An algorithm for solving some nonlinear systems with applications to extremum problems

Anca Ciurte[1], Sergiu Nedevschi[2,*] and Ioan Rasa[3]
[1,2] Department of Computer Science,Technical University of Cluj–Napoca
[3] Department of Mathematics,Technical University of Cluj–Napoca
Email: Anca.Ciurte@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro, Ioan.Rasa@math.utcluj.ro

Abstract

We consider a class of nonlinear systems for which a positive solution exists and is unique. Such systems appear quite naturally in several applications concerning difference equations. Moreover, certain extremum problems can be reduced to solving these systems. In order to solve such problems we develop a quasi-Newton algorithm which is very efficient just because the existence and uniqueness of the solution are guaranteed. Several numerical examples illustrate the general results.

2000 Mathematics Subject Classification: 65H10, 49M15, 65K99.

Keywords: nonlinear system, unique solution, quasi-Newton algorithm, extremum problems.

1 Introduction

Consider the system

$$(S_f)\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = f(x_1) \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = f(x_2) \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = f(x_n) \\ x_1 > 0, x_2 > 0, \dots, x_n > 0, \end{cases}$$

*Corresponding author:
Sergiu NEDEVSCHI
Address: 15 Constantin Daicoviciu Street, 400020 Cluj - Napoca, Romania
Tel. +40 264 401200, 401248; Tel./Fax +40 264 592055.

where $a_{ij} > 0$, $i, j = 1, \dots, n$, and $f : (0, +\infty) \to (0, +\infty)$ is a continuous function.

It appears quite naturally in several applications related to
- second, third and fourth order difference equations;
- three-point boundary value problems;
- Dirichlet problems for partial difference equations;
- periodic solutions for difference equations;
- numerical solutions for differential equations;
- steady states of complex dynamical networks.

For details see [7], where even more general systems are considered. Deep results concerning the iteration methods for weakly nonlinear systems $Ax = \Phi(x)$ can be found in [1] and the references therein; in [1] A is a complex matrix and $\Phi : \mathbb{D} \subset \mathbb{C}^n \to \mathbb{C}^n$ is a continuously differentiable function defined on a domain $\mathbb{D}$, all of them subject to suitable assumptions.

When $f(x) = x^q$ for some given $q \in \mathbb{R}$, we use the notation (S_q) instead of (S_f). It was proved in [7, Theorem 3.1] that for $q > 1$ the system (S_q) has a unique solution.

In Section 2 we present sufficient conditions on f guaranteeing the existence and uniqueness of the solution to S_f. These conditions are satisfied, in particular, by $f(x) = x^q$ for $q \in (-\infty, -1] \cup (1, +\infty)$.

A quasi-Newton algorithm for solving an extremum problem associated with (S_f) is described in Section 3. It is very efficient, when the solution exists and is unique, for example under the assumptions of Theorems 1 and 2. This *existence and uniqueness* property is really important: as G. J. McLachlan and Th. Krishnan say on page 90 of their book [5], who knows what pitfalls there may be when the algorithm is used in more complicated settings where multiple extremum points are present.

The Newton algorithm corresponding to our special setting is also described and compared with the quasi-Newton algorithm. For both of them the general convergence theorems can be applied. As proved in Section 2, our special setting has an essential feature: existence and uniqueness of the solution, and usually this particularity leads to better results.

Extremum problems which lead naturally to (S_f) are presented in Sections 3 and 4.

Numerical examples can be found in Section 5.

2 Existence and uniqueness

Let

$$s_i := a_{i1} + \cdots + a_{in}, \quad i = 1, \dots, n,$$

$$\mu := \min_i s_i, \quad \nu := \max_i s_i.$$

Theorem 1. *Let $g : (0, +\infty) \to (0, +\infty)$ be continuous and strictly increasing. Suppose there exist $z_1 > 0$, $z_2 > 0$ such that $g(z_1) = \mu$, $g(z_2) = \nu$. Let $f(x) = xg(x)$, $x > 0$. Then (S_f) has a unique solution.*

Proof. Since f is strictly increasing, (S_f) can be written under the form

$$\begin{cases} a_{11}f^{-1}(y_1) + \cdots + a_{1n}f^{-1}(y_n) = y_1 \\ \ldots \\ a_{n1}f^{-1}(y_1) + \cdots + a_{nn}f^{-1}(y_n) = y_n \end{cases}$$

where $y_i = f(x_i)$, $i = 1, \ldots, n$.

Let $m := \mu z_1$, $M := \nu z_2$. Then $0 < m \leq M$, $f(z_1) = m$, $f(z_2) = M$. Consider the set

$$K := \{y \in \mathbb{R}^n : m \leq y_j \leq M, j = 1, \ldots, n\}, \tag{1}$$

and the function $F : K \to \mathbb{R}^n$,

$$F(y) := \left(\sum_{j=1}^{n} a_{1j} f^{-1}(y_j), \ldots, \sum_{j=1}^{n} a_{nj} f^{-1}(y_j) \right), \quad y \in K. \tag{2}$$

Let $y \in K$. Then

$$f^{-1}(m) \leq f^{-1}(y_j) \leq f^{-1}(M), \quad j = 1, \ldots, n,$$

which entails

$$m = \mu z_1 = \mu f^{-1}(m) \leq s_i f^{-1}(m) \leq \sum_{j=1}^{n} a_{ij} f^{-1}(y_j) \leq \\ \leq s_i f^{-1}(M) \leq \nu f^{-1}(M) = \nu z_2 = M$$

for each $i = 1, \ldots, n$. So we have

$$m \leq \sum_{j=1}^{n} a_{ij} f^{-1}(y_j) \leq M, \quad i = 1, \ldots, n,$$

which means that $F(y) \in K$.

Summing-up, K is compact and convex, F is continuous, and $F(K) \subset K$. Now Brouwer's Theorem guarantees the existence of $y \in K$ with $F(y) = y$; obviously $\left(f^{-1}(y_1), \ldots, f^{-1}(y_n)\right)$ is a solution of (S_f).

Suppose that $u = (u_1, \ldots, u_n)$ and $v = (v_1, \ldots, v_n)$ are two distinct solutions. Then for each $i = 1, \ldots, n$ we have

$$\begin{cases} u_1 a_{i1} + \cdots + u_n a_{in} = f(u_i) \\ v_1 a_{i1} + \cdots + v_n a_{in} = f(v_i) \end{cases} \tag{3}$$

Let

$$\min_i \frac{u_i}{v_i} = \frac{u_j}{v_j}, \quad \max_i \frac{u_i}{v_i} = \frac{u_k}{v_k}, \quad i.e.,$$

$$\frac{u_j}{v_j} \leq \frac{u_l}{v_l} \leq \frac{u_k}{v_k}, \quad l = 1, \ldots, n. \tag{4}$$

From (3) we get for $i = 1, \ldots, n$,

$$\begin{cases} u_k a_{ik} + u_j a_{ij} = f(u_i) - \sum_{l \neq j,k} u_l a_{il} \\ v_k a_{ik} + v_j a_{ij} = f(v_i) - \sum_{l \neq j,k} v_l a_{il}. \end{cases} \tag{5}$$

Since u and v are distinct solutions, it is easy to see that

$$\begin{vmatrix} u_k & u_j \\ v_k & v_j \end{vmatrix} > 0$$

and so a_{ik} and a_{ij} can be determined from (5). If we do this, and if we take into account that $a_{ik} > 0$ and $a_{ij} > 0$, we get

$$v_j f(u_i) - u_j f(v_i) - \sum_{l \neq j,k} (u_l v_j - u_j v_l) a_{il} > 0,$$

$$u_k f(v_i) - v_k f(u_i) - \sum_{l \neq j,k} (v_l u_k - v_k u_l) a_{il} > 0,$$

for all $i = 1, \ldots, n$.

Combined with (4), this yields

$$v_j f(u_i) > u_j f(v_i); \quad u_k f(v_i) > v_k f(u_i), \quad i = 1, \ldots, n. \tag{6}$$

Accordingly, we get

$$v_j u_j g(u_j) > u_j v_j g(v_j); \quad u_k v_k g(v_k) > v_k u_k g(u_k),$$

which entails $g(u_j) > g(v_j)$ and $g(v_k) > g(u_k)$. Since g is strictly increasing, we get $u_j > v_j$ and $v_k > u_k$. Thus $u_j v_k > v_j u_k$, and this contradicts (4).

So Theorem 1 is proved.

Example 1. Let $q > 1$ and $g(x) = x^{q-1}$, $x > 0$. Then $f(x) = x^q$. According to Theorem 1, the system (S_q) has a unique solution. As mentioned in the Introduction, this result was obtained in [7, Theorem 3.1].

Theorem 2. *Let $h : (0, +\infty) \to (0, +\infty)$ be continuous and decreasing. Let $f(x) = \frac{h(x)}{x}$, $x > 0$, and suppose there exist $0 < m \leq M$ such that $f^{-1}(m) = \frac{M}{\nu}$, $f^{-1}(M) = \frac{m}{\mu}$. Then (S_f) has a unique solution.*

Proof. By using m and M, let us consider the set K and the function F given by (1) and (2). As in the proof of Theorem 1, one can verify that $F(K) \subset K$. Then the existence of a solution to (S_f) follows from Brouwer's Theorem.

Suppose that (S_f) has two distinct solutions u and v. As in the proof of Theorem 1, we derive the inequalities (6). Now they imply

$$\frac{f(u_j)}{u_j} > \frac{f(v_j)}{v_j}, \quad \frac{f(v_k)}{v_k} > \frac{f(u_k)}{u_k}. \tag{7}$$

Since $\frac{f(x)}{x}$ is strictly decreasing, we get $u_j < v_j$ and $v_k < u_k$, which gives $h(u_j) \geq h(v_j)$ and $h(v_k) \geq h(u_k)$. Therefore $u_j f(u_j) \geq v_j f(v_j)$ and $v_k f(v_k) \geq u_k f(u_k)$, i.e.,

$$u_j v_k f(u_j) f(v_k) \geq v_j u_k f(v_j) f(u_k). \tag{8}$$

On the other hand, from (6) we deduce

$$v_j f(u_k) > u_j f(v_k), \quad u_k f(v_j) > v_k f(u_j) \tag{9}$$

which contradicts (8). This concludes the proof of Theorem 2.

Example 2. Let $q < -1$ and $h(x) = x^{q+1}$, $x > 0$. The assumptions of Theorem 2 are satisfied with

$$m = \left(\mu^{q^2} \nu^q\right)^{1/(q^2-1)}, \quad M = \left(\mu^q \nu^{q^2}\right)^{1/(q^2-1)}.$$

Consequently, (S_q) has a unique solution.

Example 3. Let now $q = -1$. The system (S_q) becomes

$$\begin{cases} a_{11}x_1^2 + a_{12}x_1x_2 + \cdots + a_{1n}x_1x_n = 1 \\ \vdots \\ a_{n1}x_1x_n + a_{n2}x_2x_n + \cdots + a_{nn}x_n^2 = 1 \\ x_1 > 0, \ldots, x_n > 0. \end{cases} \tag{10}$$

The i^{th} equation is

$$a_{ii}x_i^2 + x_i \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j - 1 = 0.$$

Since $x_i > 0$, we get

$$x_i = \frac{2}{\left(\left(\sum_{j=1, j\neq i}^n a_{ij}x_j\right)^2 + 4a_{ii}\right)^{1/2} + \sum_{j=1, j\neq i}^n a_{ij}x_j} \tag{11}$$

Let $U := \{x \in \mathbb{R}^n | 0 \leq x_i \leq \frac{1}{\sqrt{a_{ii}}}, i = 1, \ldots, n\}$. Consider the function $G : U \to \mathbb{R}^n$, $G(x) = (G_1(x), \ldots, G_n(x))$, where $G_i(x)$ is the right-hand side of (11). Then obviously the system (10) is equivalent to the equation

$$G(x) = x, \quad x \in U. \tag{12}$$

On the other hand, it is easy to see that $G(U) \subset U$. Since G is continuous and U is compact and convex, Brouwer's Theorem implies the existence of a solution to (12). So the existence of a solution to (S_q) is proved also for $q = -1$.

The uniqueness of the solution follows from Theorem 2, since the corresponding proof of the uniqueness does not require the existence of m and M.

Remark 1. *For $-1 < q \leq 1$ it is easy to construct systems of type (S_q) having several solutions; see also [7, Rem. 3.4].*

3 A quasi-Newton algorithm for the system (S_f)

Assume that $a_{ji} = a_{ij}$, $i, j = 1, \ldots, n$. For $|q| > 1$ consider the function

$$f_q(x) := \sum_{i,j=1}^{n} a_{ij}x_ix_j - \frac{2}{q+1} \sum_{i=1}^{n} x_i^{q+1},$$

defined on the set $P_n := \{x \in \mathbb{R}^n | x_1 > 0, \ldots, x_n > 0\}$.
For $q = -1$ let $f_{-1}(x) := \sum_{i,j=1}^n a_{ij}x_ix_j - 2\sum_{j=1}^n \log x_j$ defined on the same set P_n.

For all $q \in (-\infty, -1] \cup (1, +\infty)$ we have

$$\frac{\partial f_q(x)}{\partial x_i} = 2\sum_{j=1}^{n} a_{ij}x_j - 2x_i^q, \quad i = 1, \ldots, n,$$

so that the unique stationary point of f_q is the unique solution of the system (S_q). By examining the behavior of f_q near the boundary of P_n we see that for $q > 1$ f_q attains a maximum, while for $q \leq -1$ it attains a minimum in P_n.

Theorem 3. *A quasi-Newton algorithm for finding the extremum points of f_q is described by*

$$x_r^{(k+1)} = x_r^{(k)} \left(1 + \frac{\sum_{j=1}^{n} a_{rj}x_j^{(k)} - \left(x_r^{(k)}\right)^q}{q\left(x_r^{(k)}\right)^q - a_{rr}x_r^{(k)}} \right), \qquad k \geq 0, \tag{13}$$

for $r = 1, \ldots, n$.

Proof. Let $t \in P_n$ be given. The construction of the quasi-Newton algorithm is based on approximating f_q in a neighborhood of t by a "restricted" polynomial w of the form described in (14). Of course, the approximation by a complete polynomial of second degree will lead to the Newton algorithm; see, e.g., [6, Sect. 3.4]. See also Remarks 3 and 4 below.

Consequently, we shall determine a function

$$w(x) = \sum_{i=1}^{n} v_i x_i^2 + 2\sum_{i=1}^{n} u_i x_i + c \tag{14}$$

such that

$$w(t) = f_q(t) \tag{15}$$

$$\frac{\partial w}{\partial x_i}(t) = \frac{\partial f_q}{\partial x_i}(t), \qquad i = 1, \ldots, n, \tag{16}$$

$$\frac{\partial^2 w}{\partial x_i^2}(t) = \frac{\partial^2 f_q}{\partial x_i^2}(t), \qquad i = 1, \ldots, n. \tag{17}$$

From (17) we get

$$v_i = a_{ii} - qt_i^{q-1}, \qquad i = 1, \ldots, n. \tag{18}$$

Now (16) and (18) imply

$$u_i = \sum_{j=1}^{n} a_{ij}t_j - a_{ii}t_i + (q-1)t_i^q, \qquad i = 1, \ldots, n. \tag{19}$$

The number c can be determined from (15), and the resulting function w approximates f_q in the neighborhood of t.

Let $x^{(0)} \in P_n$ be given. We construct the iterates $x^{(j)}$ as follows. Suppose $x^{(k)}$ was determined. Consider the function w associated to $t = x^{(k)}$; its extremum point is

$$\left(-\frac{u_1}{v_1}, \ldots, -\frac{u_n}{v_n} \right).$$

Then we take $x^{(k+1)}$ to be this point; see also [6, Sect. 3.4]. According to (18) and (19) we get

$$x_r^{(k+1)} = \frac{\sum_{j=1}^n a_{rj} x_j^{(k)} - a_{rr} x_r^{(k)} + (q-1)\left(x_r^{(k)}\right)^q}{q\left(x_r^{(k)}\right)^{q-1} - a_{rr}}$$

for $r = 1, \ldots, n$. This is equivalent to (13).

Corollary 1. *Suppose that for the sequence* $\left(x^{(k)}\right)_{k\geq 0}$ *given by (13) one has*

$$\lim_{k\to\infty} x_r^{(k)} = x_r^* > 0, \qquad r = 1, \ldots, n, \tag{20}$$

Then $x^* = (x_1^*, \ldots, x_n^*)$ *is a solution of the system* (S_q).

Proof. Indeed, under the assumption (20) we get from (13):

$$\sum_{j=1}^n a_{rj} x_j^* = (x_r^*)^q, \quad r = 1, \ldots, n.$$

Remark 2. *Our extensive numerical experiments show that the algorithm described in Theorem 3 converges whenever one starts with small values (if* $q \leq -1$*), respectively large values (if* $q > 1$*) of* $x_1^{(0)}, \ldots, x_n^{(0)}$*, even if the matrix* $(a_{ij})_{i,j=1,\ldots,n}$ *is not symmetric and* n *is large; see Figure 1.*

Remark 3. *Obviously the algorithm can be generalized in order to approximate the solution of* (S_f)*. It becomes*

$$x_r^{(k+1)} = x_r^{(k)}\left(1 + \frac{\sum_{j=1}^n a_{rj} x_j^{(k)} - f(x_r^{(k)})}{\left(f'(x_r^{(k)}) - a_{rr}\right) x_r^{(k)}}\right), \quad r = 1, \ldots, n, \tag{21}$$

and provides the extremum point of the function

$$F(x_1, \ldots, x_n) := \sum_{i,j=1}^n a_{ij} x_i x_j - 2\sum_{i=1}^n \varphi(x_i)$$

where $\varphi'(s) = f(s)$*,* $s \in (0, +\infty)$*.*

Remark 4. *In order to find the extremum point of the above function* F *we can use also the Newton algorithm (see, e.g., [6, Sect. 3.4.]). In our specific setting it can be described as follows.*

Let $x = (x_1, \ldots, x_n)^t$ *be a column vector. Consider the matrix*

$$A(x) := \begin{pmatrix} a_{11} - f'(x_1) & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} - f'(x_2) & \ldots & a_{2n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \ldots & a_{nn} - f'(x_n) \end{pmatrix}$$

and let $\Phi(x) := (f(x_1) - x_1 f'(x_1), \ldots, f(x_n) - x_n f'(x_n))^t$*.*

Starting with an initial solution $x^{(0)}$, the iterations are described by

$$x^{(k+1)} := \left(A(x^{(k)})\right)^{-1} \Phi(x^{(k)}), \quad k \geq 0. \tag{22}$$

Since the quasi-Newton algorithm (21) does not require to invert matrices, it is simpler than the Newton algorithm (22). Being more precise, the Newton algorithm requires a smaller number of iterations, especially in the case of small systems. For large problems, the quasi-Newton algorithm is more efficient with respect to the computing time. Figure 1 displays the computing times with (21) and respectively (22) for $f(x) = x^{-10}$. The involved matrices and initial solutions are randomly generated, with increasing dimensions.

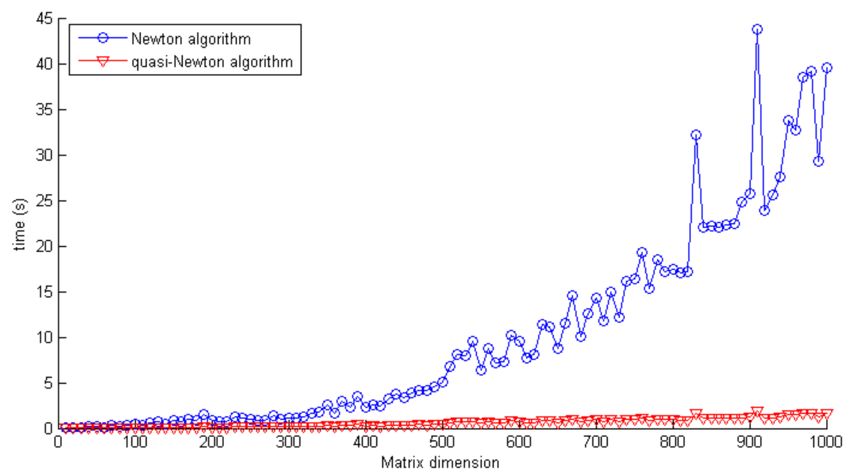

Figure 1: *Computing times with (21) and respectively (22).*

4 An extremum problem

Let $c_i = (c_{i1}, \ldots, c_{im}) \in \mathbb{R}^m$, $c_{ij} > 0$, $i = 1, \ldots, n$; $j = 1, \ldots, m$. Let $p_i > 0$ be also given, $i = 1, \ldots, n$.

Consider the function

$$f(x_1, \ldots, x_m) = \prod_{i=1}^{n} (c_{i1}x_1 + \cdots + c_{im}x_m)^{p_i}$$

defined for $x \in \mathbb{R}^m$ with $x_1 > 0, \ldots, x_m > 0$ and $\sum_{j=1}^{m} x_j^2 = 1$.
Then $\log f(x) = \sum_{i=1}^{n} p_i \log(c_{i1}x_1 + \cdots + c_{im}x_m)$ and the Lagrange function associated with $\log f(x)$ is $L = \sum_{i=1}^{n} p_i \log s_i - \lambda(x_1^2 + \cdots + x_m^2 - 1)$, where $s_i := c_{i1}x_1 + \cdots + c_{im}x_m$.

The stationary points for L are solutions of

$$\begin{cases} \frac{p_1}{s_1} c_{11} + \frac{p_2}{s_2} c_{21} + \cdots + \frac{p_n}{s_n} c_{n1} = 2\lambda x_1 \\ \vdots \\ \frac{p_1}{s_1} c_{1m} + \frac{p_2}{s_2} c_{2m} + \cdots + \frac{p_n}{s_n} c_{nm} = 2\lambda x_m \\ x_1^2 + \cdots + x_m^2 = 1. \end{cases} \tag{23}$$

Multiply the i^{th} equation by x_i, $i = 1, \dots, m$, and then add; the result will be

$$p_1 + \cdots + p_n = 2\lambda.$$

Let $t_i := \frac{p_i}{s_i}$, $P := p_1 + \cdots + p_n$.
The first m equations of (23) become

$$\begin{cases} c_{11}t_1 + \cdots + c_{n1}t_n = Px_1 \\ \vdots \\ c_{1m}t_1 + \cdots + c_{nm}t_n = Px_m. \end{cases} \tag{24}$$

Fix an $i \in \{1, \dots, n\}$. Multiply the j^{th} equation of (24) by c_{ij}, $j = 1, \dots, m$, and add to obtain

$$(c_1|c_i)t_1 + \cdots + (c_n|c_i)t_n = Ps_i \tag{25}$$

where $(c_j|c_i) := \sum_{k=1}^{m} c_{jk}c_{ik}$.
Since $s_i = p_i/t_i$, we get

$$\begin{cases} (c_1|c_1)t_1 + \cdots + (c_n|c_1)t_n = Pp_1t_1^{-1} \\ \vdots \\ (c_1|c_n)t_1 + \cdots + (c_n|c_n)t_n = Pp_nt_n^{-1}. \end{cases} \tag{26}$$

After dividing the i^{th} equation by Pp_i we get a system of type (S_q) with $q = -1$. We know that it has a unique solution with $t_1 > 0, \dots, t_n > 0$. From (24) we get the stationary point x; the corresponding value of f is

$$\prod_{i=1}^{n} \left(\frac{p_i}{t_i}\right)^{p_i}.$$

5 Examples

Example 4. Let $\alpha_i > 0$, $i = 1, 2, \dots$. Systems like

$$A_n \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix} = \begin{pmatrix} t_1^p \\ \vdots \\ t_n^p \end{pmatrix} \tag{27}$$

where

$$A_n = \begin{pmatrix} \alpha_1+\alpha_2 & -\alpha_2 & 0 & \dots & 0 & 0 \\ -\alpha_2 & \alpha_2+\alpha_3 & -\alpha_3 & \dots & 0 & 0 \\ 0 & -\alpha_3 & \alpha_3+\alpha_4 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -\alpha_n & \alpha_n+\alpha_{n+1} \end{pmatrix}$$

are inspired by the theory of snakes under the simplifying hypothesis that the snakes acts only like a membrane, i.e. $\alpha > 0$ and $\beta = 0$, see [4].
Since

$$\det A_k = \alpha_1 \dots \alpha_{k+1} \left(\frac{1}{\alpha_1} + \cdots + \frac{1}{\alpha_{k+1}}\right), \quad k \geq 1,$$

a classical result of Ostrowski (see [2, Chap. 16]) shows that the entries of A^{-1} are strictly positive. Now (27) becomes

$$A_n^{-1}\begin{pmatrix} t_1^p \\ \vdots \\ t_n^p \end{pmatrix} = \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix}. \tag{28}$$

Setting $t_i^p = x_i$ we get a system of type (S_q), with $q = 1/p$. If $p \in [-1, 0) \cup (0, 1)$, then $q \in (-\infty, -1] \cup (1, +\infty)$, and we can apply the algorithm (13) in order to solve the system (28).

In particular, if $\alpha_i = 1$, $i = 1, 2, \dots$, we are dealing with a second order Dirichlet problem like that presented in [7, Section 2].

Example 5. When $\alpha = 0$ and $\beta > 0$, we are lead to a system like (27), but with $\beta_i > 0$ and

$$A_n = \begin{pmatrix} \beta_1+4\beta_2+\beta_3 & -2\beta_2-2\beta_3 & \beta_3 & 0 & 0 & \dots & 0 \\ -2\beta_2-2\beta_3 & \beta_2+4\beta_3+\beta_4 & -2\beta_3-2\beta_4 & \beta_4 & 0 & \dots & 0 \\ \beta_3 & -2\beta_3-2\beta_4 & \beta_3+4\beta_4+\beta_5 & -2\beta_4-2\beta_5 & \beta_5 & \dots & 0 \\ 0 & \beta 4 & -2\beta_4-2\beta_5 & \beta_4+4\beta_5+\beta_6 & -2\beta_5-2\beta_6 & \dots & 0 \\ \vdots & & & & & & \\ 0 & 0 & 0 & 0 & 0 & \dots & \beta_n+4\beta_{n+1}+\beta_{n+2} \end{pmatrix}$$

Such a system can be solved as in Example 4.

In particular, if $\beta_i = 1$, $i = 1, 2, \dots$, this is a fourth order difference equation of type discussed in [7, Section 2].

Example 6. Here is an example when $\alpha_i = \beta_i = 1$. The system corresponding to (27) is

$$\begin{pmatrix} 8 & -5 & 1 & 0 & 0 & 0 \\ -5 & 8 & -5 & 1 & 0 & 0 \\ 1 & -5 & 8 & -5 & 1 & 0 \\ 0 & 1 & -5 & 8 & -5 & 1 \\ 0 & 0 & 1 & -5 & 8 & -5 \\ 0 & 0 & 0 & 1 & -5 & 8 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{pmatrix} = \begin{pmatrix} t_1^{-1/2} \\ t_2^{-1/2} \\ t_3^{-1/2} \\ t_4^{-1/2} \\ t_5^{-1/2} \\ t_6^{-1/2} \end{pmatrix} \tag{29}$$

Here $p = -\frac{1}{2}$; setting $t_i^{-1/2} = x_i$, we get the following system of type (S_q) with $q = 1/p = -2$:

$$\begin{pmatrix} 0.3159 & 0.3708 & 0.3263 & 0.2451 & 0.1530 & 0.0650 \\ 0.3708 & 0.7377 & 0.7222 & 0.5635 & 0.3576 & 0.1530 \\ 0.3263 & 0.7222 & 1.0005 & 0.8566 & 0.5635 & 0.2451 \\ 0.2451 & 0.5635 & 0.8566 & 1.0005 & 0.7222 & 0.3263 \\ 0.1530 & 0.3576 & 0.5635 & 0.7222 & 0.7377 & 0.3708 \\ 0.0650 & 0.1530 & 0.2451 & 0.3263 & 0.3708 & 0.3159 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} x_1^{-2} \\ x_2^{-2} \\ x_3^{-2} \\ x_4^{-2} \\ x_5^{-2} \\ x_6^{-2} \end{pmatrix} \tag{30}$$

Starting with the initial solution

$$x^{(0)} = (0.00004 \quad 0.0001 \quad 0.0001 \quad 0.0002 \quad 0.0005 \quad 0.0001)$$

we obtain the solution of (30)

$$x^{(31)} = (0.9600 \quad 0.6962 \quad 0.6206 \quad 0.6206 \quad 0.6962 \quad 0.9600).$$

Now the solution of (29) is obtained from $t_i = x_i^{-2}$, $i = 1, \ldots, 6$:

$$t = (1.0850 \quad 2.0633 \quad 2.5963 \quad 2.5963 \quad 2.0633 \quad 1.0850).$$

Example 7. Let $g(x) = e^x - 1$ and $f(x) = xg(x)$, $x \in (0, +\infty)$. The assumption if Theorem 1 are satisfied, hence the system S_f has a unique solution.

As an example, consider the system

$$\begin{pmatrix} 225.90 & 147.09 & 82.88 & 142.50 & 191.99 \\ 147.09 & 200.99 & 105.75 & 79.87 & 119.30 \\ 82.88 & 105.75 & 98.93 & 47.69 & 40.64 \\ 142.50 & 79.87 & 47.69 & 152.40 & 139.01 \\ 191.99 & 119.30 & 40.64 & 139.01 & 191.86 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} x_1(e^{x_1} - 1) \\ x_2(e^{x_2} - 1) \\ x_3(e^{x_3} - 1) \\ x_4(e^{x_4} - 1) \\ x_5(e^{x_5} - 1) \end{pmatrix}$$

and choose the initial solution

$$x^{(0)} = (61.27 \quad 30.08 \quad 79.81 \quad 79.56 \quad 78.10).$$

By using the quasi-Newton algorithm (21) we get the solution

$$x^{(82)} = (6.64 \quad 6.47 \quad 5.99 \quad 6.34 \quad 6.52).$$

Example 8. Consider the function

$$f(x_1, x_2, x_3) = (x_1 + 2x_2 + 3x_3)^4 (6x_1 + 4x_2 + 2x_3)^2 (4x_1 + 9x_2 + x_3)^3$$

We want to find its maximum point subject to $x_1^2 + x_2^2 + x_3^2 = 1$, $x_1, x_2, x_3 > 0$.

The system (26) becomes

$$\begin{pmatrix} 0.3889 & 0.5556 & 0.6944 \\ 1.1111 & 3.1111 & 3.4444 \\ 0.9259 & 2.2963 & 3.6296 \end{pmatrix} \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} t_1^{-1} \\ t_2^{-1} \\ t_3^{-1} \end{pmatrix}.$$

The algorithm (13), applied to this system, gives the solution

$$t_1 = 1.1721, \quad t_2 = 0.2960, \quad t_3 = 0.3353.$$

Now (24) yields

$$x_1 = 0.4766, \quad x_2 = 0.7274 \quad x_3 = 0.4938. \tag{31}$$

This is the required maximum point.

It is instructive to have another look at our extremum problem. Indeed, it is equivalent to the problem of finding the maximum point of

$$z(x_1, x_2) =$$

$$(x_1 + 2x_2 + 3\sqrt{1 - x_1^2 - x_2^2})^4 (6x_1 + 4x_2 + 2\sqrt{1 - x_1^2 - x_2^2})^2 (4x_1 + 9x_2 + \sqrt{1 - x_1^2 - x_2^2})^3$$

subject to $x_1 > 0$, $x_2 > 0$, $x_1^2 + x_2^2 < 1$.

Passing to polar coordinates

$$x_1 = \rho \cos\theta, \quad x_2 = \rho \sin\theta, \quad 0 < \rho < 1, \; 0 < \theta < \frac{\pi}{2}$$

we obtain a function z having the graph as in Fig. 2.
The maximum point corresponds to

$$\rho = 0.8696, \quad \theta = 0.9907,$$

which leads to (31).

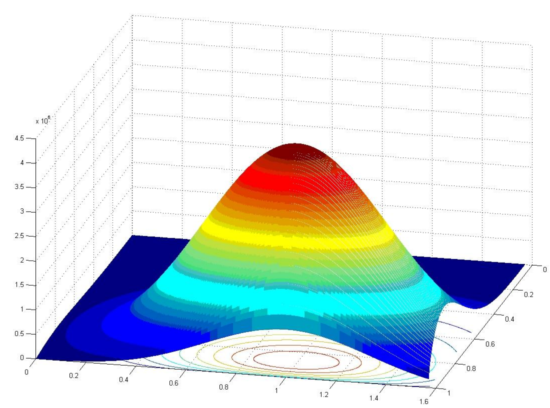

Figure 2: *The graph of the function z*

6 Concluding remarks

As seen above, the algorithm described in Theorem 3 and Remark 3 works very well in order to solve systems of type (S_f) and the corresponding extremum problems. It provides an accurate approximation of the exact solution in a small number of iterations. This is explained, apart from the convergence theorems for general systems of nonlinear equations, by the essential feature of our special setting: the existence and uniqueness of the solution, proved in our article. The rate of convergence is governed by the general rules of quasi-Newton algorithms.

On the other hand, in the general setting of the Expectation -Maximization Algorithm (see [5]), we developed in [3] a generalization of the EMML and ISRA algorithms for solving the Positron Emission Tomography problem. Similar algorithms for solving the system (S_f) will be discussed in a forthcoming paper.

References

[1] Z.-Z. Bai, X. Yang, On HSS-based iteration methods for weakly nonlinear systems, *Appl. Numer. Math.* 59(2009) 2923-2936.

[2] R. Bellman, *Introduction to Matrix Analysis*, MacGraw-Hill, London, 1960.

[3] A. Ciurte, S. Nedevschi, I. Rasa, A generalization of the EMML and ISRA algorithms for solving linear systems, *J. Comput. Anal. Appl.* 12(2010) 799-816.

[4] M. Kass, A. Witkin, D. Terzopoulos, Snakes: Active contour models, *Int. J. Computer Vision* 1 (1988) 321-331.

[5] G. J. McLachlan, Th. Krishnan, *The EM Algorithm and Extensions*, Wiley, 2008.

[6] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, 2005.

[7] G. Zhang, W. Feng, On the number of positive solutions of a nonlinear algebraic system, *Linear Alg. Appl.* 422(2007) 404-421.