Research Article

Three New Optimal Fourth-Order Iterative Methods to Solve Nonlinear Equations

Gustavo Fernández-Torres¹ and Juan Vásquez-Aquino²

¹ Petroleum Engineering Department, UNISTMO, 70760 Tehuantepec, OAX, Mexico
² Applied Mathematics Department, UNISTMO, 70760 Tehuantepec, OAX, Mexico

Correspondence should be addressed to Gustavo Fernández-Torres; gusfer@sandunga.unistmo.edu.mx

Received 21 November 2012; Revised 24 January 2013; Accepted 2 February 2013

1. Introduction

One of the most important problems in numerical analysis is solving nonlinear equations. To solve these equations, we can use iterative methods such as Newton's method and its variants. Newton's classical method for a single nonlinear equation \( f(x) = 0 \), where \( r \) is a single root, is written as

\[
x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},
\]

which converges quadratically in some neighborhood of \( r \).

Taking \( y_n = x_n - f(x_n)/f'(x_n) \), many modifications of Newton's method were recently published. In [1], Noor and Khan presented a fourth-order optimal method as defined by

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

which uses three functional evaluations.

In [2], Cordero et al. proposed a fourth-order optimal method as defined by

\[
z_n = x_n - \frac{f(x_n) + f(y_n)}{f'(x_n)},
\]

\[
w_n = z_n - \frac{f^2(y_n) (2 f(x_n) + f(y_n))}{f^2(x_n) f'(x_n)},
\]

which also uses three functional evaluations.

Chun presented a third-order iterative formula [3] as defined by

\[
x_{n+1} = x_n - \frac{3 f(x_n) f'(x_n) + f(x_n) f'(\phi(x_n))}{2 f'(x_n) + \frac{1}{2} f''(x_n)},
\]

which uses three functional evaluations, where \( \phi \) is any iterative function of second order.
Li et al. presented a fifth-order iterative formula in [4] as defined by
\[ u_{n+1} = x_n - \frac{f(x_n) - f(x_n - f(x_n)/f'(x_n))}{f'(x_n)}, \]
(5)
\[ x_{n+1} = u_{n+1} - \frac{f(u_{n+1})}{f'(x_n - f(x_n)/f'(x_n))}, \]
(6)
which uses five functional evaluations.

The main goal and motivation in the development of new methods are to obtain a better computational efficiency. In other words, it is advantageous to obtain the highest possible convergence order with a fixed number of functional evaluations per iteration. In the case of multipoint methods without memory, this demand is closely connected with the optimal order considered in the Kung-Traub conjecture.

**Kung-Traub’s Conjecture** (see [5]). Multipoint iterative methods (without memory) requiring \( n + 1 \) functional evaluations per iteration have the order of convergence at most \( 2^n \).

Multipoint methods which satisfy Kung-Traub’s conjecture (still unproved) are usually called optimal methods; consequently, \( p = 2^n \) is the optimal order.

The computational efficiency of an iterative method of order \( p \), requiring \( n \) function evaluations per iteration, is most frequently calculated by Ostrowski-Traub’s efficiency index [6] \( E = p^{1/n} \).

On the case of multiple roots, the quadratically convergent modified Newton’s method [7] is
\[ x_{n+1} = x_n - \frac{mf(x_n)}{f'(x_n)}, \]
(7)
where \( m \) is the multiplicity of the root.

For this case, there are several methods recently presented to approximate the root of the function. For example, the cubically convergent Halley’s method [8] is a special case of the Hansen-Patrick’s method [9]
\[ x_{n+1} = x_n - \frac{f(x_n)}{(m + 1)/2m f'(x_n) - f(x_n) f''(x_n)/2f'(x_n)}. \]
(8)

Osada [10] has developed a third-order method using the second derivative:
\[ x_{n+1} = x_n - \frac{1}{2} m (m + 1) u_n + \frac{1}{2} (m - 1) \frac{f'(x_n)}{f''(x_n)}, \]
(9)
where \( u_n = f(x_n)/f'(x_n) \).

Another third-order method [11] based on King’s fifth-order method (for simple roots) [12] is the Euler-Chebyshev’s method of order three
\[ x_{n+1} = x_n - \frac{m(3 - m)}{2} \frac{f(x_n)}{f'(x_n)} + \frac{m^2 f^2(x_n) f''(x_n)}{2f'(x_n)} + \frac{m^3 f^3(x_n)}{f'(x_n)}. \]
(10)

Recently, Chun and Neta [13] have developed a third-order method using the second derivative:
\[ x_{n+1} = x_n - \left(2m^2 f^2(x_n) f''(x_n)ight) \times \left(m(3 - m) f(x_n) f'(x_n) f''(x_n)ight) + \frac{m^2 f^2(x_n) f''(x_n)}{f'(x_n)} \]
(11)

All previous methods use the second derivative of the function to obtain a greater order of convergence. The objective of the new method is to avoid the use of the second derivative.

The new methods are based on a mixture of Lagrange’s and Hermite’s interpolations. That is to say not only Hermite’s interpolation. This is the novelty of the new methods. The interpolation process is a conventional tool for iterative methods; see [5, 7]. However, this tool has been applied recently in several ways. For example, in [14], Cordero and Torregrosa presented a family of Steffensen-type methods of fourth-order convergence for solving nonlinear smooth equations by using a linear combination of divided differences to achieve a better approximation to the derivative. Zheng et al. [15] proposed a general family of Steffensen-type methods with optimal order of convergence by using Newton’s iteration for the direct Newtonian interpolation. In [16], Petković et al. investigated a general way to construct multipoint methods for solving nonlinear equations by using inverse interpolation. In [17], Đumić et al. presented a new family of three-point derivative free methods by using a self-correcting parameter that is calculated applying the secant-type method in three different ways and Newton’s interpolatory polynomial of the second degree.

The three new methods (for simple roots) in this paper use three functional evaluations and have fourth-order convergence; thus, they are optimal methods and their efficiency index is \( E = 1.587 \), which is greater than the efficiency index of Newton’s method, which is \( E = 1.414 \). In the case of multiple roots, the method developed here is cubically convergent and uses three functional evaluations without the use of second derivative of the function. Thus, the method has better performance than Newton’s modified method and the above methods with efficiency index \( E = 1.4422 \).

### 2. Development of the Methods

In this paper, we consider iterative methods to find a simple root \( r \in I \) of a nonlinear equation \( f(x) = 0 \), where \( f : I \rightarrow \mathbb{R} \) is a scalar function for an open interval \( I \). We suppose that \( f(x) \) is sufficiently differentiable and \( f'(x) \neq 0 \) for \( x \in I \), and since \( r \) is a simple root, we can define \( g = f^{-1} \) on \( I \). Taking \( x_0 \in I \) closer to \( r \) and supposing that \( x_n \) has been chosen, we define
\[ z_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]
(11)
\[ K = f(x_n), \quad L = f(z_n). \]
2.1. First Method FAM1. Consider the polynomial

\[ p_2(y) = A(y - K)(y - L) + B(y - K) + C, \]  

with the conditions

\[ p_2(K) = g(K) = x_n, \quad p_2(L) = g(L) = z_n, \]

\[ p_2'(K) = g'(K) = \frac{1}{f'(x_n)}. \]  

Solving simultaneously the conditions (13) and using the common representation of divided differences for Hermite's inverse interpolation,

\[ g[f(x_n)] = x_n, \]

\[ g[f(x_n), f(x_n), f(z_n)] = \frac{1}{f'(x_n)} \left( \frac{f(x_n) - f(z_n)}{f(z_n) - f(x_n)} \right), \]

\[ g[f(x_n), f(z_n)] = \frac{z_n - x_n}{f(z_n) - f(x_n)}. \]

Consequently, we find

\[ A = g[f(x_n), f(x_n), f(z_n)], \]

\[ B = g[f(x_n), f(z_n)], \quad C = g[f(x_n)]. \]

and the polynomial (12) can be written as

\[ p_2(y) = g[f(x_n), f(x_n), f(z_n)](y - f(x_n)) \]

\[ \times (y - f(z_n)) + g[f(x_n), f(z_n)](y - f(x_n)) \]

\[ \times (y - f(z_n)) + g[f(x_n)]. \]

If we are making \( y = 0 \) in (16) we have a new iterative method (FAMI)

\[ x_{n+1} = g[f(x_n)] - g[f(x_n), f(z_n)]f(x_n) \]

\[ + g[f(x_n), f(x_n), f(z_n)]f(x_n)g(z_n). \]

It can be written as

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

\[ z_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

which uses three functional evaluations and has fourth-order convergence.

2.2. Second Method FAM2. Consider the polynomial

\[ p_3(y) = A(y - K)^2(y - L) \]

\[ + B(y - K)(y - L) + C(y - K) + D, \]

with the conditions

\[ p_3(K) = g(K) = x_n, \quad p_3(L) = g(L) = z_n, \]

\[ p_3'(K) = g'(K) = \frac{1}{f'(x_n)}. \]

Taking \( B = A(L - 2K) \), we have \( p_3(y) = A^*y^3 + B^*y + C^* \). Solving simultaneously the conditions (20) and using the common representation of divided differences for Hermite's inverse interpolation, we find

\[ A = g[f(x_n), f(x_n), f(z_n)], \]

\[ B = g[f(x_n), f(x_n), f(z_n)], \]

\[ C = g[f(x_n), f(z_n)], \quad D = g[f(x_n)], \]

and the polynomial (19) can be written as

\[ p_3(y) = g[f(x_n)] \]

\[ + \frac{g[f(x_n), f(x_n), f(z_n)]}{f(z_n) - 2f(x_n)} \times (y - f(x_n))^2 \]

\[ \times (y - f(z_n)) + g[f(x_n), f(z_n)](y - f(x_n)) \]

\[ + g[f(x_n), f(x_n), f(z_n)](y - f(x_n)) \]

\[ \times (y - f(z_n))(y - f(z_n)). \]

Then, if we are making \( y = 0 \) in (23) we have our second iterative method (FAMI)

\[ x_{n+1} = g[f(x_n)] - g[f(x_n), f(z_n)]f(x_n) \]

\[ + g[f(x_n), f(x_n), f(z_n)]f(x_n)g(z_n) \]

\[ - \frac{g[f(x_n), f(x_n), f(z_n)]}{f(z_n) - 2f(x_n)}f^2(x_n)f(z_n). \]

It can be written as

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

\[ - \frac{f^2(x_n)f(x_n)f(z_n) - 3f(x_n)}{(f(z_n) - f(x_n))f'(x_n)} \]

\[ - \frac{f^2(x_n)}{(f(z_n) - f(x_n))f'(x_n)}f'(x_n), \]

\[ z_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

which uses three functional evaluations and has fourth-order convergence.
2.3. Third Method FAM3. Consider the polynomial

\[ p_3(y) = A(y-K)^2(y-L) + B(y-K)(y-L) + C(y-K) + D, \]

with the conditions

\[ p_3(K) = g(K) = x_n, \quad p_3(L) = g(L) = z_n, \]

\[ p_3'(K) = g'(K) = \frac{1}{f'(x_n)}, \]

\[ p_3''(K) = -\frac{2f(z_n)}{f^2(x_n)f'(x_n)}, \]

which uses three functional evaluations and has third-order convergence.

Making \( y = 0 \) in (31), we have

\[ x_{n+1} = g\left[ f(x_n) \right] - g\left[ f(x_n), f(z_n) \right] f(x_n) \]

\[ + g\left[ f(x_n), f(x_n), f(z_n) \right] f(x_n) f(z_n) \]

\[ - \left( g\left[ f(x_n), f(x_n) \right] \left[ f(z_n)f^2(x_n) \right] \right) \]

\[ + g\left[ f(x_n), f(x_n), f(z_n) \right] \left( \frac{f(z_n) - f(x_n)}{f(z_n) - f(x_n)} \right) \]

\[ \times f^2(x_n)f(z_n). \]

It can be written as

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

\[ - \frac{f^3(z_n)f(x_n) - 2f(x_n)f^2(x_n)}{(f(z_n) - f(x_n))^3} f'(x_n), \]

\[ z_n = x_n - \frac{f(x_n)}{f'(x_n)}, \]

which uses three functional evaluations and has fourth-order convergence.

2.4. Method FAM4 (Multiple Roots). Consider the polynomial

\[ p_m(x) = A(x-x_n + w)^m, \]

where \( m \) is the multiplicity of the root and \( p_m(x) \) verify the conditions

\[ p_m(x_n) = f(x_n), \quad p_m(z_n) = f(z_n), \]

with \( z_n = x_n - m(f(x_n)/f'(x_n)). \)

Solving the system, we obtain

\[ w = \frac{u_m(z_n-x_n)}{1-u_m}, \]

where

\[ u_m = \left[ \frac{f(x_n)}{f'(x_n)} \right]^{1/m}. \]

Thus, we have

\[ x_{n+1} = x_n - w, \]

that can be written as

\[ x_{n+1} = z_n + \frac{mf(x_n)}{f'(x_n)} \frac{1}{1-u_m}, \]

\[ z_n = x_n - m\frac{f(x_n)}{f'(x_n)} \quad u_m = \left[ \frac{f(x_n)}{f'(x_n)} \right]^{1/m}, \]

which uses three functional evaluations and has third-order convergence.
3. Analysis of Convergence

**Theorem 1.** Let \( f : I \subseteq \mathbb{R} \to \mathbb{R} \) be a sufficiently differentiable function, and let \( r \in I \) be a simple zero of \( f(x) = 0 \) in an open interval \( I \), with \( f'(x) \neq 0 \) on \( I \). If \( x_0 \in I \) is sufficiently close to \( r \), then the methods FAM1, FAM2, and FAM3, as defined by (18), (25), and (33), have fourth-order convergence.

**Proof.** Following an analogous procedure to find the error in Lagrange's and Hermite's interpolations, the polynomials (12), (19), and (26) in FAM1, FAM2, and FAM3, respectively, have the error

\[
\varepsilon_{n+1} = \left[ \frac{g'''(\xi)}{3!} - \beta A \right] \left( f'(r) + f''(r)(x_n - r) \right)
\]

for some \( \xi \in I \). The function, and let

\[
E(y) = g(y) - p_n(y) = \left[ \frac{g'''(\xi)}{3!} - \beta A \right] (y - K)^3 (y - L),
\]

(40)

Thus, \( r = x_{n+1} = - \left[ \frac{g'''(\xi)}{3!} - \beta A \right] K^2 L \)

\[
x_{n+1} = - \left[ \frac{g'''(\xi)}{3!} - \beta A \right] f'(x_n) f'(z_n),
\]

(41)

with \( x_{n+1} - r = \varepsilon_{n+1} \). Since \( z_n \) was taken from Newton's method, we know that \( z_n - r = (f''(\xi_2)/2f'('\xi_4'))^2 + O(\varepsilon_{n+1}^2) \). Then, we have

\[
\varepsilon_{n+1} = \left[ \frac{g'''(\xi)}{3!} - \beta A \right] \left( f''(\xi_2) f'(z_n - r) \right)^2
\]

(42)

Now, in FAM1, we take \( \beta = 0 \), then,

\[
\xi_{n+1} = \frac{g'''(\xi)}{3!} \left( f''(\xi_2) \right)^2.
\]

(43)

In FAM2 and FAM3, we take \( \beta = 1 \), then,

\[
\varepsilon_{n+1} = \left[ \frac{g'''(\xi)}{3!} - A \right] \left( f''(\xi_2) f'(z_n) \right)^2
\]

(44)

Thus, FAM1, FAM2, and FAM3 have fourth-order convergence.

**Theorem 2.** Let \( f : I \subseteq \mathbb{R} \to \mathbb{R} \) be a sufficiently differentiable function, and let \( r \in I \) be a zero of \( f(x) = 0 \) with multiplicity \( m \) in an open interval \( I \). If \( x_0 \in I \) is sufficiently close to \( r \), then the method FAM4 defined by (12), (20) is cubically convergent.

**Proof.** The proof is based on the error of Lagrange's interpolation. Suppose that \( z_n \) has been chosen. We can see that

\[
f(x) - p_m(x) = \frac{f'''(\xi_1) - p_m''(\xi_1)}{2} (x - x_n) (x - z_n),
\]

(45)

for some \( \xi_1 \in I \).

\[
- \frac{p_m''(\xi_2)}{2} (r - x_{n+1}) = \frac{f'''(\xi_1) - p_m''(\xi_1)}{2} (r - x_n) (r - z_n),
\]

(46)

with \( \xi_1, \xi_2 \in I \).

\[
\varepsilon_{n+1} = \frac{f'''(\xi_1) - p_m''(\xi_1)}{2p_m''(\xi_2)} e_m^2
\]

(47)

for some \( \xi_3, \xi_4 \in I \).

Thus,

\[
\varepsilon_{n+1} = \frac{f'''(\xi_1) - p_m''(\xi_1)}{2p_m''(\xi_2)} e_m^2
\]

(48)

Therefore, FAM4 has third-order convergence.

Note that \( p_m'' \) is not zero for \( m \geq 2 \) and this fact allows the convergence of \( \lim_{n \to \infty} (\varepsilon_{n+1}/\varepsilon_m^2) \).

4. Numerical Analysis

In this section, we use numerical examples to compare the new methods introduced in this paper with Newton's classical method (NM) and recent methods of fourth-order convergence, such as Noor's method (NOM) with \( E = 1.587 \) in [1], Cordero's method (CM) with \( E = 1.587 \) in [2], Chun's third-order method (CHM) with \( E = 1.442 \) in [3], and Li's fifth-order method (ZM) with \( E = 1.379 \) in [4] in the case of simple roots. For multiple roots, we compare the method developed here with the quadratically convergent Newton's modified method (NMM) and with the cubically convergent Halley's method (HM), Osada's method (OM), Euler-Chebyshev's method (ECM), and Chun-Neta's method (CNM). Tables 2 and 4 show the number of iterations (IT) and the number of functional evaluations (NOFE). The results obtained show that the methods presented in this paper are more efficient.

All computations were done using MATLAB 2010. We accept an approximate solution rather than the exact root, depending on the precision (\( e \)) of the computer. We use
Table 1: List of functions for a single root.

<table>
<thead>
<tr>
<th>Function</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x) = x^3 + 4x^2 - 10 )</td>
<td>1.365230013414096845760868290</td>
</tr>
<tr>
<td>( f_2(x) = \cos(x) - x )</td>
<td>0.73908513321516064165531208767</td>
</tr>
<tr>
<td>( f_3(x) = \sin(x) - (1/2)x )</td>
<td>1.8954942670339809471440357381</td>
</tr>
<tr>
<td>( f_4(x) = \sin^2(x) - x^2 + 1 )</td>
<td>1.40491648215324143710673</td>
</tr>
<tr>
<td>( f_5(x) = xe^x - \sin^2(x) + 3 \cos(x) + 5 )</td>
<td>1.2076478271309189270094167584</td>
</tr>
<tr>
<td>( f_6(x) = x^2 - e^x - 3x + 2 )</td>
<td>0.257530285439860760455367304944</td>
</tr>
<tr>
<td>( f_7(x) = (x - 1)^3 - 2 )</td>
<td>2.2599210498948731647672106073</td>
</tr>
<tr>
<td>( f_8(x) = e^{-x} + \cos(x) )</td>
<td>1.746139530408012417650703089</td>
</tr>
</tbody>
</table>

Table 2: Comparison of the methods for a single root.

<table>
<thead>
<tr>
<th>Function</th>
<th>( x_0 )</th>
<th>IT</th>
<th>NOFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x) )</td>
<td>NM</td>
<td>NOM</td>
<td>CHM</td>
</tr>
<tr>
<td>( f_2(x) )</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>( f_3(x) )</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>( f_4(x) )</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>( f_5(x) )</td>
<td>1.3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>( f_6(x) )</td>
<td>-1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>( f_7(x) )</td>
<td>2</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>( f_8(x) )</td>
<td>3</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>( f_9(x) )</td>
<td>3.5</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>( f_{10}(x) )</td>
<td>1</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>( f_{11}(x) )</td>
<td>2</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>( f_{12}(x) )</td>
<td>0.5</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

the following stopping criteria for computer programs: (i) \( |x_{n+1} - x_n| < \epsilon \), (ii) \( |f(x_{n+1})| < \epsilon \). Thus, when the stopping criterion is satisfied, \( x_{n+1} \) is taken as the exact computed root \( r \). For numerical illustrations in this section, we used the fixed stopping criterion \( \epsilon = 10^{-18} \).

We used the functions in Tables 1 and 3.

The computational results presented in Tables 2 and 4 show that, in almost all of the cases, the presented methods converge more rapidly than Newton's method, Newton's modified method, and those previously presented for the case of simple and multiple roots. The new methods require less number of functional evaluations. This means that the new methods have better efficiency in computing process than Newton's method as compared to other methods, and furthermore, the method FAM3 produces the best results. For most of the functions we tested, the obtained methods behave at least with equal performance compared to the other known methods of the same order.

5. Conclusions

In this paper, we introduce three new optimal fourth-order iterative methods to solve nonlinear equations. The analysis of convergence shows that the three new methods have fourth-order convergence; they use three functional evaluations, and thus, according to Kung-Traub’s conjecture, they are optimal methods. In the case of multiple roots, the method developed here is cubically convergent and uses three
Table 3: List of functions for a multiple root.

<table>
<thead>
<tr>
<th>$f(x)$</th>
<th>$x_0$</th>
<th>IT</th>
<th>NOFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1(x)$</td>
<td>$(x^3 + 4x^2 - 10)^3$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_2(x)$</td>
<td>$(\sin^2(x) - x^2 + 1)^2$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_3(x)$</td>
<td>$(x^2 - e^x - 3x + 2)^5$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_4(x)$</td>
<td>$(\cos(x) - x)^3$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_5(x)$</td>
<td>$((x - 1)^3 - 1)^5$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_6(x)$</td>
<td>$(x e^x - \sin^2(x) + 3 \cos(x) + 5)^4$,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_7(x)$</td>
<td>$(\sin(x) - (1/2)x)^2$.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Numerical analysis shows that these methods have better performance as compared with Newton’s classical method, Newton’s modified method, and other recent methods of third- (multiple roots) and fourth-order (simple roots) convergence.

Acknowledgments

The authors wish to acknowledge the valuable participation of Professor Nicole Mercier and Professor Joelle Ann Labrecque in the proofreading of this paper. This paper is the result of the research project “Análisis Numérico de Métodos Iterativos Óptimos para la Solución de Ecuaciones No Lineales” developed at the Universidad del Istmo, Campus Tehuantepec, by Researcher-Professor Gustavo Fernández-Torres.

References


