



**ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA**  
"Antonio Ruberti"  
**CONSIGLIO NAZIONALE DELLE RICERCHE**

**A. Germani, C. Manes, P. Palumbo, M. Sciandrone**

**A HIGHER ORDER METHOD FOR THE  
SOLUTION OF NONLINEAR SCALAR EQUATIONS**

**R. 585 Giugno 2003**

**Alfredo Germani** – Dipartimento di Ingegneria Elettrica, Università degli Studi dell'Aquila, 67040 Monteluco (L'Aquila), Italy and Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy. Email: [germani@ing.univaq.it](mailto:germani@ing.univaq.it).

**Costanzo Manes** – Dipartimento di Ingegneria Elettrica, Università degli Studi dell'Aquila, 67040 Monteluco (L'Aquila), Italy and Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy. Email: [manes@ing.univaq.it](mailto:manes@ing.univaq.it).

**Pasquale Palumbo** – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy. Email: [palumbo@iasi.rm.cnr.it](mailto:palumbo@iasi.rm.cnr.it).

**Marco Sciandrone** – Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, 00185 Roma, Italy. Email: [sciandro@iasi.rm.cnr.it](mailto:sciandro@iasi.rm.cnr.it).

ISSN: 1128–3378

Collana dei Rapporti  
dell'Istituto di Analisi dei Sistemi ed Informatica, CNR  
viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161

fax ++39-06-7716461

email: [iasi@iasi.rm.cnr.it](mailto:iasi@iasi.rm.cnr.it)

URL: <http://www.iasi.rm.cnr.it>

## Abstract

A new iterative method for the approximation of the root of a nonlinear function  $f$  in one variable is proposed. It is based on a suitable polynomial model of order  $n$  derived from the Taylor series expansion around the current point  $x_k$ . At each iteration a linear system of dimension  $n$  has to be solved. It is shown that the coefficient matrix of the linear system is nonsingular if and only if the first derivative of  $f$  at  $x_k$  is not null. Moreover, it is proved that the method is locally convergent with convergence rate greater than or equal to  $n + 1$ . Finally, an easily implementable scheme is provided, in which the solution of the linear system is computed by exploiting specific properties of the coefficient matrix.

*Key words:* Root-finding algorithms, Newton's method, Higher order methods, Rate of convergence.



## 1. Introduction

In this paper it will be considered the problem of solving a nonlinear equation in one variable, that is:

$$\text{given } f : \mathbb{R} \rightarrow \mathbb{R}, \quad \text{find } x^* \in \mathbb{R} \text{ such that } f(x^*) = 0, \quad (1.1)$$

where it is assumed that the nonlinear map  $f$  is of  $C^n$ -class ( $n \geq 1$ ) in its domain, that is it admits at least  $n$ -times continuous derivatives. Newton's method, as well known, consists in constructing a sequence of approximate solutions by considering at each step the zero of the first order Taylor approximation of  $f$  at the current point  $x_k$ , named  $p_1(f; x_k)$ :

$$p_1(f; x_k)(x) = f(x_k) + f^{(1)}(x_k)(x - x_k), \quad (1.2)$$

where  $f^{(1)}(x_k)$  is the first order derivative of  $f$  at  $x_k$  (round brackets at superscript are used to distinguish the order of derivatives from powers). Newton's iteration consists in setting:

$$x_{k+1} = x_k - \frac{1}{f^{(1)}(x_k)} f(x_k). \quad (1.3)$$

It is well-known that Newton's method has local quadratic convergence, provided that  $f^{(1)}(x^*) \neq 0$  and  $f^{(1)}(x)$  is Lipschitz continuous in a neighborhood containing the root  $x^*$ .

In the relevant literature, higher order methods are referred to algorithms having convergence rate greater than two [7]. Some higher order methods have been derived attempting to extend the Newton's idea of solving the linear equation  $p_1(f; x_k) = 0$ , by considering the  $n$ -th order Taylor approximation of the function  $f$  around the current point  $x_k$ , named  $p_n(f; x_k)$ :

$$p_n(f; x_k)(x) = f(x_k) + f^{(1)}(x_k)(x - x_k) + \frac{1}{2!} f^{(2)}(x_k)(x - x_k)^2 + \dots + \frac{1}{n!} f^{(n)}(x_k)(x - x_k)^n, \quad (1.4)$$

where  $f^{(i)}(x)$  is the  $i$ -th derivative of  $f(x)$ . Note that  $p_n(f; x_k)(x) = 0$  not always admits solutions in the real field. Moreover, the solutions can be analytically computed only for  $n \leq 4$ . For these reasons, no higher order method is based on the computation of the exact solution of the polynomial equation  $p_n(f; x_k)(x) = 0$ . Several recursive schemes have been proposed which allow to define usable methods, essentially based on the computation of solutions of linear equations closely related to  $p_n(f; x_k)(x) = 0$ . The most important, to our knowledge, are the classical Halley's [4] and Chebyshev's [6] methods, that are locally convergent with convergence rate greater than or equal to  $n + 1$ . For  $n = 2$ , Halley's iteration takes the form

$$x_{k+1} = x_k - \frac{2f(x_k)f^{(1)}(x_k)}{2(f^{(1)}(x_k))^2 - f^{(2)}(x_k)f(x_k)},$$

and sufficient global convergence conditions for this method can be found in [1] and [5], while the Chebyshev's iteration consists in setting

$$x_{k+1} = x_k - \frac{1}{f^{(1)}(x_k)} f(x_k) - \frac{f^{(2)}(x_k)f^2(x_k)}{(f^{(1)}(x_k))^2}.$$

We note that different methods can generate different basins of attraction, so that the availability of more methods is important both from a theoretical and a practical point of view, and this motivates the current interest in finding new higher order methods (see, e.g., [2] and [3]).

4.

In this work we propose a higher order method based on an *embedded-relaxed* approach. The problem of computing a root of  $f$  is *embedded* in that of determining a solution of the following system:

$$f^i(x) = 0, \quad i = 1, \dots, n, \quad (1.5)$$

which, of course, has the same solutions of the original problem. Then, an  $n$ -degree Taylor expansion of the functions  $f, f^2, \dots, f^n$  around the step  $x_k$  is considered, so that the system (1.5) is approximated by  $p_n(f^i; x_k)(x) = 0, i = 1, \dots, n$ , that is:

$$\begin{bmatrix} f(x_k) \\ \vdots \\ f^n(x_k) \end{bmatrix} + \begin{bmatrix} f^{(1)}(x_k) & \cdots & \frac{f^{(n)}(x_k)}{n!} \\ \vdots & \ddots & \vdots \\ (f^n)^{(1)}(x_k) & \cdots & \frac{(f^n)^{(n)}(x_k)}{n!} \end{bmatrix} \begin{bmatrix} x - x_k \\ \vdots \\ (x - x_k)^n \end{bmatrix} = 0, \quad (1.6)$$

according to the notation:

$$(f^i)^{(j)}(x_k) = (f^i(x))^{(j)} \Big|_{x=x_k}, \quad i, j = 1, \dots, n. \quad (1.7)$$

The novelty of the method here presented is to *relax* the problem described in (1.6) in the following one:

$$\begin{bmatrix} f(x_k) \\ \vdots \\ f^n(x_k) \end{bmatrix} + \begin{bmatrix} f^{(1)}(x_k) & \cdots & \frac{f^{(n)}(x_k)}{n!} \\ \vdots & \ddots & \vdots \\ (f^n)^{(1)}(x_k) & \cdots & \frac{(f^n)^{(n)}(x_k)}{n!} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = 0, \quad (1.8)$$

by the formal substitution  $y_i = (x - x_k)^i$ , i.e. neglecting the constraints relating the variables  $y_i$ , so that the polynomials become affine functions in the new variables. Finally, denoting with  $y_{1,k}$  the value of  $y_1$ , solution of (1.8) at step  $k$ , the next iterate is given by:

$$x_{k+1} = x_k + y_{1,k}. \quad (1.9)$$

We prove that the coefficient matrix of the linear system (1.8) is nonsingular if and only if  $f^{(1)}(x_k) \neq 0$ . This means that the method has the same applicability condition of the standard Newton's method. Moreover, under the usual assumption that  $f^{(1)}(x^*) \neq 0$ , we prove that the algorithm is locally convergent with convergence rate greater than or equal to  $n + 1$ . We note that, for  $n = 2$ , the method coincides with the Chebyshev's method.

The paper is organized as follows: the next section is devoted to introduce some preliminary results concerning the proposed methodology; in section three the higher order algorithm is developed; in section four the main theorem concerning the convergence rate is proved and, finally, in section five an easy implementable scheme is described.

## 2. Preliminary results

In this section some preliminary results concerning Taylor polynomials are reported, which will be used in the sequel for the definition and the convergence analysis of the method proposed in the paper.

Given the polynomials  $\alpha(x) = \sum_{i=0}^n a_i x^i$ ,  $\beta(x) = \sum_{j=0}^n b_j x^j$ , consider the polynomial of degree  $2n$

$$\gamma(x) = \alpha(x)\beta(x) = \sum_{h=0}^n \left( \sum_{k=0}^h a_k b_{h-k} \right) x^h + \sum_{h=n+1}^{2n} \left( \sum_{k=h-n}^n a_k b_{h-k} \right) x^h = \sum_{h=0}^{2n} c_h x^h. \quad (2.1)$$

The computation of the coefficients  $c_h$  can be organized as a matrix product as follows:

$$[c_0 \ c_1 \ \cdots \ c_{2n}] = [a_0 \ a_1 \ \cdots \ a_n] \begin{bmatrix} b_0 & b_1 & \cdots & b_n & 0 & \cdots & 0 \\ 0 & b_0 & \ddots & b_{n-1} & b_n & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & b_0 & \cdots & b_{n-1} & b_n \end{bmatrix}. \quad (2.2)$$

**Lemma 2.1.** *Let  $f, g : \mathbb{R} \mapsto \mathbb{R}$  be  $n$ -continuously differentiable nonlinear maps. Then:*

$$p_n(fg; \bar{x}) = p_n(p_n(f; \bar{x})p_n(g; \bar{x}); \bar{x}). \quad (2.3)$$

**Proof.** Considering the Taylor expansion, according to the notation (1.7),  $f$  and  $g$  have the following representation:

$$f(x) = \sum_{i=0}^n \frac{f^{(i)}(\bar{x})}{i!} (x - \bar{x})^i + R_1(n, \bar{x}, x), \quad g(x) = \sum_{j=0}^n \frac{g^{(j)}(\bar{x})}{j!} (x - \bar{x})^j + R_2(n, \bar{x}, x), \quad (2.4)$$

where

$$\lim_{|x-\bar{x}| \rightarrow 0} \frac{|R_h(n, \bar{x}, x)|}{|x - \bar{x}|^n} = 0, \quad h = 1, 2. \quad (2.5)$$

Using (2.4) we can write

$$f(x)g(x) = \sum_{h=0}^n H_h(\bar{x})(x - \bar{x})^h + R_3(n, \bar{x}, x), \quad (2.6)$$

with

$$H_h(\bar{x}) = \sum_{k=0}^n \frac{f^{(k)}(\bar{x})}{k!} \cdot \frac{g^{(h-k)}(\bar{x})}{(h-k)!}. \quad (2.7)$$

and

$$R_3 = \sum_{h=n+1}^{2n} \tilde{H}_h(\bar{x})(x - \bar{x})^h + R_1 \sum_{j=0}^n \frac{g^{(j)}(\bar{x})}{j!} (x - \bar{x})^j + R_2 \sum_{i=0}^n \frac{f^{(i)}(\bar{x})}{i!} (x - \bar{x})^i + R_1 R_2, \quad (2.8)$$

6.

where

$$\tilde{H}_h(\bar{x}) = \sum_{k=h-n}^n \frac{f^{(k)}(\bar{x})}{k!} \cdot \frac{g^{(h-k)}(\bar{x})}{(h-k)!}. \quad (2.9)$$

Note that  $R_3$  is infinitesimal of the same order of  $R_1$  and  $R_2$ , so that, from (2.6) we get

$$p_n(fg; \bar{x})(x) = \sum_{h=0}^n H_h(\bar{x})(x - \bar{x})^h. \quad (2.10)$$

On the other hand, we have

$$p_n(f; \bar{x})p_n(g; \bar{x}) = \sum_{h=0}^n H_h(\bar{x})(x - \bar{x})^h + \sum_{h=n+1}^{2n} \tilde{H}_h(\bar{x})(x - \bar{x})^h, \quad (2.11)$$

so that, by definition we obtain

$$p_n\left(p_n(f; \bar{x})p_n(g; \bar{x}); \bar{x}\right) = \sum_{h=0}^n H_h(\bar{x})(x - \bar{x})^h. \quad (2.12)$$

Then, recalling (2.10), the thesis is proved. ■

In the sequel, we indicate by  $F_{k,n}(\bar{x})$  the row vector containing the coefficients of  $p_n(f^k; \bar{x})$ , i.e.,

$$F_{k,n}(\bar{x}) = \left[ f^k(\bar{x}) \quad (f^k)^{(1)}(\bar{x}) \quad \dots \quad \frac{1}{n!} (f^k)^{(n)}(\bar{x}) \right]. \quad (2.13)$$

**Lemma 2.2.** *Consider an  $n$ -continuously differentiable function  $f$  in a neighborhood of a given point  $\bar{x}$  and define the following matrix:*

$$A_n(f; \bar{x}) = \begin{bmatrix} f(\bar{x}) & f^{(1)}(\bar{x}) & \frac{f^{(2)}(\bar{x})}{2} & \dots & \frac{f^{(n)}(\bar{x})}{n!} \\ 0 & f(\bar{x}) & f^{(1)}(\bar{x}) & \dots & \frac{f^{(n-1)}(\bar{x})}{(n-1)!} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & f(\bar{x}) & f^{(1)}(\bar{x}) \\ 0 & \dots & \dots & 0 & f(\bar{x}) \end{bmatrix}. \quad (2.14)$$

Then, for any  $k \geq 0$  the following identity holds:

$$F_{k+1,n}(\bar{x}) = F_{k,n}(\bar{x})A_n(f; \bar{x}). \quad (2.15)$$

**Proof.** Taking into account that  $F_{0,n}(\bar{x}) = [1 \ O_{1 \times n}]$ , equation (2.15) is true for  $k = 0$ . Assume  $k > 0$ : by lemma 2.1 we can write

$$p_n(f^{k+1}; \bar{x}) = p_n(f^k f; \bar{x}) = p_n\left(p_n(f^k; \bar{x})p_n(f; \bar{x}); \bar{x}\right). \quad (2.16)$$

By definition,  $p_n\left(p_n(f^k; \bar{x})p_n(f; \bar{x}); \bar{x}\right)$  is the polynomial of degree  $n$  obtained considering the first  $n + 1$  terms of the polynomial  $p_n(f^k; \bar{x})p_n(f; \bar{x})$ , so that, by (2.2), it follows that (2.15) holds. ■

The following lemma will be precious in the proof of the convergence of the method and in the derivation of an efficient implementation.

**Lemma 2.3.** *Consider an  $n$ -continuously differentiable function  $f$  in a neighborhood of a given point  $\bar{x}$ , so that the following matrix is defined:*

$$Q_n(f; \bar{x}) = \begin{bmatrix} F_{0,n}(\bar{x}) \\ F_{1,n}(\bar{x}) \\ \vdots \\ F_{n,n}(\bar{x}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ f(\bar{x}) & f^{(1)}(\bar{x}) & \cdots & \frac{f^{(n)}(\bar{x})}{n!} \\ \vdots & \vdots & \ddots & \vdots \\ f^n(\bar{x}) & (f^n)^{(1)}(\bar{x}) & \cdots & \frac{(f^n)^{(n)}(\bar{x})}{n!} \end{bmatrix}. \quad (2.17)$$

Then, it results:

$$Q_n(f; \bar{x}) = \begin{bmatrix} C_n \\ C_n A_n(f; \bar{x}) \\ \vdots \\ C_n A_n^n(f; \bar{x}) \end{bmatrix} \quad \text{with} \quad C_n = [1 \quad O_{1 \times n}], \quad (2.18)$$

and the matrix  $Q_n(f; \bar{x})$  admits the following decomposition:

$$Q_n(f; \bar{x}) = L_n(f; \bar{x}) U_n(f; \bar{x}), \quad (2.19)$$

where  $L_n(f; \bar{x})$  is the lower triangular matrix defined as:

$$[L_n(f; \bar{x})]_{i,j} = \begin{cases} 0, & i < j, \\ \binom{i-1}{j-1} f^{i-j}(\bar{x}), & i \geq j, \end{cases} \quad i, j = 1, \dots, n+1, \quad (2.20)$$

and  $U_n(f; \bar{x})$  is the upper triangular matrix defined as:

$$U_n(f; \bar{x}) = \begin{bmatrix} C_n \\ C_n (A_n(f; \bar{x}) - f(\bar{x}) I_{n+1}) \\ \vdots \\ C_n (A_n(f; \bar{x}) - f(\bar{x}) I_{n+1})^n \end{bmatrix}. \quad (2.21)$$

Moreover the determinant of  $Q_n(f; \bar{x})$  is:

$$\det Q_n(f; \bar{x}) = \left( f^{(1)}(\bar{x}) \right)^{\frac{n(n+1)}{2}}. \quad (2.22)$$

**Proof.** Equation (2.18) is a straightforward consequence of lemma 2.2. According to (2.14), matrix  $A_n(f; \bar{x})$  has the eigenvalue  $f(\bar{x})$  with  $n+1$  multiplicity, that means there exists a similitude transformation of matrix  $A_n(f; \bar{x})$  in the Jordan form [8]. Then a nonsingular matrix  $U_n(f; \bar{x})$  is obtained, whose inverse is given by the generalized eigenvectors of  $A_n(f; \bar{x})$ , so that:

$$\tilde{A}_n(f; \bar{x}) = U_n(f; \bar{x}) A_n(f; \bar{x}) U_n^{-1}(f; \bar{x}) = \begin{bmatrix} f(\bar{x}) & 1 & 0 & \cdots & 0 \\ 0 & f(\bar{x}) & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & f(\bar{x}) & 1 \\ 0 & \cdots & \cdots & 0 & f(\bar{x}) \end{bmatrix}. \quad (2.23)$$

It will be proved that such a matrix  $U_n(f; \bar{x})$  is the one of (2.21). By naming

$$\tilde{C}_n(f; \bar{x}) = C_n U_n^{-1}(f; \bar{x}), \quad L_n(f; \bar{x}) = \begin{bmatrix} \tilde{C}_n(f; \bar{x}) \\ \tilde{C}_n(f; \bar{x}) \tilde{A}_n(f; \bar{x}) \\ \vdots \\ \tilde{C}_n(f; \bar{x}) \tilde{A}_n^n(f; \bar{x}) \end{bmatrix}, \quad (2.24)$$

it results:

$$Q_n(f; \bar{x}) = L_n(f; \bar{x}) U_n(f; \bar{x}). \quad (2.25)$$

Such  $U_n(f; \bar{x})$  can be found solving the equation  $U_n(f; \bar{x}) A_n(f; \bar{x}) = \tilde{A}_n(f; \bar{x}) U_n(f; \bar{x})$ . Subtracting the term  $f(\bar{x}) U_n(f; \bar{x})$  from both sides we get  $U_n(f; \bar{x}) A_n(f; \bar{x}) - f(\bar{x}) U_n(f; \bar{x}) = \tilde{A}_n(f; \bar{x}) U_n(f; \bar{x}) - f(\bar{x}) U_n(f; \bar{x})$ , that can be rewritten as

$$U_n(f; \bar{x}) (A_n(f; \bar{x}) - f(\bar{x}) I_{n+1}) = (\tilde{A}_n(f; \bar{x}) - f(\bar{x}) I_{n+1}) U_n(f; \bar{x}). \quad (2.26)$$

Exploiting the particular structure of matrices  $A_n(f; \bar{x}) - f(\bar{x}) I_{n+1}$  and  $\tilde{A}_n(f; \bar{x}) - f(\bar{x}) I_{n+1}$  (note that both of them are nilpotent), it can be checked that matrix  $U_n(f; \bar{x})$  given by (2.21) solves the identity (2.26) (note that  $(A_n(f; \bar{x}) - f(\bar{x}) I_{n+1})^{n+1} = 0$ ). Moreover, the explicit computation of (2.21) gives back an upper triangular matrix with the first row equal to  $C_n$ , and the diagonal elements given by:

$$[U_n(f; \bar{x})]_{i,i} = (f^{(1)})^{i-1}(\bar{x}), \quad i = 1, \dots, n+1. \quad (2.27)$$

According to its form, also the first row of  $U_n^{-1}(f; \bar{x})$  is equal to  $C_n$ , so that, owing to (2.24) it comes that  $\tilde{C}_n(f; \bar{x}) = C_n$ . Then, by computation, matrix  $L_n(f; \bar{x})$  is given by:

$$L_n(f; \bar{x}) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ f(\bar{x}) & 1 & 0 & \cdots & 0 \\ f^2(\bar{x}) & 2f(\bar{x}) & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ f^n(\bar{x}) & \binom{n}{1} f^{n-1}(\bar{x}) & \binom{n}{2} f^{n-2}(\bar{x}) & \cdots & 1 \end{bmatrix}, \quad (2.28)$$

that is, formally, (2.20). The determinant in (2.22) comes by taking into account the diagonal elements of the matrices involved in the lower and upper decomposition of  $Q_n(f; \bar{x})$ . According to the Binet theorem:

$$\det Q_n(f; \bar{x}) = \det L_n(f; \bar{x}) \cdot \det U_n(f; \bar{x}), \quad (2.29)$$

with  $\det L_n(f; \bar{x}) = 1$  (see (2.28)) and:

$$\det U_n(f; \bar{x}) = \prod_{i=1}^n (f^{(1)}(\bar{x}))^i = (f^{(1)}(\bar{x}))^{\sum_{i=1}^n i} = (f^{(1)}(\bar{x}))^{\frac{n(n+1)}{2}}. \quad (2.30)$$

■

**Remark 2.4.** The result of lemma 2.3 is inspired by the theory of linear dynamic systems [8]. Matrix  $Q_n(f; x)$  is the *observability matrix* of the pair  $(A_n(f; x), C_n)$ , the lower triangular matrix  $L_n(f; x)$  is the representation of  $Q_n(f; x)$  in Jordan coordinates, and  $U_n(f; x)$  is the matrix that operates the change of coordinates.

### 3. The higher order algorithm

As previously mentioned in the introduction, the proposed higher order method is based at each step  $k$  on considering the  $n$  degree Taylor polynomials at  $x_k$  associated to the first  $n$  powers of the nonlinear map  $f$ . In order to determine the new iterate  $x_{k+1}$ , a first attempt could be that of setting to zero the considered polynomials, that is

$$\begin{bmatrix} f(x_k) \\ \vdots \\ f^n(x_k) \end{bmatrix} + \begin{bmatrix} f^{(1)}(x_k) & \cdots & \frac{f^{(n)}(x_k)}{n!} \\ \vdots & \ddots & \vdots \\ (f^n)^{(1)}(x_k) & \cdots & \frac{(f^n)^{(n)}(x_k)}{n!} \end{bmatrix} \begin{bmatrix} x - x_k \\ \vdots \\ (x - x_k)^n \end{bmatrix} = 0. \quad (3.1)$$

Unless trivial cases, (3.1) does not admit solution with respect to the indeterminate  $x - x_k$ . Equation (3.1) is equivalent to the following constrained linear system:

$$\bar{f}_n(x_k) + \mathcal{F}_n(x_k) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = 0, \quad y_i = (x - x_k)^i, \quad i = 1, \dots, n, \quad (3.2)$$

where

$$\bar{f}_n(x_k) = \begin{bmatrix} f(x_k) \\ \vdots \\ f^n(x_k) \end{bmatrix}, \quad \mathcal{F}_n(x_k) = \begin{bmatrix} f^{(1)}(x_k) & \cdots & \frac{f^{(n)}(x_k)}{n!} \\ \vdots & \ddots & \vdots \\ (f^n)^{(1)}(x_k) & \cdots & \frac{(f^n)^{(n)}(x_k)}{n!} \end{bmatrix}. \quad (3.3)$$

The idea of the proposed algorithm is *to relax* the nonlinear constraints on the variables  $y_i$  and to solve the linear system

$$\bar{f}_n(x_k) + \mathcal{F}_n(x_k) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = 0, \quad (3.4)$$

provided that it admits solution. By naming  $y_{1,k}$  the solution associated to the indeterminate  $y_1$ , at step  $k$ , the next iterate of the method is given by:

$$x_{k+1} = x_k + y_{1,k}. \quad (3.5)$$

Note that the method reduces to Newton's algorithm if  $n = 1$  and to Chebyshev's if  $n = 2$ . By using the definitions of  $\bar{f}_n(x_k)$  and  $\mathcal{F}_n(x_k)$  given in (3.3), the matrix  $Q_n(x_k)$  of lemma 2.3 can be decomposed as follows:

$$Q_n(f; x_k) = \begin{bmatrix} 1 & 0 \\ \bar{f}_n(x_k) & \mathcal{F}_n(x_k) \end{bmatrix}, \quad (3.6)$$

so that, according to (2.22):

$$\det \mathcal{F}_n(x_k) = \det Q_n(f; x_k) = \left( f^{(1)}(x_k) \right)^{\frac{n(n+1)}{2}}. \quad (3.7)$$

Therefore, the proposed algorithm has the same applicability of Newton's in the class of  $n$ -continuously differentiable functions. Indeed, a sufficient condition to guarantee that (3.4) admits a solution is that the matrix  $\mathcal{F}_n(x_k)$  is nonsingular, which is equivalent to require, according to (3.7), that the first derivative is nonzero in the current point  $x_k$ .

In the following section it will be shown the local convergence properties and the convergence rate of the algorithm.

#### 4. Convergence results

This section is devoted to prove that the proposed higher order method achieves the goal of finding a root of a nonlinear map with a convergence rate increasing with the order  $n$  of the Taylor expansion. This is shown by using the iterative function coming from (3.5), that is:

$$\Phi_n(x) = x - [1 \ O_{1 \times (n-1)}] \mathcal{F}_n^{-1}(x) \bar{f}_n(x). \quad (4.1)$$

**Theorem 4.1.** *Consider the algorithm described by the iterative function  $\Phi_n(x)$  in (4.1) for an  $(n+1)$ -continuously differentiable function  $f$ , with  $f^{(1)} \neq 0$  on an open neighborhood containing  $x^*$ , with  $x^*$  such that  $f(x^*) = 0$ . Then  $x^*$  is a point of attraction of the algorithm, whose convergence rate is at least  $n+1$ . Moreover, the asymptotic error constant is given by:*

$$\left| \frac{\Phi_n^{(n+1)}(x^*)}{(n+1)!} \right| = \frac{|\phi_n(x^*)|}{n+1}, \quad \phi_n(x) = \frac{1}{n!} [1 \ O_{1 \times (n-1)}] \mathcal{F}_n^{-1}(x) \bar{f}_n^{(n+1)}(x). \quad (4.2)$$

**Proof.** The proof is based on the well-known result (see, e.g., [7]), for which the algorithm locally converges to  $x^*$ , with convergence rate at least  $n+1$ , if and only if the iterative function  $\Phi_n(x)$  defined in (4.1) is at least of order  $n+1$ , that is:

$$\begin{aligned} \Phi_n(x^*) &= x^*, \\ \Phi_n^{(i)}(x^*) &= 0, \quad 1 \leq i \leq n. \end{aligned} \quad (4.3)$$

According to its definition,  $f$  and all its powers vanish in  $x^*$ , so that  $\bar{f}_n(x^*) = 0$  and therefore

$$\Phi_n(x^*) = x^* - [1 \ O_{1 \times (n-1)}] \mathcal{F}_n^{-1}(x^*) \bar{f}_n(x^*) = x^*. \quad (4.4)$$

The theorem is proved verifying that:

$$\begin{aligned} \Phi_n^{(i)}(x) &= [O_{1 \times (n-i)} \ \psi_{i,n-i+1}(x) \ \cdots \ \psi_{i,n}(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n(x), \quad i = 1, \dots, n-1, \\ \Phi_n^{(n)}(x) &= [\psi_{n,1}(x) \ \psi_{n,2}(x) \ \cdots \ \psi_{n,n}(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n(x), \end{aligned} \quad (4.5)$$

with

$$\psi_{i,n-i+1}(x) = (-1)^{i-1} \frac{n!}{(n-i+1)!} \phi_n(x), \quad i = 1, 2, \dots, n, \quad (4.6)$$

where  $\phi_n(x)$  is defined in (4.2). The functions  $\psi_{i,j}(x)$ ,  $j = n-i+1, \dots, n$  are suitably defined functions: the first index in  $\psi_{i,j}$  identifies the derivative order of the iterative function to which the term belongs, while the second is for the position in the row vector. Note that definition (4.6) gives only the first nonzero element of the row in (4.5). Equations (4.5) and (4.6) are proved by induction. For the development of the proof, it is important to point out the following structure of the coefficient matrix  $\mathcal{F}_n(x)$ :

$$\mathcal{F}_n(x) = \left[ \bar{f}_n^{(1)}(x) \ \frac{1}{2!} \bar{f}_n^{(2)}(x) \ \cdots \ \frac{1}{n!} \bar{f}_n^{(n)}(x) \right]. \quad (4.7)$$

as it can be verified looking at the definitions (3.3).

Let  $i = 1$ . Then, taking into account the first derivative of the iterative function it follows

$$\Phi_n^{(1)}(x) = 1 - [1 \ O_{1 \times (n-1)}] \left\{ (\mathcal{F}_n^{-1}(x))^{(1)} \bar{f}_n(x) + \mathcal{F}_n^{-1}(x) \bar{f}_n^{(1)}(x) \right\}. \quad (4.8)$$

Owing to the structure (4.7) it easily comes that:

$$\mathcal{F}_n^{-1}(x) \bar{f}_n^{(1)}(x) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (4.9)$$

in that  $\bar{f}_n^{(1)}(x)$  is the first column of matrix  $\mathcal{F}_n$ . Then (4.8) becomes:

$$\Phi_n^{(1)}(x) = -[1 \ O_{1 \times (n-1)}] (\mathcal{F}_n^{-1}(x))^{(1)} \bar{f}_n(x). \quad (4.10)$$

As regards the matrix  $(\mathcal{F}_n^{-1}(x))^{(1)}$ , we have:

$$\frac{d}{dx} [\mathcal{F}_n \mathcal{F}_n^{-1}] = \mathcal{F}_n^{(1)} \mathcal{F}_n^{-1} + \mathcal{F}_n (\mathcal{F}_n^{-1})^{(1)} = 0 \quad \Rightarrow \quad (\mathcal{F}_n^{-1})^{(1)} = -\mathcal{F}_n^{-1} \mathcal{F}_n^{(1)} \mathcal{F}_n^{-1}, \quad (4.11)$$

from which the first derivative in (4.10) becomes:

$$\Phi_n^{(1)}(x) = [1 \ O_{1 \times (n-1)}] \left( \mathcal{F}_n^{-1}(x) \mathcal{F}_n^{(1)}(x) \right) \mathcal{F}_n^{-1}(x) \bar{f}_n(x). \quad (4.12)$$

On the other hand, from (4.7) we have:

$$\begin{aligned} \mathcal{F}_n^{(1)}(x) &= \left[ \bar{f}_n^{(2)}(x) \quad \frac{1}{2!} \bar{f}_n^{(3)}(x) \quad \dots \quad \frac{1}{n!} \bar{f}_n^{(n+1)}(x) \right] \\ &= \mathcal{F}_n(x) \begin{bmatrix} 0 & \dots & \dots & 0 & 0 \\ 2 & \ddots & & \vdots & 0 \\ 0 & 3 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & n & 0 \end{bmatrix} + \frac{1}{n!} \bar{f}_n^{(n+1)}(x) [O_{1 \times (n-1)} \ 1] \end{aligned} \quad (4.13)$$

which implies

$$\mathcal{F}_n^{-1}(x) \mathcal{F}_n^{(1)}(x) = \begin{bmatrix} 0 & \dots & \dots & 0 & 0 \\ 2 & \ddots & & \vdots & 0 \\ 0 & 3 & \ddots & \vdots & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \dots & 0 & n & 0 \end{bmatrix} + \frac{1}{n!} \mathcal{F}_n^{-1}(x) \bar{f}_n^{(n+1)}(x) [O_{1 \times (n-1)} \ 1] \quad (4.14)$$

Substitution of (4.14) in (4.12), recalling definition (4.2), yields

$$\Phi_n^{(1)}(x) = [O_{1 \times (n-1)} \ \phi_n(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n(x), \quad (4.15)$$

that is equation (4.5) for  $i = 1$ , and this proves the first step of the induction.

12.

Now, let (4.5) and (4.6) be true for a given  $1 \leq i < n$ . Then:

$$\begin{aligned} \Phi_n^{(i+1)}(x) &= [O_{1 \times (n-i)} \ \psi_{i,n-i+1}^{(1)}(x) \ \cdots \ \psi_{i,n}^{(1)}(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n(x) \\ &\quad + [O_{1 \times (n-i)} \ \psi_{i,n-i+1}(x) \ \cdots \ \psi_{i,n}(x)] (\mathcal{F}_n^{-1}(x))^{(1)} \bar{f}_n(x) \\ &\quad + [O_{1 \times (n-i)} \ \psi_{i,n-i+1}(x) \ \cdots \ \psi_{i,n}(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n^{(1)}(x). \end{aligned} \quad (4.16)$$

The last term is equal to zero, according to (4.9) and to the fact that  $i < n$ . By using (4.11), the second term in (4.16) becomes:

$$- [O_{1 \times (n-i)} \ \psi_{i,n-i+1}(x) \ \cdots \ \psi_{i,n}(x)] \left( \mathcal{F}_n^{-1}(x) \mathcal{F}_n^{(1)}(x) \right) \mathcal{F}_n^{-1}(x) \bar{f}_n. \quad (4.17)$$

Substitution of (4.14) in (4.17) gives

$$\begin{aligned} &- [O_{1 \times (n-i)} \ \psi_{i,n-i+1}(x) \ \cdots \ \psi_{i,n}(x)] \begin{bmatrix} 0 & \cdots & \cdots & 0 & * \\ 2 & \ddots & & \vdots & * \\ 0 & 3 & \ddots & \vdots & * \\ \vdots & \ddots & \ddots & 0 & * \\ 0 & \cdots & 0 & n & * \end{bmatrix} \mathcal{F}_n^{-1}(x) \bar{f}_n(x) \\ &= - [O_{1 \times (n-i-1)} \ (n-i+1)\psi_{i,n-i+1}(x) \ * \ \cdots \ *] \mathcal{F}_n^{-1}(x) \bar{f}_n(x), \end{aligned} \quad (4.18)$$

with the asterisks standing for generic elements whose explicit evaluation is not relevant for the proof. Note that the scalar  $(n-i+1)\psi_{i,n-i+1}(x)$  in (4.18) occupies the  $(n-i)$ -th position in the row, so that substituting the second term of (4.16) with (4.18), equation (4.5) comes out, written for  $i+1$ , with:

$$\psi_{i+1,n-i}(x) = -(n-i+1)\psi_{i,n-i+1}(x) = (-1)^i n(n-1) \cdots (n-i+2)(n-i+1)\phi_n(x), \quad (4.19)$$

which proves equation (4.6). From (4.5), equation (4.3) is clearly verified, for  $1 \leq i \leq n$ . Taking into account (4.5), for  $i = n$  we can write:

$$\Phi_n^{(n)}(x) = [\psi_{n,1}(x) \ \cdots \ \psi_{n,n}(x)] \mathcal{F}_n^{-1}(x) \bar{f}_n(x), \quad (4.20)$$

from which it follows

$$\begin{aligned} \Phi_n^{(n+1)}(x) &= [\psi_{n,1}(x)^{(1)} \ \cdots \ \psi_{n,n}(x)^{(1)}] \mathcal{F}_n^{-1}(x) \bar{f}_n(x) \\ &\quad + [\psi_{n,1}(x) \ \cdots \ \psi_{n,n}(x)] \left( (\mathcal{F}_n^{-1}(x))^{(1)} \bar{f}_n(x) + \mathcal{F}_n^{-1}(x) \bar{f}_n^{(1)}(x) \right). \end{aligned} \quad (4.21)$$

Evaluating (4.21) in  $x^*$  and considering (4.9), it comes:

$$\Phi_n^{(n+1)}(x^*) = [\psi_{n,1}(x^*) \ \cdots \ \psi_{n,n}(x^*)] \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \psi_{n,1}(x^*),$$

so that, recalling (4.6), we get  $\psi_{n,1}(x^*) = (-1)^{n-1} n! \phi_n(x^*)$ , that is equation (4.2). ■

## 5. Implementation Issues

In this section it is shown that the proposed algorithm can be easily implemented suitably taking into account the particular structure of matrices involved in the computations.

Observe that the application of the algorithm requires at each step  $k$  the solution of the linear problem described by (3.4), which is here reported:

$$\mathcal{F}_n(x_k) \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = -\bar{f}_n(x_k). \quad (5.1)$$

By naming  $y_k = (y_{1,k} \cdots y_{n,k})^T$  the solution of the above system, the component  $y_{1,k}$  updates the step:  $x_{k+1} = x_k + y_{1,k}$ . Below a recursive computation scheme for computing  $y_{1,k}$  is derived. Note that

$$\begin{bmatrix} 1 & 0_{1 \times n} \\ \bar{f}_n(x_k) & \mathcal{F}_n(x_k) \end{bmatrix} \begin{bmatrix} 1 \\ y_{1,k} \\ \vdots \\ y_{n,k} \end{bmatrix} = \begin{bmatrix} 1 \\ 0_{n \times 1} \end{bmatrix}, \quad (5.3)$$

and hence, recalling (3.6), at each step  $k$  of the algorithm it is required to compute the vector  $Y_k = (1 \ y_{1,k} \cdots y_{n,k})^T$  such that

$$Q_n(f; x_k) Y_k = \begin{bmatrix} 1 \\ 0_{n \times 1} \end{bmatrix}. \quad (5.2)$$

This is made to exploit the result of lemma 2.3, which states that  $Q_n(f; x_k)$  can be decomposed as the product of the lower and upper triangular matrices  $L_n(f; x_k)$ ,  $U_n(f; x_k)$ . It follows that the problem (5.2) can be decomposed into two simpler problems, and the solution can be found as follows:

Step 1. Solve for  $W$  the linear system

$$L_n(f; x_k) W = \begin{bmatrix} 1 \\ 0_{n \times 1} \end{bmatrix}. \quad (5.4)$$

Step 2. Solve for  $Y$  the linear system

$$U_n(f; x_k) Y = W_k, \quad \text{with } W_k \text{ solution of (5.4)}. \quad (5.5)$$

Taking into account the structure of  $L_n(f; x_k)$ , the explicit solution of (5.4) is given by

$$W_k = [1 \ -f(x_k) \ \cdots \ (-f(x_k))^n]^T, \quad (5.6)$$

i.e., naming  $W_k(j)$  the  $j$ -th element of vector  $W_k$ , it is:

$$W_k(j) = (-f(x_k))^{j-1}, \quad j = 1, \dots, n+1. \quad (5.7)$$

14.

By naming  $[L_n(f; x_k)]_j$  the  $j$ -th row of matrix  $L_n$ ,  $j = 1, \dots, n+1$  the  $j$ -th row of matrix  $L_n$ , the structure of the solution  $W_k$  can be easily verified by checking that  $[L_n(f; x_k)]_1 W_k = 1$  and for  $j > 1$ :

$$\begin{aligned} [L_n(f; x_k)]_j W_k &= \sum_{i=1}^{n+1} [L_n(f; x_k)]_{j,i} W_k(i) = \sum_{i=1}^j \binom{j-1}{i-1} f^{j-i}(x_k) (-f(x_k))^{i-1} \\ &= (f(x_k) - f(x_k))^{j-1} = 0. \end{aligned} \quad (5.8)$$

The solution of (5.5) can be recursively computed by exploiting the triangular structure of matrix  $U_n(f; x_k)$ . To this aim, let us explicit the computation of the terms  $[U_n(f; x_k)]_{i,j}$ . First, note that:

$$(A_n(f; x_k) - f(x_k)I_{n+1})_{i,j} = \begin{cases} 0 & i \geq j, \\ \frac{f^{(j-i)}(x_k)}{(j-i)!} & i < j, \end{cases} \quad i, j = 1, \dots, n+1. \quad (5.9)$$

From this and from (2.21), the computation of the elements of  $U_n$  is recursively derived for  $i \leq j$ :

$$\begin{aligned} [U_n(f; x_k)]_{1,1} &= 1, \\ [U_n(f; x_k)]_{i,j} &= \sum_{h=i-1}^{j-1} [U_n(f; x_k)]_{i-1,h} \cdot \frac{f^{(j-h)}(x_k)}{(j-h)!}. \end{aligned} \quad (5.10)$$

Exploiting the triangular structure of  $U_n(f; x_k)$ , the problem (5.5) is recursively solved as follows

$$\begin{aligned} y_{n,k} &= \frac{(-f(x_k))^n}{[U_n(f; x_k)]_{n+1,n+1}} \\ y_{i,k} &= \frac{1}{[U_n(f; x_k)]_{i+1,i+1}} \left( (-f(x_k))^i - \sum_{j=i+2}^{n+1} [U_n(f; x_k)]_{i+1,j} y_{j-1,k} \right) \end{aligned}$$

where  $i$  goes from  $n-1$  to 1. Note that  $y_{0,k} = 1$  and that  $y_{1,k}$  updates the step of the algorithm.

Summarizing, the steps of the algorithm can be put in the following form:

```

0. Choose a starting point  $x_0$  and set  $k = 0$ ;
1.  $U = O_{(n+1) \times (n+1)}$ ,  $[U]_{1,1} = 1$ ;
2. for  $i = 1$  to  $n + 1$ 
    $a_i = \frac{f^{(i-1)}(x_k)}{(i-1)!}$ ;  $b_i = (-f(x_k))^{i-1}$ ;
   end_for
3. for  $i = 2$  to  $n + 1$ ,
   for  $j = i$  to  $n + 1$ ,
      $[U]_{i,j} = \sum_{h=i-1}^{j-1} [U]_{i-1,h} a_{j-h+1}$ ;
   end_for
   end_for
4.  $y_{n,k} = \frac{b_{n+1}}{[U]_{n+1,n+1}}$ ;
5. for  $h = 1$  to  $n - 1$ ,
    $i = n - h$ ;
    $y_{i,k} = \frac{1}{[U]_{i+1,i+1}} \left( b_{i+1} - \sum_{j=i+2}^{n+1} [U]_{i+1,j} y_{j-1,k} \right)$ ;
   end_for
6.  $x_{k+1} = x_k + y_{1,k}$ ;
7.  $k = k + 1$ ; goto 1

```

## 6. Conclusions

In this work a new methodology for finding the roots of a nonlinear function has been proposed. The algorithm is based on the Taylor expansion of a given order  $n$  of the function and of its powers up to the same order  $n$ . As required by the Newton's algorithm, for the application of the method it is necessary that at each step the first derivative of the function is not zero. Increasing the order of the Taylor approximation, a faster algorithm is achieved: it is proved that the  $n$ -order algorithm has a convergence rate of order at least  $n + 1$ . Future research will be devoted to the study of the application of the proposed method to the problem of solving systems of nonlinear equations.

## References

- [1] J.A. Ezquerro and M.A. Hernandez, *On a convex acceleration of Newton's method*, Journ. Opt. Th. Appl., Vol. 100, pp. 311–326, 1999.
- [2] M. Frontini and E. Sormani, *Some variants of Newton's method with third-order convergence*, Appl. Math. Comp., Vol. 140, pp. 419–426, 2003.
- [3] J. Gerlach, *Accelerated convergence in Newton's method*, SIAM Rev., Vol. 2, pp. 272–276, 1994.
- [4] E. Halley, *A new and general method of finding the root of equations*, Philos. Trans. Roy. Soc. London, Vol. 18, pp. 136–194, 1694.
- [5] M.A. Hernandez, *A note on Halley's method*, Num. Math., Vol. 59, pp. 273–276, 1991.
- [6] E. Schröder, *Über unendlich viele Algorithmen zur Auflösung der Gleichungen*, (On infinitely many algorithms for solving equations), Math. Ann., Vol. 2, pp. 317–365, 1870.
- [7] J. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, N.J., 1964.
- [8] L.A. Zadeh and C.A. Desoer, *Linear Systems Theory: a State Space Approach*, Mcraw-Hill, New York, 1963.