Journal of Computational and Applied Mathematics 236 (2011) 1259–1266

Contents lists available at SciVerse ScienceDirect

Journal of Computational and Applied Mathematics

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

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

Miquel Grau-Sánchez *, Àngela Grau, Miquel Noguera

Technical University of Catalonia, Department of Applied Mathematics II, Campus Nord, Jordi Girona 1-3, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 9 December 2010

MSC:
65H10
65Y20
41A58

Keywords:
Order of convergence
System of nonlinear equations
Iterative methods
Computational efficiency index
Computational order of convergence

ABSTRACT

In this paper two new iterative methods are built up and analyzed. A generalization of the efficiency index used in the scalar case to several variables in iterative methods for solving systems of nonlinear equations is revisited. Analytic proofs of the local order of convergence based on developments of multilineal functions and numerical concepts that will be used to illustrate the analytic results are given. An approximation of the computational order of convergence is computed independently of the knowledge of the root and the necessary time to get one correct decimal is studied in our examples.

1. Introduction

There is no doubt that Newton's method is one of the best root-finding methods for solving nonlinear equations, $F(x) = 0$. Recent results improving the classical formula at the expense of an additional evaluation of the function, an additional evaluation of the first derivative or a change in the point of evaluation can be found in the literature on the subject (see [1–4] and references therein). In those works the order of convergence and the efficiency index in the neighborhood of a simple root have been improved. In this paper 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, F' and $(F')^{-1}$.

The concept of the computational efficiency index [1,5] is revisited introducing a necessary parameter in order to take into account the computational cost of all the computations needed and reduced to multiplication units. The preceding technique to prove the local order of convergence and the computational efficiency index are illustrated with several examples in which generalizations of the one-dimensional case to m-dimensions are carried out.

Because of the increase in the number of applications where it is required to use a higher level of numeric precision [6], numerical experiments presented with multi-precision arithmetics facilities are most appropriate in a modern large-scale scientific computing environment. An approximation of the computational order of convergence (ACOC) is obtained when the root is unknown. These results confirm theoretical results obtained previously.

2. Notation and basic results

Let $F : D \subseteq \mathbf{R^m} \longrightarrow \mathbf{R^m}$ be Fréchet-differentiable in D, and with its differential continuous. For any $x_n \in \mathbf{R^m}$ lying in a neighborhood of a simple zero, $\alpha \in \mathbf{R^m}$, of the system $F(x) = 0$ we can apply Taylor's formula and assuming that there

* Corresponding author.
E-mail addresses: miquel.grau@upc.edu (M. Grau-Sánchez), angela.grau@upc.edu (À. Grau), miquel.noguera@upc.edu (M. Noguera).

0377-0427/$ – see front matter
doi:10.1016/j.cam.2011.08.008

exists $\Gamma = F'(\alpha)^{-1}$, we have

$$F(x_n) = F(\alpha + e_n) = F'(\alpha)\left[e_n + \sum_{k=2}^{3} A_k e_n^k + O_4\right], \tag{1}$$

where e_n is the local error $e_n = x_n - \alpha$, e_n^p denotes $(e_n, \overset{p}{\cdots}, e_n)$, $A_k = \frac{1}{k!}\Gamma F^{(k)}(\alpha),\ k \geq 2,$ and $O_{p+1} = O(e_n^{p+1})$. Note that $\Gamma \in \mathscr{L}(\mathbf{R}^m)$ and $F^{(k)}(\alpha) \in \mathscr{L}_k(\mathbf{R}^m, \mathbf{R}^m)$. Moreover, we can express the differential of first order as:

$$F'(x_n) = F'(\alpha)\left[I + \sum_{k=2}^{3} k A_k e_n^{k-1} + O_3\right], \tag{2}$$

where I is the identity. From (2), after developing in a formal series expansion we get

$$F'(x_n)^{-1} = \left[I - 2A_2 e_n + (4A_2^2 - 3A_3)e_n^2 + O_3\right]\Gamma. \tag{3}$$

We begin the definitions of the iterative methods in this paper with Newton's method that we can write as

$$x_{n+1}^{(1)} = \Phi_1(x_n) = x_n - F'(x_n)^{-1}F(x_n). \tag{4}$$

The expression of the error $E_1 = x_{n+1}^{(1)} - \alpha$ in terms of e_n is built up subtracting α from both sides of (4) and taking into account (1) and (3). Namely,

$$\begin{aligned} E_1 &= e_n - \left[I - 2A_2 e_n + (4A_2^2 - 3A_3)e_n^2 + O_3\right]\left[e_n + \sum_{k=2}^{3} A_k e_n^k + O_4\right] \\ &= A_2 e_n^2 + 2(A_3 - A_2^2)e_n^3 + O_4, \end{aligned} \tag{5}$$

where we use the following notation: $A_2^\ell e_n^{\ell+1} = (A_2 e_n)^{\ell-1}\left(A_2 e_n^2\right)$. The result (5) agrees with the classical asymptotical constant in the one-dimensional case.

Without using norms we can define the local order of convergence for an iterative method without memory as follows. The local order of convergence of an iterative method is at least $p \in \mathbf{N}$ if and only if there exists a p-linear function $L \in \mathscr{L}_p(\mathbf{R}^m, \mathbf{R}^m)$ such that

$$x_{n+1} - \alpha = L e_n^p + O_{p+1}. \tag{6}$$

Notice that if we apply norms to both sides of (6) then we obtain that there exists $\tilde{L}$ such that $\|e_{n+1}\| \leq \|\tilde{L}\|\,\|e_n\|^p$.

3. Variants of Newton's method

In this section, using the results presented in the previous one, a known variant of Newton's method with local order of convergence equal to three [1,7] is analyzed. We explicitly give its vectorial error equation in which appears a 3-linear application instead of the asymptotical error constant used in the one-dimensional case. Furthermore, another two new iterative methods of local order 4 and 5 respectively are presented and they are analyzed applying the same analysis.

The first variant that substitutes the derivative of $F(x_n)$ by the harmonic mean of the derivatives of F at the points x_n and $x_{n+1}^{(1)}$ is

$$x_{n+1}^{(2)} = \Phi_2(x_n, x_{n+1}^{(1)}) = x_n - \frac{1}{2}\left[F'(x_n)^{-1} + F'(x_{n+1}^{(1)})^{-1}\right]F(x_n), \tag{7}$$

that we call the harmonic mean Newton's method. From (3) we obtain

$$F'(x_{n+1}^{(1)})^{-1} = [I - 2A_2E_1 + (4A_2^2 - 3A_3)E_1^2 + O(E_1^3)]\Gamma.$$

Moreover, from the preceding expression, (5) and the developments of $F'(x_n)$, $F'(x_{n+1}^{(1)})$, if we define $E_2 = x_{n+1}^{(2)} - \alpha$ then yields

$$\begin{aligned} E_2 &= e_n - \frac{1}{2}\left[2I - 2A_2(e_n + E_1) + \left(4A_2^2 - 3A_3\right)e_n^2 + O_3\right][e_n + A_2 e_n^2 + A_3 e_n^3 + O_4] \\ &= \frac{1}{2}A_3 e_n^3 + O_4. \end{aligned}$$

Note that we get the same expression as the one obtained for the one-dimensional case in [1,7].

Generalizing a one-dimensional iterative method derived from [1], the third scheme presented in this section is

$$x_{n+1}^{(3)} = \Phi_3(x_n, x_{n+1}^{(1)}) = x_{n+1}^{(1)} - 2F'(x_n)^{-1}F(z_n), \tag{8}$$

where $z_n = \alpha + \varepsilon_n = x_{n+1}^{(1)} - \frac{1}{2}F'(x_n)^{-1}F(x_{n+1}^{(1)})$ and we call it Traub's method. The error results in

$$E_3 = x_{n+1}^{(3)} - \alpha = E_1 - 2F'(x_n)^{-1}F(z_n), \tag{9}$$

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

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

and

$$\varepsilon_n = z_n - \alpha = E_1 - \frac{1}{2}\left[I - 2A_2e_n + (4A_2^2 - 3A_3)e_n^2 + O_3\right]\left[E_1 + 2A_2E_1^2 + O(E_1^3)\right]$$
$$= \frac{1}{2}A_2e_n^2 + A_3e_n^3 + \frac{1}{2}\left(3A_4 - 5A_2^3\right)e_n^4 + O_5. \tag{10}$$

Finally, from (9) and (10), we obtain $E_3 = \frac{9}{2}A_2^3e_n^4 + O_5$.

A new three-step iterative method that is a modification of the harmonic mean method (7) is presented. It can be written as

$$x_{n+1}^{(4)} = \Phi_4(x_n, x_{n+1}^{(1)}, x_{n+1}^{(2)}) = x_{n+1}^{(2)} - F'(x_{n+1}^{(1)})^{-1}F(x_{n+1}^{(2)}). \tag{11}$$

The vectorial error equation is

$$E_4 = x_{n+1}^{(4)} - \alpha = E_2 - \left[I - 2A_2E_1 + O(E_1^2)\right][E_2 + O(E_2^2)]$$
$$= 2A_2E_1E_2 + O(E_2^2) = 8A_2\left(A_2e_n^2\right)\left(A_3e_n^3\right) + O_6.$$

We can sum up this section with the following theorem.

Theorem 3.1. *The iterative methods Φ_3 and Φ_4, defined in (8) and (11) respectively, have local order of convergence at least 4 and 5 and their vectorial error difference equations can be written as*

$$E_3 = e_{n+1}^{(3)} = \frac{9}{2}A_2^3e_n^4 + O_5,$$
$$E_4 = e_{n+1}^{(4)} = 8A_2\left(A_2e_n^2\right)\left(A_3e_n^3\right) + O_6.$$

4. On the computational efficiency index

In this section, the traditional way to present the computational efficiency index of iterative methods (see [1,4,5]) is revisited and adapted for systems of nonlinear equations. It will be presented in general form for previous four methods, and some comparisons are given.

When dealing with a system of nonlinear equations, the total operational cost per iteration is the sum of the evaluations of functions (the function and the derivatives involved) and the operational cost of doing a step of the iterative method. Therefore, for systems with m nonlinear equations and m unknowns, we suggest the following definition of the computational efficiency index (CEI) of an iterative method of order of convergence ρ

$$\mathrm{CEI}(\mu_0, \mu_1, m) = \rho^{1/\mathcal{C}(\mu_0,\mu_1,m)}, \tag{12}$$

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

$$\mathcal{C}(\mu_0, \mu_1, m) = \mu_0 a0m + \mu_1 a1m^2 + P(m). \tag{13}$$

In (13), $a0$ and $a1$ represent the number of evaluations of the scalar functions of $F(x)$ and $F'(x)$ respectively, $P(m)$ 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. Notice that for $\mu_0 = \mu_1 = 1$ and $P(m) = 0$, (12) is reduced to the classic efficiency index of an iterative method, $EI = \rho^{1/\nu}$, where ν represents the number of evaluations of the scalar component functions necessary to apply a step of an iterative method.

According to that, an estimation of the factors $\mu_{0,1}$ is claimed. To do this, we express the cost of the evaluation of the elementary functions in terms of products, which depends on the computer, the software and the arithmetics used. In [8,9] comparison studies between a multi-precision library, MPFR, and other computing libraries can be found. In Tables 1–2 an (own) estimation of the cost of the elementary functions in product units is shown, where the running time of one product is measured in milliseconds.

In Table 1 the values presented have been rounded to 5 unities because of the huge variability obtained in the different repetitions carried out. On the contrary, in Table 2 averages are shown, since variability has been very low and, besides, the compilator of C++ used secures that the function `clock()` gives exactly the CPU time invested by the program. Table 2 shows that some relative values with respect to the product are lower in multiple precision than in double precision, although the absolute time spent for a product is much higher in multiple precision.

Table 2
Computational cost of elementary functions computed with a program written 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 on an Intel®Xeon E5420, 2.5 GHz, 6 MB cache, where $x = \sqrt{3} - 1$ and $y = \sqrt{5}$.

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

Table 3
Coefficients used in (13) and (14), local order of convergence and computational cost of the iterative methods Φ_ℓ, $1 \le \ell \le 4$.

Method	$a0$	$a1$	$p0$	$p1$	$p2$	ρ	$\mathcal{C}(\mu_0, \mu_1, m)$
Φ_1	1	1	0	1	0	2	$m(2m^2+3(2\mu_1+k+1)m+6\mu_0+3k-5)/6$
Φ_2	1	2	1	2	0	3	$m(2m^2+3(2\mu_1+k+1)m+3\mu_0+3k-2)/3$
Φ_3	3	1	2	1	2	4	$m(2m^2+3(2\mu_1+k+5)m+18\mu_0+15k-5)/6$
Φ_4	2	2	1	2	1	5	$m(2m^2+3(2\mu_1+k+2)m+6\mu_0+6k-5)/3$

In the first iterative method Φ_1, that is Newton's method, instead of computing the inverse operator we solve a linear system, where we have $m(m-1)(2m-1)/6$ products and $m(m-1)/2$ quotients in the decomposition LU and $m(m-1)$ products and m quotients in the resolution of two triangular linear systems. If we suppose that a quotient is equivalent to k products, then

$$P(m) = \frac{m(m-1)(2m+5)}{6} + k\frac{m(m+1)}{2} = \frac{m(2m^2+3(k+1)m+3k-5)}{6}.$$

In general, we denote by $p0$ the number of scalar products per iteration, and $p1$ the number of complete resolutions of a linear system (LU decomposition and resolution of two triangular systems). We call $p2$ the number of resolutions of two triangular systems when LU decomposition is computed in another step in the same iteration. Observe that $p1 = a1$. The total number of products is

$$P(m) = \frac{m(2p1m^2 + (3p1(k+1) + 6p2)m + 6p0 + p1(3k-5) + 6p2(k-1))}{6}. \tag{14}$$

Table 3 shows, for each iterative method analyzed in this paper, Φ_1–Φ_4, the number of evaluations of the scalar functions of $F(x)$, $a0$, the number of evaluations of the scalar functions of $F'(x)$, $a1$, the number of products per scalar of each iteration, $p0$, the number of resolutions of linear systems per iteration, $p1$, and the number of resolutions of two triangular systems, $p2$. Moreover, in Table 3 the local order of convergence, ρ, and the computational cost, $\mathcal{C}(\mu_0, \mu_1, m)$, are presented.

4.1. Comparison between the methods presented

If we denote the computational efficiency indices of Φ_i by $\mathrm{CEI}_i(\mu_0, \mu_1, m)$ then from (12) we define the ratio

$$R_{i,j} = \frac{\log \mathrm{CEI}_i(\mu_0, \mu_1, m)}{\log \mathrm{CEI}_j(\mu_0, \mu_1, m)} = \frac{\log(\rho_i)\mathcal{C}_j(\mu_0, \mu_1, m)}{\log(\rho_j)\mathcal{C}_i(\mu_0, \mu_1, m)}, \tag{15}$$

where

$$\begin{aligned}\mathcal{C}_\ell(\mu_0, \mu_1, m) &= 2p1_\ell m^2 + (6\mu_1 a1_\ell + 3p1_\ell(1+k) + p2_\ell)m \\ &\quad + (6\mu_0 a0_\ell + 6p0_\ell + p1_\ell(3k-5) + 6p2_\ell(k-1)), \quad \ell = i, j.\end{aligned} \tag{16}$$

For $R_{i,j} > 1$ the iterative method Φ_i is more efficient than Φ_j. From the preceding equation (15), (16) and taking into account that the border between two computational efficiencies is given by $R_{i,j} = 1$, this boundary can be expressed by the equation of a quadric that is written as

$$\mu_0 = am\mu_1 + bm^2 + cm + d, \tag{17}$$

where

$$\begin{aligned}
a &= -\frac{\log(\rho_i)a1_j - \log(\rho_j)a1_i}{\log(\rho_i)a0_j - \log(\rho_j)a0_i}, \\
b &= -\frac{1}{3}\frac{\log(\rho_i)p1_j - \log(\rho_j)p1_i}{\log(\rho_i)a0_j - \log(\rho_j)a0_i}, \\
c &= -\frac{1}{2}\frac{\log(\rho_i)(2p2_j + p1_j(k+1)) - \log(\rho_j)(2p2_i + p1_i(k+1))}{\log(\rho_i)a0_j - \log(\rho_j)a0_i}, \\
d &= -\frac{1}{6}\left[\frac{\log(\rho_i)(6p2_j(k-1) + p1_j(3k-5) + 6p0_j)}{\log(\rho_i)a0_j - \log(\rho_j)a0_i} - \frac{\log(\rho_j)(6p2_i(k-1) + p1_i(3k-5) + 6p0_i)}{\log(\rho_i)a0_j - \log(\rho_j)a0_i}\right],
\end{aligned}$$

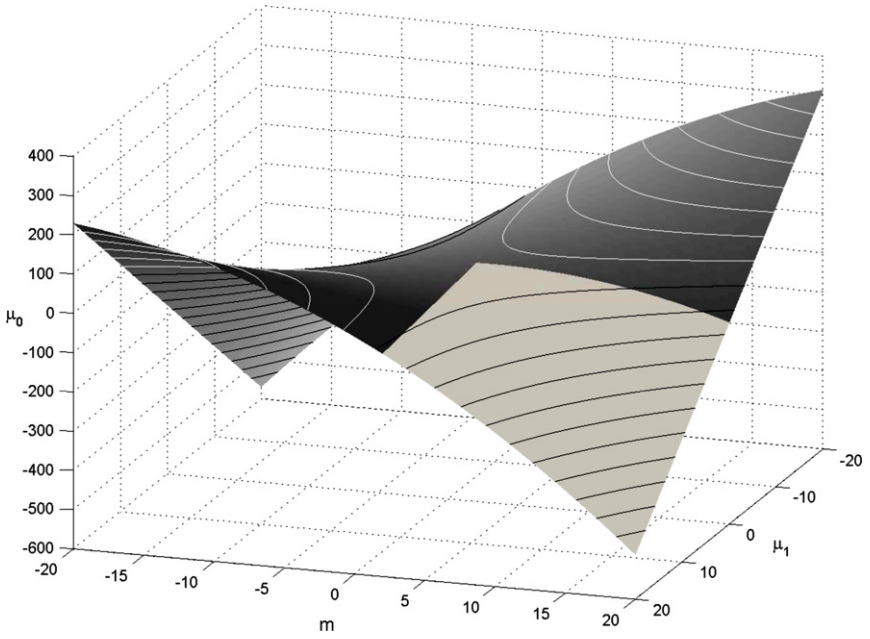

Fig. 1. Boundary surface of the best computational efficiency index, $R_{1,4} = 1$ (15) where $k = 1$.

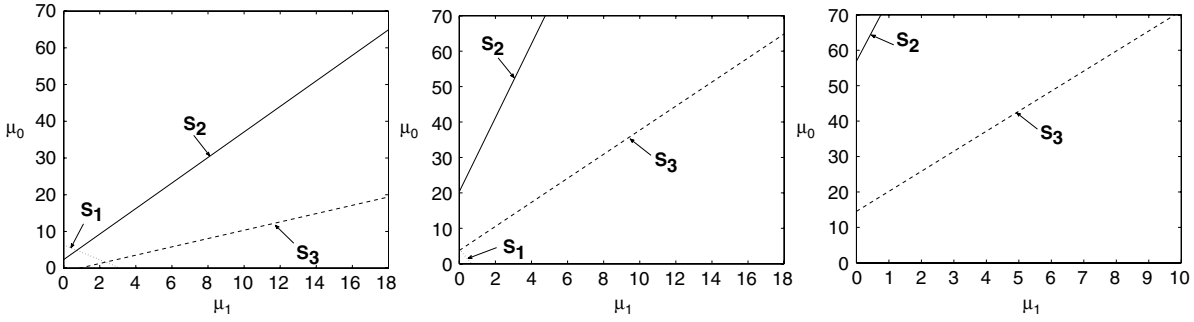

Fig. 2. The boundary straight lines in (μ_1, μ_0)-plain for $m = 2, 6, 10$ respectively, for $k = 1$.

and $\log(\rho_i)a0_j - \log(\rho_j)a0_i \neq 0$. Note that a and b do not depend on k. Recalling that $p1 = a1$, the eigenvalues of (17) are

$$\lambda_{1,2} = \frac{1}{2}\left[b \pm \sqrt{a^2 + b^2}\right] = \frac{a}{6}\left(1 \pm \sqrt{10}\right).$$

It is clear that λ_1 is a positive real number and λ_2 is always a negative real number. Consequently, the surface given in (17) is a hyperbolical paraboloid. If the variables are ordered in the following way (μ_1, m), then the corresponding eigenvectors of $\lambda_{1,2}$ are

$$v_{1,2} = \left(\frac{a}{2\lambda_{1,2}}, 1\right) = \left(-\frac{1}{3}\left(1 \mp \sqrt{10}\right), 1\right),$$

respectively. Finally, the vertex of the boundary surface is

$$m = 0, \qquad \mu_0 = d, \qquad \mu_1 = -\frac{c}{a},$$

that is a function of k. In Fig. 1 the boundary surface between the computational efficiency of the iterative methods Φ_1 and Φ_4 is shown. If we restrict the definition of the variables to real domain, that is, $\mu_0 > 0$, $\mu_1 > 0$ and $m \geq 2$, we get the light gray portion of the surface in Fig. 1.

In order to compare the computational efficiency index of the iterative method Φ_4, that's to CEI_4, with the CEI of the other three methods, in Fig. 2 in the (μ_1, μ_0)-plain for $m = 2, 6, 10$ respectively, for $k = 1$, we present the boundary between CEI_4 and CEI_1, $R_{1,4} = 1$, in a dotted line (S_1), the boundary $R_{2,4} = 1$ in a solid line (S_2) and the boundary $R_{3,4} = 1$ in a dashed line (S_3).

Table 4
Numerical results for the system $F_1(x, y, z) = 0$.

	MAPLE $(\mu_0, \mu_1) = (1.7, 0.7)$				MPFR $(\mu_0, \mu_1) = (1.7, 0.7)$		
	I	CEI	τ	$\hat{\rho}_I \pm \Delta\hat{\rho}_I$	I	CEI	τ
Φ_1	10	1.02506	0.0922	$2 \pm 7.961 \cdot 10^{-5}$	12	1.01891	0.0269
Φ_4	4	1.02395	0.1665	$5 \pm 2.434 \cdot 10^{-2}$	5	1.01794	0.0699

S_2 divides the maximum efficiency regions between Φ_4 and Φ_2, being $\mathrm{CEI}_4 > \mathrm{CEI}_2$ below S_2. In the same way S_3 divides the maximum efficiency regions Φ_4 and Φ_3, being $\mathrm{CEI}_4 > \mathrm{CEI}_3$ above S_3. Lastly, we have $\mathrm{CEI}_4 > \mathrm{CEI}_1$ above S_1. Note that S_1 disappears when $m > 7$, leaving only three regions.

We summarize the results concerning these computational efficiency indexes in the following theorem.

Theorem 4.1.

1. *For all* $m \geq 2$ *we have:*
   (a) $\mathrm{CEI}_4 > \mathrm{CEI}_2$ *for* $L_{2,4}(\mu_1, m) > \mu_0$,
   (b) $\mathrm{CEI}_4 > \mathrm{CEI}_3$ *for* $L_{3,4}(\mu_1, m) < \mu_0$.
2. *For all* $m \leq 7$ *we have that* $\mathrm{CEI}_4 > \mathrm{CEI}_1$ *for* $L_{1,4}(\mu_1, m) < \mu_0$.
3. *For all* $m > 7$ *we have always that* $\mathrm{CEI}_4 > \mathrm{CEI}_1$,

where

$$L_{1,4}(\mu_1, m) = -m\mu_1 + \frac{2\ln 2/5m^2 + 3\ln 8/5m + \ln 2/5}{3\ln 5/4},$$

$$L_{2,4}(\mu_1, m) = \frac{2\ln 3/5m\mu_1}{\ln 5/9} + \frac{2\ln 3/5m^2 + 3\ln 27/25m + \ln 3/5}{3\ln 5/9},$$

$$L_{3,4}(\mu_1, m) = \frac{\ln 16/5m\mu_1}{\ln 125/16} + \frac{\ln 16/5m^2 + 9\ln 4/25m + \ln 4/3125}{3\ln 125/16}.$$

5. Numerical results

The numerical computations listed in Tables 4–7 were performed on two different multi-precision arithmetics: the MAPLE computer algebra system where `Digits := 1024` and the MPFR library of C++ with 4096 digits of mantissa. As it is shown in Tables 1–2 the computational cost of the quotient respect to the product is $k = 1, 2.5$ respectively (see Table 3).

The classical stopping criterium $\|e_I\| = \|x_I - \alpha\| < 0.5 \cdot 10^{-\varepsilon}$, with $\varepsilon = 800$ for MAPLE and $\varepsilon = 4000$ for MPFR, is replaced by

$$E_I = \frac{\|\hat{e}_I\|}{\|\hat{e}_{I-1}\|} < 10^{-\eta}, \tag{18}$$

where $\hat{e}_I = x_I - x_{I-1}$ and $\eta = \frac{\rho-1}{\rho^2}\varepsilon$, since from $\|e_{I+1}\| \approx C\|e_I\|^\rho + \cdots$, we have $\|e_I\| \approx E_I^{\rho^2/(\rho-1)}$ (see [10]). Notice that criterium (18) is independent of the knowledge of the root. Furthermore, in all computations we have substituted the computational order of convergence (COC) [4] by an approximation (ACOC) denoted by $\hat{\rho}_I$ [10] and defined as follows

$$\hat{\rho}_I = \frac{\ln E_I}{\ln E_{I-1}}.$$

According to the definition of the computational cost (13), an estimation of the factors μ_0 and μ_1 is claimed. To do this, we express the cost of the evaluation of the elementary functions in terms of products [8,9], which depends on the machine, the software and the arithmetics used. In Tables 1–2 an estimation of the cost of the elementary functions in product units is shown, where the running time of one product is measured in milliseconds.

We present four numerical examples corresponding to different situations for the regions shown in Fig. 2. Tables 4–7 show the results obtained comparing the efficiency of the pairs (Φ_1, Φ_4), (Φ_2, Φ_4), (Φ_3, Φ_4) and Φ_4 with all remainder methods. In each table the number of necessary iterations I, the computational efficiency index CEI, the necessary computational time to get one correct decimal of the solution τ, in MAPLE and MPFR are presented. Finally, ACOC and a higher bound of ACOC is also shown.

Case 1. We begin with the function defined by

$$F_1(x, y, z) = \begin{cases} x^2 + y^2 + z^2 - 9, \\ xyz - 1, \\ x + y - z^2. \end{cases}$$

Table 5
Numerical results for $F_2(x, y) = 0$.

	MAPLE $(\mu_0, \mu_1) = (11.5, 1.0)$				MPFR $(\mu_0, \mu_1) = (67.5, 1.8)$		
	I	CEI	τ	$\hat{\rho}_I \pm \Delta\hat{\rho}_I$	I	CEI	τ
Φ_2	6	1.02471	0.3024	$3 \pm 2.625 \cdot 10^{-4}$	7	1.00641	0.2888
Φ_4	4	1.02260	0.5504	$5 \pm 6.370 \cdot 10^{-2}$	5	1.00514	0.3131

Table 6
Numerical results for the system defined by Hammerstein's integral equation.

	MAPLE $(\mu_0, \mu_1) = (11, 1.3)$				MPFR $(\mu_0, \mu_1) = (12.5, 4.6)$		
	I	CEI	τ	$\hat{\rho}_I \pm \Delta\hat{\rho}_I$	I	CEI	τ
Φ_3	4	1.001927	0.7919	$4 \pm 1.376 \cdot 10^{-4}$	5	1.001321	0.1935
Φ_4	3	1.001847	1.1574	$5 \pm 4.046 \cdot 10^{-3}$	4	1.001112	0.2952

Table 7
Numerical results for the trigonometrical system (19).

	MAPLE $(\mu_0, \mu_1) = (90.0, 22.5)$				MPFR $(\mu_0, \mu_1) = (113.0, 29.0)$		
	I	CEI	τ	$\hat{\rho}_I \pm \Delta\hat{\rho}_I$	I	CEI	τ
Φ_1	9	1.000917	0.6222	$2 \pm 1.300 \cdot 10^{-71}$	11	1.000459	1.1849
Φ_2	6	1.000951	0.8232	$3 \pm 5.203 \cdot 10^{-19}$	7	1.000467	2.3516
Φ_3	4	1.000915	1.1920	$4 \pm 5.753 \cdot 10^{-6}$	5	1.000467	1.3811
Φ_4	4	1.001052	0.5984	$5 \pm 3.9175 \cdot 10^{-6}$	5	1.000524	0.8539

We test the convergence of the methods Φ_1 and Φ_4 towards the root $\alpha \approx (2.4914, 0.2427, 1.6535)^t$, where the initial approximation is $x_0 = (3.0, 1.0, 2.0)^t$. Note that in Table 4, scheme Φ_1 spends the minimum time in getting one correct decimal in the computation of the numerical solution. This result is in correspondence with the theoretical definition of CEI and confirms the relating regions of Fig. 2.

Case 2. Here we solve the system of nonlinear equations $F_2(x, y) = 0$, where

$$F_2(x, y) = \begin{cases} \ln(y) - x^2 + xy, \\ \ln(x) - y^2 + xy. \end{cases}$$

We test the convergence of the methods towards the root $\alpha = (1.01.0)^t$, where the initial approximation is $x_0 = (0.5, 1.5)^t$. The results are shown in Table 5 and confirm the behavior of related regions of Fig. 2. Moreover, the iterative method Φ_2 presents the maximum value of CEI and minimum value of τ for the two arithmetics used.

Case 3. The example that we consider is the following mixed Hammerstein's integral equation [11]:

$$x(s) = 1 + \frac{1}{5}\int_0^1 G(s, t)x(t)^3 \mathrm{d}t, \quad s \in [0, 1],$$

where $x \in C[0, 1]$, $t \in [0, 1]$, and the kernel G is $G(s, t) = \begin{cases} (1-s)t, & t \le s, \\ s(1-t), & s \le t. \end{cases}$

Firstly, we write this integral equation as $F(x) = 0$, where $F : C[0, 1] \to C[0, 1]$ and

$$F(x)(s) = x(s) - 1 - \frac{1}{5}\int_0^1 G(s, t)x(t)^3 \mathrm{d}t, \quad s \in [0, 1].$$

Secondly, it is discretized to transform it into a finite dimensional problem. To do this, we use the Gauss–Legendre formula to approximate an integral

$$\int_0^1 f(t)\mathrm{d}t \approx \sum_{j=1}^{m} \varpi_j f(t_j),$$

where the abscissas t_j and the weights ϖ_j are determined for $m = 8$. Denoting the approximation of $x(t_i)$ by x_i ($i = 1, 2, \ldots, 8$) we obtain the system of nonlinear equations

$$x_i = 1 + \frac{1}{5}\sum_{j=1}^{8} a_{ij}x_j^3, \quad \text{where } a_{ij} = \begin{cases} \varpi_j t_j(1 - t_i) & \text{if } j \le i, \\ \varpi_j t_i(1 - t_j) & \text{if } j < i, \end{cases} \quad i = 1, 2, \ldots, 8.$$

Now, the previous nonlinear system is written in the matrix form by

$$F(\overline{x}) = \overline{x} - \overline{1} - \frac{1}{5}A\hat{x},$$

where $F : \mathbf{R}^8 \longrightarrow \mathbf{R}^8$, $\bar{x} = (x_1, x_2, \ldots, x_8)^t$, $\bar{1} = (1, 1, \ldots, 1)^t$, $A = (a_{ij})$ and $\hat{x} = (x_1^3, x_2^3, \ldots, x_8^3)^t$. We test the convergence of the methods towards the root $\alpha = (x_1^*, x_2^*, \ldots, x_8^*)^t$, where

$$x_1^* = x_8^* = 1.002096\ldots, \qquad x_2^* = x_7^* = 1.009900\ldots, \qquad x_3^* = x_6^* = 1.019727\ldots, \qquad x_4^* = x_5^* = 1.026436\ldots.$$

The results are shown in Table 6 with the initial point $\bar{x}_0 = \bar{1}$. The chosen example gives a set of values of m, μ_0 and μ_1 corresponding to a better value of CEI_3 than CEI_4 and consequently a better value of time τ.

Case 4. In this example we present a system of nonlinear equations where the parameters of the efficiency are introduced in this paper, say CEI and τ, are better for the iterative method Φ_4, and the corresponding region in Fig. 2 lies between straight lines S_2 and S_3. Setting the following trigonometric equations

$$x_i - \cos\left(2x_i - \sum_{j=1}^{4} x_j\right) = 0, \quad 1 \le i \le 4, \tag{19}$$

we test the convergence of the methods Φ_1, Φ_2, Φ_3 and Φ_4 respectively, towards the root $\alpha \approx (0.514933, 0.514933, 0.514933, 0.514933)^t$. Taking $x_0 = (1.0, 1.0, 1.0, 1.0)^t$, we get the results shown in Table 7 that confirm the theoretical results presented above.

6. Concluding remarks

A generalization to several variables of a technique used to compute analytically the error equation of iterative methods for one variable is presented. The key idea is to use formal power series. Two modified methods for solving systems of nonlinear equations whose order of convergence are higher than that of other well-known competitive methods are analyzed.

The main result of this paper is the presentation of a generalization of the efficiency index used in the scalar case to several variables. We have analyzed and compared the computational efficiency index of two well known methods, Newton's method and the harmonic mean method, with two new methods called Traub's method and the modified harmonic mean method. In order to analyze this parameter for the two methods we have studied the problem from two points of view: numerical and geometrical.

Numerical examples that illustrate the theoretical results presented in this paper are also given. An approximation of the root with high precision is obtained if we use multi-precision arithmetics. In this paper we have used two different softwares: MAPLE 13 and MPFR library in C++. To illustrate the technique presented four examples have been studied and completely solved. In each one, the fourth iterative method has been carried out and compared with the other three.

An approximation of the computational order of convergence has been computed independently of the knowledge of the root. Moreover, a new way to compare execution time is presented. Namely, we have computed the necessary time to get one correct decimal of the solution which is the ratio between the necessary execution time to accomplish the stopping criterium and the total number of correct decimals obtained.

Acknowledgment

This work has been supported by the project MTM2011-28636-C02-01 of the Spanish Ministry of Science and Innovation.

References

[1] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
[2] M. Grau, J.L. Díaz-Barrero, An improvement of the Euler–Chebyshev iterative method, J. Math. Anal. Appl. 315 (2006) 1–7.
[3] M. Grau-Sánchez, Improvement of the efficiency of some three-step iterative like-Newton methods, Numer. Math. 107 (2007) 131–146.
[4] S. Weerakoon, T.G.I. Fernando, A variant of Newton's method with accelerated third-order convergence, Appl. Math. Lett. 13 (2000) 87–93.
[5] A.M. Ostrowski, Solutions of Equations and System of Equations, Academic Press, New York, 1960.
[6] D.H. Bailey, J.M. Borwein, High-precision computation and mathematical physics, in: XII Advanced Computing and Analysis Techniques in Physics Research, 2008, http://crd.lbl.gov/~dhbailey/dhbpapers/dhb-jmb-acat08.pdf.
[7] A.Y. Özban, Some new variants of Newton's method, Appl. Math. Lett. 17 (2004) 677–682.
[8] L. Fousse, G. Hanrot, V. Lefèvre, P. Pélissier, P. Zimmermann, MPFR: a multiple-precision binary floating-point library with correct rounding, ACM Trans. Math. Software 33 (2) (2007) Art. 13 (15 pp).
[9] http://www.mpfr.org/mpfr-2.1.0/timings.html.
[10] M. Grau-Sánchez, M. Noguera, J.M. Gutiérrez, On some computational orders of convergence, Appl. Math. Lett. 23 (2010) 472–478.
[11] A.D. Polyanin, A.V. Manzhirov, Handbook of Integral Equations, CRC Press, Boca Ratón, 1998.