

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/298640623>

Weighted Gaussian correction of Newton-type methods for solving nonlinear systems

Article · January 2016

CITATIONS

2

READS

90

3 authors:



Alicia Cordero

Universitat Politècnica de València

215 PUBLICATIONS 2,401 CITATIONS

[SEE PROFILE](#)



Juan R. Torregrosa

Universitat Politècnica de València

286 PUBLICATIONS 2,701 CITATIONS

[SEE PROFILE](#)



Maria Penkova

Instituto Tecnológico de Santo Domingo

28 PUBLICATIONS 338 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Iterative methods without memory - Complex Dynamics [View project](#)

Weighted Gaussian correction of Newton-type methods for solving nonlinear systems

by

ALICIA CORDERO^a, JUAN R. TORREGROSA^a AND MARÍA P. VASSILEVA^b

Abstract

A new technique to design predictor-corrector methods for solving nonlinear equations or nonlinear systems is presented. With Newton's scheme as a predictor and any Gaussian quadrature as a corrector we construct, by using weight function procedure, iterative schemes of order four, with independence of both the number of nodes used in the quadrature and the orthogonal polynomials employed. These methods are obtained by assuming some conditions on the weight function related to the weights and nodes of the corresponding Gaussian quadrature. These methods are optimal, in the sense of Kung-Traub conjecture, in one-dimensional case. Some numerical tests allow us to confirm the theoretical results and show that the proposed methods need less computational time than well-known procedures, such as Newton's and Jarratt's schemes.

Key Words: Nonlinear system of equations, Gaussian quadrature, Pseudocomposition, Weight function procedure, Multipoint method, Optimal order, Efficiency.

2010 Mathematics Subject Classification: Primary: 65H10, Secondary: 65H05, 65D30.

1 Introduction

Different applied problems in many fields of science, engineering and technology require to find the solutions of a nonlinear system of equations $F(x) = 0$ with $F : D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^m$. The best known method for finding a zero ξ of F (being the Jacobian matrix of the function F evaluated at ξ , $F'(\xi)$, nonsingular), as it is very simple and efficient, is Newton's scheme, whose iterative expression is

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)}), \quad k = 0, 1, \dots$$

Multipoint iterative methods for solving nonlinear equations appeared for the first time in Ostrowski's book [7] and then they were extensively studied in

Traub's text [11] and some papers published in the 1960s and 1970s as for example [5]. We can see an interesting review of these methods in the book of Petković et al. [8]. The reason for the revived interest in this area is the property of multipoint methods to overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency, which is of great practical importance. The multipoint methods are introduced in the beginning to get the highest possible order of convergence using a fixed number of function evaluations. This is closely related to the optimal order of convergence in the sense of the Kung-Traub's conjecture [6]. In fact, studying the optimal convergence rate of multipoint schemes, Kung and Traub in [6] conjectured that multipoint methods without memory for solving nonlinear equations, which use $n + 1$ functional evaluations per iteration, have order of convergence at most 2^n .

Although there exist some robust and efficient methods for solving nonlinear systems, the design of new and efficient schemes for solving nonlinear systems is an interesting goal (see, for example, [1, 9, 13, 4] and the references therein). The authors in [3] designed a general procedure called pseudocomposition. Initially, multipoint third-order methods were generated by using Newton's scheme as a predictor and Gaussian quadratures as a corrector. Some conditions on nodes and coefficients of the quadrature were necessary to reach the third order of convergence, so only a subset of the Gaussian quadratures could be used (for example, by using Gauss-Chebyshev quadrature we do not reach order of convergence three, it is only linear). More later, we generalized the procedure in order to be able to use other methods as a predictor, obtaining high-order schemes with the Gaussian quadratures as a corrector,

$$x^{(k+1)} = x^{(k)} - 2 \left[\sum_{i=1}^n \omega_i F'(\eta_i^{(k)}) \right]^{-1} F(x^{(k)}), \quad (1.1)$$

where $\eta_i^{(k)} = \frac{(1 + \tau_i)y^{(k)} + (1 - \tau_i)x^{(k)}}{2}$, being $y^{(k)}$ the predictor step and ω_i and τ_i , $i = 1, \dots, n$, are the weights and nodes, respectively, of the orthogonal polynomial of degree n that define the corresponding Gaussian quadrature.

The aim of this paper is to improve the pseudocomposition technique for obtaining iterative methods (optimal in the scalar case) by using any Gaussian quadrature. The weight function approach is a known procedure to design optimal methods for solving scalar equations. The introduction of the weight function technique allows us to reach order of convergence four with no restriction on the orthogonal family of polynomials used in the Gaussian quadrature, nor in the number of nodes employed. This technique is adaptive to the orthogonal family, in the sense of the conditions on the weight function and its derivatives are expressed in terms of some variables that characterize the different Gaussian families of nodes and weights.

The structure of the rest of the manuscript is as follows: in Section 2 we design the general procedure to obtain fourth-order predictor-corrector methods

from any Gaussian quadratures and show some weight functions verifying the conditions imposed in the main Theorem, from different orthogonal polynomials. These methods are checked in the numerical section, in order to establish their applicability. The paper finishes with some conclusions and the references used in it.

2 Design and convergence analysis of the proposed class

By using a damped Newton's method as a predictor and a weight function procedure in the corrector step, we design the following set of families of two-point schemes:

$$\begin{aligned} y^{(k)} &= x^{(k)} - \beta \left[F'(x^{(k)}) \right]^{-1} F(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - 2H(u^{(k)}) \left[\sum_{i=1}^n \omega_i F'(\eta_i^{(k)}) \right]^{-1} F(x^{(k)}), \end{aligned} \quad (2.1)$$

where β is a non-zero real parameter and $u^{(k)} = [F'(x^{(k)})]^{-1} \sum_{i=1}^n \omega_i F'(\eta_i^{(k)})$.

Let us notice that weight function $H(u)$, used in expression (2.1), is a matrix function of matrix variable. Specifically, if $X = \mathbb{R}^{m \times m}$ denotes the Banach space of real square matrices of size $m \times m$, then we can define $H : X \rightarrow X$ such that its Frechet derivatives satisfy:

- a) $H'(u)(v) = H_1 uv$, where $H' : X \rightarrow \mathcal{L}(X)$ and $H_1 \in \mathbb{R}$,
- b) $H''(u, v)(w) = H_2 uvw$, where $H'' : X \times X \rightarrow \mathcal{L}(X)$ and $H_2 \in \mathbb{R}$.

Let us remark that $\mathcal{L}(X)$ denotes the space of linear mappings from X to itself.

Function H should be determined in such a way that the order of convergence of the two-point method (2.1) is four. Let us assume that $x^{(k)}$ is sufficiently close to the zero ξ , then $u^{(k)}$ is close enough to the identity matrix I of size $m \times m$. Now, let us consider the first terms of the Taylor expansion of the function H around I ,

$$H(u^{(k)}) \approx H(I) + H_1(u^{(k)} - I) + \frac{1}{2}H_2(u^{(k)} - I)^2. \quad (2.2)$$

The convergence of the proposed iterative scheme will be demonstrated by using the procedure introduced in [2]. We also introduce the following notation,

$\sigma = \sum_{i=1}^n \omega_i$ and $\sigma_j = \frac{1}{\sigma} \sum_{i=1}^n \omega_i \tau_i^j$, $j = 1, 2, \dots$, that allows us to simplify the demonstration and analyze the conditions under which the methods converge with fourth-order of convergence.

Theorem 1. *Let $\xi \in D$ be a zero of a sufficiently differentiable function $F : D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^m$. Let us also suppose that the initial estimation $x^{(0)}$ is close enough to*

the solution ξ and $F'(\xi)$ is nonsingular. The methods of family (2.1) have order of convergence four if

$$\beta = \frac{4(1 + \sigma_1)}{3(1 + 2\sigma_1 + \sigma_2)},$$

and a sufficiently differentiable matrix function H is chosen satisfying the conditions

$$\begin{aligned} H(I) &= \frac{\sigma}{2}I, \quad H_1 = \frac{\sigma(1 + 2\sigma_1 + 4\sigma_1^2 + 3\sigma_2)}{8(1 + \sigma_1)^2}, \\ H_2 &= -\frac{3\sigma(-1 - 4\sigma_1 - 2\sigma_1^2 + 4\sigma_1^3 - 4\sigma_2 - 8\sigma_1\sigma_2 + 2\sigma_1^2\sigma_2 - 3\sigma_2^2)}{8(1 + \sigma_1)^4} \end{aligned}$$

and H''' is a bounded operator, where σ and σ_j , $j = 1, 2$ depend on the Gaussian quadrature used.

Proof. By developing in Taylor series functions $F(x^{(k)})$ and $F'(x^{(k)})$ we obtain

$$\begin{aligned} F(x^{(k)}) &= F'(\xi)(e_k + C_2e_k^2 + C_3e_k^3 + C_4e_k^4) + \mathcal{O}[e_k^5], \\ F'(x^{(k)}) &= F'(\xi)(I + 2C_2e_k + 3C_3e_k^2 + 4C_4e_k^3 + 5C_5e_k^4) + \mathcal{O}[e_k^5], \end{aligned}$$

where $C_q = (1/q!)[F'(\xi)]^{-1}F^{(q)}(\xi)$, $q \geq 2$ and $e_k = x^{(k)} - \xi$.

Taking into account that $[F'(x^{(k)})]^{-1}F'(x^{(k)}) = I$, we obtain

$$[F'(x^{(k)})]^{-1} = [I + X_2e_k + X_3e_k^2 + X_4e_k^3 + X_5e_k^4][F'(\xi)]^{-1} + \mathcal{O}[e_k^5], \quad (2.3)$$

where

$$\begin{aligned} X_2 &= -2C_2, \\ X_3 &= 4C_2^2 - 3C_3, \\ X_4 &= 6C_3C_2 - 8C_2^3 + 6C_2C_3 - 4C_4, \\ X_5 &= 16C_2^5 - 12C_3C_2^2 - 12C_2C_3C_2 + 8C_4C_2 + 9C_3^2 - 12C_2^2C_3 + 8C_2C_4 - 5C_5. \end{aligned}$$

So,

$$\begin{aligned} y^{(k)} &= x^{(k)} - \beta [F'(x^{(k)})]^{-1} F(x^{(k)}) \\ &= \xi + D_1e_k + D_2e_k^2 + D_3e_k^3 + D_4e_k^4 + \mathcal{O}[e_k^5], \end{aligned}$$

where

$$\begin{aligned} D_1 &= 1 - \beta, \\ D_2 &= \beta C_2, \\ D_3 &= 2\beta (C_3 - C_2^2), \\ D_4 &= \beta (4C_2^3 - 4C_2C_3 - 3C_3C_2 + 3C_4). \end{aligned}$$

Now,

$$\begin{aligned} \eta_i^{(k)} - \xi &= \frac{(1 + \tau_i)y^{(k)} + (1 - \tau_i)x^{(k)}}{2} - \xi \\ &= \frac{1}{2} [(1 + D_1 - \tau_i + D_1\tau_i)e_k + (1 + \tau_i)(D_2e_k^2 + D_3e_k^3 + D_4e_k^4)] \\ &\quad + \mathcal{O}[e_k^5]. \end{aligned}$$

Expanding again in Taylor series, we obtain:

$$\begin{aligned}
K &= \sum_{i=1}^n \omega_i F'(\eta_i^{(k)}) \\
&= \sum_{i=1}^n \omega_i F'(\xi) \left[I + 2C_2(\eta_i^{(k)} - \xi) + 3C_3(\eta_i^{(k)} - \xi)^2 + \dots \right] \quad (2.4) \\
&= \sigma F'(\xi) [I + B_1 e_k + B_2 e_k^2 + B_3 e_k^3 + B_4 e_k^4] + \mathcal{O}[e_k^5],
\end{aligned}$$

where

$$\begin{aligned}
B_1 &= b_{1,1} C_2, \\
B_2 &= b_{2,1} C_2^2 + b_{2,2} C_3, \\
B_3 &= b_{3,1} C_3 C_2 - b_{3,2} C_2^3 + b_{3,3} C_2 C_3 + b_{3,4} C_4, \\
B_4 &= b_{4,1} C_5 + b_{4,2} C_2 C_4 + b_{4,3} C_2^2 C_3 + b_{4,4} C_3^2 + b_{4,5} C_2^4 \\
&\quad + b_{4,6} C_2 C_3 C_2 + b_{4,7} C_3 C_2^2 + b_{4,8} C_4 C_2,
\end{aligned}$$

and the coefficients $b_{i,j}$ are

$$\begin{aligned}
b_{1,1} &= 2 - \beta - \beta\sigma_1, \quad b_{2,1} = \beta(1 + \sigma_1), \\
b_{2,2} &= 3 - 3\beta + \frac{3}{4}\beta^2 - \frac{3}{2}\beta(2 - \beta)\sigma_1 + \beta^2\sigma_2, \\
b_{3,1} &= 2b_{2,1}, \quad b_{3,2} = -b_{3,1}, \quad b_{3,3} = \frac{3}{2}\beta(2 - \beta - 2(1 - \beta)\sigma_1 - \beta\sigma_2), \\
b_{3,4} &= \frac{1}{2}((2 - \beta)^2 - 3(2 - \beta)^2\beta\sigma_1 + 3\beta^2(2 - \beta)\sigma_2 - \beta^2\sigma_3), \\
b_{4,1} &= \frac{5}{16}((2 - \beta)^4 - 4\beta(2 - \beta)^3\sigma_1 + 6\beta^2(2 - \beta)^2\sigma_2 - 4\beta^3(2 - \beta)\sigma_3 + \beta^4\sigma_4) \\
b_{4,2} &= \frac{3}{4}\beta((2 - \beta)^2 + (2 - \beta)(2 - 3\beta)\sigma_1 + \beta(3\beta - 4)\sigma_2 + \beta^2\sigma_3), \\
b_{4,3} &= \frac{3}{4}(1 + 2\sigma_1 + 5\sigma_2)\beta^2 - 3\beta(2 - \beta - 2(1 - \beta)\sigma_1), \\
b_{4,4} &= 3\beta(2 - \beta - 2(1 - \beta)\sigma_1 - \beta\sigma_2), \\
b_{4,5} &= 4b_{2,1}, \quad b_{4,6} = -b_{4,5}, \quad b_{4,7} = -3b_{2,1}, \quad b_{4,8} = -b_{4,7}.
\end{aligned}$$

Moreover,

$$u^{(k)} = \left[F'(x^{(k)}) \right]^{-1} K = I + U_1 e_k + U_2 e_k^2 + U_3 e_k^3 + U_4 e_k^4 + \mathcal{O}[e_k^5],$$

where $U_1 = B_1 + X_2$, $U_2 = B_2 + X_2 B_1 + X_3$, $U_3 = B_3 + X_2 B_2 + X_3 B_1 + X_4$ and $U_4 = B_4 + X_2 B_3 + X_3 B_2 + X_4 B_1 + X_5$.

By using Taylor expansion of $H(u^{(k)})$ around I , we obtain

$$H(u^{(k)}) = h_0 + h_1 e_k + h_2 e_k^2 + h_3 e_k^3 + h_4 e_k^4 + \mathcal{O}[e_k^5],$$

where $h_0 = H(I)$, $h_1 = H_1U_1$, $h_2 = \frac{1}{2}H_2U_1^2 + H_1U_2$, $h_3 = H_2U_1U_2 + H_1U_3$ and $h_4 = \frac{1}{2}H_2U_2^2 + H_2U_1U_3 + H_1U_4$.

Again, we assume that $K^{-1} = \frac{1}{\sigma}(I + Y_2e_k + Y_3e_k^2 + Y_4e_k^3 + Y_5e_k^4)[F'(\xi)]^{-1} + \mathcal{O}[e_k^5]$, where

$$\begin{aligned} Y_2 &= -B_1, \\ Y_3 &= -B_2 + B_1^2, \\ Y_4 &= B_3 + B_1B_2 + B_2B_1 - B_1^3, \\ Y_5 &= -B_4 + B_1B_3 + B_2^2 - B_1^2B_2 + B_3B_1 - B_1B_2B_1 - B_2B_1^2 + B_1^3. \end{aligned}$$

So, now we can calculate

$$L(x^{(k)}) = H(u^{(k)})K^{-1} = \frac{1}{\sigma}(h_0 + L_1e_k + L_2e_k^2 + L_3e_k^3 + L_4e_k^4)[F'(\xi)]^{-1} + \mathcal{O}[e_k^5],$$

where

$$\begin{aligned} L_1 &= h_0Y_2 + h_1, \\ L_2 &= h_0Y_3 + h_1Y_2 + h_3, \\ L_3 &= h_0Y_4 + h_1Y_3 + h_2Y_2 + h_3, \\ L_4 &= h_0Y_5 + h_1Y_4 + h_2Y_3 + h_3Y_2 + h_4. \end{aligned}$$

Then,

$$M(x^{(k)}) = L(x^{(k)})F(x^{(k)}) = \frac{1}{\sigma}(M_1e_k + M_2e_k^2 + M_3e_k^3 + M_4e_k^4) + \mathcal{O}[e_k^5],$$

where $M_1 = h_0$, $M_2 = h_0L_1 + h_1$, $M_3 = h_0L_2 + h_1L_1 + h_2$ and $M_4 = h_0L_3 + h_1L_2 + h_2L_1 + h_3$.

Finally, we get the error equation of the proposed iterative scheme

$$e_{k+1} = e_k - 2M(x^{(k)}) = \left(I - \frac{2M_1}{\sigma}\right)e_k - \frac{2}{\sigma}(M_2e_k^2 + M_3e_k^3 + M_4e_k^4) + \mathcal{O}[e_k^5].$$

The conditions to be fulfilled by the function $H(u)$ to have order of convergence at least four are determined by solving the system of simultaneous equations:

$$I - \frac{2M_1}{\sigma} = 0, \quad M_2 = 0 \quad \text{and} \quad M_3 = 0.$$

From the first equation we obtain the condition that the value of the function $H(t)$ at the identity matrix must be equal to the sum of the weights corresponding to orthogonal polynomial of the Gaussian quadrature used, divided by two $H(I) = \frac{\sigma}{2}I$. From the second and third equations, we obtain

$$\beta = \frac{4(1 + \sigma_1)}{3(1 + 2\sigma_1 + \sigma_2)}, \quad H_1 = \frac{\sigma(-1 + \beta(1 + \sigma_1))}{2\beta(1 + \sigma_1)}$$

and

$$H_2 = -\frac{3\sigma(-1 - 4\sigma_1 - 2\sigma_1^2 + 4\sigma_1^3 - 4\sigma_2 - 8\sigma_1\sigma_2 + 2\sigma_1^2\sigma_2 - 3\sigma_2^2)}{8(1 + \sigma_1)^4}.$$

By substituting the obtained value of β , we get $H_1 = \frac{\sigma(1 + 2\sigma_1 + 4\sigma_1^2 - 3\sigma_2)}{8(1 + \sigma_1)^2}$ and the resulting expression of the error is

$$e_{k+1} = (m_{4,1}C_2^3 + m_{4,2}C_2C_3 + m_{4,3}C_3C_2 + m_{4,4}C_4)e_k^4 + \mathcal{O}[e_k^5], \quad (2.5)$$

where

$$\begin{aligned} m_{4,1} &= \frac{7 + 14\sigma_1 - 8\sigma_1^2 + 15\sigma_2}{3(1 + 2\sigma_1 + \sigma_2)}, \\ m_{4,2} &= \frac{-29 - 174\sigma_1 - 288\sigma_1^2 + 8\sigma_1^3 + 312\sigma_1^4 + 144\sigma_1^5 + 64\sigma_1^6}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)} \\ &\quad + \frac{-147\sigma_2 - 588\sigma_1\sigma_2 - 612\sigma_1^2\sigma_2 - 48\sigma_1^3\sigma_2 - 120\sigma_1^4\sigma_2}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)} \\ &\quad + \frac{-90\sigma_2^2 - 270\sigma_1\sigma_2^2 + 108\sigma_1^2\sigma_2^2 - 82\sigma_2^3}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)}, \\ m_{4,3} &= \frac{11 + 66\sigma_1 + 54\sigma_1^2 - 224\sigma_1^3 - 384\sigma_1^4 - 144\sigma_1^5 - 64\sigma_1^6}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)} \\ &\quad + \frac{111\sigma_2 + 444\sigma_1\sigma_2 + 432\sigma_1^2\sigma_2 - 24\sigma_1^3\sigma_2 + 120\sigma_1^4\sigma_2}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)} \\ &\quad + \frac{117\sigma_2^2 + 234\sigma_1\sigma_2^2 - 126\sigma_1^2\sigma_2^2 + 81\sigma_2^3}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)}, \\ m_{4,4} &= \frac{2 + 12\sigma_1 + 42\sigma_1^2 + 56\sigma_1^3 + 24\sigma_1^4 - 12\sigma_2 + 36\sigma_1^2\sigma_2}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)} \\ &\quad + \frac{24\sigma_1^3\sigma_2 + 18\sigma_2^2 + 36\sigma_1\sigma_2^2 + 18\sigma_1^2\sigma_2^2 - 16(1 + \sigma_1)^3\sigma_3}{18(1 + \sigma_1)^2(1 + 2\sigma_1 + \sigma_2)}. \end{aligned}$$

□

Table 1 shows the values of σ , σ_1 , σ_2 and β , depending on the weights and the nodes of the orthogonal polynomials used in the Gaussian quadrature and also the corresponding values of the weight function and its derivatives, which are used in the iterative scheme.

Let us remark that the class of methods designed allows the use of any Gaussian quadrature. In fact, when the technique of pseudocomposition was defined in [3], based also in Gaussian quadrature, the Chebyshev orthogonal polynomials could not be employed, as they did not verify the hypothesis of the main Theorem. Nevertheless, with this new design, all the orthogonal polynomials can be applied and all of them derive optimal methods in the scalar case. In practice, we

| Quadrature | No. nodes | σ | σ_1 | σ_2 | β | $H(I)$ | H_1 | H_2 |
|-----------------|-----------|----------|------------|------------|---------|------------------|---------|----------|
| Gauss-Chebyshev | 1 | π | 0 | 0 | 4/3 | $\frac{\pi}{2}I$ | $\pi/8$ | $3\pi/8$ |
| Gauss-Legendre | 1 | 2 | 0 | 0 | 4/3 | I | 1/4 | 3/4 |
| Gauss-Lobatto | 1 | 2 | 0 | 0 | 4/3 | I | 1/4 | 3/4 |
| | 2 | 2 | 0 | 1 | 2/3 | I | -1/2 | 6 |
| Gauss-Radau | 1 | 2 | -1 | 0 | 0 | $\frac{1}{2}I$ | - | - |
| | 2 | 2 | 0 | 1/3 | 1 | I | 0 | 2 |

Table 1: Nodes, $H(I)$, H_1 and H_2 for the different Gaussian quadratures

use only the quadratures with one and two nodes because, according to Theorem 1, the order of convergence is independent of the number of nodes.

For the one dimensional case, according to the Kung-Traub's conjecture [6], the obtained fourth-order methods are optimal (in case of Gauss-Radau with one node, classical Newton's method is obtained). Other new methods are obtained in the rest of cases. In Table 2 we show the weight functions used for each iterative scheme coming from the respective Gaussian quadrature rules and we also establish the notation used further. Let us note that the iterative method coming from Gauss-Lobatto with 1 node is the same as the resulting from the application of Gauss-Legendre with also 1 node. In this case, both coincide with the fourth-order procedure recently published by Sharma et al. in [10]. Let us note that other weight functions should derive in other new schemes. Indeed, we include the procedure GR2, coming from Gauss-Radau quadratures, with 2 nodes.

Let us also remark that the order of convergence of methods from Gaussian quadrature but without weight functions (see [3]) depends on the family of orthogonal polynomials used. In fact, in [3] it is proved that Chebyshev polynomials induce a linear family of iterative schemes, meanwhile the rest of orthogonal polynomials used in this work induce iterative methods of order three. However, when weight-function procedure is introduced in the design process, the order of the obtained families is always four, not depending on the orthogonal family used.

| Quadrature | N. of nodes | Name | $H(u)$ |
|-----------------|-------------|------|--|
| Gauss-Chebyshev | 1 | GC1 | $\frac{\pi}{16} \frac{5-12u+15u^2}{u^2}$ |
| Gauss-Legendre | 1 | GLe1 | $\frac{1}{8} (9-4u+3u^2)$ |
| Gauss-Lobatto | 2 | GLo2 | $\frac{9}{2} - \frac{13}{2}u + 3u^2$ |
| Gauss-Radau | 2 | GR2 | $u^2 - 2u + 2$ |

Table 2: Notation and weight functions for the different Gaussian quadratures

For example, the iterative expression of the method obtained by using the

Gauss-Legendre quadrature with one node is

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{4}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= \frac{1}{2}(x^{(k)} + y^{(k)}), \\ x^{(k+1)} &= x^{(k)} - \frac{9}{8}[F'(z^{(k)})]^{-1}F(x^{(k)}) + D[F'(x^{(k)})]^{-1}F(x^{(k)}), \end{aligned}$$

where $D = (\frac{1}{2}I - \frac{3}{8}[F'(x^{(k)})]^{-1}[F'(z^{(k)})]^{-1})$.

3 Numerical tests for selected problems

In the following, we apply the proposed class of iterative schemes with order of convergence four to estimate the solution of some particular equations and systems of nonlinear equations. The obtained results will be compared with some known methods existing in the literature with the same theoretical order of convergence and Newton's scheme denoted by NM.

In particular, we use the fourth-order iterative scheme designed by Jarratt in [5]

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - \frac{1}{2}MN[F'(x^{(k)})]^{-1}F(x^{(k)}), \end{aligned}$$

which we denote by JM, where $M = [3F'(y^{(k)}) - F'(x^{(k)})]^{-1}$ and $N = 3F'(y^{(k)}) + F'(x^{(k)})$; the iterative scheme

$$\begin{aligned} y^{(k)} &= x^{(k)} - \frac{2}{3}[F'(x^{(k)})]^{-1}F(x^{(k)}), \\ x^{(k+1)} &= x^{(k)} - \frac{1}{2}T[F'(x^{(k)})]^{-1}F(x^{(k)}), \end{aligned}$$

published in [10] by Sharma et al. with fourth-order of convergence that we denote by SHM, where $T = -I + \frac{9}{4}[F'(y^{(k)})]^{-1}F'(x^{(k)}) + \frac{3}{4}[F'(x^{(k)})]^{-1}F'(y^{(k)})$, and the fourth-order iterative scheme

$$\begin{aligned} y^{(k)} &= x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \\ z^{(k)} &= x^{(k)} - [F'(x^{(k)})]^{-1}[F(x^{(k)}) + F(y^{(k)})], \\ x^{(k+1)} &= y^{(k)} - [F'(z^{(k)})]^{-1}F(y^{(k)}), \end{aligned}$$

designed by Abad et al. in [1] and denoted by ABM.

Numerical computations have been carried out using variable precision arithmetic, in MATLAB 14a. The stopping criterion used has been $\|x^{(k+1)} - x^{(k)}\| < tol$ or $\|F(x^{(k+1)})\| < tol$. For every method, we count the number of iterations needed to reach the wished tolerance (tol), we calculate the approximated computational order of convergence *ACOC* by using the formula

$$p \approx ACOC = \frac{\ln(\|x^{(k+1)} - x^{(k)}\|/\|x^{(k)} - x^{(k-1)}\|)}{\ln(\|x^{(k)} - x^{(k-1)}\|/\|x^{(k-1)} - x^{(k-2)}\|)},$$

(if it is not stable, it is marked in the table with '-') and the error estimation made with the last values of $\|x^{(k+1)} - x^{(k)}\|$ and $\|F(x^{(k+1)})\|$. For the estimation of computational cost we use the average time of 100 runs of each algorithm in a processor Intel(R) Core (TM) i5-3210M CPU .50 GHz (64-bit Machine), OS X V.10.9.4, RAM Memory 16 GB, 1600 MHz DDR3.

3.1 Performance on scalar equations

Let us remark that proposed methods appearing in Table 2 are optimal, in the sense of Kung-Traub conjecture, when they are applied on scalar equations. It is also the case of known methods NM, JM and SHM.

To check the numerical behavior of these methods, an applied chemical problem is analyzed: when the flow within a round-section pipe is analyzed, different models are used (see, for example, [12]); these models show experimental relationship among the different variables in the flow transport in a pipe, such as Reynolds number Re with the longitude, inner diameter and rugosity of the pipe ε_r and its friction factor f_f .

Colebrook-White equation is one of the more precise and wide-rank ways to calculate the friction factor f_f associated to a pipe, but is an implicit function that must be solved in an iterative way,

$$\frac{1}{\sqrt{f_f}} = -2.0 \log_{10} \left(\frac{\varepsilon_r}{3.7065} + \frac{2.5226}{Re \sqrt{f_f}} \right)$$

In the following a particular case is shown, corresponding to the values of Reynolds number $Re = 4 \cdot 10^3$ and rugosity factor $\varepsilon_r = 1 \cdot 10^{-4}$ (in this case, the friction factor is $f_f \approx 0.0401$). The behavior of Newton's method and also the proposed schemes and other known fourth-order procedures is shown on this problem.

We will use different initial estimation x_0 in all the analyzed methods. We show in Table 3 the performance of the new and known methods by means of the following items: the number of iterations *iter*, the error estimation of the last iteration, $|x_{k+1} - x_k|$ and $|f(x_{k+1})|$ (being $f(x) = 0$ the nonlinear equation whose root is the friction factor), the approximated order of convergence *ACOC* and the mean elapsed time (e-time) in seconds, calculated by the mean of the cputime elapsed after 100 executions of each method. Variable precision arithmetics with

32 digits has been used, and the tolerance used in the stopping criterium has been $tol = 10^{-16}$. When a method does not converge, it is marked in the table with 'nc'.

| Method | x_0 | iter | $ x_{k+1} - x_k $ | $ f(x_{k+1}) $ | ACOC | e-time (sec) |
|--------|-------|------|-------------------|----------------|--------|--------------|
| NM | 0.07 | 6 | 2.6220e-11 | 8.9484e-19 | 2.0020 | 0.2017 |
| | 0.1 | nc | - | - | - | - |
| JM | 0.07 | 3 | 1.677e-15 | 1.6992e-30 | 4.0769 | 0.1524 |
| | 0.1 | 3 | 1.32e-10 | 1.3374e-25 | 4.1342 | 0.1530 |
| SHM | 0.07 | 4 | 3.3485e-16 | 3.3926e-31 | 4.0061 | 0.1991 |
| | 0.1 | nc | - | - | - | - |
| ABM | 0.07 | 5 | 3.3536e-12 | 2.0277e-37 | - | 0.2893 |
| | 0.1 | nc | - | - | - | - |
| GC1 | 0.07 | 3 | 5.6034e-7 | 4.1223e-20 | 4.0445 | 0.1527 |
| | 0.1 | 4 | 1.7318e-6 | 3.8113e-18 | 4.0945 | 0.2017 |
| GLe1 | 0.07 | 4 | 3.3485e-16 | 3.3926e-31 | 4.0061 | 0.2095 |
| | 0.1 | nc | - | - | - | - |
| GLo2 | 0.07 | 4 | 5.2558e-8 | 4.4819e-23 | 4.0908 | 0.1979 |
| | 0.1 | nc | - | - | - | - |
| GR2 | 0.07 | 4 | 1.3064e-11 | 1.3236e-26 | 4.0134 | 0.2042 |
| | 0.1 | nc | - | - | - | - |

Table 3: Performance of the methods for $Re = 4 \cdot 10^3$ and $\varepsilon_r = 1 \cdot 10^{-4}$

In Table 3 it can be observed that the performance of the proposed methods can improve the results obtained by Newton' and other known methods, not only in terms of number of iterations, but also in terms of convergence to the solution (when the initial estimation is not very close to the solution and Newton fails) and execution time.

3.2 Performance on multidimensional problems

In this section, to force proposed methods on more exigent problems, numerical computations have been carried out using variable precision arithmetics with 2000 digits of mantissa and the stopping criterion uses $tol = 10^{-700}$.

For every method, we count the number of iterations needed to reach the wished tolerance, we calculate the approximated computational order of convergence $ACOC$, the error estimation made with the last values of $\|x^{(k+1)} - x^{(k)}\|$ and $\|F(x^{(k+1)})\|$ and the mean execution time after 100 runs of each algorithm.

The nonlinear functions $F(x)$, the desired zeros ξ and the initial estimations $x^{(0)}$ used for the tests, joint with the numerical results obtained are described in the following.

Problem 1. Let us consider the two-dimensional nonlinear function

$$F_1(x_1, x_2) = (\exp x_1 \exp x_2 + x_1 \cos x_2, x_1 + x_2 - 1)^T,$$

whose approximated solution is $\xi \approx (3.471, -2.471)^T$. We will use the initial estimation $x^{(0)} = (3, -2)^T$ in all the analyzed methods. We show in Table 4 the performance of the new and known methods by means of the following items: the number of iterations $iter$, the error estimation of the last iteration, $\|x^{(k+1)} - x^{(k)}\|$ and $\|F(x^{(k+1)})\|$ and the approximated order of convergence ACOC. Moreover, in Table 5 we are going to show the behavior of the methods in the first three iterations, in order to compare their stability from the same initial estimation.

| Method | iter | $\ x^{(k+1)} - x^{(k)}\ $ | $\ F(x^{(k+1)})\ $ | ACOC | e-time (sec) |
|--------|------|---------------------------|--------------------|--------|--------------|
| NM | 8 | 0.8069e-397 | 0.4802e-794 | 2.0000 | 3.6836 |
| JM | 4 | 0.3958e-253 | 0.5734e-1014 | 4.0000 | 2.5851 |
| SHM | 4 | 0.7993e-251 | 0.3015e-1005 | 4.0000 | 3.0093 |
| ABM | 4 | 0.2395e-400 | 0.3812e-1604 | 4.0000 | 2.7418 |
| GC1 | 5 | 0.4332e-255 | 0.1634e-1022 | 4.0000 | 1.8790 |
| GLe1 | 4 | 0.5652e-251 | 0.3015e-1005 | 4.0000 | 1.8706 |
| GLo2 | 5 | 0.1205e-998 | 0.3544e-998 | 4.0000 | 2.2229 |
| GR2 | 4 | 0.4642e-250 | 0.1515e-1001 | 4.0000 | 1.8604 |

Table 4: Performance of the methods for F_1 with $x^{(0)} = (3, -2)^T$

| | NM | JM | SHM | ABM |
|-------------------------|------------|------------|------------|------------|
| $\ x^{(1)} - x^{(0)}\ $ | 0.4675 | 0.4704 | 0.6652 | 0.6656 |
| $\ F(x^{(1)})\ $ | 0.9213e-2 | 0.7415e-3 | 0.7420e-3 | 0.6279e-5 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.3128e-2 | 0.2521e-3 | 0.3563e-3 | 0.3019e-5 |
| $\ F(x^{(2)})\ $ | 0.7167e-5 | 0.9431e-15 | 0.1195e-14 | 0.9628e-24 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.2437e-5 | 0.3207e-15 | 0.3597e-15 | 0.4630e-24 |
| $\ F(x^{(3)})\ $ | 0.4379e-11 | 0.2471e-62 | 0.8055e-62 | 0.5325e-99 |
| | GC1 | GLe1 | GLo2 | GR2 |
| $\ x^{(1)} - x^{(0)}\ $ | 0.6650 | 0.4704 | 0.4704 | 0.4704 |
| $\ F(x^{(1)})\ $ | 0.7412e-3 | 0.7420e-3 | 0.1470e-1 | 0.7422e-3 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.3564e-3 | 0.2523e-3 | 0.2524e-3 | 0.2523e-3 |
| $\ F(x^{(2)})\ $ | 0.7480e-15 | 0.1195e-14 | 0.7424e-3 | 0.1321e-14 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.3597e-15 | 0.4063e-15 | 0.4924e-15 | 0.4493e-16 |
| $\ F(x^{(3)})\ $ | 0.7763e-63 | 0.8055e-62 | 0.1448e-14 | 0.1330e-61 |

Table 5: Error estimations for F_1 with $x^{(0)} = (3, -2)^T$

In this case, methods ABM, GC1 and GR2 show the best behavior as its error estimations are the lowest ones, being the differences between the results obtained by all the methods very small. However, when the elapsed time of the methods is observed, proposed schemes GR2, GLe1 and GC1 show to be faster than known methods.

Problem 2. Now, let us consider the four-dimensional nonlinear function

$F_2(x_1, x_2, x_3, x_4) = (x_2x_3 + x_4(x_2 + x_3), x_1x_3 + x_4(x_1 + x_3), x_1x_2 + x_4(x_1 + x_2), x_1x_2 + x_1x_3 + x_2x_3 - 1)^T$, whose exact zero is $\xi = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{-1}{2\sqrt{3}})^T$. The initial estimation in this case is $x^{(0)} = (1, 1, 1, 1)^T$.

| Method | iter | $\ x^{(k+1)} - x^{(k)}\ $ | $\ F(x^{(k+1)})\ $ | ACOC | e-time (sec) |
|--------|------|---------------------------|--------------------|--------|--------------|
| NM | 10 | 0.6502e-582 | 0.5507e-1168 | 2.0041 | 6.6116 |
| JM | 5 | 0.6502e-582 | 0.2697e-2007 | 4.0678 | 2.4154 |
| SHM | 5 | 0.7647e-496 | 0.2772e-1989 | 4.0800 | 4.6017 |
| ABM | 5 | 0.3976e-180 | 0.3538e-727 | 4.0543 | 4.8256 |
| GC1 | 5 | 0.3181e-196 | 0.4482e-792 | 4.0495 | 2.0121 |
| GLe1 | 5 | 0.7647e-495 | 0.2772e-1990 | 4.0800 | 5.1078 |
| GLo2 | 5 | 0.9448e-450 | 0.1162e-1808 | 4.0882 | 4.0882 |
| GR2 | 5 | 0.1845e-469 | 0.1302e-1887 | 4.0844 | 4.9185 |

Table 6: Performance of the methods for F_2 with $x^{(0)} = (1, 1, 1, 1)^T$

| | NM | JM | SHM | ABM |
|-------------------------|------------|------------|------------|------------|
| $\ x^{(1)} - x^{(0)}\ $ | 0.1014e+1 | 0.1425e+1 | 0.1394e+1 | 0.1490e+1 |
| $\ F(x^{(1)})\ $ | 0.1202e+1 | 0.1280 | 0.2003 | 0.1738e-1 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.4213 | 0.6040e-1 | 0.9356e-1 | 0.7961e-2 |
| $\ F(x^{(2)})\ $ | 0.1280 | 0.6784e-7 | 0.1093e-5 | 0.1765e-10 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.5978e-1 | 0.3351e-7 | 0.5394e-6 | 0.8615e-10 |
| $\ F(x^{(3)})\ $ | 0.1280e-2 | 0.9544e-34 | 0.1886e-28 | 0.3450e-43 |
| | GC1 | GLe1 | GLo2 | GR2 |
| $\ x^{(1)} - x^{(0)}\ $ | 0.1014e+1 | 0.1425e+1 | 0.1394e+1 | 0.1490e+1 |
| $\ F(x^{(1)})\ $ | 0.7162e-1 | 0.2003 | 0.2517 | 0.2280 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.3424e-1 | 0.9356e-1 | 0.1168 | 0.1061 |
| $\ F(x^{(2)})\ $ | 0.1218e-10 | 0.1093e-5 | 0.4565e-5 | 0.2448e-5 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.6028e-10 | 0.5394e-6 | 0.2252e-5 | 0.1208e-5 |
| $\ F(x^{(3)})\ $ | 0.2457e-46 | 0.1886e-28 | 0.1033e-25 | 0.6573e-27 |

Table 7: Error estimations for F_2 with $x^{(0)} = (1, 1, 1, 1)^T$

At a glance of Tables 6 and 7 we deduce that, in terms of the error estimation, the best behavior corresponds to GC1 method, followed shortly by ABM. In this case, both methods gets lower precision in their calculations than the rest, but GC1 is in fact the best one in elapsed time of computation.

Problem 3 Let us work now on a three-dimensional function $F_3(x_1, x_2, x_3) = (x_1^2 + x_2^2 + x_3^2 - 9, x_1x_2x_3 - 1, x_1 + x_2 - x_3^2)^T$, whose estimated solution is $\xi \approx (2.140, -2.090, -0.2235)^T$. In all cases, we use $x^{(0)} = (2, -1.5, -0.5)^T$ as an initial estimation.

In Tables 8 and 9 we show that the performance of the new method GC1 is the best, as well in precision of the obtained results as in the elapsed time. The

| Method | iter | $\ x^{(k+1)} - x^{(k)}\ $ | $\ F(x^{(k+1)})\ $ | ACOC | e-time (sec) |
|--------|------|---------------------------|--------------------|--------|--------------|
| NM | 10 | 0.4822e-477 | 0.3078e-954 | 2.0002 | 4.0797 |
| JM | 5 | 0.3163e-476 | 0.2516e-1906 | 4.0009 | 2.9017 |
| SHM | 6 | 0.1125e-283 | 0.8107e-1136 | 3.9999 | 4.9535 |
| ABM | 6 | 0.2985e-222 | 0.1590e-890 | 4.0001 | 4.0185 |
| GC1 | 5 | 0.4387e-551 | 0.2703e-2007 | 3.9896 | 2.4532 |
| GLe1 | 6 | 0.1125e-283 | 0.8107e-1136 | 3.9999 | 5.3051 |
| GLo2 | 6 | 0.4290e-188 | 0.2604e-753 | 3.9999 | 4.6598 |
| GR2 | 6 | 0.4548e-231 | 0.2723e-925 | 3.9999 | 4.9226 |

Table 8: Performance of the methods for F_3 with $x^{(0)} = (2, -1.5, -0.5)^T$

approximated order of convergence ACOC and the number of iterations are the same for all the methods. This scheme is followed by Jarratt's one, and there exist a great difference among this two procedures and the rest. This conduct is stable in the iterative process, as can be deduced comparing both tables.

| | NM | JM | SHM | ABM |
|-------------------------|------------|------------|------------|------------|
| $\ x^{(1)} - x^{(0)}\ $ | 0.9711 | 0.6994 | 0.8155 | 0.4342 |
| $\ F(x^{(1)})\ $ | 0.1307e+1 | 0.1115 | 0.5665 | 0.7801 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.2829 | 0.3669e-1 | 0.1607 | 0.2763 |
| $\ F(x^{(2)})\ $ | 0.1154 | 0.2895e-6 | 0.2338e-3 | 0.2847e-2 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.3702e-1 | 0.8282e-7 | 0.6779e-4 | 0.8422e-3 |
| $\ F(x^{(3)})\ $ | 0.1681e-2 | 0.1347e-28 | 0.1101e-16 | 0.1017e-12 |
| | GC1 | GLe1 | GLo2 | GR2 |
| $\ x^{(1)} - x^{(0)}\ $ | 0.6409 | 0.8155 | 0.1017e+1 | 0.9008 |
| $\ F(x^{(1)})\ $ | 0.1081 | 0.5665 | 0.1433e+1 | 0.9221 |
| $\ x^{(2)} - x^{(1)}\ $ | 0.2839 | 0.1607 | 0.3701 | 0.2503 |
| $\ F(x^{(2)})\ $ | 0.5502e-7 | 0.2338e-3 | 0.6380e-2 | 0.1439e-2 |
| $\ x^{(3)} - x^{(2)}\ $ | 0.1097e-7 | 0.6779e-4 | 0.1842e-2 | 0.4176e-3 |
| $\ F(x^{(3)})\ $ | 0.6796e-33 | 0.1101e-16 | 0.9056e-11 | 0.1982e-13 |

Table 9: Error estimation for F_3 with $x^{(0)} = (2, -1.5, -0.5)^T$

From results in Tables 4 to 9, it can be deduced that, in general, a very good performance of the presented methods is shown. The Gauss-Chebyshev scheme shows to be the best in the tests, in terms of the error and time of computing, but this behavior can depend on the problem to be solved. Let us remark that elapsed times are similar and have a strong dependence of the problem.

4 Conclusions

In this paper, we have presented a general procedure to design fourth-order predictor-corrector methods for solving nonlinear equations or systems, based on any Gaussian

quadrature. For one-dimensional case the obtained methods are optimal in the sense of Kung-Traub conjecture. Numerical tests, for scalar and vectorial functions, have shown the good performance of the proposed methods in terms of error estimation and mean execution times. Specifically, for scalar example, the lowest time corresponds to methods JM and GC1. Moreover, in the analyzed multidimensional cases, the best execution time has been the one of GR2 for Problem 1 and GC1 for Problems 2 and 3.

Acknowledgement *This research was supported by Spanish grant MTM2014-52016-C2-2-P and FONDOCYT 2014-1C1-088 República Dominicana. The authors would like to thank the anonymous reviewers for their helpful suggestions and comments.*

References

- [1] ABAD M.F., CORDERO A., TORREGROSA J.R., Fourth- and Fifth-Order Methods for Solving Nonlinear Systems of Equations: An Application to the Global Positioning System, *Abstract and Applied Analysis*, **2013** Article ID 586708 (2013) 10 pages, doi:10.1155/2013/586708
- [2] CORDERO A., HUESO J.L., MARTÍNEZ E., TORREGROSA J.R., A modified Newton-Jarratt's composition, *Numerical Algorithms* **55** (2010), 87–99
- [3] CORDERO A., TORREGROSA J.R., VASSILEVA M.P., Pseudocomposition: A technique to design predictor-corrector methods for systems of nonlinear equations, *Applied Mathematics and Computation* **218** (2012), 11496–11504
- [4] GHORBANZADEH M., SOLEYMANI F., A Quartically Convergent Jarratt-Type Method for Nonlinear System of Equations, *Algorithms* **8** (2015), 415–423
- [5] JARRATT P., Some fourth order multipoint iterative methods for solving equations, *Math. Comput.* **20** (1966), 434–437
- [6] KUNG H.T., TRAUB J.F., Optimal order of one-point and multi-point iterations, *J. Assoc. Comput. Math.* **21** (1974), 643–651
- [7] OSTROWSKI A.M., *Solution of equations and systems of equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1964
- [8] PETKOVIĆ M., NETA B., PETKOVIĆ L., DŽUNIĆ J., *Multipoint methods for solving nonlinear equations*, Elsevier, Amsterdam, 2013
- [9] SHARMA J.R., ARORA H., Efficient Jarratt-like methods for solving systems of nonlinear equations, *Calcolo* **51** (2014), 193–210
- [10] SHARMA J.R., GUHA R.K., SHARMA R., An efficient fourth order weighted-Newton method for systems of nonlinear equations, *Numerical Algorithms* **62** (2013), 307–323
- [11] TRAUB J.F., *Iterative Methods for the Solution of Equations*, Chelsea Publishing Company, New York, 1982
- [12] WHITE F., *Fluid Mechanics*, McGraw-Hill, Boston 2003
- [13] XIAO X.Y., YIN H.W., Increasing the order of convergence for iterative methods to solve nonlinear systems, *Calcolo* doi:10.1007/s10092-015-0149-9

Received: 02.03.2015

Revised: 20.08.2015

Accepted: 09.11.2015

(a) Instituto de Matemáticas Multidisciplinar,
Universitat Politècnica de València,
46022, Valencia, Spain
E-mail: {acordero,jrtorre}@mat.upv.es

(b) Instituto Tecnológico de Santo Domingo (INTEC),
República Dominicana
E-mail: maria.penkova@intec.edu.do