Real effectiveness of iterative method for solving nonlinear equations

G Calderón$^{1,3}$, J Villamizar M$^{2,3}$ and J C Carrillo E$^3$

$^1$Departamento de Matemáticas, Universidad de Los Andes, Mérida, Venezuela
$^2$Facultad de Ingeniería, Universidad de Los Andes, Mérida, Venezuela
$^3$Escuela de Matemáticas, Universidad Industrial de Santander, Bucaramanga, Colombia

E-mail: gcalderon@matematicas.uis.edu.co

Abstract. In this paper is defined a strategy to analyze the practical effectiveness of given iterative methods for solving nonlinear equations; which considers the CPU time used for convergence to the root with a given accuracy. Through this time the difference of two populations mean between each pair of these methods is estimated. From this study the practical effectiveness of each method and its computational cost is obtained.

1. Introduction

One of the most important problems in Numerical analysis is to solve nonlinear equations, which is usually done by standard iterative methods such as Newton, secant and bisection. Among the measures of efficiency of these methods are the order of convergence and the efficiency ratio, the second being a measure of a balance between the order of convergence of the iterative method and the number of functions that are evaluated in each iteration (see references given in this paper and the citations given therein).

However, either of these two classic ways of measuring the efficiency of an iterative method is biased. Indeed, the fact that certain iterative methods have a higher order of convergence does not necessarily imply that it converges in less time under an accepted tolerance.

In a similar way it happens to the efficiency ratio, because the time of evaluation of a function or its derivatives are often different, especially for non-linear functions. Because of this, in recent years there have been developed many numerical techniques which modifications of these methods are, among which is the predictor-corrector methods type, where the predictor is usually the method of Newton.

Although in terms of this efficiency is important to improve the order of convergence, which achieve the predictor-corrector methods, does not mean that they be in practice, where it is considered that an efficient iterative method will be which use the least computer time and has the largest radius of convergence for a particular type of problem.

In order to analyze the practical implementation of the mentioned iterative methods, this paper considers the CPU time used by each method to achieve convergence to the root with a given accuracy. That is, for each method, a sample mean of the time used is defined to solve the problem on the basis of non-linear functions (population sample) and a finite number of initial conditions, taken over some interval containing the root of such functions is defined. To validate these sample means and differentiate between each pair of these methods are considered confidence intervals [1,2]. From this study we conclude on the applicability of each method and its computational cost.
Due to the large number of new methods discussed in the literature (see references), to validate the proposed classification to define some methods that come from changes to the method of Newton and that only require evaluating the first derivative are selected. In particular, those methods with a higher order of convergence and increased efficiency index than the method of Newton are selected.

The rest of the article is structured as follows. The second section briefly describes the iterative methods to be classified, and their order of convergence and its efficiency ratio, leaving the details to the reference given for each method. A proposed classification of methods is defined in the third section; then in the fourth section the numerical experimentation and the classification obtained are presented. Finally, the results are analyzed together with references work.

2. Iterative methods

Let \( f: D \subseteq \mathbb{R} \to \mathbb{R} \) be a scalar function and \( D \) an open interval. The problem of solving the nonlinear equation \( f(\alpha) = 0 \) is considered, which is assumed to have a single root \( \alpha \) in \( D \), that is, \( f(\alpha) = 0 \) and \( f'(\alpha) \neq 0 \). Methods for the case of multiple roots will not be considered.

For any iterative method to solve this nonlinear equation, its efficiency index is defined as \( I_e = r/d \) where \( r \) is the order of convergence's method and \( d \) is the number of iterations required functions evaluated by the method [3]. For example, as the order of convergence of the Newton method is quadratic then its efficiency index equal to \( 2^{1/2} \approx 1.414 \).

Iterative methods to be classified in this work are the predictor-corrector type, where for all of them the iterative method of Newton acts as the predictor (Equation 1).

\[
y_n = x_n - \frac{f(x_n)}{f'(x_n)}, \quad (1)
\]

which converges quadratically in some neighborhood of \( \alpha \) [3]. In this kind of methods, the correction terms are defined as follows.

Corrector 1 (Changbum [4], 2008):

\[
x_{n+1} = x_n - \frac{f^2(x_n)}{f'(x_n)[f(x_n)-f(y_n)]} \quad (M_1)
\]

Corrector 2 (Zhongyong et al. [5], 2011):

\[
x_{n+1} = y_n - \left[1 + \left(\frac{f(y_n)}{f(x_n)}\right)^2\right] \frac{f(y_n)}{f'(y_n)} \quad (M_2)
\]

Corrector 3 (Zhongyong et al. [5], 2011):

\[
\begin{align*}
z_n &= y_n - \left[1 + \left(\frac{f(y_n)}{f(x_n)}\right)^2\right] \frac{f(y_n)}{f'(y_n)}, \\
x_{n+1} &= z_n - \left[1 + 2 \left(\frac{f(y_n)}{f(x_n)}\right)^2 + 2 \left(\frac{f(y_n)}{f'(y_n)}\right)\right] \frac{f(x_n)}{f'(y_n)}. \quad (M_3)
\end{align*}
\]

Corrector 4 (Kou et al. [6], 2007):

\[
x_{n+1} = x_n - \frac{f^2(x_n)+f^2(y_n)}{f'(x_n)[f(x_n)-f(y_n)]} \quad (M_4)
\]

Corrector 5 (Kou et al. [6], 2007):
\[ x_{n+1} = x_n + \frac{f(x_n)+f(y_n)}{f'(x_n)} - 2 \frac{f^2(x_n)}{f'(x_n)[f(x_n)-f(y_n)]} \] (M_5)

Corrector 6 (Traub [7], 1977):
\[ x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} \] (M_6)

Corrector 7 (Kou and Wang [8], 2010):
\[ H_2(x_n, y_n) = \frac{f(y_n)}{f(x_n)-2f(y_n)}, \]
\[ z_n = y_n - H_2(x_n, y_n)(x_n - y_n), \]
\[ H_3(x_n, y_n) = \frac{f(x_n)}{f'(x_n)-3f(x_n)}, \] (M_7)
\[ x_{n+1} = z_n - [(1 + H_2(x_n, y_n))^2 +(1 + 4H_2(x_n, y_n))H_3(x_n, y_n)] \frac{f(x_n)}{f'(x_n)}. \]

Corrector 8 (Changbum and Yoon Mee [9], 2008):
\[ x_{n+1} = y_n - \frac{2f(x_n)-f(y_n)}{f'(x_n)} \frac{f(y_n)}{f'(x_n)} \] (M_8)

Corrector 9 (Liping and Xia [10], 2010):
\[ z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{4f(x_n)-f(y_n)}{4f(x_n)-9f(y_n)}, \]
\[ x_{n+1} = z_n - \frac{f(x_n)}{f'(x_n)} \frac{8f(y_n)}{4f(x_n)-11f(y_n)} + \frac{f(x_n)}{f'(x_n)+f(x_n)} + 1 + \frac{4f(x_n)}{f'(x_n)+f(x_n)} \] (M_9)

Corrector 10 (Sharma and Guha [11], 2007):
\[ z_n = y_n - \frac{f(x_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n)-2f(y_n)}, \] (M_10)
\[ x_{n+1} = z_n - \frac{f(x_n)}{f'(x_n)} \frac{f(x_n)+f(y_n)}{f(x_n)-f(y_n)}. \]

Corrector 11 (Changbum and Beny [12], 2012):
\[ z_n = y_n - \frac{f(y_n)}{f'(x_n)} \frac{1}{(1-f(y_n)/f(x_n))^2}, \] (M_11)
\[ x_{n+1} = z_n - \frac{f(x_n)}{f'(x_n)} \frac{1}{[1-f(y_n)/f(x_n)]^2}. \]

Table 1 shows the classification of these iterative methods according to their convergence, and efficiency index. In particular, as a position exchange of these methods is observed when classified according to their efficiency index (see last column of the table), the question is whether such classification changes or another arises when is considered the CPU time used by each one of these same methods. Following are provided answers to this question.
Table 1. Iterative methods classification according to its convergence order \( r \). The terms \( Nf \) and \( Nf' \) represent the number of evaluations of the functions \( f \) and \( f' \) respectively, in each iteration.

| Method | \( r \) | \( Nf \) | \( Nf' \) | \( I_{ef} = r^{1/d} \) |
|--------|--------|--------|--------|------------------|
| \( M_1 \) | 9 | 3 | 2 | 1.552 |
| \( M_7 \) y \( M_8 \) | 8 | 3 | 1 | 1.682 |
| \( M_{10} \) y \( M_{11} \) | 6 | 3 | 1 | 1.565 |
| \( M_2 \) | 5 | 2 | 2 | 1.495 |
| \( M_4 , M_5 \) y \( M_8 \) | 4 | 2 | 1 | 1.587 |
| \( M_6 \) | 4 | 2 | 2 | 1.414 |
| \( M_1 \) | 3 | 2 | 1 | 1.442 |
| Newton | 2 | 1 | 1 | 1.414 |

3. Classification strategy

Suppose we have a set of \( n \) functions \( f_1, ..., f_n \), where each function \( f_j : [a_j, b_j] \subset \mathbb{R} \rightarrow \mathbb{R} \) has a single root \( \alpha_j \) on the interval \( (a_j, b_j) \). It is further assumed that the \( m \) given iterative methods \( M_1, ..., M_m \) are going to be sorted as follows. It is assumed the method \( M_i (i = 1, ..., m) \) estimates the root \( \alpha_j \) of the function \( f_j (j = 1, ..., n) \) from the initial data \( \hat{x}_{jk} \) \((k = 1, ..., l)\) and at a time \( t_{ij} \). Under these conditions it is obtained that the estimates of the roots of the functions \( f_1, ..., f_n \) are made on times \( t_{1jk}, ..., t_{njk} \), respectively, and its corresponding average time is \( T_{ik} \), with \( k \) fixed. Table 2 meets these time averages.

Table 2. For a fixed \( k \) \((k = 1, ..., l)\), the method \( M_i \) \((i = 1, ..., m)\), with initial data \( \hat{x}_{jk} \), estimates the root of function \( f_j \) on the interval \([a_j, b_j]\) \((j = 1, ..., n)\) at time \( t_{ijk} \).

| Method | Function | Average times |
|--------|----------|--------------|
| \( M_1 \) | \( t_{11}^k \) \( t_{12}^k \) \( ... \) \( t_{1n}^k \) | \( T_{1k} = \frac{1}{n} \sum_{j=1}^{n} t_{1j}^k \) |
| \( M_2 \) | \( t_{21}^k \) \( t_{22}^k \) \( ... \) \( t_{2n}^k \) | \( T_{2k} = \frac{1}{n} \sum_{j=1}^{n} t_{2j}^k \) |
| \( M_m \) | \( t_{m1}^k \) \( t_{m2}^k \) \( ... \) \( t_{mn}^k \) | \( T_{mk} = \frac{1}{n} \sum_{j=1}^{n} t_{mj}^k \) |

In this way, for all initial data or node \( \hat{x}_{jk} \) in \([a_j, b_j]\) the average times \( T_{1k}, ..., T_{mk} \) \((k = 1, ..., l)\) of the respective methods \( M_1, ..., M_m \) are obtained (see Table 3). From these times the methods \( M_1, ..., M_m \) are classified.

Table 3. Scheme to obtain the average time of each method for a total of \( k \) nodes on each interval.

| Method | Node | Average times |
|--------|------|--------------|
| \( M_1 \) | \( \hat{x}_1 \) \( \hat{x}_2 \) \( ... \) \( \hat{x}_l \) |
| \( M_2 \) | \( T_{11} \) \( T_{12} \) \( ... \) \( T_{1l} \) |
| \( M_2 \) | \( T_{21} \) \( T_{22} \) \( ... \) \( T_{2l} \) |
| \( M_m \) | \( T_{m1} \) \( T_{m2} \) \( ... \) \( T_{ml} \) |
3.1. Statistical test

Each of the methods $M_i$ has a real media $\mu_i$, which comes from averaging all CPU times used by each method, which arise when using each point of the interval $[a_i, b_i]$ (infinite points), the domain of the function $f_i$. As these times cannot be obtained, the times $t_k^i$ for $k$ points $[a_i, b_i]$ (finite sample) are calculated. From that times a sample mean $\bar{\mu}_i$ is obtained, which is an estimate of $\mu_i$.

Because an estimate of the true mean of each method is obtained, the next consequences follow. First, it can happen that two $M_{i_1}$ and $M_{i_2}$ methods have sample means $\bar{\mu}_{i_1}$ and $\bar{\mu}_{i_2}$ very close, but whose sample means $\mu_{i_1}$ and $\mu_{i_2}$ are estimates of a same real average $\mu_i$ of methods $M_{i_1}$ and $M_{i_2}$, that is, the real average of $M_{i_1}$ and $M_{i_2}$ are such that $\mu_{i_1} = \mu_{i_2} = \mu_i$. It may also happen that the sample means $\bar{\mu}_{i_1}$ and $\bar{\mu}_{i_2}$ are very close but they are estimates of different averages. This uncertainty generated by the estimate these real averages is solved by confidence intervals' theory.

The confidence intervals estimate consists of determining a couple of $LI$ and $LS$ values which set up the interval $[LI, LS]$, in such a way that for a given probability of $1 - \rho$ (called confidence level), the condition of Equation 2.

$$P(LI \leq \mu \leq LS) = 1 - \rho$$

is fulfilled in relation to the estimate of the true mean $\mu$. This condition ensures with this probability, the true average is contained in the interval $[LI, LS]$, which is built [1]. For this reason, this interval is called the confidence interval.

In the case of one true mean, the confidence interval of error produces just error limits on the average. The values in the interval must be seen as fair values given the experimental data. In the case of a difference between two true means, the interpretation can be extended to a comparison of the two means. For example, if one has a great confidence that a difference $\mu_{i_1} - \mu_{i_2}$ is positive, actually it follows that $\mu_{i_1} > \mu_{i_2}$ with little risk of falling into an error.

In our case, it will be built intervals with a confidence level of 99% for the difference of two upcoming mean to each other, $\mu_{i+1} - \mu_i$ $(i = 1, \ldots, m - 1)$, such that the Equation 3.

$$LI < \mu_{i+1} - \mu_i < LS$$

(3)

Where $LI$ and $LS$ are presented by Equations 4 and 5, respectively.

$$LI = (\bar{\mu}_{i+1} - \bar{\mu}_i) - 2.58 \sqrt{\frac{s_{i+1}^2}{n_{i+1}} + \frac{s_i^2}{n_i}}$$

(4)

$$LS = (\bar{\mu}_{i+1} - \bar{\mu}_i) + 2.58 \sqrt{\frac{s_{i+1}^2}{n_{i+1}} + \frac{s_i^2}{n_i}}$$

(5)

and $s_{i+1}$ and $s_i$ are the standard deviations in each of the samples of sizes $n_{i+1}$ and $n_i$, respectively [8]. In our case, $n_i = n_{i+1}$ for all $M_i$ method.

4. Numerical experiments

The condition of Equation 6, For the numerical experiments stopping criterion.

$$|f(x_n)| \leq \varepsilon \quad \text{or} \quad |x_{n+1} - x_n| \leq \varepsilon,$$

(6)

is implemented with a tolerance $\varepsilon = 10^{-8}$. It is considered that all methods converge with one of these stopping criteria and not the maximum number of iterations.

To implement the strategy of classification $M_i$ iterative methods considered in the second section, it is used a base of $n = 50$ functions. The closed interval where the root exists of each of these functions
is defined in such a way that all iterative methods converge to this root. In each of these intervals is considered a partition of \( I = 1000 \) nodes and is obtained for each method \( M_i \) a sample of \( I \) values of computation times. From the sample of each method the mean and standard deviation are calculated and a table is built in ascending from the means of samples of each method (see Table 4).

In Table 4, columns 6 and 7 show the lower limit LI and the upper limit LS, which are needed to construct confidence intervals for each of the differences of two consecutive mean (see column 8 in Table 4). For example, the third row shows Equation 7.

\[
0.6180 < \mu_4 - \mu_3 < 0.1088,
\]

which means one has a confidence level of 99% that the difference of the actual means of the methods \( M_4 \) and \( M_3 \) is a point in the interval (0.6180,0.1088). Also, one can ensure with the same level of confidence that \( \mu_4 > \mu_3 \). At the same time, one cannot make a difference between real average \( \mu_3 \) and \( \mu_4 \) (see row 4 in Table 4), i.e, it cannot be concluded that \( \mu_3 < \mu_4 \) or \( \mu_4 < \mu_3 \) by the fact that \( \mu_3 - \mu_4 \) belongs to the interval \((-0.2256,0.2451)\). Similar conclusions can be drawn for the rest of the differences of two means shown in Table 4.

| Method | \( \bar{\mu} \times 10^{-3} \)seg | \( DE \times 10^{-3} \)seg | Order | \( I_{ef} \) | LI \( \times 10^{-4} \) | LS \( \times 10^{-4} \) | \( \mu_{i} - \mu_{j} \) |
|--------|---------------------------------|-----------------------------|-------|------------|-----------------|-----------------|-----------------|
| \( M_2 \) | 0.5785                          | 0.1733                      | 8     | 1.6818     | \( \mu_1 - \mu_7 \) |
| \( M_4 \) | 0.6423                          | 0.2204                      | 3     | 1.4422     | 0.4093          | 0.8667          | \( \mu_3 - \mu_1 \) |
| \( M_5 \) | 0.7275                          | 0.1852                      | 4     | 1.5874     | 0.6180          | 0.1088          | \( \mu_4 - \mu_6 \) |
| \( M_9 \) | 0.7285                          | 0.2211                      | 9     | 1.5518     | -0.2256         | 0.2451          | \( \mu_3 - \mu_4 \) |
| \( M_0 \) | 0.8073                          | 0.2032                      | 8     | 1.6818     | 0.5428          | 0.1033          | \( \mu_9 - \mu_3 \) |
| \( M_6 \) | 0.8766                          | 0.1955                      | 4     | 1.4142     | 0.4633          | 0.9234          | \( \mu_6 - \mu_4 \) |
| \( M_{11} \) | 0.8890                          | 0.2353                      | 5     | 1.4953     | -0.1259         | 0.3733          | \( \mu_2 - \mu_6 \) |
| \( M_8 \) | 0.8981                          | 0.1803                      | 8     | 1.5874     | -0.1508         | 0.3303          | \( \mu_5 - \mu_2 \) |
| Newton | 0.9342                          | 0.1768                      | 2     | 1.4142     | 0.1553          | 0.5674          | \( \mu_N - \mu_5 \) |
| \( M_6 \) | 1.4082                          | 0.2139                      | 4     | 1.5874     | 4.5132          | 4.9660          | \( \mu_8 - \mu_6 \) |
| \( M_{10} \) | 1.6673                          | 0.2332                      | 6     | 1.5661     | 2.3330          | 2.8493          | \( \mu_{10} - \mu_8 \) |
| \( M_{11} \) | 72.320                          | 3.2099                      | 6     | 1.5661     | 703.90          | 709.16          | \( \mu_{11} - \mu_{10} \) |

It is important to note that the classification of methods based on the estimated average varies significantly to the one given in Table 1. For example, \( M_4 \) method is of higher order and with an efficiency index \( I_{ef} \) greater than the \( M_1 \) method. However, the CPU time for convergence of \( M_4 \) method is greater than the time of \( M_1 \) method. This phenomenon is repeated in many cases, showing that many of the newly introduced methods, even if they have high convergence orders or greater efficiency index, are not necessarily effective in practice.

5. Concluding remarks

Due to the constant appearance in the literature of new iterative methods with both higher convergence order and efficiency index, it is important to classify them according to a real effectiveness, CPU time. Because of this there has given a simple strategy on how to achieve this classification. The results show that there are methods with a high order of convergence or a great index of efficiency, or both inclusive, which need more CPU time than other methods with lower order of convergence. Furthermore, outside the framework of the given classification, the numerical experimentation allows to conclude that the interval of convergence of the methods that have higher order convergence is much lower than the methods with a lower order of convergence or index of efficiency, for example, the Newton method. Thus, while a method with a high-order of convergence achieves a good classification according to the
mean, it loses interest because of its need for an initial approximation extremely close to the root; this was the case with the of $M_7$ and $M_9$ methods.

Although the classification was made for iterative methods applied to scalar functions with simple root, and from the method of Newton, the same can be applied to other methods and other configurations of the problem.

Finally, it would be vital that when introducing a new method, and its efficiency according to the order of convergence or efficiency ratio is promoted, is considered the classification given in this paper. This would help relieve some of the large number of new iterative methods are presented in the literature and to have a more realistic classification of them.

Acknowledgements
J. Villamizar M. acknowledges financial support from UIS, under Doctoral Academic Commission. G. Calderón y J.C. Carrillo E. gratefully acknowledges financial support provided by VIE-UIS, under Grant No 2415.

References
[1] Ronal W, Rymond H, and Sharon M 1999 Probabilidad y estadística para ingenieros (México: Prentice-Hall S.A.)
[2] Juan Carlos Correa y Esperanza Sierra 2003 Intervalos de confianza para la comparación de dos proporciones. Revista Colombiana de Estadística 26 (1) 61-75
[3] Ostrowski A 1973 Solution of Equations in Euclidean and Banach Space (New York, Academic Press)
[4] Changbum C 2008 Some fourth-order iterative methods for solving nonlinear equations Applied Mathematics and Computation 195(1) 454-459
[5] Zhongyong H, Liu G, and Li T 2011 An iterative method with ninth-order convergence for solving nonlinear equations International Journal of Contemporary Mathematical Sciences 6(1) 17-23
[6] Kou J, Li Y, and Wang X 2007 A composite fourth-order iterative method for solving non-linear equations Applied Mathematics and Computation 184(2) 471-475
[7] Traub J 1977 Iterative Methods for the solution of equations (New York: Chelsea Publishing Company)
[8] Kou J and Wang X 2010 Some improvements of Ostrowski’s method Applied Mathematics Letters 23(1) 92-96
[9] Changbum C and Yoon Mee H 2008 Some fourth-order modifications of newton's method Applied Mathematics and Computation 197(1) 654-658
[10] Liping L and Xia W Eighth-order methods with high efficiency index for solving nonlinear equations Applied Mathematics and Computation 215(1) 3449-3454
[11] Sharma J and Guha R A 2007 family of modified Ostrowski methods with accelerated sixth order convergence Applied Mathematics and Computation 190(1) 111-115
[12] Changbum C and Beny N 2012 A new sixth-order scheme for nonlinear equations Applied Mathematics Letters 25(1) 185-189