On iterative methods to solve nonlinear equations

M. Àngela Grau Gotés

Barcelona, January 2015

Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya. BarcelonaTech

On iterative methods to solve nonlinear equations

M. Àngela Grau Gotés

Doctoral program in Applied Mathematics

On iterative methods to solve nonlinear equations

Thesis submitted by

M. Àngela Grau Gotés

for the degree of Doctor of Mathematics in the

Universitat Politècnica de Catalunya

Thesis Advisor

Dr. Miquel Grau Sánchez

Barcelona, January 2015

Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya. BarcelonaTech

M. Àngela Grau Gotés
Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya. BarcelonaTech
Jordi Girona 1-3, Edifici Omega, despatx 439, 08034 Barcelona, Spain
<angela.grau@upc.edu>

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola de Doctorat

Acta de qualificació de tesi doctoral

Curs acadèmic:

Nom i cognoms

Programa de doctorat

Unitat estructural responsable del programa

Resolució del Tribunal

Reunit el Tribunal designat a l'efecte, el doctorand / la doctoranda exposa el tema de la seva tesi doctoral titulada

_____

_____.

Acabada la lectura i després de donar resposta a les qüestions formulades pels membres titulars del tribunal, aquest atorga la qualificació:

☐ NO APTE ☐ APROVAT ☐ NOTABLE ☐ EXCEL·LENT

(Nom, cognoms i signatura) President/a	(Nom, cognoms i signatura) Secretari/ària	
(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal	(Nom, cognoms i signatura) Vocal

_____, _____ d'/de _____ de _____

El resultat de l'escrutini dels vots emesos pels membres titulars del tribunal, efectuat per l'Escola de Doctorat, a instància de la Comissió de Doctorat de la UPC, atorga la MENCIÓ CUM LAUDE:

☐ SÍ ☐ NO

(Nom, cognoms i signatura) President de la Comissió Permanent de l'Escola de Doctorat	(Nom, cognoms i signatura) Secretari de la Comissió Permanent de l'Escola de Doctorat

Barcelona, _____ d'/de _____ de _____

— A tots els que estan a prop meu —

Santi, Jaume, Joan, Núria

Abstract

Many of the problems in experimental sciences and other disciplines can be expressed in the form of nonlinear equations. The solution of these equations is rarely obtained in closed form. With the development of computers, these problems can be addressed by numerical algorithms that approximate the solution. Specifically, fixed point iterative methods are used, which generate a convergent sequence presumably to the solution of the equation or system of equations. Since J.F. Traub, (*Iterative methods for the solution of equations*, Prentice-Hall, N.J. 1964) initiated the qualitative as well the quantitative analysis of iterative methods in the 1960s, iterative methods for nonlinear systems has been a constant interesting field of study to numerical analysts.

Our contribution in this field is the analysis and the construction of new iterative methods, by improving the order of convergence and computational efficiency either of these or other known methods. To study new iterative methods that we have proposed, we reviewed, analyzed and improved classic concepts of computational order of convergence, the error equation and the computational cost of an iterative method for both an equation for a system of nonlinear equations. Specifically, we have worked on the following points:

- We computed the local order of convergence for known two-step and new multi-step iterative methods by means of expansions in formal developments in power series of the functions F, the Jacobian operator, the inverse Jacobian operator, the divided difference operator and its inverse operator.
- We generated some measures that approximate the order of convergence. Four new variants to compute computational order of convergence (COC) are given: one requires the value of the root, whilst the other three do not.
- We constructed families of iterative schemes that are variants of Newton's method

and Chebyshev's method and improve the order and the efficiency.

- We studied several families of modified Secant method (Secant, Kurchatov and Steffensen), evaluated variants of these methods and choose the most efficient.
- We generalized the concepts of efficiency index and computational efficiency for a nonlinear equations to a systems of nonlinear equations. It has been termed *Computational Efficiency Index* (CEI).
- We considered that in iterative process using variable precision, the accuracy will increase as the computation proceeds. The final result will be obtained as precisely as possible, depending on the computer and the software.
- We expressed the cost of evaluating elementary functions in terms of products. This cost depends on the computer, the software and the arithmetic that we used. The above numerical calculations were performed in the algebraic system called MAPLE.
- We presented a new way to compare elapsed time for different iterative schemes. This consists in estimating the time required to achieve a correct decimal of the solution by the method selected. That is, we measured the relationship between time to fulfill the stop criterion and the total number of correct decimals obtained by method.

The five papers selected for this compendium were published in scientific journals in the area of applied mathematics. The impact factor of these journals is, in all cases, in the first third according to the classification of the *Journal of Citation Reports (JCR).*

Agraïments

Agraeixo en primer lloc al meu director de tesi:
Dr. Miquel Grau Sánchez,
per la seva paciència i suport,
per totes les seves correccions precises;
però sobretot per la seva guia, confiança i paraules d'alè.

A tots els meus professors:
en especial en Dr. Miquel Noguera i Batlle,
qui m'ha mostrat com de complex pot ser el món dels càlculs.

A totes les persones que han cregut en mi,
que m'han donat l'oportunitat de créixer acadèmicament
fins arribar al desenvolupament d'aquesta tesi.

M. Àngela Grau Gotés

Contents

Notation

f	*nonlinear equation*
F	*nonlinear system of equations*
ρ	*order of convergence*
α	*correct root o exact solution*
x_n	*the n-th iterate*
e_n	*error in the n-th iterate (COC, CLOC)*
d_n	*approximation of the number of correct figures in the n-th iterate*
$\mathcal{O}$	*big-O notation*
o	*little-o notation*
$\widehat{e_n}$	*error parameter in the n-th iterate related to ACOC and ACLOC*
$\widetilde{e_n}$	*error parameter in the n-th iterate related to ECOC and ECLOC*
$\breve{e_n}$	*error parameter in the n-th iterate related to PCOC and PCLOC*
$\overline{\rho}_n$	*Computational Order of Convergence (COC)*
$\widehat{\rho}_n$	*Approximated Computational Order of Convergence (ACOC)*
$\widetilde{\rho}_n$	*Extrapolated Computational Order of Convergence (ECOC)*
$\breve{\rho}_n$	*Petković Computational Order of Convergence (PCOC)*
$\overline{\lambda}_n$	*Computational Local Order of Convergence (CLOC)*
$\widehat{\lambda}_n$	*Approximated Computational Local Order of Convergence (ACLOC)*
$\widetilde{\lambda}_n$	*Extrapolated Computational Local Order of Convergence (ECLOC)*
$\breve{\lambda}_n$	*Petković Computational Local Order of Convergence (PCLOC)*
θ	*number of evaluations of functions*
$\rho^{1/\theta}$	*Efficiency Index (EI)*
ω	*number of operations, expressed in product units*
$\rho^{1/\omega}$	*Computational Efficiency (CE)*

a_0	*number of evaluations of the scalar functions* F
a_1	*number of evaluations of the scalar functions* F'
μ_0	*ratio between products and evaluations in* F
μ_1	*ratio between products and evaluations in* F'
ℓ	*cost of one quotient in products*
$\mathcal{C}$	*computational Cost*
$\rho^{1/\mathcal{C}}$	*Computational Efficiency Index (CEI)*
$\mathcal{L}_n$	*approximation of the mantissa length in the* n*-th iterate in terms of* e_n
$\widehat{\mathcal{L}}_n$	*approximation of the mantissa length in the* n*-th iterate in terms of* $\widehat{e_n}$
$\widetilde{\mathcal{L}}_n$	*approximation of the mantissa length in the* n*-th iterate in terms of* $\widetilde{e_n}$
$\breve{\mathcal{L}}_n$	*approximation of the mantissa length in the* n*-th iterate in terms of* $\breve{e}_n$

Compendium of publications

The work presented in this thesis is a compendium of publications consisting of five papers. Its originality is measured by the level of scientific journals in the area of applied mathematics which has been published. The impact factor of these journals is, in all cases, the first third.

The references for the five peer-reviewed papers are given below:

1 M. Grau-Sánchez, A. Grau, M. Noguera, J.R. Herrero,
On new computational local orders of convergence,
Appl. Math. Lett. **25** (2012) 2023–2030.

2 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández, M. Noguera,
Analyzing the efficiency of some modifications of the Secant method,
Comput. Math. Appl. **64** (2012) 2066–2073.

3 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández,
Construction of derivative-free iterative methods from Chebyshev's method,
Anal. Appl. **11** (2013) 1350009 (16 pp.).

4 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández,
On the efficiency of two variants of Kurchatov's method for solving nonlinear systems,
Numer. Algor. **64** (2013) 685–698.

5 J.A. Ezquerro, M. Grau-Sánchez, A. Grau, M.A. Hernández-Verón,
A new class of secant-like methods for solving nonlinear systems of equations,
Commun. Appl. Math. Comput. Sci. **9** (2014) 201–213.

On new computational local orders of convergence[1]

M. Grau-Sánchez, **A. Grau**, **M. Noguera**, **José R. Herrero**

Four new variants of the computational order of convergence (COC) of a one-point iterative method with memory for solving nonlinear equations are presented. Furthermore, the way to approximate the new variants of the local order of convergence is analyzed. Three of the new definitions given here do not involve the unknown root. Numerical experiments using adaptive arithmetic with multiple precision and stopping criteria are implemented without using any known root.

Analyzing the efficiency of some modifications of the Secant method[2]

J. A. Ezquerro, **A. Grau**, **M. Grau-Sánchez**, **M. A. Hernández**, **M. Noguera**

Some modifications of the Secant method for solving nonlinear equations are revisited and the local order of convergence is found in a direct symbolic computation. To do this, a development of the inverse of the first-order divided differences of a function of several variables in two points is presented. A generalization of the efficiency index used in the scalar case to several variables is also analyzed in order to use the most competitive algorithm.

[1] Applied Mathematics Letters **25** (2012) 2023–2030.
[2] Computers and Mathematics with Applications **64** (2012) 2066–2073.

Construction of derivative-free iterative methods from Chebyshev's method[3]

J. A. Ezquerro, **A. Grau**, **M. Grau-Sánchez**, **M. A. Hernández**

From some modifications of Chebyshev's method, we consider a uni-parametric family of iterative methods that are more efficient than Newton's method, and we then construct two iterative methods in a similar way to the Secant method from Newton's method. These iterative methods do not use derivatives in their algorithms and one of them is more efficient than the Secant method, which is the classical method with this feature. A rigorous study of local and semi-local convergence of these methods is presented. Furthermore, we provide a technique for estimating the computational cost of any iterative method. This measure allows us to choose the most efficient method.

On the efficiency of two variants of Kurchatov's method for solving nonlinear systems[4]

J.A. Ezquerro, **A. Grau**, **M. Grau-Sánchez**, **M.A. Hernández**

We consider Kurchatov's method and construct two variants of this method for solving systems of nonlinear equations and deducing their local R-orders of convergence in a direct symbolic computation. We also propose a generalization to several variables of the efficiency used in the scalar case, and analyze the efficiencies of the three methods when they are used to solve systems of nonlinear equations.

[3] Analysis and Applications **11** (2013) 1350009 (16 pp.)
[4] Numerical Algorithms **64** (2013) 685–698.

A new class of Secant-like methods for solving nonlinear systems of equations[5]

J.A. Ezquerro, **A. Grau**, **M. Grau-Sánchez**, **M.A. Hernández-Verón**

We twice apply an idea by Hernández and Rubio (2002) for constructing a one-parameter family of Secant-like methods. As a result, we define a two-parameter family of Secant-like methods for solving nonlinear systems of equations. We analyze the efficiency of this new family and conclude that the Kurchatov method, which is one member of the family, is the most efficient. We illustrate this with Troesch's problem.

[5]Communications in Applied Mathematics and Computational Science **9** (2014) 201–213.

Chapter 1

Introduction

The search for solutions to nonlinear equations is one of the mathematical problems that appear most frequently in many scientific disciplines. It is well-known that the resolution of this kind of problems involves knowledge of the roots of an equation or a system of nonlinear equations. In general, it is not possible to find the exact solution, and numerical techniques need to be used to approximate them. The most common tools are iterative methods that, starting from one or several initial points, build up a sequence converging to an approximate solution of the equation. The key idea of the work presented in this report is try to obtain variants of the classical Secant and Newton's methods to improve their convergence.

1.1 Historical background

The two iterative methods that have been used most commonly to approximate roots of nonlinear equations are the Secant method and Newton's method. Below, a historical overview of both methods is given. Then, the state of the art of iterative methods from the nineteenth century to the present is reviewed.

1.1.1 Secant method

The Secant method –also known as Regula Falsi or the method of chords– is one of the most commonly used iterative methods for solving nonlinear equations. Traces of this scheme date back thirty centuries, to before Newton formulated the iterative method. The

first evidence of the Regula Falsi method is from Egypt in the eighteenth century BC, but it was also used in other civilizations in later times. For example, this algorithm was studied by Hindu mathematicians in third century BC, and by Chinese mathematicians in the second century BC, as shown in the *Vaishali Ganit* and *The Nine Chapters on the Mathematical Art*, texts, respectively [51]. Moreover, in the sixteenth century, the student Parameśvara presented a Sanskrit document in which the Secant method was used in techniques to correct planetary longitudes [54].

This method is based on a divided difference operator. The term *divided difference* was coined by the English mathematician A. De Morgan in 1842. He gave this name to the operator employed by A.M. Ampère (Lyon, 1775) in his work. Ampère was the first to focus on, which he called the *fonction interpolaire* (1826) – term used in French literature of the 1800s – At this time, divided differences appeared in the generation of interpolation polynomials. In 1840, L.A. Cauchy estimated the error in polynomial interpolation, using the results obtained by Ampère [32].

1.1.2 Newton's method

Three hundred years ago, Sir Isaac Newton wrote a method to approximate the roots of the equation $f(x) = 0$ in a manuscript entitled *Analysis per aequationes numero terminorum infinitos* (1669), published at the end of 1711 by W. Jones, and in *De methodus fluxionum et serierum infinitorum* (written in 1671, translated and published as *Method of Fluxions* in 1736 by J. Colson). Newton's description differs substantially from the modern scheme, because he only applied the method to polynomials. He computed a sequence of polynomials rather than successive approximations x_n, and only arrived at an approximation for the root α at the end. Newton may have derived his method from a similar, but less precise, method by F. Vieta. The essence of the Vieta method can be found in the work of the Persian mathematician, Sharaf al–Din al–Tusi, while his successor Jamshid al–Kashi used a form of Newton's method to solve $x^P - N = 0$ to find roots of N.

Later, in his manuscript *Analysis æquationum universalis: seu ad æquationes algebraicas resolvendas methodus generalis, & expedita, ex nova infinitarum serierum methodo, deducta ac demonstrata* (1690), J. Raphson presented a method for approximating roots of polynomials in which an operational process similar to that suggested by Newton is simplified.

Finally, in 1740, T. Simpson described Newton's method as an iterative method for solving general nonlinear equations using fluxional calculus. In the same publication, Simpson provided a generalization to systems of two equations, and noted that Newton's method could be used for solving optimization problems by setting the gradient to zero.

In 1879 in the Newton-Fourier imaginary problem, A. Calyley was the first to note the difficulties in generalizing Newton's method to complex roots of polynomials with degrees greater than 2 and complex initial values. This led to studies of the theory of iterations of rational functions (a set of references can be found in [32, 41, 63, 64]).

1.1.3 Around the twentieth century

We begin this period of time with the work of E. Schröder (1870, [56]). He presented a family of iterative methods to compute zeros of nonlinear functions of one variable. Two kinds of algorithms can be distinguished: the first is a recurrent construction of iterative schemes from Newton's method; and the second is derived using a suitable development to partial fractions restricted to a rational function whose roots would be approximated. Bernoulli's method can be regarded as a method of the second kind.

Later, a major development in automatic computations took place in the USA. In 1946, A. S. Householder joined the Mathematics Division of the Oak Ridge National Laboratory, where he was appointed chair in 1948. It is during this time that his interests shifted toward numerical analysis and he published a paper on this topic [34]. He was a leader in the field of numerical methods during the second half of the twentieth century, when an excellent survey of existing methods was carried out [35]. In 1952, L. V. Kantorovich [39] generalized Newton's method to Banach spaces and undertook a study of semilocal convergence. A. M. Ostrowski (1960, [49]) addressed the problem of solving nonlinear equations and systems of equations from different perspectives. Furthermore, he was the first to define *efficiency* using local convergence order and computational cost. One iterative method built up by him has been revisited frequently in the last decade, and it hs been generalized to several variables. Currently, a key reference for all researchers of this field is the work of J. F. Traub (1964, [57]). He used the concept of the iteration function and a technique to obtain the order of convergence using Taylor developments. Moreover, he redefined *the efficiency index* that has been presented by Ostrowski. Without a doubt, the main reference in this memory is Traub's book. A book by J. M. Ortega and W. C. Rheinboldt (1970, [48]) is another interesting reference for analyzing and studying the convergence of schemes for solving systems of nonlinear equations.

From 1964 to 1984, multi-step methods were widely considered. These, consist of increasing the number of steps per iteration to obtain higher efficiency, P. Jarratt [37], R. F. King [40] and B. Neta [45–47] have worked to obtain more efficient multi-step algorithms. F. A. Potra and V. Ptak (1984, [55]) presented a unified theory of convergence, based on what is known as the *method of nondiscrete mathematical induction*, which they applied to study in detail the rate of convergence of iterative processes.

In a very well-known paper from the twenty-first century, the authors S. Weerakoon and T. G. I. Fernando (2000, [62]) proposed a third order improvement to the iteration of Newton's method. Two important aspects of this study are: (1) the new method does not require computation of the second derivative of the function to carry out iterations; (2) the computational order of convergence (COC) is defined. Since its publication, many authors have obtained other multi-point methods using (1) techniques. However, the –Weerakoon and Fernando– paper has been cited frequently due to (2). Some papers offer a new construction of their algorithms, using the technique given in [62], but previously known and written by Traub [57], as claimed by L. D. Petkovic and M. S. Petkovic (2007, [53]). Other authors like M. Grau-Sánchez and J. L. Diaz-Barrero [20] and J. Kou [42] have improved the order of the Ostrowski method [49] from 4^{th} to 6^{th} and 7^{th} respectively. A thesis by D. K. R. Babajee (2010 , [5]) reviews the historical development of multi-point methods for finding simple roots of equations from Traub's period to 2010. Our contributions in this field, such as those described above, can be found in [16, 22, 25, 26].

Recently, research on improving the convergence order of numerical methods has become very active. In general, a increase in order implies an increase in the number of evaluations of functions in each iteration. To address the problem of solving nonlinear equations in which the function has no derivative, one part of this report analyzes methods without derivatives, using a divided difference operator as a substitute. A key reference to work on divided differences is [11]. We have contributed to this analysis with the papers [15, 24, 30].

1.1.4 High-precision

In the last twenty years, computations have generally been performed using high-precision software packages. For a small but growing sector of the scientific computing world, the 64-bit and 80-bit IEEE floating-point arithmetic formats currently provided in most computer systems are not sufficient. A survey of some interesting applications of high-precision arithmetic can be found in [6, 7].

Efficient packages are available for performing to any desired precision basic arithmetic operations, square and n-th roots, and for evaluating the most transcendental functions [8,9]. Such computations are provided by high-precision software packages that include high-level language translation modules to minimize the conversion effort [7,19,58,59]. Mathematical software packages such as MAPLE include facilities for arbitrarily high precision, but for some applications researchers rely on internet-available software, such as the GNU multi-precision package, ARPREC [4], MPFR [43] and MPFR++ [44].

1.2 Summary of the work in this report

Nonlinear equations must be solved to model problems in the real world. Due to the development of computers, problems can be addressed using numerical algorithms that approximate solutions. Our contribution in this field is the analysis and construction of iterative methods that improve known methods either in convergence order or computational efficiency.

More precisely, we highlight among others the following points:

- Construction of families of iterative schemes that are variants of the Newton and Chebyshev methods [12, 16, 25, 26].
- Study of several families of modified Secant methods (Secant, Kurchatov and Steffensen) [12–15, 24].
- Definition of new parameters of order: CLOC, ACLOC, ECLOC and PCLOC based on the known computational order of convergence (COC, [62]). A generalization to systems of equations of parameters: COC, ACOC, ECOC [24, 27, 31].
- A generalization of the computational efficiency to higher dimensions, named computational efficiency index (CEI) [15, 16, 24–26] and its application [12–14]
- Resolution using discretization of differential and integro-differential equations and other nonlinear models that appear in a lot of problems in the real world [13–16, 26].

From a theoretical perspective, in the aforementioned papers we began building up families of iterative methods that are variants of well-known ones. We have selected the methods with the highest order of convergence. We then compared with classical methods using CEI. Numerical examples and applications that support the theoretical results are given.

We found that the elapsed times agrees with the theoretical CEIs, and the values of the computational order that we obtained were sufficiently close to the local order of convergence claimed in the first part. We used a multiple precision arithmetic to carry out the computations.

Chapter 2

Iterative methods and local order of convergence

The aim of this chapter is to provide a survey of theoretical results and numerical tools for some iterative schemes, to approximate solutions of nonlinear equations. Namely, we examine the concept of iterative methods and their local order of convergence. Finally, we provide a detailed presentation of the error difference equation for the one-dimensional case and the multi-dimensional case, and its relations with the orders using the indicial polynomial.

2.1 Iteration functions

Many problems that arise in computational sciences and other disciplines can be formulated by means of an equation like the following:

$$\mathcal{F}(x) = 0\,, \tag{2.1}$$

where $\mathcal{F} : D \subset \mathbb{X} \longrightarrow \mathbb{Y}$ is a continuous operator, defined on a nonempty convex subset D of a Banach space $\mathbb{X}$ with values in a Banach space $\mathbb{Y}$. In this study, we address the problem of approximating a local unique solution $\alpha \in \mathbb{X}$ of equation (2.1). Since the exact solution of this equation can rarely be found, we need to use iterative techniques to approximate it to the desired precision from one or several initial approximations. This procedure generates a sequence of approximations of the solution searched.

A classification of iteration functions, according to the previous information required carry them out, can be found in Traub [57]. We build up a sequence $\{x_n\}_{n\geq 1}$ in a Banach space $\mathbb{X}$ using the initial conditions $x_{-k},\dots,x_{-1},x_0$, $0\leq k\leq j-1$. Traub's classification of the iteration functions is the following.

Type I Term x_{n+1} is obtained using only the information at x_n and no other information. That is,

$$x_{n+1}=\Phi(x_n)\,,\quad n\geq 0\,. \tag{2.2}$$

The function Φ is called a *one-point iteration function* and equation (2.2) is called the one-point iterative method.

Type II Term x_{n+1} is obtained using the information at x_n and the previous information at $x_{n-1},\dots,x_{n-j}$. Namely,

$$x_{n+1}=\Phi(x_n\,;x_{n-1},\dots,x_{n-j})\,,\quad n\geq 0\,,\quad j\geq 1\,. \tag{2.3}$$

Function Φ is called a *one-point iteration function with memory* and equation (2.3) is called the one-point iterative method with memory (j points). The semicolon in (2.3) is written to distinguish the information from new data from the previously used information.

Type III Term x_{n+1} is determined by new information at x_n and previous information at $\varphi_1=\varphi_1(x_n)$, $\varphi_2=\varphi_2(\varphi_1,x_n),\dots,$ $\varphi_r=\varphi_r(\varphi_{r-1},\dots,\varphi_1,x_n)$, $r\geq 1$. That is,

$$x_{n+1}=\Phi\left(x_n\,,\varphi_1\,,\dots,\varphi_r\right)\,,\quad n\geq 0\,,\quad r\geq 1\,. \tag{2.4}$$

Here, function Φ is called a *multi-point iteration function* without memory and equation (2.4) is called the multi-point iterative method without memory (r steps).

Type IV Term x_{n+1} is obtained from the new information at x_n and the previous information at
$\varphi_1=\varphi_1(x_n\,;x_{n-1},\dots,x_{n-j}),\dots,\varphi_r=\varphi_r(x_n,\varphi_1,\dots,\varphi_{r-1}\,;x_{n-1},\dots,x_{n-j})$. Namely,

$$x_{n+1}=\Phi\left(x_n\,,\varphi_1\,,\dots,\varphi_r;x_{n-1}\,,\dots,x_{n-j}\right)\,,\quad n\geq 0\,,\quad r\geq 1\,,\quad j\geq 1. \tag{2.5}$$

Function Φ is called a *multi-point iteration function with memory* and (2.5) is called a multi-point iteration method with memory (r steps and j points).

2.2 One-dimensional case

In particular, when the Banach spaces in (2.1) are $\mathbb{X}=\mathbb{Y}=\mathbb{R}$, we have to solve the simplest and most classical nonlinear problem. Namely, let $f:I\subseteq\mathbb{R}\to\mathbb{R}$ be a nonlinear function.

We have to approximate a simple root α of the equation

$$f(x) = 0, \tag{2.6}$$

where I is a neighborhood of α. An approximation of α is usually obtained by means of an iterative function, of type **I , II , III** or **IV** defined in (2.2),(2.3),(2.4) or (2.5) whereby a sequence $\{x_n\}_{n\geq 1}$ is considered converging to α.

Definition 1. *The sequence $\{x_n\}$ is said to converge to α with order of convergence $\rho \in \mathbb{R}$, $\rho \geq 1$, if a positive real constant $C \neq 0$ and $C \neq \infty$ exists such that*

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^{\rho}} = C, \tag{2.7}$$

where $e_n = x_n - \alpha$ is the error *in the n-th iterate, and the constant C is called the* asymptotic error constant *(see [57]).*

The *local order of convergence of an iterative method* in a neighborhood of a root is the order of its corresponding sequence generated by the iterative function and the corresponding initials approximations. For iterative methods without memory, the local order is a positive integer. The convergence is said to be linear if $\rho = 1$, quadratic if $\rho = 2$, cubic if $\rho = 3$, and, in general, superlinear if $\rho > 1$, superquadratic if $\rho > 2$, and so on.

To avoid higher order terms in some relations, which do not influence the convergence order, we employ the $\mathcal{O}$–notation and the o–notation. If the sequences $\{\xi_n\}_{n\geq 1}$ and $\{\zeta_n\}_{n\geq 1}$ are null sequences and the sequence $\xi_n/\zeta_n \to K$ when $n \to \infty$, where K is a nonzero constant, we shall write $\xi_n = \mathcal{O}(\zeta_n)$ or $\xi_n \sim \zeta_n$. If the sequence $\xi_n/\zeta_n \to 0$ when $n \to \infty$, we shall write $\xi_n = o(\zeta_n)$; in other words, ξ_n is dominated by ζ_n asymptotically. This approach significantly simplifies both the convergence analysis and presentation.

According to the previous notation the one-point iterative method without memory (2.7) can be written as

$$e_{n+1} = C\, e_n^{\rho} + \mathcal{O}\big(e_n^{\rho+1}\big), \ n \geq n_0. \tag{2.8}$$

The expression (2.8) is called the *error difference equation* for the one-point iterative method. Note that the higher order terms in (2.8) are powers of $\rho + 1$.

For the one-point iterative method without memory, an approximation of the number of correct decimal places in the n-th iterate, d_n, is given by

$$d_n \approx -\log_{10} |x_n - \alpha|. \tag{2.9}$$

From (2.8), for n large enough we have $e_{n+1} \approx C\, e_n^{\rho}$, which using logarithms, yields

$$d_{n+1} \approx -\log_{10} C + \rho \cdot d_n, \tag{2.10}$$

from which follows

$$d_{n+1} \approx \rho \cdot d_n. \tag{2.11}$$

This means that, in each iteration, the number of correct decimals is approximately the number of correct decimals of the previous iteration, multiplied by the local error.

This is in agreement with Wall's definition (1954, [61]). That is, *the local order of convergence of a one-point iteration function indicates the rate of convergence of the iteration method.* Then, Wall defines the order ρ of the iteration formula by

$$\rho = \lim_{n\to\infty} \frac{\log|e_{n+1}|}{\log|e_n|} = \lim_{n\to\infty} \frac{d_{n+1}}{d_n}. \tag{2.12}$$

This expression will be used later on when we define some parameters employed in the computation of the local order of convergence of an iterative method.

For the one-point iterative method with memory (2.3), the error difference equation is

$$e_{n+1} \;=\; C e_n^{a_1} e_{n-1}^{a_2} \dots e_{n-j+1}^{a_j} + \mathcal{O}(e_n^{a_1} e_{n-1}^{a_2} \dots e_{n-j+1}^{a_j}), \tag{2.13}$$

where a_k are nonnegative integers for $1 \le k \le j$ and $\mathcal{O}(e_n^{a_1} e_{n-1}^{a_2} \dots e_{n-j+1}^{a_j})$ represents terms with a higher order than the term $e_n^{a_1} e_{n-1}^{a_2} \dots e_{n-j+1}^{a_j}$. In this case, the order of convergence ρ, is the unique real positive root of the *indicial polynomial* (see [48,49,57,60]) of the error difference equation (2.13) given by

$$p_j(t) = t^j - a_1 t^{j-1} - \dots - a_{j-1} t - a_j. \tag{2.14}$$

Notice that $p_j(t)$ in (2.14) has a unique real positive root on account of Descartes' rule of signs. Moreover, we can write $e_{n+1} = C\, e_n^{\rho} + \mathcal{O}\big(e_n^{\rho}\big)$, $n \ge n_0$.

2.3 Muti-dimensional case

When the Banach spaces in (2.1) are $\mathbb{X} = \mathbb{Y} = \mathbb{R}^m$ we have to solve a system of nonlinear equations. Namely, let $F : D \subset \mathbb{R}^m \longrightarrow \mathbb{R}^m$ be a nonlinear function and $F \equiv (F_1, F_2, \dots, F_m)$ with $F_i : D \subseteq \mathbb{R}^m \to \mathbb{R}$, $i = 1, 2, \dots, m$, where D is an open convex domain in $\mathbb{R}^m$, so that we have to approximate a solution $\alpha \in D$ of the equation $F(x) = 0$.

Starting with a given set of initial approximations of the root α, the iteration function $\Phi : D \longrightarrow D$ of type **I , II , III** or **IV** defined by (2.2),(2.3),(2.4) or (2.5), whereby a sequence $\{x_n\}_{n\geq 1}$ is considered converging to α.

Definition 2. *The sequence $\{x_n\}$ converges to α with an order of convergence of at least $\rho \in \mathbb{R}$, $\rho \geq 1$, if a positive real constant $0 < C < \infty$ exists such that*

$$\|e_{n+1}\| \leq C\|e_n\|^{\rho}, \tag{2.15}$$

where $e_n = x_n - \alpha$ is the error in the n-th iterate, and the constant C is called the asymptotic error constant (see [57]). Here the norm used is the maximum norm.

The local order of convergence of an iterative method in a neighborhood of a root is the order of the corresponding sequence generated (in $\mathbb{R}^m$) by the iterative function Φ and the corresponding initials approximations.

If we do not use norms, a definition of the local order of convergence for the one-point iterative method without memory can be considered as follows. The local order of convergence is $\rho \in \mathbb{N}$ if a ρ–linear function $C \in \mathscr{L}\left(\mathbb{R}^m \times \overset{\rho}{\cdots} \times \mathbb{R}^m, \mathbb{R}^m\right) \equiv \mathscr{L}_{\rho}\left(\mathbb{R}^m, \mathbb{R}^m\right)$ exists such that

$$e_{n+1} = C\, e_n^{\rho} + \mathcal{O}\left(e_n^{\rho+1}\right), \quad n \geq n_0 \tag{2.16}$$

where e_n^{ρ} is $(e_n, \overset{\rho}{\cdots}, e_n) \in \mathbb{R}^m \times \overset{\rho}{\cdots} \times \mathbb{R}^m$.

Some measures of the asymptotic rate of convergence of sequences and iterative processes are considered by Ortega and Rheinboldt [48]; we will consider the R-order for two reasons. Firstly, the R-order of Φ at α is independent of the norm. Secondly, it follows that when $0 < C < \infty$ exists for some $\rho \in [1, \infty)$ from (2.15), then ρ is the R-order of convergence of the iterative method defined by Ortega and Rheinboldt [48]. Moreover, the local order ρ of (2.16) is also the *R-order* of convergence of the method.

For the one-point iterative method with memory, the error difference equation can be expressed by

$$e_{n+1} = C e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j} + \mathcal{O}(e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j}), \tag{2.17}$$

where $C \in \mathscr{L}_{a_1+\cdots+a_j}\left(\mathbb{R}^m, \mathbb{R}^m\right)$ and a_k are nonnegative integers for $1 \leq k \leq j$. The expression $\mathcal{O}(e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j})$ represents terms with a higher order than $e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j}$.

As in the one-dimensional case, we can write the equation associated with (2.17), $p_j(t) = t^j - a_1 t^{j-1} - \cdots - a_{j-1} t - a_j$. If we apply Descartes' rule of signs to the previous polynomial, there is a unique real positive root ρ that coincides with the local order of convergence (see [48, 60]).

Chapter 3

Computational estimation of the local order of convergence

Here, we present numerical parameters that allow us to assess the previosly studied theoretical order of convergence. These measures are used to check the theoretical local order of convergence. The parameter computational order of convergence (COC) is used in most studies published after Weerakoon and Fernando [62]. However, this parameter can only be used when the root α is known. To overcome this problem, the following parameters have been introduced:

ACOC Approximated computational order of convergence by Hueso *et al.* (2009, [36]) .
ECOC Extrapolated computational order of convergence by Grau *et al.* (2009, [29]) .
PCOC Pétkovic computational order of convergence by Petković (2011 [52]) .

The paper by Grau *et al.* (2010, [31]) examines the relations between the parameters COC, ACOC and ECOC and the theoretical convergence order of iterative methods without memory.

Subsequently, using Wall's definition of the order (2.12), four new parameters (CLOC, ACLOC, ECLOC and PCLOC) were given in [27] to check this order. Note that the last three parameters do not need knowledge of the root.

Generalizations of COC, ACOC and ECOC from the one-dimensional case to the multi-dimensional one can be found in [23]. They will be presented in detail in the sequel.

3.1 Computational order of convergence and its variants

Let $\{x_n\}_{n\geq 1}$ be a sequence of real numbers converging to α. It is obtained by carrying out one iteration function in $\mathbb{R}$, starting with an initial approximation x_0 or $x_{-j+1}, \ldots, x_{-1}, x_0$, of the root α of (2.6). Let $\{e_n\}_{n\geq 1}$ be the sequence of errors given by $e_n = x_n - \alpha$. If functions (2.2)–(2.5) have local order of convergence ρ, then from (2.10) we have

$$\log|e_n| \approx \rho \cdot \log|e_{n-1}| + \log C,$$
$$\log|e_{n-1}| \approx \rho \cdot \log|e_{n-2}| + \log C.$$

If we substract the second expression from the first one we get

$$\rho \approx \frac{\log|e_n \,/\, e_{n-1}|}{\log|e_{n-1} \,/\, e_{n-2}|}\,. \tag{3.1}$$

This expression is the same as that described in papers by Weerakoon and Fernando [62].

Definition 3. *The value $\overline{\rho}_n$, computed by*

$$\overline{\rho}_n = \frac{\log|e_n \,/\, e_{n-1}|}{\log|e_{n-1} \,/\, e_{n-2}|}\,, \quad e_n = x_n - \alpha\,, \quad n \geq 3\,, \tag{3.2}$$

is called the computational order of convergence and is denoted by COC.

The first variant of COC involves the parameter $\widehat{e}_n = x_n - x_{n-1}$.

Definition 4. *The approximated computational order of convergence (ACOC) of a sequence of iterates $\{x_n\}_{n\geq 1}$ and is defined by*

$$\widehat{\rho}_n = \frac{\log|\widehat{e}_n \,/\, \widehat{e}_{n-1}|}{\log|\widehat{e}_{n-1} \,/\, \widehat{e}_{n-2}|}\,, \quad \widehat{e}_n = x_n - x_{n-1}\,, \quad n \geq 4. \tag{3.3}$$

The second variant of COC is obtained using Aitken's extrapolation procedure [1]. That is, from iterates x_{n-2}, x_{n-1}, x_n, the approximation $\widetilde{\alpha}_n$ of the root α can be obtained.

Definition 5. *The extrapolated computational order of convergence (ECOC) of a sequence of iterates $\{x_n\}_{n\geq 1}$ that is defined by*

$$\widetilde{\rho}_n = \frac{\log|\widetilde{e}_n \,/\, \widetilde{e}_{n-1}|}{\log|\widetilde{e}_{n-1} \,/\, \widetilde{e}_{n-2}|}\,, \quad n \geq 5, \tag{3.4}$$

where $\widetilde{e}_n = x_n - \widetilde{\alpha}_n$ and $\widetilde{\alpha}_n$ is given by

$$\widetilde{\alpha}_n = x_n - \frac{(\delta x_{n-1})^2}{\delta^2 x_{n-2}}, \quad \delta x_n = x_{n+1} - x_n. \tag{3.5}$$

Sequences $\{\widetilde{\rho}_n\}_{n\geq 5}$ and $\{\widehat{\rho}_n\}_{n\geq 4}$ converge to ρ. The details of the preceding claim can be found in [31] where the relations between the error e_n and $\widetilde{e}_n$ and $\widehat{e}_n$ are also described.

Finally, Petković [52] uses the following parameter

$$\breve{\rho}_n = \frac{\log|\breve{e}_n|}{\log|\breve{e}_{n-1}|}, \qquad \breve{e}_n = \frac{f(x_n)}{f(x_{n-1})}, \quad n \geq 2, \tag{3.6}$$

that we call the Petković computational order of convergence (PCOC).

From a computational viewpoint, ACOC has the lowest computational cost, followed by PCOC. Inspired by (2.12) given in [61], in our study [27] we present four new parameters that are described in the following section.

3.2 New parameters to compute the order of convergence

3.2.1 Definitions

Given the sequence $\{x_n\}_{n\geq 1}$ of iterates converging to α with order ρ, we consider the sequences of errors $e_n = x_n - \alpha$ and error parameters $\widehat{e}_n = x_n - x_{n-1}$, $\widetilde{e}_n = x_n - \widetilde{\alpha}_n$ and $\breve{e}_n = \frac{f(x_n)}{f(x_{n-1})}$ defined previously in (3.3),(3.4),(3.6).

Definition 6. *From the preceding, we define the following sequences* $\{\overline{\lambda}_n\}_{n\geq 2}$, $\{\widehat{\lambda}_n\}_{n\geq 3}$, $\{\widetilde{\lambda}_n\}_{n\geq 4}$ *and* $\{\breve{\lambda}_n\}_{n\geq 2}$ *:*

• Computational local order of convergence or parameter CLOC is the general term of the sequence $\{\overline{\lambda}_n\}_{n\geq 2}$ given by

$$\overline{\lambda}_n = \frac{\log|e_n|}{\log|e_{n-1}|}. \tag{3.7a}$$

• Approximated computational local order of convergence or parameter ACLOC is the general term of the sequence $\{\widehat{\lambda}_n\}_{n\geq 3}$ defined by

$$\widehat{\lambda}_n = \frac{\log|\widehat{e}_n|}{\log|\widehat{e}_{n-1}|}. \tag{3.7b}$$

• Extrapolated computational local order of convergence or parameter ECLOC is the general term of the sequence $\{\widetilde{\lambda}_n\}_{n\geq 4}$ given by

$$\widetilde{\lambda}_n = \frac{\log|\widetilde{e}_n|}{\log|\widetilde{e}_{n-1}|}. \tag{3.7c}$$

• Petković computational local order of convergence or parameter PCLOC is the general term of the sequence $\{\breve{\lambda}_n\}_{n\geq 2}$ defined by

$$\breve{\lambda}_n = \frac{\log|f(x_n)|}{\log|f(x_{n-1})|}\,. \tag{3.7d}$$

Note the analogy between $\overline{\lambda}_n$ and the definitions given by Wall in [61] and Tornheim in [60]. Knowledge of α is needed to obtain $\overline{\lambda}_n$; but is not required to obtain $\widehat{\lambda}_n$, $\widetilde{\lambda}_n$ and $\breve{\lambda}_n$. The new parameters CLOC, ACLOC, ECLOC and PCLOC have a lower computational cost than their predecessors. A detailed description of their convergence can be found in our studies [27, 28].

3.2.2 Convergence of new sequences of parameters

In the case of iterative methods to obtain approximates of the root α of $f(x) = 0$, where $f : I \subset \mathbb{R} \to \mathbb{R}$, the error difference equation is given by

$$e_{n+1} = C\, e_n^{\rho}\left(1 + \mathcal{O}(e_n^{\sigma})\right), \quad 0 < \sigma < 1, \tag{3.8}$$

where C is the asymptotic error constant. With the additional hypothesis on the order, say $\rho \geq (1+\sqrt{5})/2$, in [27] the relations between ρ and the parameters $\overline{\lambda}_n$, $\widehat{\lambda}_n$, $\widetilde{\lambda}_n$ and $\breve{\lambda}_n$ are presented. Namely,

$$\overline{\lambda}_n = \rho\left(1 + \mathcal{O}\left(\frac{\log|C|}{\rho\log|e_{n-1}|}\right)\right), \quad n \geq 1, \tag{3.9a}$$

$$\widehat{\lambda}_n = \rho\left(1 + \mathcal{O}\left(\frac{\log|C|}{\rho\log|e_{n-2}|}\right)\right), \quad n \geq 2, \tag{3.9b}$$

$$\widetilde{\lambda}_n = \rho\left(1 + \mathcal{O}\left(\frac{\log|C|}{(2\rho-1)\log|e_{n-2}| + \log|C|}\right)\right), \quad n \geq 2, \tag{3.9c}$$

$$\breve{\lambda}_n = \rho\left(1 + \mathcal{O}\left(\frac{\log|\Gamma C|}{\rho\log|e_{n-1}|}\right)\right), \quad n \geq 1, \quad \Gamma = f'(\alpha). \tag{3.9d}$$

Notice that the convergence of the sequences $\overline{\lambda}_n$, $\widehat{\lambda}_n$, $\widetilde{\lambda}_n$, $\breve{\lambda}_n$ towards ρ may be derived from (3.9a), (3.9b), (3.9c) and (3.9d). Moreover, when built up the sequences $\overline{\lambda}_n$ and $\widehat{\lambda}_n$ we realized that the error term in (3.9a) and (3.9b) is the same, but shifted one step as tested in [27].

3.2.3 Relations between error and error parameters

The following inverse error difference equation is presented in [27]. That is,

$$e_n^{\rho} = C^{-1}\, e_{n+1}\left(1 + \mathcal{O}(\, e_{n+1}^{\sigma/\rho})\right). \tag{3.10}$$

Using (3.8) and (3.10), and the definitions of $\widehat{e}_n$, $\widetilde{e}_n$ and $\breve{e}_n$, we obtain the following theoretical approximations of e_n. Namely,

$$e_n \approx C^{\frac{1}{1-\rho}} \left(\frac{\widehat{e}_n}{\widehat{e}_{n-1}}\right)^{\rho^2/(\rho-1)} \text{ or } e_n \sim \left(\frac{\widehat{e}_n}{\widehat{e}_{n-1}}\right)^{\rho^2/(\rho-1)} \quad n \geq 3\,, \tag{3.11a}$$

$$e_n \approx C^{\frac{\rho-1}{2\rho-1}} \left(\widetilde{e}_n\right)^{\rho^2/(2\rho-1)} \text{ or } e_n \sim \left(\widetilde{e}_n\right)^{\rho^2/(2\rho-1)} \quad n \geq 3, \tag{3.11b}$$

$$e_n \approx C^{\frac{1}{1-\rho}} \left(\breve{e}_n\right)^{\rho/(\rho-1)} \text{ or } e_n \sim \left(\breve{e}_n\right)^{\rho/(\rho-1)} \quad n \geq 2. \tag{3.11c}$$

From the preceding (3.11a), (3.11b) and (3.11c), we can obtain bounds of the error to predict the number of correct figures and establish a stopping criterion, all without knowledge of the root α.

$f(x)$	α	x_0	$\{x_{-1}\,,\, x_0\}$
$f_1(x) = x^3 - 3x^2 + x - 2$	2.89328919630449778890635 6	2.5	$\{2.25, 2.60\}$
$f_2(x) = x^3 + \cos x - 2$	1.17257796475397001267333 3	1.5	$\{1.50, 2.50\}$
$f_3(x) = 2\sin x + 1 - x$	2.38006127313933901721254 8	2.5	$\{1.00, 2.00\}$
$f_4(x) = (x+1)\, e^{x-1} - 1$	0.55714559899761141685867 2	1.0	$\{0.00, 0.75\}$
$f_5(x) = e^{x^2+7x-30} - 1$	3.0	2.94	$\{2.90, 3.10\}$
$f_6(x) = e^{-x} + \cos x.$	1.74613953040801241765070 3	1.5	$\{1.60, 1.90\}$
$f_7(x) = x - 3\ln x$	1.85718386020783533645698 1	2.0	$\{1.00, 2.00\}$

Table 3.1: Test functions, their roots and the initial points considered

3.2.4 Numerical tests

The convergence of the new parameters has been tested in six iterative schemes with a local convergence order equal to 2, 3, 4, $(1+\sqrt{5}\,)/2$, $1+\sqrt{2}$ and $1+\sqrt{3}$ respectively, in a set of seven real functions that are shown in Table 3.1. The first three methods are one-point iterative methods without memory, known as the Newton's method, Chebyshev

method [22] and the Schröder method [56]. The other three are iterative methods with memory, namely the Secant method and two of its variants (see [30]).

They are defined by

$$\phi_1(x_n) = x_n - u(x_n), \tag{3.12}$$

$$\phi_2(x_n) = \phi_1(x_n) - \frac{1}{2} L(x_n)\, u(x_n), \tag{3.13}$$

$$\phi_3(x_n) = \phi_2(x_n) - \left(\frac{1}{2} L(x_n)^2 - M(x_n)\right) u(x_n), \tag{3.14}$$

$$\phi_4(x_n) = x_n - [x_{n-1}, x_n]_f^{-1} f(x_n), \tag{3.15}$$

$$\phi_5(x_n) = \phi_4(x_n) - [x_n, \phi_4(x_n)]_f^{-1} f(\phi_4(x_n)), \tag{3.16}$$

$$\phi_6(x_n) = \phi_4(x_n) - [x_n, 2\phi_4(x_n) - x_n]_f^{-1} f(\phi_4(x_n)), \tag{3.17}$$

where

$$u(x) = \frac{f(x)}{f'(x)}, \quad L(x) = \frac{f''(x)}{f'(x)}\, u(x), \quad M(x) = \frac{f'''(x)}{3!\, f'(x)}\, u(x)^2 \quad [x, y]_f^{-1} = \frac{y - x}{f(y) - f(x)}.$$

The numerical results can be found in [27]. For each method from (3.12) to (3.17) and each function in Table 3.1, we applied the four techniques with adaptive multi-precision arithmetic (see below) derived from relations (3.11a), (3.11b) and (3.11c) and the desired precision, which for this study is 10^{-2200}. The number of iterations required to obtain the desired precision and the values of iterated points $x_1, \ldots, x_I$ are the same. Table 3.2 shows the number of iterations needed to compute the root. In addition, the last four columns show the interval with the minimum and maximum error obtained in the computation of the corresponding CLOC, ECLOC, ACLOC or PCLOC for the seven tested functions.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	$\mathcal{I}(\bar{\lambda})$	$\mathcal{I}(\widetilde{\lambda})$	$\mathcal{I}(\widehat{\lambda})$	$\mathcal{I}(\check{\lambda})$
ϕ_1	12	11	10	11	12	10	11	[2.8e−5, 1.1e−3]	[3.7e−5, 1.5e−3]	[5.6e−5, 2.2e−3]	[6.0e−5, 1.2e−3]
ϕ_2	8	7	6	7	8	6	7	[8.9e−5, 3.3e−3]	[1.0e−4, 6.0e−3]	[1.4e−4, 9.9e−3]	[2.1e−4, 4.5e−3]
ϕ_3	6	6	5	6	6	5	5	[8.8e−6, 1.3e−2]	[2.0e−5, 3.0e−2]	[3.5e−5, 5.1e−2]	[1.6e−3, 1.2e−2]
ϕ_4	17	18	16	16	18	14	16	[8.1e−6, 5.8e−4]	[7.9e−6, 6.8e−4]	[1.2e−5, 9.4e−4]	[3.2e−5, 5.5e−3]
ϕ_5	9	9	9	8	10	7	8	[5.5e−5, 3.0e−3]	[1.0e−4, 4.4e−3]	[1.3e−4, 7.2e−3]	[1.9e−3, 3.3e−3]
ϕ_6	8	8	7	7	8	6	7	[3.6e−5, 3.7e−3]	[1.4e−4, 1.6e−2]	[1.6e−4, 1.7e−2]	[2.8e−4, 4.5e−3]

Table 3.2: Min-max interval for error bounds

From these numerical tests, we can conclude that CLOC gives the best approximation of the theoretical order of convergence of an iterative method. However, knowledge of the root is required. Conversely, as we can see in the definitions of ACLOC (3.7b), ECLOC

(3.7c) and PCLOC (3.7d), these parameters do not involve the expression of the root α. Actually, in real problems we want to approximate a root that is not known in advance. For practical purposes, (see Table 3.2) we recommend ECLOC, because it presents the best approximation of the local order (see [27]). Nevertheless, PCLOC is a good practical parameter in many cases, because it requires fewer additional computations.

3.3 Multi-dimensional case

A generalization to several variables of some parameters is carried out to approximate the local order of the iterative method presented in the previous sections. To define the new parameters, we substitute the absolute value by the maximum norm, and all computations are done using the components of the vectors. Let $\{x_n\}_{n\in\mathbb{N}}$ be a convergence sequence of $\mathbb{R}^m$ towards $\alpha \in \mathbb{R}^m$, where $x_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(m)})^t$ and $\alpha = (\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(m)})^t$. We consider the vector sequence of the error $e_n = x_n - \alpha$ and the following vector sequences of parameters:

$$\widehat{e}_n = x_n - x_{n-1}\,, \qquad \widetilde{e}_n = \max_{1\le r\le m} \left| \frac{\left(\delta x_{n-1}^{(r)}\right)^2}{\delta^2 x_{n-2}^{(r)}} \right| \tag{3.18}$$

where $\delta x_n = x_{n+1} - x_n$. Notice that $\widetilde{e}_n$ is the δ^2-Aitken procedure applied to the components of x_{n-1}, x_n and x_{n+1}, and all parameters are independent of knowledge of the root.

Definition 7. *Let* $\{\overline{\rho}_n\}_{n\ge3}$, $\{\widehat{\rho}_n\}_{\ge4}$, $\{\widetilde{\rho}_n\}_{\ge5}$, $\{\overline{\lambda}_n\}_{n\ge2}$, $\{\widehat{\lambda}_n\}_{\ge3}$, $\{\widetilde{\lambda}_n\}_{\ge4}$ *and* $\{\breve{\lambda}_n\}_{n\ge2}$ *be the following real sequences:*

- Parameters COC, $\{\overline{\rho}_n\}_{n\ge3}$, and CLOC, $\{\overline{\lambda}_n\}_{n\ge2}$:

$$\overline{\rho}_n = \frac{\log\left(\|e_n\|/\|e_{n-1}\|\right)}{\log\left(\|e_{n-1}\|/\|e_{n-2}\|\right)}\,,\ n\ge3\,, \qquad \overline{\lambda}_n = \frac{\log\|e_n\|}{\log\|e_{n-1}\|}\,,\ n\ge2\,. \tag{3.19a}$$

- Parameters ACOC, $\{\widehat{\rho}_n\}_{n\ge4}$, and ACLOC, $\{\widehat{\lambda}_n\}_{n\ge3}$:

$$\widehat{\rho}_n = \frac{\log\left(\|\widehat{e}_n\|/\|\widehat{e}_{n-1}\|\right)}{\log\left(\|\widehat{e}_{n-1}\|/\|\widehat{e}_{n-2}\|\right)}\,,\ n\ge4\,, \qquad \widehat{\lambda}_n = \frac{\log\|\widehat{e}_n\|}{\log\|\widehat{e}_{n-1}\|}\,,\ n\ge3\,. \tag{3.19b}$$

- Parameters ECOC, $\{\widetilde{\rho}_n\}_{n\ge5}$, and ECLOC, $\{\widetilde{\lambda}_n\}_{n\ge4}$:

$$\widetilde{\rho}_n = \frac{\log\left(\|\widetilde{e}_n\|/\|\widetilde{e}_{n-1}\|\right)}{\log\left(\|\widetilde{e}_{n-1}\|/\|\widetilde{e}_{n-2}\|\right)}\,,\ n\ge5\,, \qquad \widetilde{\lambda}_n = \frac{\log\|\widetilde{e}_n\|}{\log\|\widetilde{e}_{n-1}\|}\,,\ n\ge4\,. \tag{3.19c}$$

• Parameters PCOC, $\{\breve{\rho}_n\}_{n\geq 1}$, and PCLOC, $\{\breve{\lambda}_n\}_{n\geq 1}$:

$$\breve{\rho}_n = \frac{\|F(x_n)\|/\|F(x_{n-1})\|}{\|F(x_{n-1})\|/\|F(x_{n-2})\|}\,,\ n\geq 3\,, \qquad \breve{\lambda}_n = \frac{\log\|F(x_n)\|}{\log\|F(x_{n-1})\|}\,,\ n\geq 2\,. \tag{3.19d}$$

Approximations COC, ACOC and ECOC have been used in Grau *et al.* (2011, [23]). A complete study of these parameters has been carried out to compute the local convergence for four iterative methods and seven systems of nonlinear equations. In our studies on iterative methods for systems of linear equations, the computation will be the approach the order for one of these five parameters (3.19b), (3.19c) or (3.19d). The (3.19b) and (3.19d) parameters are the most efficient, and (3.19c) and (3.19a) parameters are the most accurate.

Chapter 4

Development of two inverse operators

In this chapter we consider, first of all, the Taylor development of a function in order to express the theoretical error equation of iterative algorithms with and without memory. By developing in a formal series expansion, we derive an expression of the inverse derivative operator (inverse of Jacobian) and the inverse divided difference operator. These are the most commonly used operators in the recent literature on iteration algorithms.

4.1 The vector error difference equation

Here we present a generalization to several variables of a technique used to compute analytically the error equation of iterative methods without memory for one variable.

We consider iterative methods to find a simple root of a system of nonlinear equations

$$F(x) = 0\,,$$

where $F : D \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^m$ is sufficiently differentiable and D is an open convex domain in $\mathbb{R}^m$. We assume that the solution of $F(x) = 0$ is $\alpha \in D$, at which $F'(\alpha)$ is non singular.

The key idea is to use formal power series. The vector expression of the error equation obtained by carrying out this procedure, is

$$e_{n+1} = G\left(F'(\alpha),\, F''(\alpha), \ldots\right) e_n^{\rho} + \mathcal{O}\left(e_n^{\rho+1}\right),$$

where ρ is a nonnegative integer. If the iterative scheme is with memory *(j points)* we obtain Grau *et al.* (2011, [25]):

$$e_{n+1} = H\left(F'(\alpha),\, F''(\alpha), \ldots\right) e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j} + \mathcal{O}(e_n^{a_1} e_{n-1}^{a_2} \ldots e_{n-j+1}^{a_j}),$$

where a_k are nonnegative integers for $1 \leq k \leq j$.

4.2 Notation

To obtain the vector equation of the error, we need some known results that, for ease of reference, are included in the following. Let $F : D \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^m$ be sufficiently differentiable (Fréchet-differentiable) in D, and therefore with continuous derivatives. If we consider the kth derivative of F at $a \in \mathbb{R}^m$, we have the k-linear function

$$\begin{aligned} F^{(k)}(a) : \mathbb{R}^m \times \overset{k}{\cdots} \times \mathbb{R}^m &\longrightarrow \mathbb{R}^m \\ (h_1, \ldots, h_k) &\longmapsto F^{(k)}(a)\,(h_1, \ldots, h_k). \end{aligned}$$

That is, $F^{(k)}(a) \in \mathscr{L}\left(\mathbb{R}^m \times \overset{k}{\cdots} \times \mathbb{R}^m, \mathbb{R}^m\right) \equiv \mathscr{L}_k\left(\mathbb{R}^m, \mathbb{R}^m\right)$.

It has the following properties:

P1. $F^{(k)}(a)\,(h_1, \ldots, h_{k-1}, \cdot\,) \in \mathscr{L}\,(\mathbb{R}^m, \mathbb{R}^m) \equiv \mathscr{L}\,(\mathbb{R}^m)$.

P2. $F^{(k)}(a)\,(h_{\sigma(1)}, \ldots, h_{\sigma(k)}) = F^{(k)}(a)\,(h_1, \ldots, h_k)$, where σ is any permutation of the set $\{1, 2, \ldots k\}$.

Notice that from P1 and P2 we can use the following notation:

N1. $F^{(k)}(a)\,(h_1, \ldots, h_k) = F^{(k)}(a)\,h_1 \cdots h_k$. For $h_j = h$, $1 \leq j \leq k$, we write $F^{(k)}(a)\,h^k$.

N2. $F^{(k)}(a)\,h^{k-1}\,F^{(l)}(a)\,h^l = F^{(k)}(a)\,F^{(l)}(a)\,h^{k+l-1}$.

Hence, we can also express $F^{(k)}(a)\,(h_1, \ldots, h_k)$ as

$$F^{(k)}(a)\,(h_1, \ldots, h_{k-1})\,h_k = F^{(k)}(a)\,(h_1, \ldots, h_{k-2})\,h_{k-1}\,h_k = \ldots = F^{(k)}(a)\,h_1 \cdots h_k.$$

For any $q = a + h \in \mathbb{R}^m$ lying in a neighborhood of $a \in \mathbb{R}^m$, assuming that $[F'(a)]^{-1}$ exists, and taking into account the previous notation, we write Taylor's formulae in the following way:

$$F(a+h) = F(a) + F'(a)h + \frac{1}{2!}F^{(2)}(a)\,h^2 + \cdots + \frac{1}{(q-1)!}F^{(q-1)}(a)\,h^{q-1} + \mathcal{O}_q\,, \tag{4.1}$$

$$= F(a) + F'(a)\left(h + \sum_{k=2}^{q-1} A_k(a)\,h^k + \mathcal{O}_q\right), \tag{4.2}$$

where $A_k(a) = \dfrac{1}{k!}\left[F'(a)\right]^{-1} F^{(k)}(a) \in \mathscr{L}_k\left(\mathbb{R}^m, \mathbb{R}^m\right),\ 2 \le k \le q-1,$ and $\mathcal{O}_q = \mathcal{O}(h^q)\,.$

4.3 Symbolic computation of the inverse of the Jacobian operator

We assume that $F : D \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^m$ has at least q-order derivatives with continuity on D for any $x \in \mathbb{R}^m$ lying in a neighborhood of a simple zero, $\alpha \in D$, of the system $F(x) = 0$. We can apply Taylor's formulae to $F(x)$. By setting $e = x - \alpha$, as the local order and assuming that $[F'(\alpha)]^{-1}$ exists, we have

$$F(x) = F(\alpha + e) = \Gamma\left(e + \sum_{k=2}^{q-1} A_k\, e^k\right) + \mathcal{O}_q\,, \tag{4.3}$$

where

$$A_k = A_k(\alpha)\,,\ k \ge 2\,,\ \text{with}\ \ \Gamma = F'(\alpha)\,,\ e^k = (e, \overset{k}{\cdots}, e) \in \mathbb{R}^m \times \overset{k}{\cdots} \times \mathbb{R}^m\,.$$

Moreover, from (4.3) noting the identity by I, the derivatives of $F(x)$ can be written as

$$F'(x) = \Gamma\left(I + \sum_{k=2}^{q-1} k\,A_k\,e^{k-1}\right) + \mathcal{O}_q\,, \tag{4.4}$$

$$F''(x) = \Gamma\left(\sum_{k=2}^{q-2} k\,(k-1)\,A_k\,e^{k-2}\right) + \mathcal{O}_{q-1}\,, \tag{4.5}$$

$$F'''(x) = \Gamma\left(\sum_{k=3}^{q-3} \frac{k!}{(k-3)!}\,A_k\,e^{k-3}\right) + \mathcal{O}_{q-2}\,, \tag{4.6}$$

and so forth up to order q.

By developing in a formal series expansion of e the inverse of $F'(x)$ is

$$\left[F'(x)\right]^{-1} = \left(I + \sum_{j=1}^{4} K_j e^j + \mathcal{O}_5\right)\Gamma^{-1}\,, \tag{4.7}$$

$$\begin{aligned}
\text{with}\quad K_1 &= -2\,A_2\,,\\
K_2 &= 4\,A_2^2 - 3\,A_3\,,\\
K_3 &= -8\,A_2^3 + 6\,A_2\,A_3 + 6\,A_3\,A_2 - 4\,A_4\,,\\
K_4 &= 16\,A_2^4 - 12\,A_2^2\,A_3 - 12\,A_2\,A_3\,A_2 - 12\,A_3\,A_2^2 + 8\,A_2\,A_4 + 8\,A_4\,A_2\,,
\end{aligned}$$

where we use the following notation: $A_k^q\,e^{q+1} = (A_k\,e)^{q-1}\,\left(A_k\,e^2\right)$.

4.3.1 Example: Newton's method

We consider Newton's method that we can write as

$$X = x - F'(x)^{-1}\,F(x). \tag{4.8}$$

The expression of the error $E = x - \alpha$ in terms of e is built up by subtracting α from both sides of (6.9) and taking into account (4.4) and (4.7). Namely,

$$\begin{aligned}
E &= e - \left(I + \sum_{j=1}^{3} K_j e^j + \mathcal{O}_4\right)\Gamma^{-1}\,\Gamma\left[e + \sum_{k=2}^{4} A_k\,e^k + \mathcal{O}_5\right]\\
E &= A_2\,e^2 + 2\,(A_3 - A_2^2)\,e^3 + (3\,A_4 - 4\,A_2 A_3 - 3\,A_3 A_2 + 4\,A_2^3)\,e^4 + \mathcal{O}_5\,.
\end{aligned} \tag{4.9}$$

The result (4.9) agrees with the classical asymptotic constant in the one-dimensional case and states that Newton's method has at least local order 2. Note that the terms $A_2 A_3$ and $A_3 A_2$ are noncommutative.

4.4 Development of the inverse of the first-order divided differences of a function of several variables

We assume that $F : D \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}^m$ has, at least, *qth*-order derivatives with continuity on D. We consider the first divided difference operator of F in $\mathbb{R}^m$ as a mapping

$$\begin{aligned}
[-,-;F]:\ D\times D &\longrightarrow \mathcal{L}(\mathbb{R}^m,\mathbb{R}^m)\\
(x+h,x) &\longrightarrow [x+h,x\,;\,F]\,,
\end{aligned}$$

which, for all $x, x+h \in D$, is defined by

$$[x+h\,,x\,;\,F]\,h = F(x+h) - F(x)\,, \tag{4.10}$$

where $\mathcal{L}(\mathbb{R}^m,\mathbb{R}^m)$ denotes the set of bounded linear functions (see [48,55] and references therein). For F that are sufficiently differentiable in D, we can write:

$$F(x+h) - F(x) = \int_x^{x+h} F'(z)\,dz \;= \int_0^1 F'(x+th)\,dt. \tag{4.11}$$

By developing $F'(x+th)$ in Taylor's series at the point $x \in \mathbb{R}^m$ and integrating we obtain

$$[x+h\,,x\,;\,F] = F'(x) + \frac{1}{2}F''(x)h + \cdots + \frac{1}{q!}F^{(q)}(x)h^{q-1} + \mathcal{O}(h^q) \tag{4.12}$$

By developing $F(x)$ and its derivatives in Taylor's series at the point $x = \alpha + e$ lying in a neighborhood of a simple zero, $\alpha \in D$, of the system $F(x) = 0$, assuming that Γ^{-1} exists, we obtain the expressions (4.3)–(4.6). Next, by replacing these expressions in (4.12), we obtain:

$$[y\,,x\,;\,F] = \Gamma\left(I + A_2(2e+h) + A_3(3\,e^2 + 3\,e\,h + h^2) + \dots\right), \tag{4.13}$$

or more precisely

$$[x+h\,,x\,;\,F] = \Gamma\left(\,I \,+ \sum_{k=1}^{q-1} S_k(h,e) \,+ \dots\right), \tag{4.14}$$

where

$$S_k(h,e) = A_{k+1}\sum_{j=1}^{k+1}\binom{k+1}{j} e^{k-j+1}\,h^{j-1}\,,\quad k \geq 1\,.$$

Setting $y = x+h$, $\varepsilon = y-\alpha$ and $h = \varepsilon - e$ in (4.13) or in (4.14) we obtain

$$[y\,,x\,;\,F] = \Gamma\left(I + A_2(\varepsilon+e) + A_3(\varepsilon^2 + \varepsilon\,e + e^2) + \dots\right), \tag{4.15}$$

or more precisely

$$[y\,,x\,;\,F] = \Gamma\left(\,I \,+ \sum_{k=1}^{q-1} T_k(\varepsilon,e) \,+ \mathcal{O}_q(\varepsilon,e)\right), \tag{4.16}$$

where $T_k(\varepsilon,e) = \;A_{k+1}\sum_{j=0}^{k}\varepsilon^{k-j}\,e^j\,,\;\; k \geq 1$, and $\mathcal{O}_q(\varepsilon,e)$ according to the next definition.

Definition 8. *We say that:*
a function depending on ε *and* e *is an* $\mathcal{O}_q(\varepsilon, e)$ *if it is an* $\mathcal{O}(\varepsilon^{q_0}\, e^{q_1})$ *where* $q_0 + q_1 = q$, $q_0 \geq 0$, $q_1 \geq 0$.

If we expand in formal power series of e and ϵ, the inverse of the divided difference given in (4.15) or in (4.16) may be written as:

$$[y\,, x\,;\, F]^{-1} = \Big(I - A_2(\epsilon + e) - A_3(\varepsilon^2 + \varepsilon\, e + e^2) + A_2(\epsilon + e)A_2(\epsilon + e) + O_3(\varepsilon, e)\Big)\,\Gamma^{-1}\,. \tag{4.17}$$

Notice that equation (4.17) is written explicitly until the $2nd$-degree in ε and e, while in each specific circumstance it will be adapted and reduced to the necessary terms, with an effective contribution to the computation of the local order of convergence.

This development of the divided difference operator (4.17) was first used in our study Grau *et al.* (2011, [24]).

4.4.1 Example: Secant method

The generic case, (4.15), (4.16) or (4.17) can be adapted to different cases. One example is the well-known iterative method, the Secant method, which is defined by the algorithm:

$$x_{n+1} = x_n - [x_{n-1}\,, x_n\,;\, F]^{-1}\, F(x_n)\,, \quad x_0\,, x_1 \in D. \tag{4.18}$$

If $y = x_{n-1}$ and $x = x_n$ in (4.15), then we obtain an expression of the operator $[x_{n-1}\,, x_n\,;\, F]$ in terms of $e_{n-1} = x_{n-1} - \alpha$ and $e_n = x_n - \alpha$. If we expand in formal power series of e_{n-1} and e_n the inverse of the divided difference operator in the Secant method we obtain:

$$[x_{n-1}, x_n; F]^{-1} = \big(I - A_2\,(e_{n-1} + e_n) + (A_2^2 - A_3)e_{n-1}^2 + \mathcal{O}(e_{n-1}^2)\big)\,\Gamma^{-1}\,, \tag{4.19}$$

where $A_2^2 e_{n-1}^2 = (A_2 e_{n-1})^2$. The expression of the error $e_{n+1} = x_{n+1} - \alpha$ in terms of e_n and e_{n-1} for the Secant method is built up by subtracting α from both sides of (6.1). Taking into account (4.3) and (4.19), we have

$$\begin{aligned} e_{n+1} &= e_n - \Big[I - A_2\,(e_{n-1} + e_n) + (A_2^2 - A_3)e_{n-1}^2 + \mathcal{O}(e_{n-1}^2)\Big] \cdot \Big[e_n + A_2 e_n^2 + \mathcal{O}(e_n^3)\Big] \\ &= A_2\, e_{n-1}\, e_n + (A_3 - A_2^2)e_{n-1}^2\, e_n + o\,(e_{n-1}^2\, e_n)\,, \end{aligned} \tag{4.20}$$

where the indicial polynomial (see (2.17)) of the error difference equation (4.20) is $t^2 - t - 1 = 0$, with only one positive real root, which is the R-order of convergence of

the Secant method, $\phi = (1 + \sqrt{5})/2$. This result agrees with the classical asymptotic constant in the one-dimensional case, and states that the Secant method has at least local order ϕ.

Another complete expression of the error expression for the Secant method can be found in our study Grau *et al.* (2011, [30]).

Chapter 5

Efficiency indices

We are interested in comparing iterative processes for approximating a solution α of a system of nonlinear equations. In the scalar case, the parameters of the efficiency index (EI) and computational efficiency (CE) are possible indicators of the efficiency of the scheme. The concept of computational efficiency index (CEI) is introduced in our work. We considerer the computational efficiency index as a generalization to the multi-dimensional case. We then show the power of this parameter by applying it to some numerical examples.

5.1 Efficiency index and computational efficiency

To compare iterative methods for solving scalar nonlinear equations, the efficiency index suggested by Ostrowski ([49], 1960, p. 20) is widely used.

Definition 9. (Ostrowski, 1960) *The efficiency index of an iterative method is defined by*

$$EI = \rho^{1/\theta}, \tag{5.1}$$

where θ represents the number of the evaluations of functions required to carry out the method per iteration, and ρ is the local order of convergence of the method (see [49]).

Another classical measure of efficiency for iterative methods applied to scalar nonlinear equations is the computational efficiency proposed by Traub [57]. Note that the number of arithmetic operations required to evaluate the iteration function can differ greatly between two problems solved by the same scheme. If it is considered relevant to account for this cost, the term computational efficiency should be used.

Definition 10. (Traub, 1964) *The computational efficiency of an iterative method is defined by*

$$CE = \rho^{1/\omega}, \tag{5.2}$$

where ρ is the local order of convergence of the method and ω the number of operations, expressed in product units, that are needed to compute each iteration without considering the evaluations of the functions (see [57]).

The efficiency index for Newton's method is $2^{1/2} \approx 1.414$ because an iteration step of Newton requires the computation of $f(x)$ and $f'(x)$, then $\omega = 2$ and the local order is $\theta = 2$. Note that the parameter (5.1) is independent of the expression of $f(x)$ and its derivative, while the parameter (5.2) aims to reflect the cost, in arithmetic operations, of solving the problem $f(x) = 0$ by a method of order ρ. For the Newton's method is $2^{1/1}$, because an iteration step of Newton requires one quotient ($\omega = 1$).

More precisely, note that an iteration step requires two actions: first the calculation of new function values; and then the combination of data to calculate the next iterate. The evaluation of functions requires invocation of subroutines, whereas the calculation of the next iterate requires only a few arithmetic operations. In general, these few arithmetic operations are not considered in the scalar case. In the case of iterative methods for systems of nonlinear equations, matrix operations include arithmetic operations, such as solving linear systems or obtaining inverse matrices. From our point of view, an efficiency indicator for systems of nonlinear equations must account for the cost of two steps.

5.2 Computational efficiency index

The traditional way to present the computational efficiency index of iterative methods (see [49, 57]) is adapted for systems of nonlinear equations. When we deal with a system of nonlinear equations, the total operational cost is the sum of the evaluations of functions (the function and the derivatives involved) and the operational cost of a step of the iterative method.

Definition 11. *The computational efficiency index (CEI) and the computational cost per iteration ($\mathcal{C}$) are defined by*

$$CEI(\mu_0, \mu_1, m, \ell) = \rho^{\frac{1}{\mathcal{C}(\mu_0, \mu_1, m, \ell)}}, \tag{5.3}$$

where $\mathcal{C}(\mu_0, \mu_1, m, \ell)$ *is the computational cost given by*

$$\mathcal{C}(\mu_0, \mu_1, m, \ell) = a_0(m)\,\mu_0 + a_1(m)\,\mu_1 + \omega(m, \ell), \tag{5.4}$$

$a_0(m)$ *represents the number of evaluations of the scalar functions* $(F_1, \ldots, F_m)$ *used in one step of the iterative scheme.*

$a_1(m)$ *is the number of evaluations of scalar functions of* F'*, say* $\dfrac{\partial F_i}{\partial x_j}$, $1 \le i, j \le m$.

$\omega(m, \ell)$ *represents the number of products needed per iteration.*

The constants μ_0 *and* μ_1 *are the ratios between the products and evaluations required to express the value of* $\mathcal{C}(\mu_0, \mu_1, m, \ell)$ *in terms of products, and* ℓ *is the cost of one quotient in products. (see [24, 25, 30, 55]).*

From the definition, we can derive the following properties:

$$CEI(\mu_0, \mu_1, m, \ell) > 1, \qquad \lim_{m \to +\infty} CEI(\mu_0, \mu_1, m, \ell) = 1. \tag{5.5}$$

Notice that for $\mu_0 = \mu_1 = 1$ and $\omega(m, \ell) = 0$, (5.3) is reduced to (5.1), that is the classic efficiency index of an iterative method, say $EI = \rho^{1/\theta}$, in the scalar case. Also observe that, if $a_0(m) = a_1(m) = 0$, (5.3) is written in the scalar case as (5.2); namely, $CE = \rho^{1/\omega}$.

According to (5.4), the factors μ_0, μ_1 can be estimated. To do this, we express the cost of evaluating the elementary functions in terms of products. This cost, depends on the machine, the software and the arithmetic used. In [19, 58, 59], comparisons are made between a multi-precision library, MPFR, and other computing libraries. Tables 5.1–5.2 show our own estimation of the cost of the elementary functions in product units, where the running time of one product is measured in milliseconds.

Software	$x * y$	x/y	$\sqrt{x}$	$\exp(x)$	$\ln(x)$	$\sin(x)$	$\cos(x)$	$\arctan(x)$
Matlab 2009b	4.5E−7 *ms*	10	55	80	145	35	50	65
Maple13 16 digits	1.2E−3 *ms*	1	10	25	45	25	20	95
Maple13 1024 digits	4.0E−2 *ms*	1	5	45	10	90	90	90
Maple13 4096 digits	3.5E−1 *ms*	1	5	50	10	105	105	100

Table 5.1: Computational cost of elementary functions computed with Matlab 2009b and Maple 13 in an Intel ®Core(TM)2 Duo CPU P8800 (32-bit machine) Microsoft Windows 7 Professional processor, where $x = \sqrt{3} - 1$ and $y = \sqrt{5}$.

The values presented in Table 5.1 were rounded to 5 units, because of the huge variability obtained in the different repetitions. In contrast, Table 5.2 gives the averages, since the

variability was very low in this case. In addition, the compilation of C++ ensured that the function `clock()` gave exactly the CPU time invested by the program. Table 5.2 shows that some relative values for the product were lower in multiple precision than in double precision, although the absolute time spent on a product is was much higher in multiple precision.

Arithmetic	$x * y$	x/y	$\sqrt{x}$	$\exp(x)$	$\ln(x)$	$\sin(x)$	$\cos(x)$	$\arctan(x)$
C++ `double`	2.3E−7 *ms*	29	29	299	180	181	192	237
C++ MPFR 1024 digits	1.16E−2 *ms*	2.4	1.7	62	57	69	65	200
C++ MPFR 4096 digits	1.04E−1 *ms*	2.5	1.7	88	66	116	113	228

Table 5.2: Computational cost of elementary functions computed with a program write in C++, compiled by `gcc(4.3.3)` for i486-linux-gnu with `libgmp (v.4.2.4)` and `libmpfr (v.2.4.0)` libraries in a processor Intel ®Xeon E5420, 2.5GHz, 6MB cache, where $x = \sqrt{3}-1$ and $y = \sqrt{5}$.

This measure of computing efficiency is clearly more satisfactory than considering only the number of iterations or only the number of evaluated functions, both of which are used widely by other authors. Any change in software or hardware requires us to recompute the elapsed time of elemental functions, quotients and products. A complete study of this parameter was carried out for all methods and examples presented in this thesis: [15, 16, 24–26].

5.2.1 On the computational cost of obtaining the inverse of a linear operator

All iterative methods for non-linear systems $F(x) = 0$ have a common point: the calculation of the inverse matrix of a linear operator. In Newton's method, the inverse of $F'(x))$ must be computed; in the Secant method the inverse of the first divided difference operator $[x, y; F]$ must be computed.

From a computational perspective, we do not need the inverse operator expression, knowing how to act is sufficient. Therefore, in all cases to obtain the inverse operator we will solve the "equivalent" linear system : $v = \Upsilon^{-1} \cdot F(x) \iff \Upsilon \cdot v = F(x)$. At this point, we have solved the linear system from the LU decomposition of the matrix associated with the linear operator Υ, and solved the two linear systems associated with a lower triangular matrix with ones in the main diagonal ($L \cdot w = F(x)$) and the other upper triangular matrix $U \cdot v = w$.

The cost in number of products of the LU factorization is $\frac{m}{6}(2m^2-3m+1)$ and the number of divisions is $\frac{m}{2}(m-1)$ where m is the number of equations. We solve a triangular linear system of equations with a cost of $m(m-1)$ products and m divisions (see [10]).

If the calculations are performed in MAPLE (see Table 5.1) the products and quotients have the same cost, that is $\ell = 1$. Then, the computation of an inverse operator adds $\frac{m}{3}(m^2 + 3m - 1)$ to the total cost $\mathcal{C}$. If we use a software where the cost ratio ℓ takes values $\ell > 1$ (see Table 5.2), then the computation of an inverse operator adds $\frac{m}{6}(2m^2 - 3(1+\ell)m + 3\ell - 5)$ to the total cost $\mathcal{C}$.

5.2.2 An example: Ostrowski's method

We analyze the computational efficiency index of two iterative methods. The first one is a generalization to several variables of Ostrowski's method; the second derived from Ostrowski's method, retrieved by freezing the inverse linear operator (which generalizes [21]). Our study was published in 2011, in memory of the mathematician Alexander M. Ostrowski on the *25th* anniversary of this death (see [26]).

Ostrowski's scheme The iterative method starts with Newton's method as first step. Namely, for $n \geq 0$ and $x_0 \in D$,

$$\begin{aligned} x_n^{(1)} &= x_n - F'(x_n)^{-1} F(x_n)\,, \\ x_n^{(2)} &= = x_n^{(1)} - \kappa^{-1} F(x_n^{(1)})\,, \quad \text{where} \quad \kappa = 2\left[x_n^{(1)}, x_n; F\right] - F'(x_n)\,, \end{aligned} \tag{5.6}$$

where the operator $[\cdot,\cdot;F]^{-1}$ is defined in Section 4.4, and the last computed term is the new iteration point; $x_{n+1} = \Theta_1(x_n; x_n^{(1)}) = x_n^{(2)}$.

Frozen Ostrowski's scheme The iterative method is a three-step iterative method, with Ostrowski's method as the two first steps. By taking $x_0 \in D$, we obtain

$$\begin{aligned} x_n^{(1)} &= x_n - F'(x_n)^{-1} F(x_n)\,, \quad n \geq 0\,, \\ x_n^{(2)} &= x_n^{(1)} - \kappa^{-1} F(x_n^{(1)})\,, \quad \text{where} \quad \kappa = 2\left[x_n^{(1)}, x_n; F\right] - F'(x_n)\,, \\ x_n^{(3)} &= x_n^{(2)} - \kappa^{-1} F(x_n^{(2)})\,. \end{aligned} \tag{5.7}$$

and the last computed term is the next iteration point, $x_{n+1} = \Theta_2(x_n; x_n^{(1)}, x_n^{(2)}) = x_n^{(3)}$.

The vector error equation can be summed up with the following theorem.

Theorem 5.2.1. *The iterative method Θ_1, defined in (5.6), has a local order of convergence of at least 4, with the following error equation*

$$e_{n+1}^{(1)} = (A_2^2 - A_3)A_2 e_n^4 + \mathcal{O}_5\,, \tag{5.8}$$

and Θ_2 , defined in (5.8), has a local R-order of convergence of at least 6 and its vector error difference equation can be written as

$$e_{n+1}^{(2)} = (2A_2^2 - A_3)(A_2^2 - A_3)A_2 e_n^6 + \mathcal{O}_7\,. \tag{5.9}$$

Recall that for nonlinear systems with m equations and m unknowns, the computational efficiency index (CEI) of an iterative method of order of convergence ρ is

$$CEI(\mu_0, \mu_1, m, \ell) = \rho^{1/\mathcal{C}(\mu_0,\mu_1,m,\ell)}, \tag{5.10}$$

where $\mathcal{C}(\mu_0, \mu_1, m)$ is the computational cost given by

$$\mathcal{C}(\mu_0, \mu_1, m) = \mu_0\, a_0(m) + \mu_1\, a_1(m) + \omega(m, \ell), \tag{5.11}$$

where the parameters $a_0(m)$, $a_1(m)$, $\omega(m, \ell)$, μ_0 and μ_1 are defined in (5.4) . We proceed as follows to obtain the value for a particular iterative method.

In this study $a_0(m)$ represents the number of evaluations of the scalar functions $(F_1, \ldots, F_m)$ used in the evaluation of F and $[y, x; F]$. When we evaluate F in any iterative function we calculate m component functions, and if we compute a divided difference then we evaluate $m(m-1)$ scalar functions, where $F(x)$ and $F(y)$ are computed separately. That is, we have $a_0(m) = m^2 + m$ for the scheme (5.6) and $a_0(m) = m^2 + 2m$ for the scheme (5.7).

The number of evaluations of scalar functions of F', say $\dfrac{\partial F_i}{\partial x_j}$, $1 \le i, j \le m$ is m^2 for any new derivative F', that is $a_1(m) = m^2$.

To the function $\omega(m, \ell)$, we must add m^2 quotients from any divided difference. In order to compute an inverse linear operator, we have $(m^3 - m)/3$ products or quotients in the decomposition LU, and m^2 products or quotients in the resolution of two triangular linear systems will be added to $\omega(m, \ell)$.

Recall that ℓ depends on the software used. The study presented in [26] was performed with the MAPLE software, that implies $\ell = 1$ (see Table 5.1). The values μ_0 and μ_1 depend on the expression of the system of nonlinear equations and the software used, but the domain is $(\mu_0, \mu_1) \in \mathcal{M} = (0, +\infty) \times (0, +\infty)$.

Taking into account the previous considerations, we have $a_0(m) = m^2 + m$, $a_1(m) = m^2$, $\omega(m) = 2(m^3 - m)/3 + 4m^2$ for the Ostrowski iterative function (5.6), that is

$$\mathcal{C}_1 = (m^2 + m)\mu_0 + m^2\mu_1 + 4m^2 + \frac{2(m^3 - m)}{3} \quad \text{and} \quad CEI_1 = 4^{1/\mathcal{C}_1}. \tag{5.12}$$

In an analogous way for (5.8), we get

$$\mathcal{C}_2 = (m^2 + 2m)\mu_0 + m^2\mu_1 + 5m^2 + \frac{2(m^3 - m)}{3} \quad \text{and} \quad CEI_2 = 6^{1/\mathcal{C}_2}. \tag{5.13}$$

In order to compare the two iterative methods Θ_1 and Θ_2, we define the ratio

$$R_{2,1} = \frac{\log CEI_2(\mu_0, \mu_1, m)}{\log CEI_1(\mu_0, \mu_1, m)} = \frac{\log(\rho_2)\,\mathcal{C}_1(\mu_0, \mu_1, m)}{\log(\rho_1)\,\mathcal{C}_2(\mu_0, \mu_1, m)}. \tag{5.14}$$

If $R_{2,1} > 1$ then the iterative method Θ_2 is more efficient than Θ_1.

We have made this comparison for the Ostrowski's method (Θ_1) and the Frozen Ostrowski's method (Θ_2), where m is a positive integer $m \geq 2$. The particular boundary $R_{2,1} = 1$ expressed by μ_0 and written as a function of μ_1 and m is

$$G_{2,1} = \frac{1}{3}\frac{2qm^2 + (3q\mu_1 + 12q - 3r)m - 2q}{r - qm - q}, \tag{5.15}$$

where $q = \ln(3/2)$ and $r = 2\ln(2)$. This function has the vertical asymptote for $m = (r - q)/q = 2.419\ldots$

Notice that the numerator of (5.15) is positive for $m \geq 1$, since for $m = 1$ it yields $3q\mu_1 + 12q - 3r > 0$. The denominator of $G_{2,1}$ is negative for $m > 2.419\ldots$. Consequently, we obtain that $\mu_0 = G_{2,1}(\mu_1, m)$ is always negative for $m \geq 3$. That is, the boundary is out of the admissible region for $m \geq 3$ and we have $\forall(\mu_0, \mu_1) \in \mathcal{M},\ CEI_2 > CEI_1$.

Finally, for $m = 2$ the boundary (5.15) is the straight line with positive slope

$$\mu_0 = \frac{2(q\mu_1 + 5q - r)}{r - 3q},$$

where $CEI_1 > CEI_2$ over it and $CEI_2 > CEI_1$ under it (see [26]).

The numerical examples given in the study presented in [26] illustrate the theoretical results presented here. Furthermore, a sequence that approximates the order of convergence is generated for the examples, and confirms in a numerical way that the order of the methods is well-deduced.

In all the computations we have substituted the computational order of convergence (COC) by the approximation ACOC defined in (3.7b). All computed approximations of ACOC have at least two correct decimals. That is, if $\widehat{\rho}_I = \rho \pm \Delta\widehat{\rho}_I$, where ρ is the local order of convergence and $\Delta\widehat{\rho}_I$ is the upper error bound of ACOC at I-step, the last computed step, then we obtain $\Delta\widehat{\rho}_I < 0.5{\cdot}10^{-4}$ for Ostrowski's method and $\Delta\widehat{\rho}_I < 0.5{\cdot}10^{-3}$ for the frozen Ostrowski's method. This means that in all computations of ACOC, we obtain at least four (three) significant digits, respectively.

Chapter 6

Two iteration families

In this chapter, we present some of the iterative methods developed in our work. For each of them, we give their function of iteration and their error equation. Moreover, the efficiency of the iterative schemes is discussed. Usually, we work with methods derived from Newton's method or the Secant method. Specifically, we summarize papers [24, 25]. Other iterative methods have been developed in papers [15, 16, 26]

In this chapter, we have used the notation established in previous sections.

6.1 Family Φ_1

This family is a k-step iterative method with memory [57]. It is a generalization of the classical Secant method [2, 48, 55]. The complete work can be found in [24].

6.1.1 Secant method

We call $\Phi_{1,1}$ to the well-known iterative Secant method. That is, by setting $x_0, x_1 \in D$, we have

$$x_{n+1} = \Phi_{1,1}(x_{n-1}, x_n) = x_n - [x_{n-1}, x_n; F]^{-1} F(x_n), \quad x_0, x_1 \in D, \tag{6.1}$$

where the operator $[x_{n-1}, x_n; F]^{-1}$ is defined in Section 4.4. This method has local order of convergence at list convergence of the Secant method, $\phi = (1 + \sqrt{5})/2$.

6.1.2 k-step iterative method

For $n \geq 0$ and for $0 \leq j \leq k-1$, given $x_{n-1}\,,\ x_n \in D$, we consider the following iterative method

$$x_n^{(j+1)} = \Phi_{1,j+1}(x_{n-1}, x_n) = x_n^{(j)} - [\,x_{n-1}, x_n; F]^{-1}\, F(x_n^{(j)})\,, \tag{6.2}$$

where $x_n^{(0)} = x_n$ and in the last step, the last computed term is the new iteration point $x_{n+1} = x_n^{(k)}$. Note that for $j = 0$, the algorithm (6.2) is the Secant method. We call these methods the *frozen Secant method* since the operator $[\,x_{n-1}, x_n; F]^{-1}$ is the same in all steps (frozen operator).

We are in a condition to state the following theorem.

Theorem 6.1.1. *The k–step iterative method (6.2) has local order of convergence*

$$\rho_k = \frac{1 + \sqrt{1 + 4\,k}}{2} \tag{6.3}$$

at least, and the vectori error difference equation can be written as

$$e_{n+1} = e_n^{(k)} = (A_2\, e_{n-1})^k\ e_n + o(e_{n-1}^k\, e_n). \tag{6.4}$$

6.1.3 Family efficiency

In view of the expressions obtained for ρ_k (6.3), the local order of the methods (6.2) increases with k. What about efficiency, does it increase or decrease?

The computational cost (5.4) of the scheme $x_n^{(k)} = \Phi_{1,k}(x_{n-1}, x_n)$ takes $\mu_1 = 0\,,\ a_1(m) = 0\,,$ $\ell = 1\,,\ \mu_0 = \mu\,,\ a_0(k,m) = m(m-1) + km\,,\ \omega(k,m) = \frac{m^3-m}{3} + (k+1)m^2$ then

$$\mathcal{C}(k,\mu,m) = m(m-1) + km\,\mu + \frac{m^3 - m}{3} + (k+1)m^2, \tag{6.5}$$

$$= \underbrace{\left(m\,\mu + m^2\right)}_{M}\ k\ + \underbrace{\left((m(m-1)\,\mu) + \frac{1}{3}(m^3 - m) + m^2\right)}_{N}. \tag{6.6}$$

And the computational efficiency index (5.3) is

$$CEI(k,\mu,m) = \rho_k^{\ \frac{1}{M\,k+N}}\,. \tag{6.7}$$

In order to compare the performance of the various methods, we compute the optimal point of (6.7) for the k-step iterative method. We consider $\frac{d}{dk}(\ln CEI(k,\mu,m)) = 0$, to obtain

$$\frac{1}{2}\left(\sqrt{1+4k}+1+4k\right)\ln\rho_k - k - \frac{M}{N} = 0, \tag{6.8}$$

where ρ_k is given in (6.3) as a function of k.

For a given couple (m,μ) we can solve (6.8) in terms of ρ_k using the Secant method. Subsequently we obtain k that is the value at which the computational efficiency index (6.7) attains its optimal point. The next table (6.1) shows the values of k as functions of m and μ that are positive integer solutions of (6.8).

º $\mu \setminus m$	2	3	4	5	6	7	8	9	10	20	40
0.5	2	2	2	2	3	3	3	3	3	4	7
1	2	2	2	2	3	3	3	3	3	5	7
5	1	2	2	3	3	3	4	4	4	5	8
10	1	2	2	3	3	4	4	4	4	6	9
50	1	2	3	3	3	4	4	5	5	8	12
100	1	2	3	3	4	4	4	5	5	8	13
500	1	2	3	3	4	4	4	5	5	9	15

Table 6.1: Values of k for optimal efficiency where m is the dimension of the system and μ is the ratio between products and evaluations.

6.1.4 Numerical results.

The numerical computations were performed on the MAPLE algebraic system ($\ell = 1$) with two multiple precision arithmetic libraries (`Digits:=1024` and `Digits:=2048`), depending on the local order of the iterative methods used. The factor μ is estimated. In general μ has several unit values for the polynomial function, while its value is increased for transcendental functions and trigonometric functions, since it will easily reach several tens of hundreds (see Table 5.1).

The examples were chosen to check the optimal k value effectively. In the study [24], we present three situations. In the first case, for non-linear systems with $1 < \mu < 100$ for $m = 2$ equations the Secant method is optimal, and for $m = 3$ freeze once ($k = 2$) is optimal. In the second case, with $\mu = 100$ and two situations, $m = 3$ and $m = 4$, the optimal k are 2 and 3 respectively. Finally, when we increase the number of equations in the second case, we confirm that for $m = 8$ the optimal k is 4, and with one more equation the optimal k is 5.

In the computations, we substituted the computational order of convergence (COC) by the approximation ACOC defined in (3.7b). All computed approximations of ACOC have, at least, two correct decimals. That is, if $\widehat{\rho}_I = \rho \pm \Delta\widehat{\rho}_I$, where ρ is the local order of convergence and $\Delta\widehat{\rho}_I$ is the upper error bound of ACOC at I-step, the last computed step. Then, we obtain $\Delta\widehat{\rho}_I < 0.5 \cdot 10^{-2}$. This means that in all computations of ACOC we obtain at least three significant digits.

The same line of work has been followed in another study on the application of the frozen Newton's method, in collaboration with some members of the *Universidad Politécnica de Cartagena* (UPCT). As a result, an application for digital images was presented (see [3]).

6.2 Family Φ_2

Three variants of Newton's method with local order of convergence equal to three, four and five are analyzed. We explicitly give their vector error equation and their computational efficiency index. The detailed study can be found in [25].

6.2.1 Newton's method

Let $x_0 \in D \subset \mathbb{R}^m$, Newton's method is given by

$$x_{n+1} = \Phi_{2,1}(x_n) = x_n - F'(x_n)^{-1} F(x_n), \tag{6.9}$$

where the operator $[F(x_n)]^{-1}$ is defined in Section 4.3. The expression of the error $e_{n+1} = x_{n+1} - \alpha$ in terms of $e_n = x_n - \alpha$ is

$$e_{n+1} = A_2 e_n^2 + 2(A_3 - A_2^2) e_n^3 + (3A_4 - 4A_2A_3 - 3A_3A_2 + 4A_2^3) e_n^4 + \mathcal{O}_5. \tag{6.10}$$

6.2.2 Harmonic mean Newton's method

The first variant is a two-point iteration function without memory. We take Newton's method twice, but the second time $F'(x_n)$ is replaced by the harmonic mean of the derivatives of F at the points x_n and $x_{n+1}^{(1)}$. That is, for $n \geq 0$ and $x_0 \in D \subset \mathbb{R}^m$,

$$\begin{aligned} x_n^{(1)} &= \Phi_{2,1}(x_n) = x_n - [F'(x_n)]^{-1} F(x_n), \\ x_n^{(2)} &= x_n - \frac{1}{2}\left[F'(x_n)^{-1} + F'(x_n^{(1)})^{-1}\right] F(x_n), \end{aligned} \tag{6.11}$$

and $x_{n+1} = x_n^{(2)} = \Phi_{2,2}(x_n; x_n^{(1)})$. The vector error equation is

$$e_{n+1} = \frac{1}{2} A_3 e_n^3 + \mathcal{O}_4 , \tag{6.12}$$

and the local R-order of convergence is, at least equal to three. This result agrees with those obtained in [33], [50] and [57].

6.2.3 Traub's method

We generalize the one-dimensional iterative method from [57]. By taking $x_0 \in D \subset \mathbb{R}^m$, we obtain

$$\begin{aligned} x_n^{(1)} &= \Phi_{2,1}(x_n) , \\ z_n &= x_n^{(1)} - \frac{1}{2}\left[F'(x_n)\right]^{-1} F(x_n^{(1)}) , \quad n \geq 0 , \\ x_n^{(3)} &= x_n^{(1)} - 2\left[F'(x_n)\right]^{-1} F(z_n) , \end{aligned} \tag{6.13}$$

and $x_{n+1} = x_n^{(3)} = \Phi_{2,3}(x_n; x_n^{(1)})$. And finally, we consider a three-step iterative method that is a modification of the harmonic mean method (6.11).

6.2.4 Modification of the harmonic mean Newton's method

Let $x_0 \in D \subset \mathbb{R}^m$, this is defined by $x_{n+1} = \Phi_{2,4}(x_n, x_{n+1}^{(1)}, x_{n+1}^{(2)})$ where

$$\begin{aligned} x_n^{(1)} &= \Phi_{2,1}(x_n) , \\ x_n^{(2)} &= \Phi_{2,2}(x_n; x_n^{(1)}) , \quad n \geq 0 , \\ x_n^{(4)} &= x_n^{(2)} - \left[F'(x_n^{(1)})\right]^{-1} F(x_n^{(2)}) . \end{aligned} \tag{6.14}$$

and the next iterate is $x_{n+1} = x_{n+1}^{(4)}$.

We can sum up with the following theorem.

Theorem 6.2.1. *The iterative methods $\Phi_{2,3}$ and $\Phi_{2,4}$ defined in (6.13) and in (6.14) have a local order of convergence, of at least 4 and 5, and their vector error difference equations can be written as*

$$\epsilon_{n+1} = \frac{9}{2} A_2^3 \epsilon_n^4 + \mathcal{O}_5 , \quad \epsilon_n = x_n^{(3)} - \alpha , \tag{6.15}$$

$$\varepsilon_{n+1} = 8A_2(A_2\varepsilon_n^2)(A_3\varepsilon_n^3) + \mathcal{O}_6 , \quad \varepsilon_n = x_n^{(4)} - \alpha . \tag{6.16}$$

6.2.5 Family efficiency

For nonlinear systems with m equations and m unknowns, the computational efficiency index (CEI) of an iterative method of order of convergence ρ is

$$CEI(\mu_0, \mu_1, m) = \rho^{1/\mathcal{C}(\mu_0,\mu_1,m)}, \tag{6.17}$$

where $\mathcal{C}(\mu_0, \mu_1, m)$ is the computational cost given by

$$\mathcal{C}(\mu_0, \mu_1, m) = \mu_0\, a_0(m)\, m + \mu_1\, a_1\, m^2 + \omega(m). \tag{6.18}$$

In (6.18) a_0 and a_1 represent the number of evaluations of the scalar functions of $F(x)$ and $F'(x)$ respectively, $\omega(m, \ell)$ is the number of products per iteration and μ_0 and μ_1 are the ratios between products and evaluations required to express the value of $\mathcal{C}(\mu_0, \mu_1, m)$ in terms of products.

Table 6.2 shows, for each iterative method analyzed in this paper, $\Phi_{2,1}$–$\Phi_{2,4}$, the numbers a_0 and a_1, the number of products by scalar of an iteration, p_0, the number of resolutions of linear systems per iteration, p_1, and the number of resolutions of two triangular systems, p_2. Moreover, Table 6.2 presents the local order of convergence, ρ, and the computational cost, $\mathcal{C}(\mu_0, \mu_1, m)$.

Method	a_0	a_1	p_0	p_1	p_2	ρ	$\mathcal{C}(\mu_0, \mu_1, m)$
Φ_1	1	1	0	1	0	2	$m(2m^2 + 3(2\mu_1 + \ell + 1)\, m + 6\mu_0 + 3\ell - 5)/6$
Φ_2	1	2	1	2	0	3	$m(2m^2 + 3(2\mu_1 + \ell + 1)\, m + 3\mu_0 + 3\ell - 2)/3$
Φ_3	3	1	2	1	2	4	$m(2m^2 + 3(2\mu_1 + \ell + 5)\, m + 18\mu_0 + 15\ell - 5)/6$
Φ_4	2	2	1	2	1	5	$m(2m^2 + 3(2\mu_1 + \ell + 2)\, m + 6\mu_0 + 6\ell - 5)/3$

Table 6.2: Coefficients used in (5.4), local order of convergence and computational cost of the iterative methods $\Phi_{2,k}$, $1 \le k \le 4$.

The question is, is the higher order iterative method always the most efficient?

We compare the methods presented above. If we denote the computational efficiency indices of $\Phi_{2,i}$ by $CEI_i(\mu_0, \mu_1, m)$, then from (5.3) we define the ratio

$$R_{i,j} = \frac{\log CEI_i(\mu_0, \mu_1, m)}{\log CEI_j(\mu_0, \mu_1, m)} = \frac{\log(\rho_i)\, \mathcal{C}_j(\mu_0, \mu_1, m)}{\log(\rho_j)\, \mathcal{C}_i(\mu_0, \mu_1, m)}, \tag{6.19}$$

where $\mathcal{C}_i(\mu_0, \mu_1, m)$ are the values in Table 6.2. For $R_{i,j} > 1$ the iterative method Φ_i is more efficient than Φ_j. The boundary between two computational efficiencies is made by $R_{i,j} = 1$.

We study the sets, $R_{4,1} = 1$, $R_{4,2} = 1$ and $R_{4,3} = 1$ to compare the computational efficiency index of iterative method $\Phi_{2,4}$, say CEI_4 with the CEI of the other three methods, in the (μ_1, μ_0)-plain for $m \geq 2$.

The following theorem (see [25]) summarizes the results for the computational efficiency indexes of this family.

Theorem 6.2.2. *1. For all $m > 7$ we have always that $CEI_4 > CEI_1$,*

2. For all $m \leq 7$ we have that $CEI_4 > CEI_1$ for $L_{1,4}(\mu_1 m) < \mu_0$,

3. For all $m \geq 2$ we have:

(a) $CEI_4 > CEI_2$ for $L_{2,4}(\mu_1 m) > \mu_0$,

(b) $CEI_4 > CEI_3$ for $L_{3,4}(\mu_1 m) < \mu_0$,

where

$$
\begin{aligned}
L_{1,4}(\mu_1 m) &= -m\mu_1 + \frac{2\ln 2/5m^2 + 3\ln 8/5m + \ln 2/5}{3\ln 5/4}, \\
L_{2,4}(\mu_1 m) &= \frac{2\ln 3/5m}{\ln 5/9}\mu_1 + \frac{2\ln 3/5m^2 + 3\ln 27/25m + \ln 3/5}{3\ln 5/9}, \\
L_{3,4}(\mu_1 m) &= \frac{\ln 16/5m}{\ln 125/16}\mu_1 + \frac{\ln 16/5m^2 + 9\ln 4/25m + \ln 4/3125}{3\ln 125/16}.
\end{aligned}
$$

6.2.6 Numerical results

We analyze four numerical examples corresponding to different situations for the boundary regions $R_{4,j} = 1$ $(j = 1, 2, 3)$, Theorem 6.2.2. The numerical computations were performed on two multi-precision libraries: the MAPLE computer algebra system with `Digits:= 1024`, and the MPFR library of C++ with 4096 digits of mantissa. As shown in Tables 5.1 – 5.2, the computational cost of the quotient with respect to the product is $\ell = 1, 2.5$ respectively (see Table 6.2). Furthermore, in all computations we substituted the computational order of convergence (COC) by the approximation ACOC defined in (3.7b). All computed approximations of ACOC have, at least, three correct decimals. That is, if $\widehat{\rho}_I = \rho \pm \Delta\widehat{\rho}_I$, where ρ is the local order of convergence and $\Delta\widehat{\rho}_I$ is the upper error bound of ACOC at I-step, which is the last step, then we obtain $\Delta\widehat{\rho}_I < 0.5 \cdot 10^{-3}$. This means that in all computations of ACOC we obtain at least four significant digits.

Chapter 7

Numerical strategies

This chapter describes some techniques developed for computer implementation of the algorithms. One goal is to obtain the solution of the nonlinear problem with the maximum number of significant figures and the minimum CPU time, depending on the hardware and software available. The algorithms employed to solve nonlinear equations are usually supported by testing them with numerical experiments. From a theoretical view point, when these experiments are carried out, concepts such as error, computational order of convergence and efficiency must be taken into account. For their practical implementations concepts as multi-precision, stopping criteria and fixed or adaptive arithmetic must be analyzed.

7.1 On admissible points

When we carry out an iterative procedure, we need to decide the length of the mantissa i.e. the arithmetic of the model.

Starting with a given set of initial approximations of the root α, the iteration function $\Phi : D \in \mathbb{R} \longrightarrow D$ of type I , II , III or IV defined by (2.2),(2.3),(2.4) or (2.5), generates the *set of admissible points* of iteration, $\{x_1, x_2, \ldots, x_I\}$.

Definition 12. *A point is admissible if the software arithmetic is sufficient to contain its correct figures.*

According to the above definition, the first non-admissible point x_{I+1} by the arithmetic, check

$$|x_I - \alpha| > 10^{-\eta} \quad \text{and} \quad |x_{I+1} - \alpha| < 10^{-\eta}, \tag{7.1}$$

where η is set according to the order ρ of the iterative method and the number of significant figures. Usually we set $\eta = 1024, 2048, 4096$, that implies $I > 3$ for $\rho > 4$. For our studies, we need to employ multi-precision arithmetic, otherwise the *set of admissible points* will be nearly empty.

The condition (7.1) is called the *stop criterion*. The last admissible point in (7.1) is an approximate value of α and the number of correct decimals of x_I is $D_I = \lceil -\log_{10} |x_I - \alpha| \rceil$ approximately (2.9). These criteria are not feasible when we carry out an iterative procedure, because the root α is unknown.

One computational option is to replace the values of ρ and $|e_I| = |x_I - \alpha|$ with the corresponding parameters introduced in Section (3). The relationships given in (3.11a), (3.11b) and, (3.11c) allow us to substitute $|e_I| = |x_I - \alpha|$ in (7.1) and the resulting expression does not involve the exact root. Recall, the error parameters $\widehat{e}_I = x_I - x_{I-1}$, $\widetilde{e}_I = x_I - \widetilde{\alpha}_I$ and, $\breve{e}_I = \frac{f(x_I)}{f(x_{I-1})}$ defined previously in (3.3), (3.4) and, (3.6). Moreover, we propose the following *stop criteria*, instead of (7.1):

$$\left| \frac{\widehat{e}_I}{\widehat{e}_{I-1}} \right| > 10^{-\kappa_1} \text{ and } \left| \frac{\widehat{e}_{I+1}}{\widehat{e}_I} \right| < 10^{-\kappa_1}, \quad \kappa_1 = \eta\,(\rho - 1)/\rho^2, \tag{7.2a}$$

$$|\,\widetilde{e}_I| > 10^{-\kappa_2} \text{ and } |\,\widetilde{e}_{I+1}| < 10^{-\kappa_2}, \quad \kappa_2 = \eta\,(2\rho - 1)/\rho^2, \tag{7.2b}$$

$$|\breve{e}_I| > 10^{-\kappa_3} \text{ and } |\breve{e}_{I+1}| < 10^{-\kappa_3}, \quad \kappa_3 = \eta\,(\rho - 1)/\rho. \tag{7.2c}$$

For practical implementations, we can approximate the value of the order ρ by one of the previously defined parameters: $\hat{\rho}_I$ or ACOC (3.3); $\tilde{\rho}_I$ or ECOC (3.4) instead of ρ_I or COC (3.2); and $\widehat{\lambda}_I$ or ACLOC (3.7b); $\tilde{\lambda}_I$ or ECLOC (3.7c) and $\breve{\lambda}_I$ or PCLOC (3.7d) instead of λ_I or CLOC (3.7a). Finally, in the numerical results we write $\rho_I^* = \rho \pm \Delta\rho_I^*$ where ρ is the local order of convergence and $\Delta\rho_I^*$ is a higher bound of the error in ACOC, ECOC, ACLOC, ECLOC or PCLOC.

Depending on which stop criterion is fulfilled: (7.2a), (7.2b) or (7.2c), an approximate value of the local order of convergence ρ is determined by only one of the computational values ACOC, ECOC, ACLOC, ECLOC or PCLOC. In all the experimental cases that we have worked on, we also obtained $|\,f(x_I)\,| < 10^{-\eta}$ for the last admissible iterate.

For systems of nonlinear equations, $F(x) = 0$, we have generalized the preceding definitions and techniques taking a norm $\|\cdot\|$ instead of the absolute value in all situations.

With this hypothesis, the last *admissible iteration point*, noted x_I, fulfills:

$$\|x_I - \alpha\| > 10^{-\eta} \text{ and } \|x_I - \alpha\| < 10^{-\eta}, \quad \eta = 1024, 2048, \ldots, \tag{7.3a}$$

$$\frac{\|\widehat{e}_I\|}{\|\widehat{e}_{I-1}\|} > 10^{-\kappa_1} \text{ and } \frac{\|\widehat{e}_I\|}{\|\widehat{e}_{I-1}\|} < 10^{-\kappa_1}, \quad \kappa_1 = \eta\,(\rho-1)/\rho^2, \tag{7.3b}$$

$$\|\widetilde{e}_I\| > 10^{-\kappa_2} \text{ and } \|\widetilde{e}_{I+1}\| < 10^{-\kappa_2}, \quad \kappa_2 = \eta\,(2\rho-1)/\rho^2, \tag{7.3c}$$

$$\|\breve{e}_I\| > 10^{-\kappa_3} \text{ and } \|\breve{e}_{I+1}\| < 10^{-\kappa_3}, \quad \kappa_3 = \eta\,(\rho-1)/\rho, \tag{7.3d}$$

$$\text{where } (\breve{e}_I)_r = \frac{F_r(x_I)}{F_r(x_{I-1})} \text{ for } 1 \le r \le m \text{ and } F = (F_1, F_2, \ldots, F_m).$$

We also call the relations (7.3a), (7.3b), (7.3c) and (7.3d) *stop criteria*. In terms of efficiency, the norm $\|\cdot\|_\infty$ is considered rather than the Euclidean norm $\|\cdot\|_2$.

7.2 Adaptive arithmetic

The requirement of the efficiency of the calculations performed in multi-precision has forced us to develop a new paradigm of computing. This consists of adapting the length of the mantissa to the number of significant figures for each iteration; rather than taking all possible arithmetic. The role of the local order of convergence ρ is the key concept to predicting the accuracy of the next iterate.

Starting with a given set of initial approximations of the root α, the iteration function $\Phi : D \in \mathbb{R}^m \longrightarrow D$ of type I , II , III or IV defined by (2.2),(2.3),(2.4) or (2.5) generates the the sequence of iterates, $\{x_0, x_1, ..., x_n, x_{n+1}, ...\}$ that converges to α when $n \to \infty$.
In each iteration, the number of correct decimals can be approximated by (2.9). We realize from (2.11) that the precision of the next iterate is approximately increased ρ times. That is, you need to increase the mantissa of the next iterate in ρ at least to carry out a next step of the method. Accordingly, an expression of the expected length of the mantissa for step $n+1$, in terms of $e_n = x_n - \alpha$, is

$$\mathcal{L}_{n+1} := \left[\, \rho\left(-\log_{10} \|e_n\| + j \right) \right], \tag{7.4}$$

where ρ is the order of convergence of the iterative method, $[\,x\,]$ denotes the integer part of x and j is a security term. We have numerically checked the value of j, by varying it between 1 and 50, in order to have enough accuracy in the computation of the iterates $\{x_n\}_{n\ge0}$, the errors $\{e_n\}_{n\ge0}$ and the COC $\{\widehat{\rho}_n\}_{n\ge0}$ (see [23]).

When we carry out an iterative procedure and the root is unknown, we need the values of ρ and α to compute $\mathcal{L}_{n+1}$. We replace them according to Section (3). From the relationships given in (3.11a), (3.11b) and (3.11c), we derive the following expressions

$$-\log_{10}\|e_n\| \approx -\frac{\rho^2}{\rho-1}\log_{10}\left(\frac{\|\widehat{e}_n\|}{\|\widehat{e}_{n-1}\|}\right),$$

$$-\log_{10}\|e_n\| \approx -\frac{\rho^2}{2\rho-1}\log_{10}\|\widetilde{e}_n\|,$$

$$-\log_{10}\|e_n\| \approx -\frac{\rho}{\rho-1}\log_{10}\|\breve{e}_n\|.$$

According to these expressions, when the root α is unknown, instead of (7.4), we must use one of the following forecast formulae:

$$\widehat{\mathcal{L}}_{n+1} := \left\lceil \frac{\rho^3}{\rho-1}\left(-\log_{10}\left(\frac{\|\widehat{e}_n\|}{\|\widehat{e}_{n-1}\|}\right)+j\right)\right\rceil, \text{ and } \rho\approx\widehat{\rho}_n \text{ or } \rho\approx\widehat{\lambda}_n, \tag{7.6a}$$

$$\widetilde{\mathcal{L}}_{n+1} := \left\lceil \frac{\rho^3}{2\rho-1}\left(-\log_{10}\|\widetilde{e}_n\|+j\right)\right\rceil, \text{ and } \rho\approx\widetilde{\rho}_n \text{ or } \rho\approx\widetilde{\lambda}_n, \tag{7.6b}$$

$$\breve{\mathcal{L}}_{n+1} := \left\lceil \frac{\rho^2}{\rho-1}\left(-\log_{10}\|\breve{e}_n\|+j\right)\right\rceil, \text{ and } \rho\approx\breve{\rho}_n \text{ or } \rho\approx\breve{\lambda}_n, \tag{7.6c}$$

That is $\widehat{\mathcal{L}}_{n+1}$, $\widetilde{\mathcal{L}}_{n+1}$ and $\breve{\mathcal{L}}_{n+1}$ are the forecast length of the mantissa for the iteration $n+1$. If the forecast is higher than the maximum precision η, the procedure must stop. Then, instead of (7.1), the *stop criteria* should be

$$\mathcal{L}_{n+1}>\eta, \text{ or } \widehat{\mathcal{L}}_{n+1}>\eta, \text{ or } \widetilde{\mathcal{L}}_{n+1}>\eta \text{ or } \breve{\mathcal{L}}_{n+1}>\eta. \tag{7.7}$$

In all cases we choose $n=I$, and the sequence of admissible points is $\{x_1,x_2,\ldots,x_I\}$.

In cases where $\lceil x_n\rceil>0$, we improve the formula for predicting the length of the mantissa. The expression: $\lceil\rho\,(-\log_{10}\|e_n\|+4)+\log_{10}\|x_n\|\rceil$. works appropiately.

7.3 Remarks

Here we highlight some numerical aspects taht we have not considered previously in depth, but which are important in this report.

7.3.1 Remark about j

The term j in (7.4), (7.6a), (7.6b) and (7.6c) is obtained empirically. In order to compare the performance of the formulas under conditions that were as identical as possible, we

used various methods to solve a set of test problems. We started, each problem under the same initial conditions. We have completed a study on this subject (see [23]). Seven systems of nonlinear equations were considered. We computed the solution of each system for the same set of initial approximations and for four iterative methods using the MAPLE computer algebra system. Each one was stopped when $\widehat{\mathcal{L}}_{n+1} > 2800$, or $\widetilde{\mathcal{L}}_{n+1} < 2800$ and we also obtainED $\|F(x_I)\| < 0.5 \cdot 10^{-2800}$. Finally, in (7.4), (7.6a), (7.6b) or (7.6c), we choose j such that $\|e_k\|$, $\|\hat{e}_k\|$ or $\|\tilde{e}_k\|$ respectively, and $\|F(x_k)\|$ had at least three significant digits. We realized that the minimum value that guarantees all the significant digits in x_n required is, in almost all cases, $j = 2$. But we needed to take $j \geq 4$ to guarantee a significant number of digits in COC, ACOC or ECOC, which means $\Delta\rho_n^* \leq 0.5 \cdot 10^{-2}$ at least in the approximations $\rho_n^* = \rho \pm \Delta\rho_n^*$.

7.3.2 Remark on tables of numerical results

Usually, to study and analyze an iterative method, we present a table that displays the method, the number of iterations needed I to reach the maximum precision requested, the computational elapsed time τ in seconds of MAPLE execution for these iterations, the correct decimal reached in $\mathcal{L}_I$ approximately, the computational efficiency index CEI, the time factor $TF = 1/log_{10}CEI$, and an error's higher bound in the approximations $\rho_n^* = \rho \pm \Delta\rho_n^*$. From a theoretical perspective, it is enough to consider CEI and, for practical purposes, τ is sufficient.

7.3.3 Remark on the time factor

In order to compare easily the efficiency of the iterative methods we introduced a new measure, called the time factor (TF), defined as $TF = 1/log_{10}CEI$. Its asymptotic behavior (5.5) does not allow a graphical visualization of the properties of $CEI(\mu, m)$ as a function of m; properties that can be seen in the graphical representation of TF as a function of m. In all my studies, we observed a direct relationship between the elapsed time and the time factor TF. That is, the ordering of execution times τ coincides with the ordering of the $CEI's$. After these studies, this relationship was studied in greater depth by M. Grau-Sánchez and M. Noguera, see [30] for more details.

7.3.4 Remark about the mantissa length

Note that, when we work with adaptive arithmetic the computational cost of each iteration is different, and depends on the length of the mantissa used. This technique drastically reduces the total elapsed time, even though we have to compute the number of correct figures after each iteration. In other words, the cost in adaptive arithmetic is minimum in comparison with keeping an unnecessarily huge long mantissa in the first iterations when fixed arithmetic is applied. The final project, led by M. Grau-Sánchez and performed by the student Marc Fernández, develops a technique to evaluate the computational efficiency of an algorithm using adaptive arithmetic (see [17]).

Chapter 8

Peer-reviewed papers

The articles included in this compendium of publications are:

1 M. Grau-Sánchez, A. Grau, M. Noguera, J.R. Herrero,
On new computational local orders of convergence,
Appl. Math. Lett. **25** (2012) 2023–2030.

2 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández, M. Noguera,
Analyzing the efficiency of some modifications of the Secant method,
Comput. Math. Appl. **64** (2012) 2066–2073.

3 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández,
Construction of derivative-free iterative methods from Chebyshev's method,
Anal. Appl. **11** (2013) 1350009 (16 pp.).

4 J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández,
On the efficiency of two variants of Kurchatov's method for solving nonlinear systems,
Numer. Algor. **64** (2013) 685–698.

5 J.A. Ezquerro, M. Grau-Sánchez, A. Grau; M.A. Hernández-Verón,
A new class of secant-like methods for solving nonlinear systems of equations,
Commun. Appl. Math. Comput. Sci. **9** (2014) 201–213.

Paper 1

This paper was published in Applied Mathematics Letters. The impact factor of the journal in 2012 was 1.501-Q1, according to the classification of the *Journal of Citation Reports (JCR)* [38].

M. Grau-Sánchez, A. Grau, M. Noguera, José. R. Herrero,
On new computational local orders of convergence,
Applied Mathematics Letters **25**, (2012) 2023–2030.

ATTENTION ¡
Pages 60 to 70 of the thesis are available at the editor's web
http://www.sciencedirect.com/science/article/pii/S0893965912002285

Paper 2

This paper was published in Computers and Mathematics with Applications. The impact factor of the journal in 2012 was **2.069**-Q1, according to the JCR classification [38].

J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández, M. Noguera,
Analyzing the efficiency of some modifications of the Secant method,
Computers and Mathematics with Applications **64**, (2012) 2066–2073.

Paper 3

This paper was published in Analysis and Applications. The impact factor of the journal in 2013 was 1.500-Q1, according to the JCR classification [38].

J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández, M. Noguera,
Construction of derivative-free iterative methods from Chebyshev's method,
Analysis and Applications **11** (2013) 1350009 (16 pp.).

Paper 4

This paper was published in Numerical Algorithms. The impact factor of the journal in 2013 was 1.005-Q2, according to the JCR classification [38].

J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández,
On the efficiency of two variants of Kurchatov's method for solving nonlinear systems,
Numerical Algorithms **64** (2013) 685–698.

Paper 5

This paper was published in Communications in Applied Mathematics and Computational Science. The impact factor of the journal in 2013 was 1.167-Q1, according to the JCR classification [38].

J.A. Ezquerro, A. Grau, M. Grau-Sánchez, M.A. Hernández-Verón,
A new class of Secant-like methods for solving nonlinear systems of equations,
Commun. Appl. Math. Comput. Sci. **9** (2014) 201–213.

ATTENTION ¡
Pages 118 to 134 of the thesis are available at the editor's web
http://msp.org/camcos/2014/9-2/camcos-v9-n2-p02-p.pdf

Chapter 9

Other relevant works

The kernel of the work has been to design new iterative methods that are efficient to numerically solve nonlinear equations, to analyze the convergence and the computational efficiency.

The articles included in this chapter are six. The first four were submitted to the CENAI (*Centro Nacional de Asesoramiento e Investigación*) for the *"Convocatoria Sexenios"* to evaluate my research activity. They led to a *"Favorable"* qualification for 2011. The sixth article is my first work on this subject.

6 M. Grau-Sánchez, A. Grau, M. Noguera,
Frozen divided differences scheme for solving systems of nonlinear equations,
J. Comput. Appl. Math. **235** (2011) 1739–1743,
JCR-Science Edition - 2011 - 1.112 - Q2.

7 J.A. Ezquerro, M. Grau-Sánchez, A. Grau, M.A. Hernández, M. Noguera, N. Romero,
On iterative methods with accelerated convergence for solving systems of nonlinear equations, J. Optim. Theory Appl. **151** (2011) 163–174,
JCR-Science Edition - 2011 - 1.062 - Q2.

8 M. Grau-Sánchez, A. Grau, M. Noguera,
On the computational efficiency index and some iterative methods for solving systems of nonlinear equations,
J. Comput. Appl. Math. **236** (2011) 1259–1266,
JCR-Science Edition - 2011 - 1.112 - Q2.

9 M. Grau-Sánchez, A. Grau, M. Noguera,
Ostrowski type methods for solving systems of nonlinear equations,
Appl. Math. Comput. **218** (2011) 2377–2385,
JCR-Science Edition - 2011 - 1.317 - Q1.

10 S. Amat, S. Busquier, M. Grau-Sánchez, A. Grau,
Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications,
Appl. Math. Comput. **219** (2013) 7954–7963,
JCR-Science Edition - 2013 - 1.600 - Q1.

11 M. Grau-Sánchez, A. Grau, J.L. Díaz-Barrero,
On Computational Order of Convergence of some Multi-Precision Solvers of Nonlinear Systems of Equations,
Octogon Mathematical Magazine, **21** (2013) 569–592.

Frozen divided differences scheme for solving systems of nonlinear equations[6]

M. Grau-Sánchez, **A. Grau**, **M. Noguera**

A development of an inverse first-order divided-difference operator for functions of several variables is presented, as well as a direct computation of the local order of convergence of an iterative method. A generalized algorithm of the Secant method for solving a system of nonlinear equations is studied and the maximum computational efficiency is computed. Furthermore, a sequence that approximates the order of convergence is generated for the examples and confirms in a numerical way that the order of the methods is well deduced.

On iterative methods with accelerated convergence for solving systems of nonlinear equations[7]

J. A. Ezquerro, **M. Grau-Sánchez**, **A. Grau**, **M. A. Hernández**, **M. Noguera**, **N. Romero**

We present a modified method for solving nonlinear systems of equations with a higher order of convergence than other competitive methods. We also generalize the efficiency index that is commonly used in the one-dimensional case to several variables. Finally, we show some numerical examples of applications of the theoretical results obtained in this paper.

[6] Journal of Computational and Applied Mathematics **235** (2011) 1739–1743.
[7] Journal of Optimization Theory and Applications **151** (2011) 163–174.

On the computational efficiency index and some iterative methods for solving systems of nonlinear equations[8]

M. Grau-Sánchez, **A. Grau**, **M. Noguera**

In this paper, two new iterative methods are built up and analyzed. We revisit a generalization of the efficiency index used in the scalar case to several variables in iterative methods for solving systems of nonlinear equations. We provide analytical proofs of the local order of convergence, based on developments of multi-linear functions and numerical concepts that will be used to illustrate the analytic results. An approximation of the computational order of convergence is computed independently of knowledge of the root, and the time needed to obtain one correct decimal is studied in our examples.

Ostrowski type methods for solving systems of nonlinear equations[9]

M. Grau-Sánchez, **A. Grau**, **M. Noguera**

Four generalized algorithms built up from Ostrowski's method for solving systems of nonlinear equations are written and analyzed. A development of an inverse first-order divided difference operator for functions of several variables is presented, as well as a direct computation of the local order of convergence for these variants of Ostrowski's method. Furthermore, a sequence that approximates the order of convergence is generated for the examples and confirms in a numerical way that the order of the methods is well deduced.

---

[8] Journal of Computational and Applied Mathematics **236** (2011) 1259–1266.
[9] Applied Mathematics and Computation **218** (2011) 2377–2385.

Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications[10]

S. Amat, **S. Busquier**, **M. Grau-Sánchez**, **A. Grau**

A generalized, k-step iterative application of Newton's method with the frozen derivative is studied and used to solve a system of nonlinear equations. The maximum computational efficiency is computed. A sequence that approximates the order of convergence is generated for the examples, and numerically confirms the calculation of the order of the method and computational efficiency. This type of method appears in many applications where the authors have heuristically chosen a given number of steps with frozen derivatives. An example is shown in which the total variation (TV) minimization model is approximated using the schemes described in this paper.

On Computational Order of Convergence of some Multi-Precision Solvers of Nonlinear Systems of Equations[11]

M. Grau-Sánchez, **A. Grau**, **J.L. Díaz-Barrero**

In this paper the local order of convergence used in iterative methods to solve nonlinear systems of equations is revisited, where shorter alternative analytic proofs of the order based on developments of multilineal functions are shown. Most important, an adaptive multi-precision arithmetics is used hereof, where in each step the length of the mantissa is defined independently of the knowledge of the root. Furthermore, generalizations of the one dimensional case to m-dimensions of three approximations of computational order of convergence are defined. Examples illustrating the previous results are given.

[10] Applied Mathematics and Computation **219** (2013) 7954–7963.
[11] Octogon Mathematical Magazine, **21** (2013) 569–592.

Conclusions

In order to study some iterative methods built up for us, we revisited and analyzed their convergence orders and their computational costs for the one-dimensional case, and worked out the multi-dimensional case. The study introduced new concepts or methods in each of the following steps:

- The computation of the local order of convergence for known two-step and new multi-step iterative methods is performed by means of expansions in formal developments in power series of the functions F, the Jacobian operator F', the inverse Jacobian operator $[F']^{-1}$ and the divided difference operator $[-,-;F]$, its inverse operator $[-,-;F]^{-1}$.
- Some measures that approximate the order of convergence are generated. Four new variants with lower cost than the computational order of convergence (COC) are given: CLOC, ACLOC, ECLOC and PCLOC.
- We worked out a generalization of the efficiency index used in the scalar case to several variables in iterative methods for solving systems of nonlinear equations.
- We express the cost of evaluating the elementary functions in terms of products, which depend on the computer, the software and the arithmetic. The numerical computations listed were performed on an algebraic system called MAPLE.
- A new way to compare elapsed time for different iterative schemes is presented. It consists of computing the time required to obtain one correct decimal of the solution, that is, the ratio between the elapsed time needed to accomplish the stopping criterion and the total number of correct decimals obtained for each method.

On a humble note, my contribution to the area of iterative procedures is a continuous set of small steps that together gave rise to this thesis. I think that interesting results have been achieved altogether. I recall the words of Neil Armstrong (July, 1969):

"That's one small step for man, one giant leap for mankind".

Bibliography

[1] Aitken, A., *On Bernoulli's numerical solution of algebraic equations*, Proc. Roy. Soc. Edinburgh, **46** (1926) 289–305.

[2] Amat, S.; Busquier, S.; Candela, V., *A class of quasi-Newton generalized Steffensen methods on Banach spaces*, J. Comp. Appl. Math. **149** (2002) 397–406.

[3] Amat, S.; Busquier, S.; Grau, A.; Grau-Sánchez, M., *Maximum efficiency for a family of Newton-like methods with frozen derivatives and some applications,* Appl. Math. Comput. **219** (2013) 7954–7963.

[4] `ARPREC` Software, available at `http://crd.lbl.gov/~{}dhbailey/mpdist`.

[5] Babajee, D.K.R., *Analysis of higher order variants of Newton's method and their applications to differential and integral equations in ocean acidification*, Ph. D., University of Mauritius, (2010).

[6] Bailey, D. H.; Hida, Y.; Li, X. S.; Thompson, B., *ARPREC: An arbitrary precision computation package*, Berkeley Lab., Berkeley, CA 94720, (2002). Available at `http://escholarship.org/uc/item/2rt1p2vm`.

[7] Bailey, D. H.; Barrio, R.; Borwein, J. M., *High-Precision Computation: Mathematical Physics and Dynamics*, Appl. Math. and Comp., **218** (2012), 10106–10121.

[8] Brent, R.P.; Zimmermann, P., *Modern Computer Arithmetic*, ArXiv e-prints (2010). Available at `arXiv:1004.4710v1[cs.DS]27Apr2010`

[9] Brent, R.P., *Multiple-Precision Zero-Finding Methods and the Complexity of elementary Function Evaluation*, `arXiv:1004.341v2[cs.NA]30May2010`

[10] Dalhquist, G.; Björch, Å., *Numerical Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[11] de Boor, C.R., *Divided Differences*, Surveys in Approximation Theory **1** (2005) 46-69.

[12] Ezquerro, J.A.; Grau, A.; Grau-Sánchez, M.; Hernández, M.A., *Construction of derivative-free iterative methods from Chebyshev's method,* Anal. Appl. **11** (2013) 1350009 (16 pp.).

[13] Ezquerro, J.A.; Grau, A.; Grau-Sánchez, M.; Hernández, M.A., *On the efficiency of two variants of Kurchatov's method for solving nonlinear systems,* Numer. Algor. **64** (2013) 685–698.

[14] Ezquerro, J.A.; Grau, A.; Grau-Sánchez, M.; Hernández-Verón, M.A., *A new class of secant-like methods for solving nonlinear systems of equations,* Commun. Appl. Math. Comput. Sci. **9** (2014) 201–213.

[15] Ezquerro, J.A.; Grau, A.; Grau-Sánchez, M.; Hernández, M.A.; Noguera. M., *Analysing the efficiency of some modifications of the secant method,* Comput. Math. Appl. **64** (2012) 2066–2073.

[16] Ezquerro, J.A.; Grau-Sánchez, M.; Grau, A.; Hernández, M.A.; Noguera, M.; Romero, N., *On iterative methods with accelerated convergence for solving systems of nonlinear equations,* J. Optim. Theory Appl. **151** (2011) 163–174.

[17] Fernández-Sánchez, M., *Mètodes iteratius per a la resolució d'equacions no lineals*, Master thesis (pre-Bologna period) (2012).
http://upcommons.upc.edu/pfc/handle/2099.1/15804

[18] Fernández-Sánchez, M.; Grau, A.; Grau-Sánchez, M.; Diaz-Barrero, J.L., *Results on multi-precision and adaptive arithmetic for solving nonlinear equations.*
Draft date October 2013.

[19] Fousse, L.; Hanrot, G.; Lefèvre, V.; Pélissier, P.; Zimmermann, P., *MPFR: A multiple-precision binary floating-point library with correct rounding*, ACM Trans. Math. Softw. **33** (2007). Doi:10.1145/1236463.1236468.

[20] Grau-Sánchez, M.; Díaz-Barrero, J.L., *An improvement of the Euler-Chebyshev iterative method*, J. Math. Anal. Appl. **315** (2006) 1–7.

[21] Grau-Sánchez, M.;Díaz-Barrero, J.L., *An improvement to Ostrowski root-finding method*, Appl. Math. Comput. **173** (2006) 450–456.

[22] Grau-Sánchez, M.; Díaz-Barrero, J.L., *A weighted variant family of Newton's method with accelerated third-order convergence*, Appl. Math. Comput. **186** (2007) 1005–1009.

[23] Grau-Sánchez, M.; Grau, A.; Díaz-Barrero, J.L., *On Computational Order of Convergence of some Multi-Precision Solvers of Nonlinear Systems of*

Equations, Octogon Mathematical Magazine, **21** (2013) 569–592.http://www.uni-miskolc.hu/~matsefi/Octogon/index.php?menu=archive.

[24] Grau-Sánchez, M.; Grau, A.; Noguera, M., *Frozen divided differences scheme for solving systems of nonlinear equations,* J. Comput. Appl. Math. **235** (2011) 1739–1743.

[25] Grau-Sánchez, M.; Grau, A.; Noguera, M., *On the computational efficiency index and some iterative methods for solving systems of nonlinear equations,* J. Comput. Appl. Math. **236** (2011) 1259–1266.

[26] Grau-Sánchez, M.; Grau, A.; Noguera, M., *Ostrowski type methods for solving systems of nonlinear equations,* Appl. Math. Comput. **218** (2011) 2377–2385.

[27] Grau-Sánchez, M.; Grau, A.; Noguera, M.; Herrero, J. R., *On new computational local orders of convergence,* Appl. Math. Lett. **25** (2012) 2023–2030.

[28] Grau-Sánchez, M.; Grau, A.; Noguera, M.; Herrero, J. R., *A study on new computational local orders of convergence.* At arXiv:1202.4236v1[math.NA]2012.

[29] Grau-Sánchez, M.; Gutiérrez, J.M., *Zero-finder methods derived from Obreshkov's techniques*, Appl. Math. Comput. (2009) **215** (2009) 2992-3001.

[30] Grau-Sánchez, M.; Noguera, M., *A technique to choose the most efficient method between secant method and some variants*, Appl. Math. Comput. **218** (2012) 6415–6426.

[31] Grau-Sánchez, M.; Noguera, M.; Gutiérrez, J.M., *On some computational orders of convergence*, Appl. Math. Lett. **23** (2010) 472–478.

[32] Goldstine, H.H., *A history of numerical analysis from the 16th through the 19th century*, Springer-Verlag, Heidelberg, 1977.

[33] Homeier, H.H., *A modified Newton method for rootfinding with cubic convergence*, J. Comput. Appl. Math. **157** (2003) 227–230.

[34] Householder, A.S., *Some numerical methods for solving systems of linear equations*, The American Mathematical Monthly **57** (1950) 453-459. Available at http://www.jstor.org/stable/2308297.

[35] Householder, A.S., *Principles of Numerical Analysis*, McGraw-Hill, New York, 1953.

[36] Hueso, J.L.; Martínez, E.; Torregrosa, J.R., *Third and fourth order iterative methods free from second derivative for nonlinear systems*, Appl. Math. Comput. **211** (2009) 190–197.

[37] Jarratt, P., *A review of methods for solving nonlinear algebraic equations in one variable*, Numerical Methods for Nonlinear Algebraic Equations, Ph. Rabinowitz (ed.) Gordon and Breach, London, New York, Paris, (1970) 1–26.

[38] JCR, Journal of Citation Reports of Web of Knowledge, Thomson Reuters 2013. http://go.thomsonreuters.com/jcr.

[39] Kantorovich, L.V., *Functional Analysis and Applied Mathematics*, Uspekhi Mat. Nauk **3** (1948) 89-185 (Russian). Tr. by C.D. Benster, Natl. Bur. Std. Rept. No. 1509, Washington, 1952.

[40] King, R.F., *A fifth order family of modified Newton's methods*, BIT Numerical Mathematics **11** (1971) 409–412.

[41] Kline, M., *Mathematical though from ancient to modern times*, Oxford University Press, New York, 1972.

[42] Kou, J.; Li, Y.; Wang, X., *Some variants of Ostrowski's method with seventh-order convergence*, J. Comput. Appl. Math. **209** (2007) 153–159.

[43] `MPFR` Software, available at http://www.mpfr.org.

[44] `MPFR++` Software, at http://perso.ens-lyon.fr/nathalie.revol/software.html.

[45] Neta, B., *A sixth-order family of methods for nonlinear equations*, Int. J. Comput. Math. **7** (1979) 157–161.

[46] Neta, B., *A New Iterative Method for the Solution of Systems of Nonlinear Equations*, Proc. Conf. Approximation Theory and Applications (Z. Ziegler, ed.), Academic Press (1981) 249-263.

[47] Neta, B., *On a family of multipoint methods for nonlinear equations* Int. J. Comput. Math. **9** (1981) 353–361.

[48] Ortega, J.M.; Rheinboldt, W.C., *Iterative solution of nonlinear equations in several variables* Academic Press, New York, 1970.

[49] Ostrowski, A.M., *Solutions of Equations and System of Equations*, Academic Press, New York, 1960.

[50] Özban, A.Y., *Some new variants of Newton's method*, Appl. Math. Lett. **17** (2004) 677–682.

[51] Papakonstantinou, J.M., *A Historical Development of the (n+1)-point Secant Method*, Ph. D., Rice University, Houston, 2006 available at http://hdl.handle.net/1911/20568

[52] Petković, M.S., *Remarks on "On a general class of multipoint root-finding methods of high computational efficiency"*, SIAM J. Numer. Anal. **49** (2011) 1317–1319.

[53] Petković, L.D.; Petković, M.S., *A note on some recent methods for solving nonlinear equations*, Appl. Math. Comp. **185** (2007) 368–374.

[54] Plofker, K., *An Example of the Secant Method of Iterative Approximation in a Fifteenth-Century Sanskrit Text*, Historia Mathematica **23** (1996) 246–2256.

[55] Potra, F.A.; Ptak, V., *Nondiscrete Induction and Iterative Processes*, Research Notes Math. **103**, Pitman Advanced Publishing Program, Boston (1984).

[56] Schröder, E., *Über unendlich viele Algorithmen zur Auflösung der Gleichungen*, Math. Ann. **2** (1870) 317–365. Tr. by G.W. Stewart, On Infinitely Many Algorithms for Solving Equations, (1998). Available at http://drum.lib.umd.edu/handle/1903/577.

[57] Traub, J.F., *Iterative methods for the solution of equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

[58] The MPFR library 2.1.0: timings. http://www.mpfr.org/mpfr-2.1.0/timings.html.

[59] The MPFR library 3.0.0: timings. http://www.mpfr.org/mpfr-3.0.0/timings.html.

[60] Tornheim, L., *Convergence of multipoint iterative methods*, J. ACM **11** (1964) 210–220.

[61] Wall, D.D., *The order of an iteration formula*, Math. Tables Aids Comput. **10** (1956) 167–168.

[62] Weerakoon, S.; Fernando, T.G.I., *A variant of Newton's method with accelerated third-order convergence*, Appl. Math. Lett. **13** (2000) 87–93.

[63] Yamamoto, T., *Historical developments in convergence analysis for Newton's and Newton-like methods* J. Comput. Appl. Math. **124** (2000) 1–23.

[64] T.J. Ypma, T.J., *Historical Development of the Newton-Raphson Method*, SIAM Review **37** (1995) 531–551.