = 10(-2), \beta = 2$     | 6.94856(-19) | 1.16875(-72) | 9.35474(-288) | 4.00000    | 0.3861    |
| $\alpha = 0.1, \beta = 10(-100)$ | 6.99725(-19) | 1.20186(-72) | 1.04607(-287) | 4.00000    | 0.3891    |
| LM4                              | 6.99725(-19) | 1.20186(-72) | 1.04607(-287) | 4.00000    | 0.4014    |
| KT4                              | 2.67496(-15) | 4.89633(-58) | 5.49652(-229) | 4.00000    | 0.4547    |
| LT4                              | 2.59730(-09) | 1.32300(-32) | 8.90668(-126) | 4.00000    | 0.5455    |

Table 4.7:  $f(x) = \frac{1}{x^4} - x^2 - \frac{1}{x} + 1$ , and  $x_0 = 1.5$

| Methods                          | $ f(x_1) $  | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|-------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 8.80670(-7) | 1.16956(-25) | 3.63791(-101) | 4.00000    | 0.1237    |
| $\alpha = -1, \beta = 1/2$       | 3.47899(-7) | 2.84825(-27) | 1.27962(-107) | 4.00000    | 0.1220    |
| $\alpha = 1, \beta = 1$          | 2.01785(-4) | 3.22418(-16) | 2.10109(-063) | 4.00000    | 0.1201    |
| $\alpha = 10(-2), \beta = 2$     | 2.48237(-5) | 7.38317(-20) | 5.77748(-078) | 4.00000    | 0.1162    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.38658(-5) | 6.30790(-20) | 3.07828(-078) | 4.00000    | 0.1187    |
| LM4                              | 2.38658(-5) | 6.30790(-20) | 3.07828(-078) | 4.00000    | 0.1207    |
| KT4                              | 3.68940(-5) | 3.83527(-19) | 4.47821(-075) | 4.00000    | 0.1248    |
| LT4                              | 1.93989(-4) | 2.28321(-15) | 4.38049(-059) | 4.00000    | 0.1250    |

b) Numerical comparisons of eighth order optimal iterative methods without memory

Table 4.8:  $f(x) = \log(1 + x^2) + e^x \sin(x)$ , and  $x_0 = 0.03$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.27153(-12) | 4.44007(-93) | 9.46161(-739) | 8.00000    | 0.1061    |
| $\alpha = -1, \beta = 1/2$       | 2.19410(-12) | 3.36436(-93) | 1.02818(-739) | 8.00000    | 0.1079    |
| $\alpha = 1, \beta = 1$          | 2.51577(-12) | 1.00509(-92) | 6.52362(-736) | 8.00000    | 0.1093    |
| $\alpha = 10(-2), \beta = 2$     | 2.35253(-12) | 5.87651(-93) | 8.90843(-738) | 8.00000    | 0.1092    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.35092(-12) | 5.84445(-93) | 8.52691(-738) | 8.00000    | 0.1047    |
| LM8                              | 2.35092(-12) | 5.84445(-93) | 8.52691(-738) | 8.00000    | 0.1093    |
| KT8                              | 2.33967(-11) | 6.22429(-84) | 1.56161(-664) | 8.00000    | 0.1049    |
| LT8                              | 2.12675(-11) | 2.47808(-84) | 8.41982(-668) | 8.00000    | 0.1109    |

Table 4.9:  $f(x) = x \log(1 + x \sin(x)) + e^{-1+x^2+x \cos(x)} \sin(\pi x)$ , and  $x_0 = 0.01$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 7.08005(-19) | 1.42230(-148) | 3.77259(-1186) | 8.00000    | 0.1516    |
| $\alpha = -1, \beta = 1/2$       | 6.91054(-19) | 1.17165(-148) | 7.99983(-1187) | 8.00000    | 0.1484    |
| $\alpha = 1, \beta = 1$          | 7.58874(-19) | 2.47778(-148) | 3.20036(-1184) | 8.00000    | 0.1483    |
| $\alpha = 10(-2), \beta = 2$     | 7.25302(-19) | 1.72526(-148) | 1.76823(-1185) | 8.00000    | 0.1595    |
| $\alpha = 0.1, \beta = 10(-100)$ | 7.24963(-19) | 1.71882(-148) | 1.71609(-1185) | 8.00000    | 0.1594    |
| LM8                              | 7.24963(-19) | 1.71882(-148) | 1.71609(-1185) | 8.00000    | 0.1656    |
| KT8                              | 2.36512(-18) | 6.06906(-144) | 1.14096(-1148) | 8.00000    | 0.1579    |
| LT8                              | 3.78158(-18) | 6.08458(-142) | 2.73328(-1132) | 8.00000    | 0.1735    |

Table 4.10:  $f(x) = (1 + x^2) \cos(\frac{\pi x}{2}) + \frac{\log(x^2 + 2x + 2)}{(1 + x^2)}$ , and  $x_0 = -1.1$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 4.84111(-12) | 8.67386(-98) | 9.21200(-784) | 8.00000    | 0.1015    |
| $\alpha = -1, \beta = 1/2$       | 1.15711(-10) | 9.23974(-87) | 1.52734(-695) | 8.00000    | 0.0905    |
| $\alpha = 1, \beta = 1$          | 1.50433(-10) | 7.54047(-86) | 3.00500(-688) | 8.00000    | 0.1031    |
| $\alpha = 10(-2), \beta = 2$     | 7.33852(-11) | 2.41837(-88) | 3.36386(-708) | 8.00000    | 0.0999    |
| $\alpha = 0.1, \beta = 10(-100)$ | 7.21402(-11) | 2.10899(-88) | 1.12526(-708) | 8.00000    | 0.1016    |
| LM8                              | 7.21402(-11) | 2.10899(-88) | 1.12526(-708) | 8.00000    | 0.1048    |
| KT8                              | 9.04212(-9)  | 1.96675(-69) | 9.85339(-555) | 8.00000    | 0.1015    |
| LT8                              | 1.04564(-9)  | 8.61235(-78) | 1.82410(-622) | 8.00000    | 0.1031    |

Table 4.11:  $f(x) = e^x - 1 - \frac{x}{5}$ , and  $x_0 = 0.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.53447(-9)  | 1.24378(-71) | 4.18412(-570) | 8.00000    | 0.0750    |
| $\alpha = -1, \beta = 1/2$       | 5.57321(-9)  | 6.79977(-69) | 3.33890(-548) | 8.00000    | 0.0783    |
| $\alpha = 1, \beta = 1$          | 1.123094(-5) | 3.85082(-42) | 3.53249(-334) | 8.00000    | 0.0781    |
| $\alpha = 10(-2), \beta = 2$     | 8.91387(-7)  | 2.91190(-51) | 3.77627(-407) | 8.00000    | 0.0766    |
| $\alpha = 0.1, \beta = 10(-100)$ | 8.51712(-7)  | 2.02298(-51) | 2.04918(-408) | 8.00000    | 0.0818    |
| LM8                              | 8.51712(-7)  | 2.02298(-51) | 2.04918(-408) | 8.00000    | 0.0828    |
| KT8                              | 5.79857(-6)  | 1.09894(-43) | 1.82908(-345) | 8.00000    | 0.0780    |
| LT8                              | 5.55571(-6)  | 5.87273(-44) | 9.15515(-348) | 8.00000    | 0.0797    |

Table 4.12:  $f(x) = \tan^{-1}(\exp(x+2)+1) + \tanh(\exp(-x\cos(x))) - \sin(\pi x)$ , and  $x_0 = -3.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 8.19245(-16) | 1.51673(-119) | 2.09351(-949) | 8.00000    | 0.3813    |
| $\alpha = -1, \beta = 1/2$       | 1.33869(-15) | 7.70972(-118) | 9.32064(-936) | 8.00000    | 0.3842    |
| $\alpha = 1, \beta = 1$          | 1.95852(-16) | 1.61818(-124) | 3.51402(-986) | 8.00000    | 0.4173    |
| $\alpha = 10(-2), \beta = 2$     | 5.02757(-16) | 3.05118(-121) | 5.61492(-963) | 8.00000    | 0.3826    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.07647(-16) | 3.29680(-121) | 1.04313(-962) | 8.00000    | 0.3907    |
| LM8                              | 5.07647(-16) | 3.29680(-121) | 1.04313(-962) | 8.00000    | 0.4201    |
| KT8                              | 1.11901(-07) | 1.89453(-053) | 9.11335(-420) | 8.00000    | 0.4218    |
| LT8                              | 1.90849(-03) | 9.86084(-028) | 6.84442(-222) | 8.00000    | 2.0908    |

Table 4.13:  $f(x) = \frac{1}{x^4} - x^2 - \frac{1}{x} + 1$ , and  $x_0 = 1.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 3.57866(-31) | 7.89931(-246) | 4.45187(-1963) | 8.00000    | 0.1173    |
| $\alpha = -1, \beta = 1/2$       | 6.94348(-33) | 1.58653(-259) | 1.17876(-2072) | 8.00000    | 0.1187    |
| $\alpha = 1, \beta = 1$          | 4.10362(-21) | 2.36168(-165) | 2.83897(-1319) | 8.00000    | 0.1124    |
| $\alpha = 10(-2), \beta = 2$     | 4.10362(-25) | 2.63279(-198) | 6.77888(-1583) | 8.00000    | 0.1092    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.64382(-25) | 7.00950(-199) | 1.71129(-1587) | 8.00000    | 0.1154    |
| LM8                              | 2.64382(-25) | 7.00950(-199) | 1.71129(-1587) | 8.00000    | 0.1186    |
| KT8                              | 3.20388(-27) | 2.24832(-214) | 1.32224(-1711) | 8.00000    | 0.1202    |
| LT8                              | 2.12610(-16) | 2.56973(-126) | 1.17036(-1005) | 8.00000    | 0.1187    |

c) Numerical comparisons of sixth order iterative methods with memory

Table 4.14:  $f(x) = \log(1 + x^2) + e^x \sin(x)$ , and  $x_0 = 0.03$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 4.00953(-35) | 9.03604(-206) | 1.18384(-1229) | 6.00000    | 0.1155    |
| $\alpha = -1, \beta = 1/2$       | 3.08683(-35) | 1.58107(-206) | 2.85434(-1234) | 6.00000    | 0.1108    |
| $\alpha = 1, \beta = 1$          | 8.18964(-35) | 1.01557(-203) | 3.69231(-1217) | 6.00000    | 0.1204    |
| $\alpha = 10(-2), \beta = 2$     | 5.16685(-35) | 4.86062(-205) | 3.36885(-1225) | 6.00000    | 0.1154    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.14182(-35) | 4.70680(-205) | 2.84757(-1225) | 6.00000    | 0.1141    |
| LM6                              | 5.14182(-35) | 4.70680(-205) | 2.76935(-1225) | 6.00000    | 0.1220    |
| KT6                              | 7.45663(-33) | 9.86632(-192) | 5.29457(-1145) | 6.00000    | 0.1221    |
| LT6                              | 2.37121(-32) | 1.02027(-188) | 6.47439(-1127) | 6.00000    | 0.1233    |

Table 4.15:  $f(x) = x \log(1 + x \sin(x)) + e^{-1+x^2+x \cos(x)} \sin(\pi x)$ , and  $x_0 = 0.01$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 4.65815(-56) | 6.10923(-333) | 2.19045(-1994) | 6.00055    | 0.1874    |
| $\alpha = -1, \beta = 1/2$       | 5.74638(-56) | 2.11390(-332) | 1.95793(-1991) | 6.00000    | 0.1905    |
| $\alpha = 1, \beta = 1$          | 6.17693(-57) | 1.91698(-336) | 1.81097(-2015) | 6.00000    | 0.1812    |
| $\alpha = 10(-2), \beta = 2$     | 3.37629(-56) | 1.26911(-333) | 3.57417(-1998) | 6.00000    | 0.1893    |
| $\alpha = 0.1, \beta = 10(-100)$ | 3.40255(-56) | 1.31320(-333) | 3.94881(-1998) | 6.00015    | 0.1731    |
| LM6                              | 3.40255(-56) | 1.31320(-333) | 4.34010(-1998) | 6.00000    | 0.1907    |
| KT6                              | 9.08388(-54) | 1.93450(-320) | 1.80449(-1920) | 6.00000    | 0.1860    |
| LT6                              | 1.55675(-52) | 4.90071(-313) | 4.76974(-1876) | 6.00000    | 0.1923    |

Table 4.16:  $f(x) = (1 + x^2) \cos(\frac{\pi x}{2}) + \frac{\log(x^2 + 2x + 2)}{(1 + x^2)}$ , and  $x_0 = -1.1$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 1.60260(-29) | 4.57958(-177) | 2.55988(-1061) | 5.99315    | 0.1004    |
| $\alpha = -1, \beta = 1/2$       | 1.12858(-28) | 7.57209(-174) | 1.80939(-1038) | 5.95579    | 0.1030    |
| $\alpha = 1, \beta = 1$          | 2.61662(-30) | 1.45276(-180) | 1.39959(-1082) | 6.00321    | 0.1017    |
| $\alpha = 10(-2), \beta = 2$     | 1.84543(-31) | 1.85384(-189) | 4.16701(-1138) | 6.00794    | 0.0999    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.29206(-31) | 5.51204(-190) | 5.92880(-1143) | 6.00791    | 0.1064    |
| LM6                              | 2.29206(-31) | 5.51204(-190) | 5.92880(-1143) | 6.00000    | 0.1123    |
| KT6                              | 7.46713(-27) | 1.32368(-160) | 4.10771(-963)  | 6.00000    | 0.1047    |
| LT6                              | 1.84649(-28) | 3.02668(-170) | 5.87072(-1021) | 6.00000    | 0.1047    |

Table 4.17:  $f(x) = e^x - 1 - \frac{x}{5}$ , and  $x_0 = 0.5$

| Methods                          | $ f(x_1) $    | $ f(x_2) $     | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|---------------|----------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.48714(-169) | 3.37791(-1011) | 2.64885(-6065) | 6.00345    | 0.0812    |
| $\alpha = -1, \beta = 1/2$       | 2.26855(-147) | 1.81409(-881)  | 4.30313(-5286) | 6.00006    | 0.0892    |
| $\alpha = 1, \beta = 1$          | 7.92907(-88)  | 3.15917(-523)  | 1.26338(-3135) | 6.00000    | 0.0891    |
| $\alpha = 10(-2), \beta = 2$     | 1.39507(-114) | 2.15988(-684)  | 2.97461(-4103) | 5.99998    | 0.0893    |
| $\alpha = 0.1, \beta = 10(-100)$ | 4.97679(-115) | 4.38186(-687)  | 2.63865(-4119) | 6.00000    | 0.0873    |
| LM6                              | 4.97679(-115) | 4.38186(-687)  | 2.63865(-4119) | 6.00000    | 0.0953    |
| KT6                              | 3.67710(-98)  | 1.38442(-585)  | 3.94320(-3510) | 6.00000    | 0.0950    |
| LT6                              | 5.44682(-96)  | 1.46250(-572)  | 5.48037(-3432) | 6.00000    | 0.0951    |

Table 4.18:  $f(x) = \tan^{-1}(\exp(x+2)+1) + \tanh(\exp(-x\cos(x))) - \sin(\pi x)$ , and  $x_0 = -3.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 3.83410(-30) | 9.12318(-175) | 1.65597(-1042) | 6.00000    | 0.4283    |
| $\alpha = -1, \beta = 1/2$       | 8.52813(-30) | 1.06493(-172) | 4.03770(-1030) | 6.00000    | 0.4236    |
| $\alpha = 1, \beta = 1$          | 3.50681(-31) | 5.93869(-181) | 1.40070(-1079) | 6.00000    | 0.4187    |
| $\alpha = 10(-2), \beta = 2$     | 1.71660(-30) | 7.62313(-177) | 5.84685(-1055) | 6.00000    | 0.4299    |
| $\alpha = 0.1, \beta = 10(-100)$ | 1.74453(-30) | 8.39223(-177) | 1.04720(-1054) | 5.99998    | 0.4360    |
| LM6                              | 1.74453(-30) | 8.39223(-177) | 1.04720(-1054) | 6.00000    | 0.4751    |
| KT6                              | 9.52811(-18) | 4.45833(-100) | 4.67973(-594)  | 6.00000    | 0.5171    |
| LT6                              | Div.         | Div.          | Div.           | Div.       | —         |

Table 4.19:  $f(x) = \frac{1}{x^4} - x^2 - \frac{1}{x} + 1$ ,  $x_0 = 1.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $   | $ f(x_3) $    | <i>coc</i> | Time Used |
|----------------------------------|--------------|--------------|---------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 1.04275(-12) | 1.14035(-73) | 2.17404(-440) | 6.01563    | 0.1219    |
| $\alpha = -1, \beta = 1/2$       | 1.00235(-13) | 1.13381(-78) | 1.75043(-470) | 6.03283    | 0.1250    |
| $\alpha = 1, \beta = 1$          | 1.24621(-08) | 4.51185(-51) | 2.52570(-305) | 5.99068    | 0.1187    |
| $\alpha = 10(-2), \beta = 2$     | 3.26228(-10) | 1.31226(-59) | 4.87314(-356) | 6.00116    | 0.1217    |
| $\alpha = 0.1, \beta = 10(-100)$ | 3.04748(-10) | 8.98539(-60) | 5.04015(-357) | 6.00139    | 0.1157    |
| LM6                              | 3.04748(-10) | 8.98539(-60) | 5.04015(-357) | 6.00139    | 0.1267    |
| KT6                              | 3.51323(-10) | 5.41578(-59) | 7.77345(-352) | 5.99940    | 0.1266    |
| LT6                              | 3.21532(-10) | 3.30926(-59) | 4.04604(-353) | 5.99975    | 0.1282    |

d) Numerical comparisons of twelfth order iterative methods with memory

Table 4.20:  $f(x) = \log(1 + x^2) + e^x \sin(x)$ ,  $x_0 = 0.03$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.27153(-12) | 2.31199(-137) | 2.87137(-1637) | 12.0000    | 0.2111    |
| $\alpha = -1, \beta = 1/2$       | 2.19410(-12) | 1.13464(-137) | 4.17585(-1641) | 12.0000    | 0.2096    |
| $\alpha = 1, \beta = 1$          | 2.51577(-12) | 1.70810(-136) | 1.64298(-1626) | 12.0000    | 0.2109    |
| $\alpha = 10(-2), \beta = 2$     | 2.35253(-12) | 4.66019(-137) | 1.70788(-1633) | 12.0000    | 0.2156    |
| $\alpha = 0.1, \beta = 10(-100)$ | 2.35092(-12) | 4.59760(-137) | 1.44441(-1633) | 12.0000    | 0.2141    |
| LM12                             | 2.35092(-12) | 4.59760(-137) | 1.44441(-1633) | 12.0000    | 0.2174    |
| KT12                             | 2.33967(-11) | 4.78046(-124) | 2.53240(-1476) | 12.0000    | 0.2216    |
| LT12                             | 2.12675(-11) | 8.76676(-125) | 2.11351(-1485) | 12.0000    | 0.2951    |

Table 4.21:  $f(x) = x \log(1 + x \sin(x)) + e^{-1+x^2+x \cos(x)} \sin(\pi x)$ ,  $x_0 = 0.01$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 7.08005(-19) | 4.02338(-219) | 2.69371(-2621) | 11.9961    | 0.1609    |
| $\alpha = -1, \beta = 1/2$       | 6.91054(-19) | 3.42920(-219) | 4.27278(-2621) | 11.9913    | 0.1640    |
| $\alpha = 1, \beta = 1$          | 7.58874(-19) | 9.55736(-220) | 5.37626(-2632) | 12.0072    | 0.1577    |
| $\alpha = 10(-2), \beta = 2$     | 7.25302(-19) | 3.31125(-219) | 2.67628(-2623) | 12.0000    | 0.1670    |
| $\alpha = 0.1, \beta = 10(-100)$ | 7.24963(-19) | 3.33781(-219) | 2.64476(-2623) | 12.0003    | 0.3203    |
| LM12                             | 7.24963(-19) | 3.33781(-219) | 2.64476(-2623) | 12.0003    | 0.3280    |
| KT12                             | 2.36512(-18) | 1.02983(-210) | 9.92328(-2519) | 11.9984    | 0.3141    |
| LT12                             | 1.31077(-18) | 1.31077(-209) | 5.03681(-2508) | 12.0047    | 0.2562    |

Table 4.22:  $f(x) = (1 + x^2) \cos(\frac{\pi x}{2}) + \frac{\log(x^2 + 2x + 2)}{(1 + x^2)}$ ,  $x_0 = -1.1$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 4.84111(-12) | 1.88173(-135) | 3.23766(-1622) | 12.0473    | 0.2658    |
| $\alpha = -1, \beta = 1/2$       | 1.15711(-10) | 1.23468(-123) | 1.28975(-1480) | 12.0117    | 0.2752    |
| $\alpha = 1, \beta = 1$          | 1.50433(-10) | 6.12658(-125) | 1.40440(-1498) | 12.0084    | 0.2547    |
| $\alpha = 10(-2), \beta = 2$     | 7.33852(-11) | 1.52641(-126) | 2.11654(-1514) | 11.9972    | 0.2674    |
| $\alpha = 0.1, \beta = 10(-100)$ | 7.21402(-11) | 1.34110(-126) | 4.82631(-1515) | 11.9972    | 0.2781    |
| LM12                             | 7.21402(-11) | 1.34110(-126) | 4.82631(-1515) | 11.9972    | 0.2893    |
| KT12                             | 9.04212(-9)  | 1.25373(-102) | 6.29725(-1229) | 12.0000    | 0.2625    |
| LT12                             | 1.04564(-9)  | 2.21032(-113) | 5.44406(-1358) | 12.0049    | 0.2062    |

Table 4.23:  $f(x) = e^x - 1 - \frac{x}{5}$ ,  $x_0 = 0.5$

| Methods                          | $ f(x_1) $  | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|-------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 2.53447(-9) | 3.70624(-109) | 1.93941(-1305) | 12.0013    | 0.0748    |
| $\alpha = -1, \beta = 1/2$       | 5.57321(-9) | 3.48315(-105) | 3.80222(-1258) | 11.9997    | 0.0826    |
| $\alpha = 1, \beta = 1$          | 1.23094(-5) | 1.30288(-59)  | 2.67955(-707)  | 12.0000    | 0.0813    |
| $\alpha = 10(-2), \beta = 2$     | 8.91387(-7) | 3.44579(-75)  | 4.20883(-896)  | 12.0000    | 0.1234    |
| $\alpha = 0.1, \beta = 10(-100)$ | 8.51712(-7) | 1.82898(-75)  | 1.92714(-899)  | 11.9996    | 0.1924    |
| LM12                             | 8.51712(-7) | 1.82898(-75)  | 1.92714(-899)  | 11.9994    | 0.2092    |
| KT12                             | 5.79857(-6) | 2.95423(-64)  | 9.79754(-764)  | 11.9994    | 0.2001    |
| LT12                             | 5.55571(-6) | 7.19582(-65)  | 1.55590(-771)  | 12.0002    | 0.2655    |

Table 4.24:  $f(x) = \tan^{-1}(\exp(x+2)+1) + \tanh(\exp(-x\cos(x))) - \sin(\pi(x))$ , and  $x_0 = -3.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 1.05493(-48) | 6.20606(-584) | 4.28348(-7008) | 12.0026    | 0.3564    |
| $\alpha = -1, \beta = 1/2$       | 3.32356(-51) | 4.14714(-614) | 4.11077(-7370) | 12.0021    | 0.3576    |
| $\alpha = 1, \beta = 1$          | 7.85590(-34) | 7.10315(-405) | 2.49960(-4856) | 11.9971    | 0.3609    |
| $\alpha = 10(-2), \beta = 2$     | 7.62179(-40) | 2.66888(-477) | 9.39301(-5727) | 12.0000    | 0.3642    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.99484(-39) | 1.60979(-466) | 3.26777(-5597) | 11.9996    | 0.3843    |
| LM12                             | 5.96258(-40) | 1.39019(-478) | 3.57214(-5742) | 12.0000    | 0.3578    |
| KT12                             | 8.27173(-41) | 1.53230(-485) | 2.49691(-5822) | 12.0000    | 0.3642    |
| LT12                             | 4.24940(-29) | 2.37664(-345) | 2.20973(-4140) | 12.0000    | 0.3596    |

Table 4.25:  $f(x) = \frac{1}{x^4} - x^2 - \frac{1}{x} + 1$ ,  $x_0 = 1.5$

| Methods                          | $ f(x_1) $   | $ f(x_2) $    | $ f(x_3) $     | <i>coc</i> | Time Used |
|----------------------------------|--------------|---------------|----------------|------------|-----------|
| $\alpha = -1/2, \beta = -1/2$    | 1.24354(-20) | 4.06404(-234) | 6.03624(-2796) | 12.0000    | 1.2891    |
| $\alpha = -1, \beta = 1/2$       | 2.45371(-20) | 1.32294(-230) | 7.98715(-2754) | 12.0000    | 1.2782    |
| $\alpha = 1, \beta = 1$          | 1.72032(-21) | 2.43001(-244) | 1.53372(-2918) | 12.0000    | 1.2750    |
| $\alpha = -10(-2), \beta = -1$   | 6.50586(-21) | 1.82401(-237) | 4.30425(-2836) | 12.0000    | 1.2658    |
| $\alpha = 0.1, \beta = 10(-100)$ | 5.61363(-21) | 3.15205(-238) | 3.09723(-2845) | 12.0000    | 1.5296    |
| LM12                             | 6.41932(-21) | 1.55529(-237) | 6.36622(-2837) | 12.0000    | 1.3875    |
| KT12                             | 1.30717(-07) | 7.90923(-077) | 1.91043(-0907) | 12.0000    | 1.4110    |
| LT12                             | Div.         | Div.          | Div.           | Div.       | —         |

# Chapter 5

## Conclusion

In this study several new iterative methods with and without memory are derived based on power means. Derived fourth and eighth order iterative methods are without memory in nature where as sixth and twelfth order iterative methods are with memory in nature. Fourth and eighth order proposed methods are optimal in sense of Kung-Traub conjecture [6]. Sixth and twelfth order methods are extended from fourth and eighth order methods respectively by using accelerator parameter. All proposed methods are totally derivative free. Further we numerically compare proposed iterative families with their existing counter parts and found that the developed methods are comparable and efficient as far as efficiency index is concerned. Moreover, the numerical illustrations have proved that the constructed methods are taking less time for computation then the existing ones, which is a considerable contribution in this field.

# Bibliography

- [1] Taher Lotfi, Fazlollah Soleymani, Zohreh Noori, A Kılıçman, and F Khaksar Haghani. Efficient iterative methods with and without memory possessing high efficiency indices. *Discrete Dynamics in Nature and Society*, 2014, 2014.
- [2] Peter S Bullen. *Handbook of means and their inequalities*, volume 560. Springer Science & Business Media, 2013.
- [3] Mahinder Kumar Jain. *Numerical methods for scientific and engineering computation*. New Age International, sixth edition edition, 2003.
- [4] Sam D Conte and David K Kahaner. *Numerical analysis*. John Wiley and Sons Ltd., third edition edition, 2003.
- [5] Joseph Frederick Traub. *Iterative methods for the solution of equations*. American Mathematical Soc., 1982.
- [6] HT Kung and Joseph Frederick Traub. Optimal order of one-point and multipoint iteration. *Journal of the ACM (JACM)*, 21(4):643–651, 1974.
- [7] Richard L Burden and J Douglas Faires. *Numerical analysis. 2001*. Brooks/Cole, USA, ninth edition edition, 2001.
- [8] V Kanwar, SK Tomar, Sukhjit Singh, and Sanjeev Kumar. Note on super-halley method and its variants. *Tamsui Oxford Journal of Information and Mathematical Sciences*, 28(2):191–216, 2012.
- [9] James M Ortega and Werner C Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

- [10] KC Gupta, V Kanwar, and Sanjeev Kumar. A family of ellipse methods for solving non-linear equations. *International Journal of Mathematical Education in Science and Technology*, 40(4):571–575, 2009.
- [11] Taher Lotfi and Elahe Tavakoli. On a new efficient steffensen-like iterative class by applying a suitable self-accelerator parameter. *The Scientific World Journal*, 2014, 2014.
- [12] Young Hee Geum and Young Ik Kim. An optimal family of fast 16th-order derivative-free multipoint simple-root finders for nonlinear equations. *Journal of Optimization Theory and Applications*, 160(2):608–622, 2014.
- [13] F Soleymani, PS Stanimirović, S Shateyi, and F Khaksar Haghani. Approximating the matrix sign function using a novel iterative method. In *Abstract and Applied Analysis*, volume 2014. Hindawi Publishing Corporation, 2014.
- [14] Taher Lotfi, Fazlollah Soleymani, Somayeh Sharifi, Stanford Shateyi, and F Khaksar Haghani. Multipoint iterative methods for finding all the simple zeros in an interval. *Journal of Applied Mathematics*, 2014.
- [15] Fazlollah Soleymani, Emran Tohidi, Stanford Shateyi, and F Khaksar Haghani. Some matrix iterations for computing matrix sign function. *Journal of Applied Mathematics*, 2014, 2014.
- [16] Min Surp Rhee and Young Ik Kim. A general class of optimal fourth-order multiple-root finders without memory for nonlinear equations. *Applied Mathematical Sciences*, 7(111):5537–5551, 2013.
- [17] F Soleymani and DKR Babajee. Computing multiple zeros using a class of quartically convergent methods. *Alexandria Engineering Journal*, 52(3):531–541, 2013.
- [18] Nikolay Kyurkchiev and Anton Iliev. On some multipoint methods arising from optimal in the sense of kung-traub algorithms for numerical solution of nonlinear equations. *Biomath*, 2(1):1305155, 2013.

- [19] Fazlollah Soleymani. Some high-order iterative methods for finding all the real zeros. *Thai Journal of Mathematics*, 12(2):313–327, 2013.
- [20] F Soleymani and S Shateyi. Two optimal eighth-order derivative-free classes of iterative methods. In *Abstract and Applied Analysis*, volume 2012. Hindawi Publishing Corporation, 2012.
- [21] Quan Zheng, Jingya Li, and Fengxi Huang. An optimal steffensen-type family for solving nonlinear equations. *Applied Mathematics and Computation*, 217(23):9592–9597, 2011.
- [22] H Wozniakowski. Maximal order of multipoint iterations using n evaluations. 1975.
- [23] Vinay Kanwar and Sushil Kumar Tomar. Modified families of newton, halley and chebyshev methods. *Applied Mathematics and Computation*, 192(1):20–26, 2007.
- [24] Peter Jarratt. Some fourth order multipoint iterative methods for solving equations. *Mathematics of Computation*, 20(95):434–437, 1966.
- [25] Vinay Kanwar and Sushil Kumar Tomar. Modified families of multi-point iterative methods for solving nonlinear equations. *Numerical Algorithms*, 44(4):381–389, 2007.
- [26] Ahmet Yasar Özban. Some new variants of newton’s method. *Applied Mathematics Letters*, 17(6):677–682, 2004.
- [27] Joseph Frederick Traub, Grzegorz Włodzimierz Wasilkowski, and Henryk Woźniakowski. *Information, uncertainty, complexity*. Addison-Wesley Publishing Company, Advanced Book Program/World Science Division, 1983.