ORNL/TM-10976

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Bifurcating Optimization Algorithms and Their Possible Application

J.-C. Culioli
V. Protopopescu

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-10976

Engineering Physics and Mathematics Division

BIFURCATING OPTIMIZATION ALGORITHMS AND THEIR POSSIBLE APPLICATION

J.-C. Culioli and V. Protopopescu

Published: November 1988

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0291337 3

TABLE OF CONTENTS

Abstract v

1. Introduction 1

2. Sigmoidic Algorithms 5

   2.1. Definitions and Elementary Properties 5

   2.2. Algorithms for Nonlinear Optimization 6

   2.3. Algorithms for Linear Optimization 10

3. Bifurcation and Chaotic Behavior of Sigmoidic Algorithms 13

   3.1. Nonlinear Updating 13

   3.2. Linear Updating 25

4. Summary 27

Acknowledgments 27

References 29

Appendix 31

Abstract

A new class of optimization algorithms for linear and nonlinear problems is proposed. The algorithms are based on sigmoidic updatings and can be looked upon as nonlinear multidimensional maps. The bifurcation and chaotic regimes of these maps are analyzed and their possible applications in optimization problems are indicated.

1. Introduction

Recently, a new class of optimization algorithms has been introduced [1,2,3] that are inspired both by Neural Network models (such as Hopfield's model [4]) and by classical saddle-point seeking techniques (such as Arrow-Hurwicz-Uzawa algorithms [5]). We want to present here these algorithms and their properties from a different point of view which could open the way to better and more stable numerical schemes. The main difference with "standard" optimization algorithms, lies in the use of the gradient information: a nonlinear (sigmoidic) operation is applied to the gradient before it is included in the updating part. A motivation for this updating and its connection with the standard gradient algorithm follows here on a very simple example. Suppose that we attempt to minimize the function $F(x) = \frac{1}{2}(x-1)^2$ on the real line R. A gradient algorithm would proceed along the following updating scheme:

- given an initial point x^0 and a "small" positive parameter μ,
- for every k, compute x^{k+1} according to:

$$x^{k+1} = x^k - \mu F'(x^k) = x^k - \mu (x^k - 1) = (1-\mu) x^k + \mu \qquad (1)$$

For some values of μ, the sequence defined by (1) will converge to 1 while for other values it will diverge. Indeed, we see from the last expression in (1) that as long as μ is smaller than 1, x^{k+1} is contained between x^k and 1, and thus the sequence will monotonically converge to 1. Note that the rate of convergence to the solution $x^*=1$ increases with μ. When $\mu = 1$, $x^1 = 0 + \mu = 1$, which is the fastest possible convergence.

For μ larger than 1, the algorithm will provide either an alternate convergent sequence or an unbounded sequence. In one and only one case, $\mu = 2$, it will generate an alternating sequence. To show this, let us define $\mu = 1 + \mu'$, with $\mu' > 0$ and rewrite (1) as

$$x^{k+1} = -\mu' x^k + 1 + \mu'. \tag{2}$$

We denote the mapping associated with Eq. (2) by $G : R \to R : G(x) = -\mu' x + 1 + \mu'$. The only fixed point of G, solution of $G(x) = x$, is $x^* = 1$. We are interested in the stability of this unique fixed point. Since $G'(x^*) = -\mu'$, as long as $\mu' < 1$, x^* will be stable because $|G'(x^*)| < 1$. This can also be shown by rewriting (2) in the following way: $x^{k+1} - 1 = -\mu' (x^k - 1)$, which implies $|x^{k+1} - 1| = \mu' |x^k - 1| < |x^k - 1|$. For $\mu' = 1$ (or $\mu = 2$), we can observe some "bifurcation" cycles which depend on the initial condition x^0: for example, if $x^0 = -1$, the generated sequence is $\{-1, 3, -1, 3,...\}$; if $x^0 = 2$, the sequence is $\{2, 0, 2, 0, ...\}$. In any case, only a period two oscillation will occur and the mean of the two alternating values will be x^*, the fixed point. The convergence speed is the best that can be achieved (if we exclude the case $\mu = 1$ which was more a smart guess than a real computation). Note that this oscillating behavior will only be detected on an infinite-precision computer, otherwise, round-off effects will mask this particular case.
As soon as $\mu' > 1$ (or $\mu > 2$), the gradient algorithm will diverge because for every real x, since $|G'(x)| = \mu' > 1$ and the fixed point becomes repulsive. On a computer, the generated sequence will end in a series of negative or positive overflows.

The analysis of the above example shows a very general feature of recursive algorithms, irrespective of their specific form and/or of the specific problem they are applied to. Namely, since these algorithms are most often represented by nonlinear maps, one is not surprised to see that they exhibit regions of stability, alternating sequences, and chaotic regimes. Usually one would like to use an algorithm in its stability region, in order to ensure convergence. However, one may be constrained to choose parameters that drive the algorithm outside this region (one may not always have the time to find the right parameters). In most cases, this apparent inconvenience turns out to be only apparent and sometimes, it becomes even beneficial. Namely, (i) one can recover a fair approximation of

the right answer from a bifurcating (chaotic) region, and (ii) one can do it more quickly than in the normal regime. In order to take advantage of these properties, and to design some optimization algorithms with good convergence speed on computers, one way could be to "magnify" the set on which bifurcations occur, and reduce, possibly to a zero-measure set, the set for which unbounded sequences are generated. The latter part can be easily achieved with a "hard-limiter", i.e. a bound included into the updating that will prevent the sequences $\{x^k\}$ from diverging. For example, a "hard-limited" gradient algorithm could be (with B a large positive number):

$$x^{k+1} = \mathrm{Min}\{\ \mathrm{Max}\ \{\ x^k - \mu\,(x^k - 1), -B\}, +B\}. \tag{3}$$

But it is not difficult to see that this implementation will generate, for every $\mu \geq \mu^B$ some alternating sequences of the form $\{..., B, -B, B, -B\}$, the mean of which is 0 and not x^*. The mapping associated with (3) is now given by

$$G(x) = \mathrm{Min}\{\ \mathrm{Max}\ \{\ (1-\mu)\,x + \mu\ , -B\}, +B\}.$$

Note that $x^*=1$ is still a fixed point of G. To compute μ^B, we just need to write $G(B) \leq -B$ or $B - \mu\,(B-1) \leq -B$, which leads to

$$\mu \geq \mu^B = \frac{2\ B}{B\ -\ 1} \approx 2 + \frac{1}{B}.$$

Thus a "hard-limiter" is thus not the right "device" to obtain "good" bifurcations. An alternative is a "soft-limiter". A "soft-limiter" should provide an updating $x^{k+1} = G\,(x^k, \mu)$ such that x^* is still a fixed point of G, the ranges of μ for which there are bifurcations will be "large enough", and the mean of the bifurcating values will be as close as possible to x^*. With these properties, such an updating should have a good convergence speed to (at least) an approximation of x^*. In the remainder of this paper, we shall introduce the sigmoidic algorithms as possible candidates for a "soft-limiter" approach. We shall study their bifurcating and chaotic properties and their possible utilization in linear and nonlinear optimization problems.

2. Sigmoidic Algorithms

2.1 Definitions and Elementary Properties

A one dimensional parameter-dependent mapping $g_\mu : R \to (0,1)$ defined by

$$g_\mu (y) = \frac{1}{1+ \exp(- \mu y)} \tag{4}$$

will be referred to as a *sigmoidic function*.

The most important properties of sigmoidic functions are summarized below:

$\forall\, y \in R, \forall\, \mu \in R_+^*,$

$$0 < g_\mu (y) < 1, \quad g_\mu'(y) = \mu\, g_\mu (y)\, g_\mu (-y), \quad 0 < g_\mu'(y) \leq \mu, \tag{5}$$

$$g_\mu^{-1}(z) = \mu^{-1} \log[z/(1-z)], \ \forall\, z \in (0,1).$$

We define a *sigmoidic algorithm* as a recursive algorithm that realizes the updatings via a sigmoidic function. When designing a sigmoidic algorithm, one will use a linear updating if the function to be minimized, F, is linear in x, and a nonlinear updating otherwise. As will be seen in the following, in the first case it is critical to have, beforehand, an upper bound for every variable that will be updated. In the second case, it is time-saving to have some information on the sign of the sought solution, since the nonlinear updating does not change the sign of the generated sequence $\{x^k\}$. Without any knowledge of that sign, one could choose it at random ; if the sign is wrong, the algorithm converges to 0. Thus one has to check if $x^k = 0$ is an optimal solution ; if this is not the case, one must rerun the algorithm with $x^0 = - x^0$. One could also run the algorithm twice in parallel, but this must be done for each variable and may lead to combinatorial explosion for large dimensions.

Since we are going to introduce several "primal-dual" algorithms, we shall recall the definition of the saddle-point of a functional $L : R^n \times R^m \to R$. The couple (x^*,p^*) in $R^n \times R^m$ is called a *saddle-point* of the functional L, if there exists a neighborhood $V(x^*,p^*)$ in $R^n \times R^m$, such that

$$L(x^*,p) \leq L(x^*,p^*) \leq L(x,p^*), \forall \ (x,p) \in V(x^*,p^*). \tag{6}$$

We shall call x a *primal* variable and p a *dual* variable. In optimization problems, when the functional L has the meaning of a cost function, the primal variables are usually the controls that enable to minimize a given criterion and the dual variables have the interpretation of prices associated with the constraints.

2.2 Algorithms For Nonlinear Optimization.

We consider a constrained optimization problem:

Find

$$\mathrm{Min}_x \ F(x) \tag{7-a}$$

subject to the constraint

$$Q(x) = 0 \tag{7-b}$$

with $F : R^n \to R$ convex and differentiable and $Q : R^n \to R^m$, also convex and differentiable. The Lagrangian associated with (7) is given by $L(x,p) = F(x) - \langle p,Q(x)\rangle$, where p is an element of R^m and $\langle .,. \rangle$ is the inner product in R^m. We assume that problem (7) has a solution and we denote by (x^*,p^*) the corresponding saddle-point of L. A standard saddle-point algorithm for (7) is the Arrow-Hurwicz algorithm ([5]) which updates the primal and dual variables x^k and p^k, according to

$$x^{k+1} = x^k - \mu \, [\, F'(x^k) - \langle p^k, Q'(x^k) \rangle \,] \tag{8-a}$$

$$p^{k+1} = p^k + \varepsilon \quad Q(x^{k+1}) \tag{8-b}$$

with μ and ε two small non-negative real constants. The Arrow-Hurwicz algorithm is often described as "performing one gradient-descent step on the primal and one gradient-ascent step on the dual". Figure 1 illustrates this feature.

We propose as an alternative the nonlinear sigmoidic algorithm:

$$x^{k+1} = 2 \; x^k \; g_\mu \, [\, F'(x^k) - \langle p^k, Q'(x^k) \rangle \,] \tag{9-a}$$

$$p^{k+1} = p^k + \varepsilon \; Q(x^{k+1}) \tag{9-b}$$

with $g_\mu(.)$ applied componentwise to its vector-argument. The primal variable, x^k, is updated through a nonlinear sigmoidic mapping. This updating provides, as will be seen later, a better stability. Note that if the constraint in (7) is $Q(x) \leq 0$ (the inequality is understood componentwise in R^m), then (9-b) is simply transformed into

$$p^{k+1} = \text{Max} \, \{p^k + \varepsilon \; Q(x^{k+1}), 0\}. \tag{9-b'}$$

Since $g_\mu(0) = \frac{1}{2}$, the solutions of (7) are the fixed points of algorithm (9). Their stability can be studied in the same way as was done for the non-constrained example given in the Introduction. When varying μ, one discovers similar properties concerning the speed of convergence and the approximation of the solution (see Section 2.3).

Remarks:

(i) Sometimes it is cumbersome or even impossible to get an analytical expression of the derivative of F. Since - as will be seen later - the standard algorithm gives interesting results when μ is large, we propose the following updating:

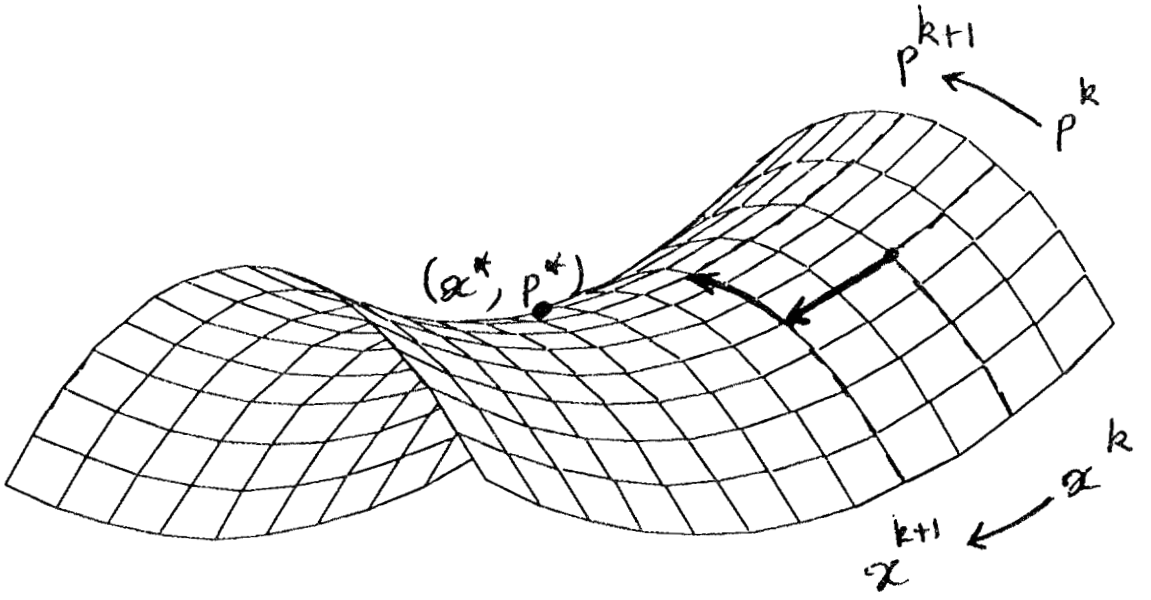
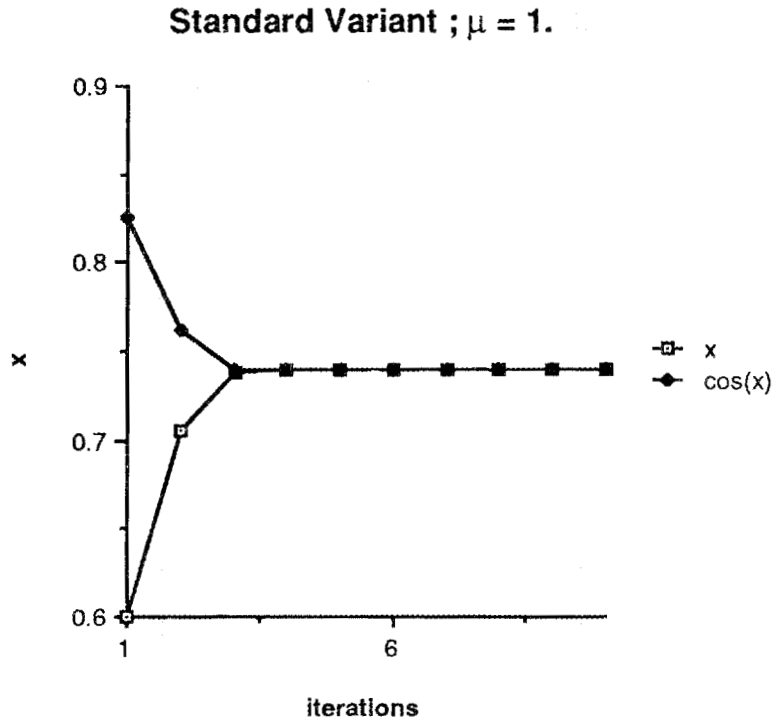


Figure 1: Arrow-Hurwicz Algorithm: "one descent step on the primal, one ascent step on the dual" : $x^{k+1} = x^k - \mu \frac{\partial L}{\partial x}(x^k,p^k)$ and $p^{k+1} = p^k + \varepsilon \frac{\partial L}{\partial p}(x^{k+1},p^k)$, with μ and ε small positive constants.

$$x^{k+1} = x^k \, g_\mu \left(-\frac{F(x^k) - F(x^{k-1})}{x^k - x^{k-1}}\right), \qquad \mu >> 1.$$

Figure 2 shows that this variant behaves essentially as well as the original algorithm . We have made this comparison using the unconstrained problem of finding the minimum of $F(x) = \frac{1}{2}(\cos(x)-x)^2$. We have chosen $\mu = 1$ for the standard variant and $\mu = 5$ for the finite-difference variant. The latter needs just a few more iterations to converge.

(ii) The reader will find in the Appendix some comments on the transformation performed on F(x) by the nonlinear sigmoidic updating (9-a).



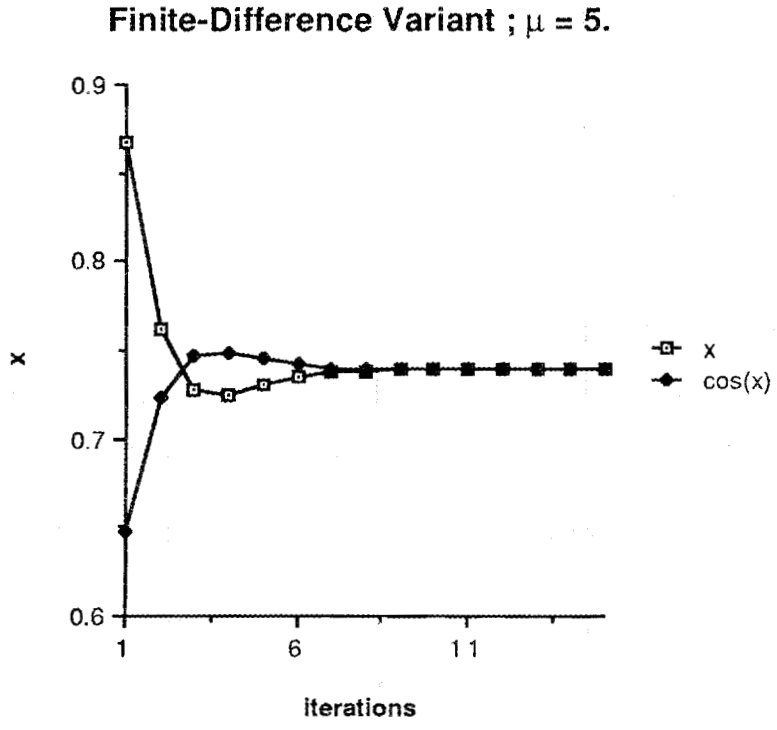


Figure 2: A Finite-Difference Version.

2.3. Algorithms For Linear Optimization.

The standard linear programming problem can be stated as follows:

Find

$$\text{Max} \quad \Sigma_i \; c_i \; x_i \tag{10-a}$$

subject to the constraints

$$x_i \geq 0, \quad i = 1, 2, ..., n. \tag{10-b}$$

$$\Sigma_i \; a_j^i \; x_i = b_j \;, \quad j = 1,2,...,m. \tag{10-c}$$

We suppose that this problem is bounded and has a solution, denoted by x^*. We denote by p^* the solution of the dual problem associated with (10) (see [6]). The Lagrangian functional is now:

$$L(x,p) = \Sigma_i \, c_i \; x_i - \Sigma_j \; p_j \; [\Sigma_i \, a_j^i \; x_i - b_j].$$

In [1], we have proposed the following updatings:

$$x_i^{k+1} = X_i \; g_\mu [c_i - \Sigma_j \; p_j^k \; a_j^i] \qquad i = 1, 2, ..., n \tag{11-a}$$

$$p_j^{k+1} = p_j^k + \varepsilon \; [\Sigma_i \; a_j^i \; x_i^{k+1} - b_j] \qquad j = 1,2, ..., m \tag{11-b}$$

applied until an optimality criterion is satisfied. Here, X_i is a bound on x_i (typically, X_i has to be greater than x_i^*).Convergence and optimality for this linear-programming algorithm have been studied in [1] when all coefficients a_i^j are positive (which is the case for the Transportation Problem or for the Assignment Problem). We recall the main results of [1]:

For ε and μ^{-1} small enough, the algorithm converges to an approximation (x^a,p^a) of the actual solution (x^*,p^*). The couple (x^a,p^a) is feasible (i.e. (10-b) and (10-c) are satisfied), and the duality-gap, defined as $| \langle c,x^a \rangle - \langle b,p^a \rangle |$, is linearly bounded by μ^{-1}.

Thus, when the parameters ε and μ^{-1} are sufficiently small, the sigmoidic algorithms (9a-b) and (11a-b) are expected to follow the "classical behavior", namely they generate sequences which converge to the optimal solution. We give in Figure 3 the trajectories of the linear algorithm corresponding to the solution of the following problem:

Find

$$\text{Max } z = 2x_1 + 4x_2 + 4x_3 - 3x_4, \quad x_i \geq 0$$

subject to the constraints

$$x_1 + x_2 + x_3 = 4 \quad \text{and} \quad x_1 + 4x_2 + x_4 = 8$$

The optimal solution is $x^* = (0,2,2,0)$, $p^* = (4,0)$ and $z^* = 16$. The algorithm parameters are $\varepsilon = 0.02$, $\mu = 5$.

In the "classical regime", the sigmoidic algorithms do not perform particularly better than the standard optimization algorithms (e.g. Simplex, Arrow-Hurwicz, etc.). However, interesting properties occur when ε and μ^{-1} can take larger values. In that case, we discover that the linear and nonlinear sigmoidic updatings act as the "soft-limiters" that we were looking for in the Introduction: they enhance the ranges of ε and μ in which bifurcations and high-speed convergence can occur and provide a greater overall stability to the algorithm. Due to the common structure of the sigmoidic updatings, the study of chaotic behavior and bifurcations leads to very similar features for both the linear or nonlinear case. Therefore we shall illustrate different essential aspects for each of these updatings by indiscriminately using either the linear or the nonlinear version.

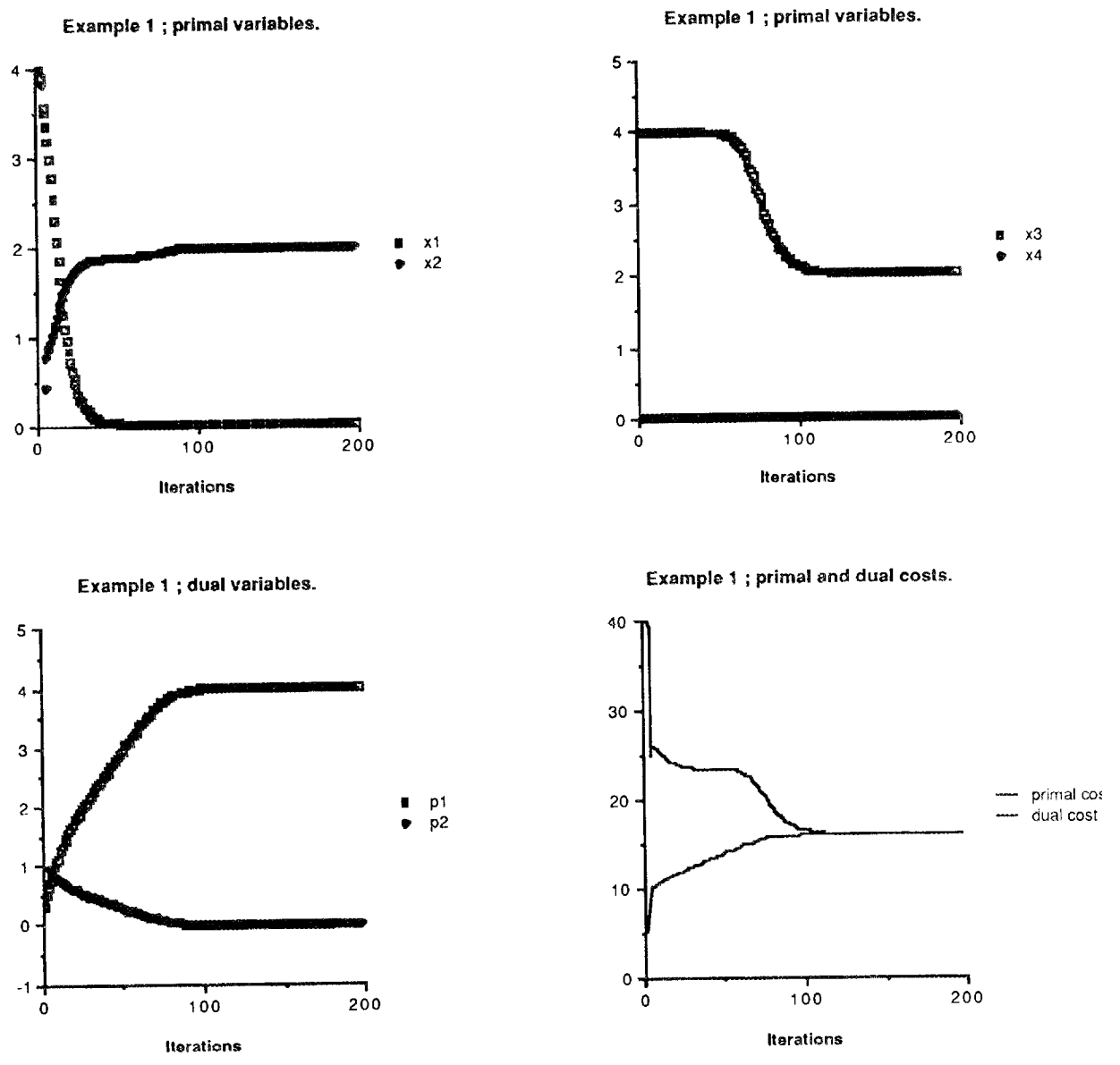


Figure 3: Trajectories of primal, dual variables and cost with the sigmoidic linear-programming algorithm.

3. Bifurcation and Chaotic Behavior of Sigmoidic Algorithms

3.1 Nonlinear Updating

Let us study the behavior of the sigmoidic updating for a single-valued function F(x) and suppose that there is no constraint, i.e. $Q \equiv 0$. In that special case, we have proposed in (9) the following "soft-limiter" updating:

$$x^{k+1} = 2\, x^k \;\; g_\mu\,(F'(x^k))$$

with $g_\mu(.)$ given by (4). We thus have

$$x^{k+1} = 2\, x^k \;\; [1 + \exp(\mu F'(x^k))]^{-1} \overset{\text{def}}{=} G(x^k,\mu)$$

We compute $G'(x,\mu) = 2\dfrac{1 + e^{\mu F'(x)} - \mu\, x\, F''(x)\, e^{\mu F'(x)}}{[\,1 + e^{\mu F'(x)}\,]^2}$.

For any critical point x* of F, since $F'(x^*) = 0$, this expression reduces to

$$G'(x^*,\mu) = 1 - \frac{\mu\, x^*}{2}\; F''(x^*).$$

We can conclude that:

- if x* is equal to zero, or $F''(x^*) = 0$, then $G'(x^*,\mu) = 1$ and the algorithm may converge to a set of limit-points containing x*.
- if $x^* > 0$ and F is locally strictly convex in x*, then $F''(x^*) > 0$, and:
  - for $\mu < [\frac{x^*}{4} F''(x^*)]^{-1}$, we have $G'(x^*,\mu) < 1$ and thus convergence to x*.
  - for $\mu \geq [\frac{x^*}{4} F''(x^*)]^{-1}$, some bifurcation and chaos may appear.

For the simple example given in the Introduction, i.e. $F(x) = \frac{1}{2}(x-1)^2$, the updating takes the form

$$x^{k+1} = G(x^k,\mu) = x^k \,.\, \frac{2}{1 + \exp(\mu(x^k-1))},$$

and $x^* = 1$ and $F''(x^*) = 1$. Thus the critical value of μ is 4. For μ larger than 4, the analytical study of the bifurcation sequences is more involved than in the "hard-limiter" case. However, we can turn to numerical experimentations. Figure 4 illustrates that:

- while μ is smaller than $\mu^0 \approx 4.0$, the sequence converges to 1.
- if μ belongs to the interval $[\mu^0, \mu^1]$, with $\mu^1 \approx 5.1$, there is a period-two limit cycle. This is the first bifurcation.
- for $\mu > \mu^1$, we enter a short region of period-four bifurcation and then a "chaotic scenario", similar to the features of nonlinear maps like Feigenbaum's quadratic map (See Refs. [7, 8]). In particular, for $\mu \approx 6.50$ and $\mu \approx 6.66$, we have respectively a period-three and a period-six limit cycle.

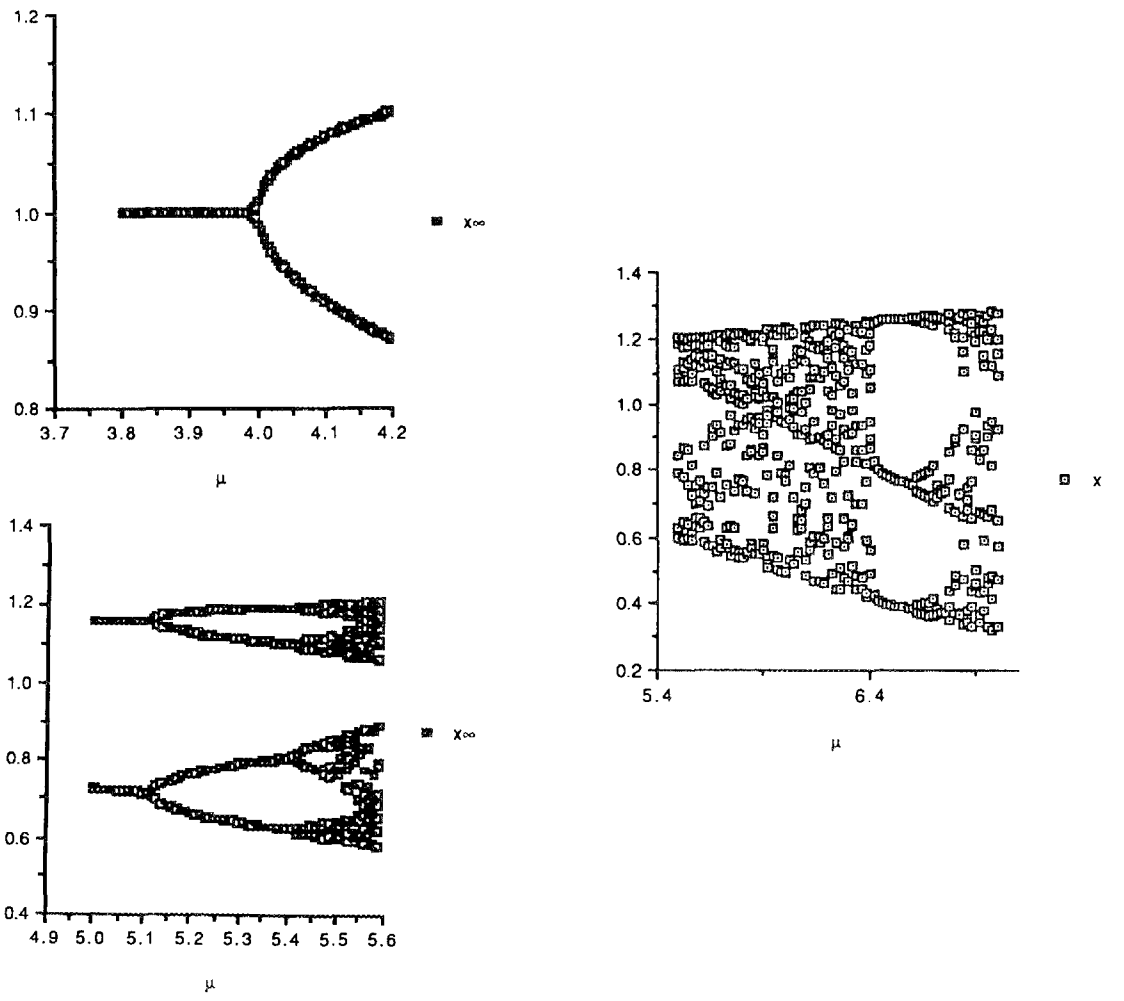


Figure 4: Bifurcation diagram. Limits of the algorithm vs the values of μ.

These experiments also confirm two computationally interesting properties already mentioned: (a) the speed of convergence (to a fixed point or to a limit-cycle) increases with the value of μ (this follows immediately from the expression for $G'(x,\mu)$); (b) the mean value of the points of the limit cycles is, for small even periods (i.e. 2 or 4), an approximation of the sought solution (Figure 5). These properties suggest that a rapid scanning of several values of μ should quickly enable one to obtain the best compromise between speed of convergence and accuracy of the solution. For the sake of completeness, we give in Figure 6 the first five iterates of G for $\mu \approx 6$.

Lyapunov Exponents.

For a one-dimensional mapping like G(x), the unique Lyapunov exponent is usually defined by [7,10]: $L(\mu) = \mathrm{Lim}_{k\to\infty} \frac{1}{k} \mathrm{Log}\, |(G^k)'(x^0)|$, with an initial point x^0 that does not lead to a pathological behavior (fixed point, repeller, etc.). According to the chain-rule, we have $L(\mu) = \mathrm{Lim}_{k\to\infty} \frac{1}{k} \Sigma_l \mathrm{Log}\, |(G^l)'(x^l)|$, with $x^l = G^l(x^0)$. Applied as such, this formula does not provide good numerical results since, due to the boundedness of the attractor, we do not have the expected exponential divergence for a large number of iterations. Instead, we observe only a transient divergence. Indeed, suppose that we start with two very close points x_0 and x_1. If, at a certain stage, the iterates x_0^k and x_1^k are such that $| x_0^k - x_1^k | \approx M = \mathrm{Sup}_x\, G(x)$, then clearly one of them is close to 0 and the other one is close to M. Let us assume $x_0^k \approx 0$ and $x_1^k \approx M$. Then $x_0^{k+1} \approx 0$ and $x_1^{k+1} \approx G(M) < M$, which means that the iterates do not diverge any more. This is illustrated by Figure 7. We have computed the Lyapunov exponents with the following technique (proposed in [9]):

- start the algorithm with two very close points x_0 and x_1
- record the iterates x_0^k and x_1^k as long as these iterates diverge
- estimate the exponent with a least mean-square algorithm.

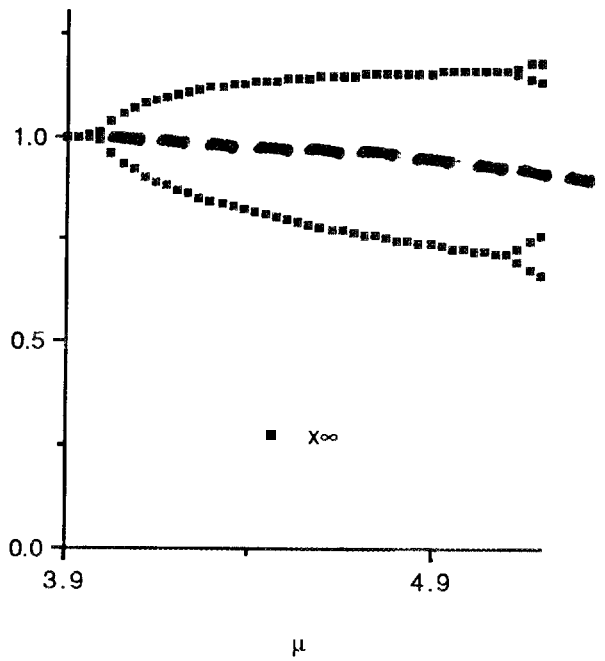
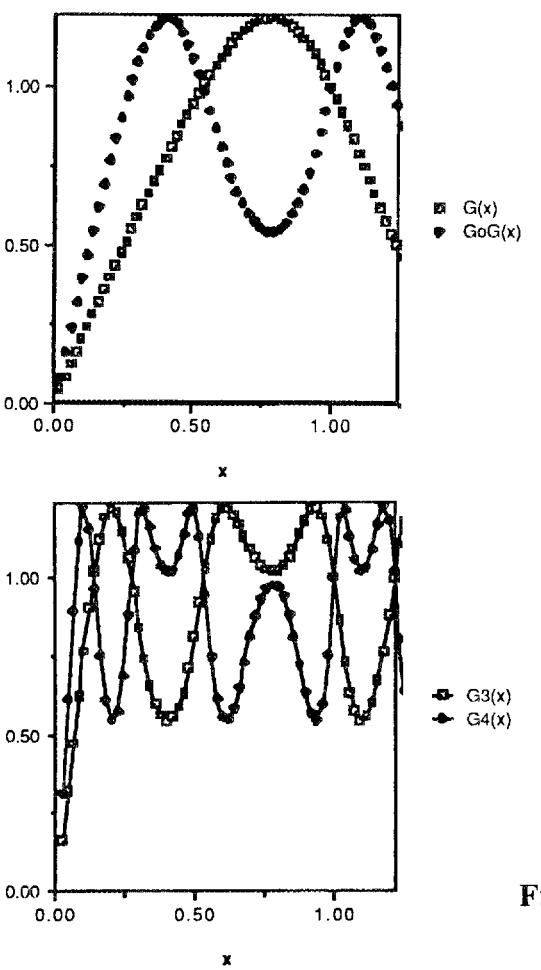


Figure 5: The mean of the small period limit cycles is an approximation of x*.



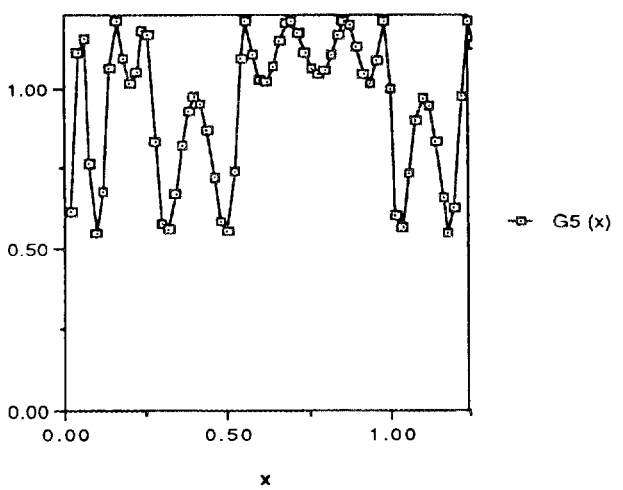


Figure 6: First five iterates of G(x) for μ = 6.

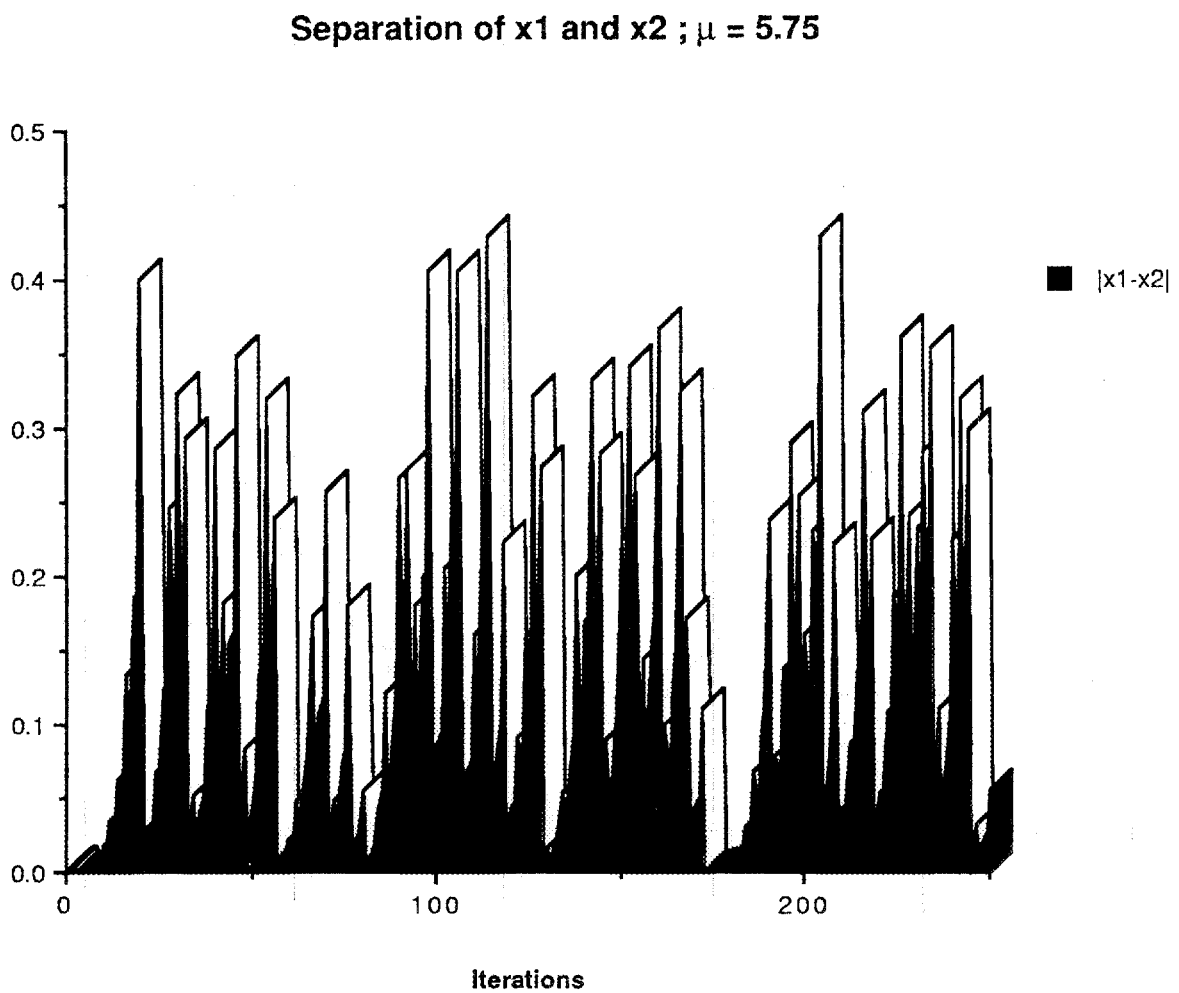


Figure 7: Successive iterates of two very close initial points.

Figure 8 indicates an estimation of the Lyapunov exponent associated with G when $\mu = 4.5$ (stable period 2 limit cycle, the exponent is negative) and $\mu = 5.75$ (chaotic behavior, the exponent is positive).

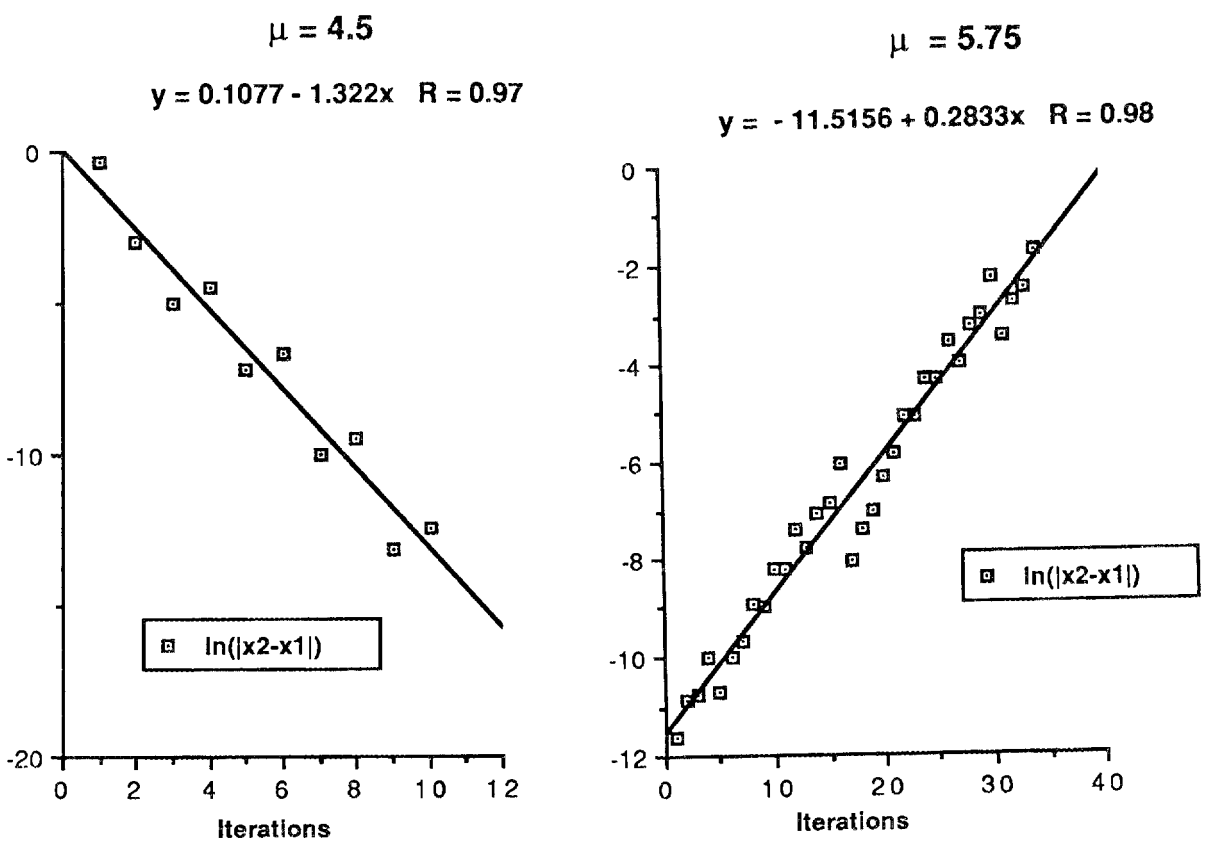


Figure 8: Lyapunov exponents.

Boundaries of the Bifurcation Map

We have also computed the boundaries of the bifurcation map shown in Figure 9. As noticed previously [11, 12], these boundaries are the successive images of the critical points of the iterates of G(x). It can be shown easily that the critical point for a given μ is obtained through the fixed point iteration: $x = 1+\frac{1}{\mu}\log(\frac{1}{\mu(x-1)})$ the convergence of which takes less than 25 iterations.

Limiting Distribution.

For a given μ, we denote by M a number larger than $\mathrm{Sup}_x\, G(x,\mu)$.

We consider $h(x) = \frac{1}{M}\chi_{[0\text{-}M]}(x)$, the uniform density on the interval [0,M]. If we associate with each point x in [0,M] its image under G, we get a new density which can be computed by the application of the Perron-Frobenius operator [13]. This operator, uniquely associated with G, is given by:

$$Ph(x) = \frac{d}{dx}\int_{G^{-1}([0,x])} h(u)\, du$$

We call $y_1(x)$ and $y_2(x)$ the two counterimages of a given x (see Figure 10), and write:

$$Ph(x) = \frac{d}{dx}\left\{ \int_0^{y_1(x)} h(u)\, du + \int_{y_2(x)}^{0} h(u)\, du \right\}$$

Or $Ph(x) = y_1'(x)\, h(y_1(x)) - y_2'(x)\, h(y_2(x))$, where the derivatives are computed numerically. After several applications of the Perron-Frobenius operator to a given density measure on [0,M], the successive images tend to approach rapidly a limiting density h* that depends on μ and is the solution of $Ph^* = h^*$. We have computed the first and second images of the uniform density on [0,M] for $\mu = 4.5$ (period-2 limit cycle), 5.75 (chaotic behavior) and 6.5 (period-3 limit cycle). For the first and third examples, the limiting

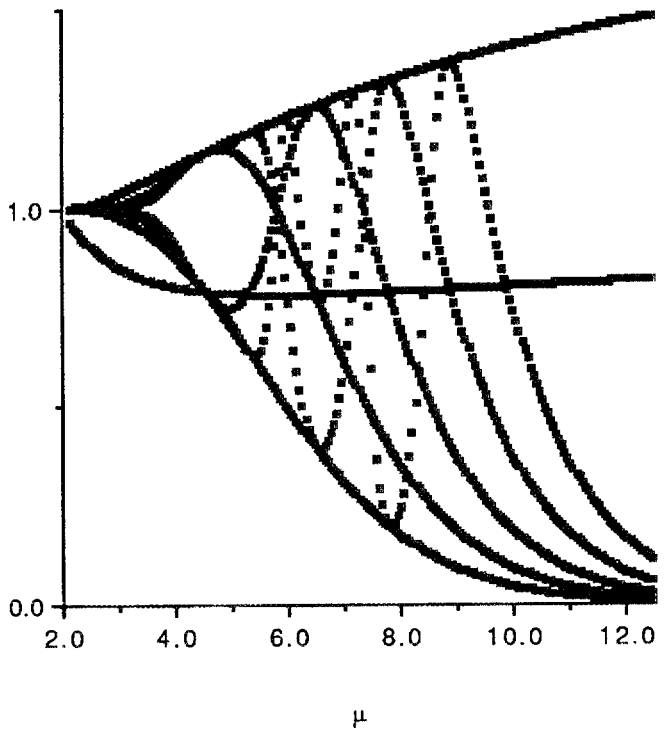


Figure 9: Boundaries of the bifurcation map.

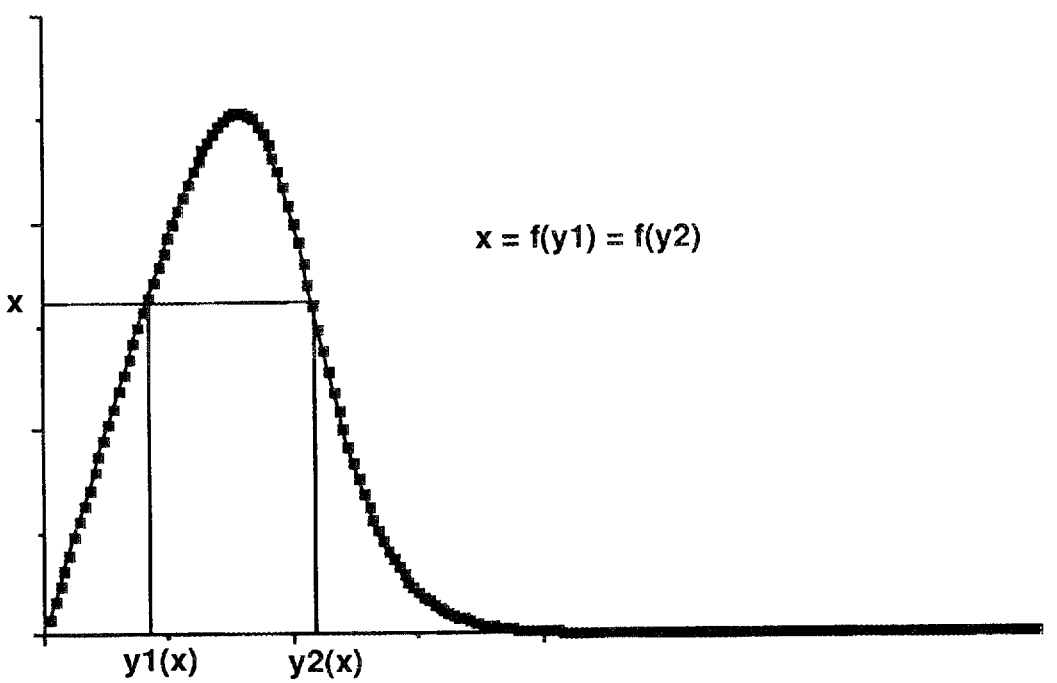


Figure 10: Counterimages of x.

distribution is obtained very quickly ($P^4 h \approx P^5 h$, see Figures 11 to 14). In the chaotic case, convergence is not obtained after 5 iterations (Figures 12 and 14) and the density function continues to generate new peaks which correspond to the images of the critical points of G(x).

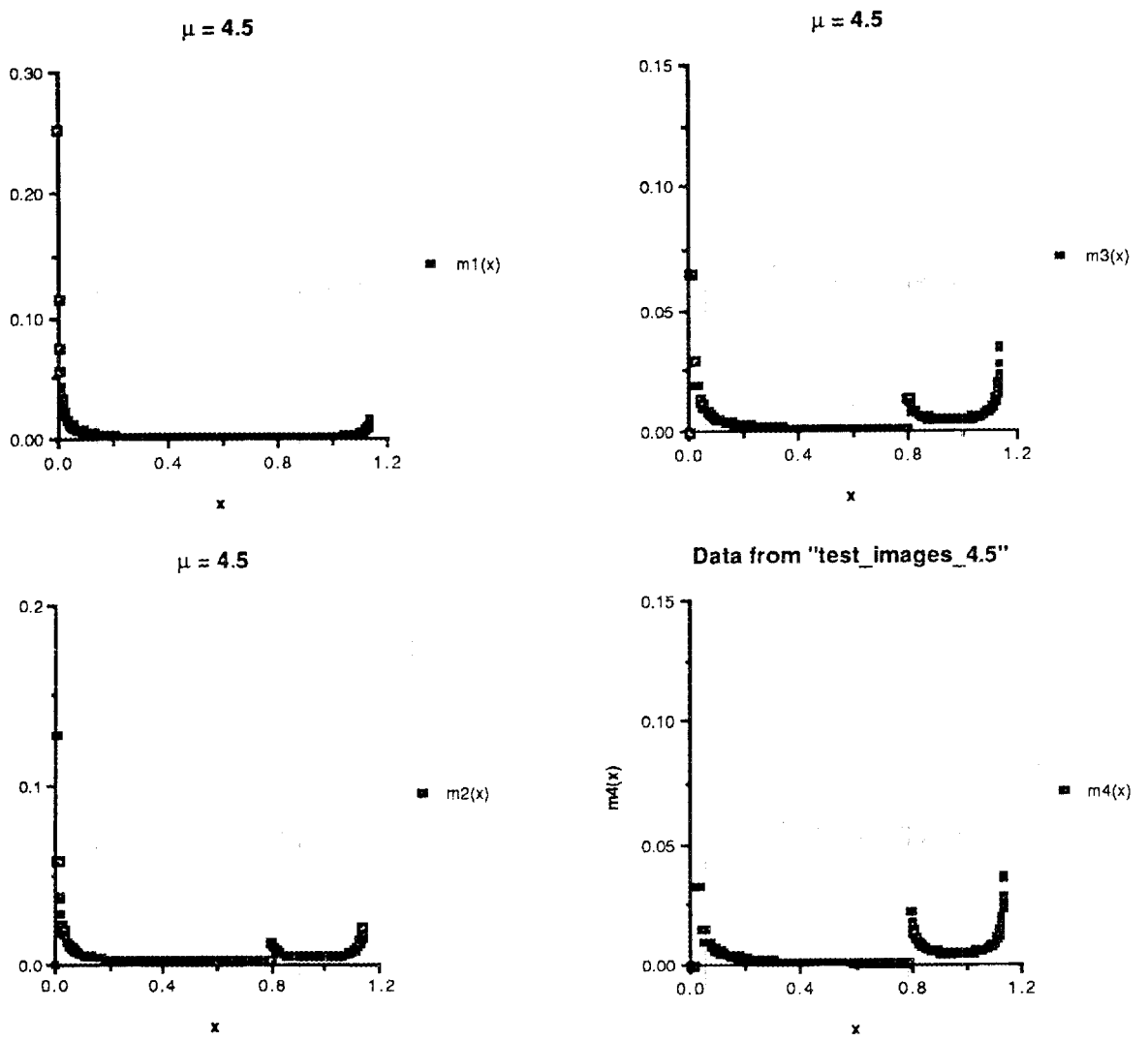


Figure 11: First four iterates of the Perron-Frobenius operator, $\mu = 4.5$

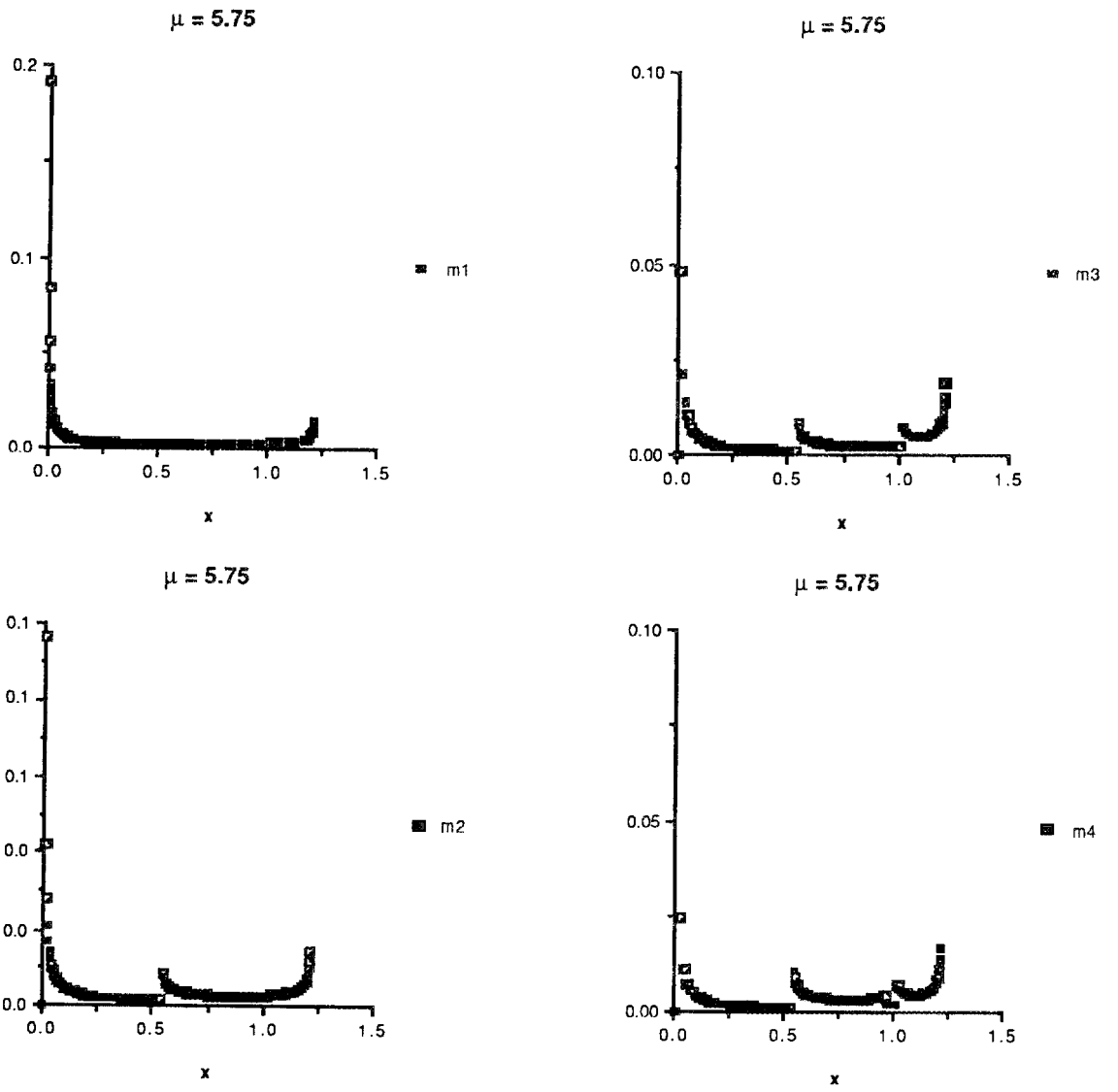


Figure 12: First four iterates of the Perron-Frobenius operator, $\mu = 5.75$

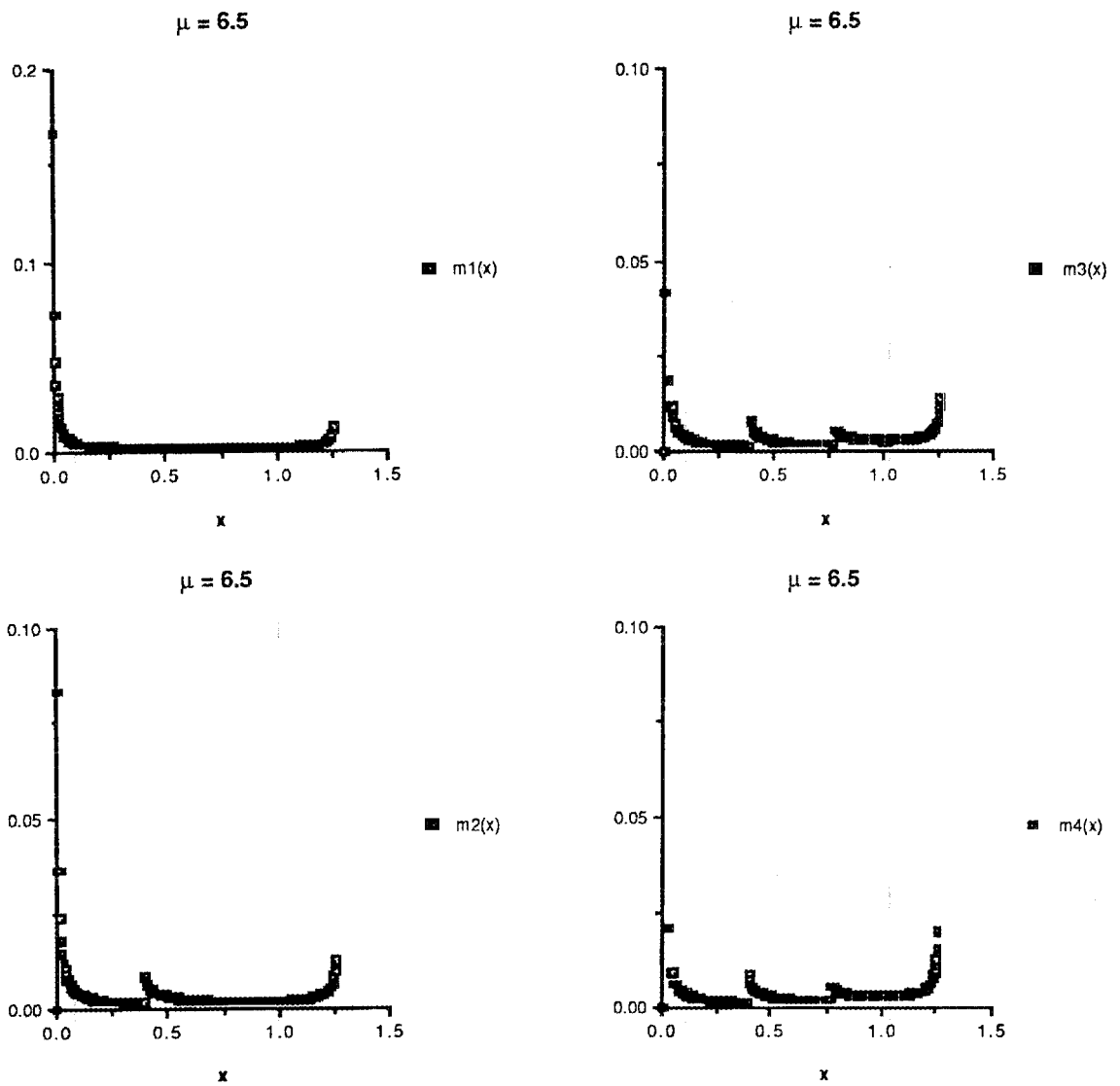


Figure 13: First four iterates of the Perron-Frobenius operator, $\mu = 6.5$

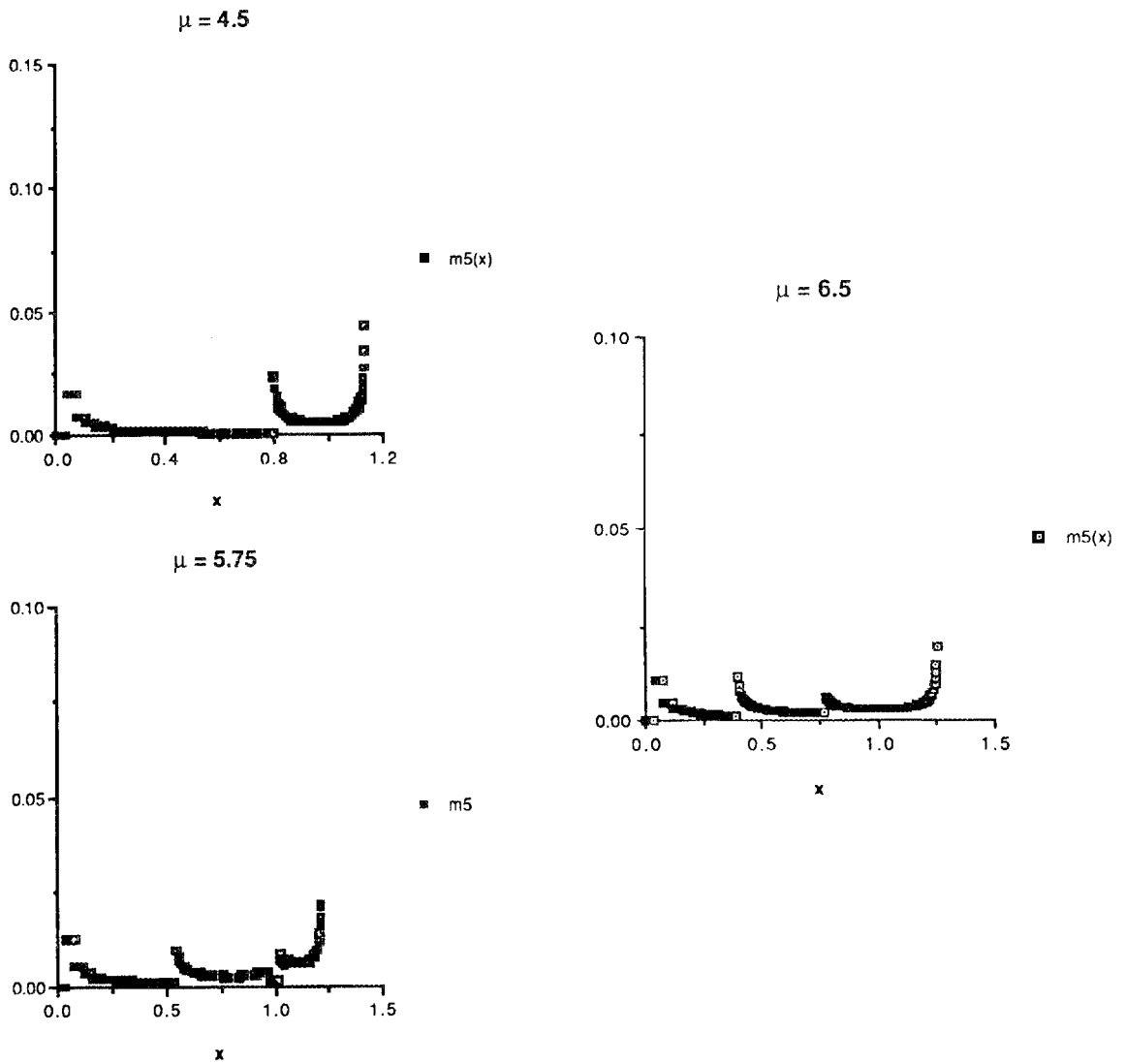


Figure 14: Fifth iterate for $\mu = 4.5$, 5.75 and 6.5.

3.2 Linear Updating.

We have shown in Section 3 the behavior of the linear-programming algorithm when ε and μ are small enough. We wish to show here that even if those parameters are not chosen optimally, one can get very quickly some useful information concerning the actual solution. If, instead of taking for μ and ε some values which ensure convergence, we take a small ε and progressively increase μ, a scenario of bifurcations and chaos appears. In Figure 15, we have typical trajectories for a large μ. In Figure 16, we also depict the bifurcation diagram for x_3 and the least square estimation of x_3 vs. $\mu \in [10,16]$ which happens to be a quasi-horizontal line $x_3 \approx a\,\mu + b$ with $a \approx 2.01$ and $b \approx -0.001$. We conclude that as μ increases, the convergence speed to the limit cycle is improved and we still get, on a large range of μ (for a given ε), a fair approximation of x^*. We also note that the quality of the approximation is better than in the nonlinear case.

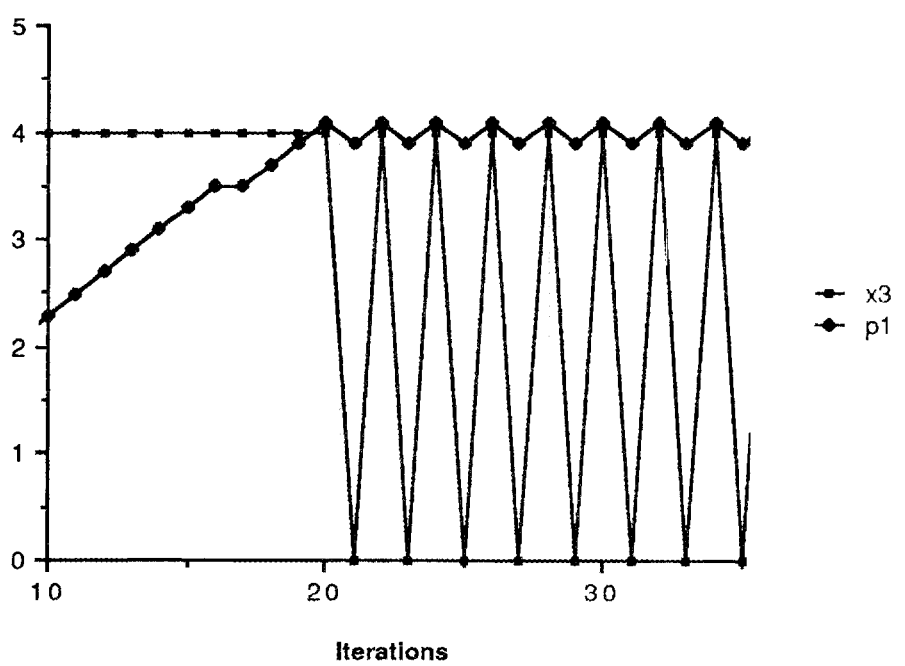


Figure 15: An example of bifurcation. Typical trajectories for large μ ($\mu > 100$).

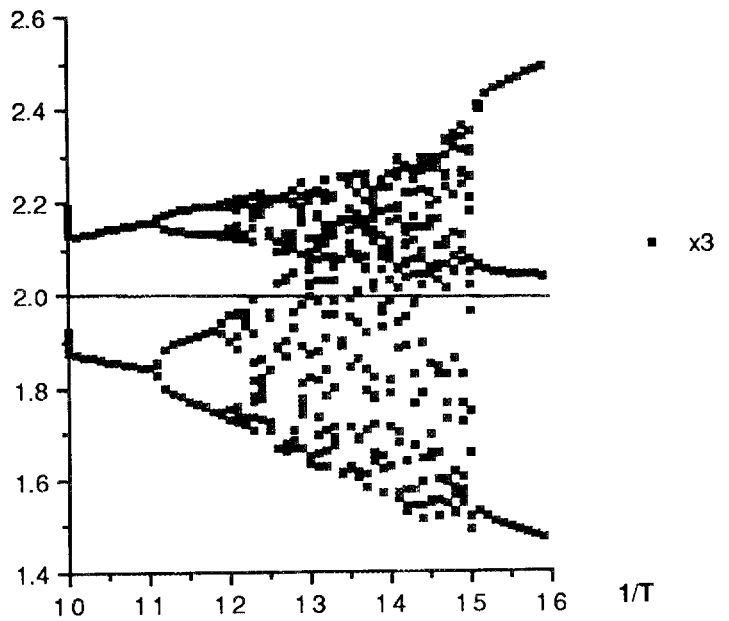


Figure 16: Bifurcation diagram and least squares estimation of x_3 vs. μ.

4. Summary

We have presented here a class of optimization algorithms based on sigmoidic updatings. In different variants, the algorithms can be applied to linear and nonlinear problems with or without constraints. All variants are in fact nonlinear multidimensional maps and as such they may display bifurcations and chaotic regimes for certain ranges of parameters. Although a priori these features are not desirable in numerical schemes, we have shown that they enhance stability regions, provide meaningful (even if partial) information on the solution of the optimization problem, and increase the speed of convergence.

Acknowledgments

This research was partially sponsored by the Office of Military Applications and the Office of Basic Energy Sciences, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, by OJCS, J6 Directorate, U.S. Department of Defense under Interagency Agreement DOE #1612-1657-A1 with Martin Marietta Energy Systems, and by an ORNL Postgraduate Research Appointment administered by Oak Ridge Associated Universities.

We gratefully thank R. C. Mann for his critical reading of the manuscript and his continuous support.

References

[1] J.-C. Culioli and V. Protopopescu, "A New Algorithm for Linear Programming that is Easy to Implement: Application to the Transportation Problem," ORNL TM 10929, ORNL Report, September 1988.

[2] J.-C. Culioli and V. Protopopescu, "A New Class of Optimization Algorithms with Bifurcations and Chaotic Behavior", presented at the SIAM National Meeting, July , 1988, Minneapolis, MN.

[3] J.-C. Culioli and V. Protopopescu, "Stability and Chaos in a Numerical Scheme Inspired by Neural Networks", workshop of the AAAS, Corvallis, Oregon, June 1988.

[4] J.J. Hopfield and D.W. Tank, "Neural Computation of Decisions in Optimization Problems", Biol. Cybern. **52** 141 (1985).

[5] K. J. Arrow, L. Hurwicz, and H. Uzawa, Studies in Linear and Nonlinear Programming, Stanford University Press, 1972.

[6] G. B. Dantzig, Linear programming and extensions, Princeton University Press, 1964.

[7] P. Collet and J.-P. Eckmann, Iterated Maps on the Interval as Dynamical Systems, Birkhaüser, Boston, 1980.

[8] M. J. Feigenbaum, "Universal Behavior in Nonlinear Systems", Los Alamos Science, **1**, 4 (1980).

[9] P. Berge, Y. Pomeau and C. Vidal, Order within Chaos: Towards a Deterministic Approach to Turbulence, John Wiley & Sons and Hermann, Paris, France, 1984.

[10] Dimensions and Entropies in Chaotic Systems, G. Mayer-Kress Ed., Proc. of Int. Workshop at the Pecos River Branch, New Mexico ; Springer-Verlag , Berlin, 1986.

[11] R. V. Jensen and C. R. Myers, "Images of the Critical Points of Nonlinear Maps", Phys. Rev. A **32** 1222 (1985).

[12] E. M. Oblow, "Supertracks, Supertrack Functions and Chaos in the Quadratic Map", Phys. Lett. A **128** 406 (1988).

[13] A. Lasota and M. C. Mackey, Probabilistic Properties of Deterministic Systems, Cambridge University Press, Cambridge, 1985.

Appendix

Implicit transformation due to the nonlinear sigmoidic updating.

Suppose we minimize a convex but not necessarily strictly convex real function F(x). We assume in the following discussion than the sought optimal value x* is positive ($x^* \geq 0$). A standard gradient descent algorithm would be:

- choose an initial point x^0 and a threshold Δ ;
- at step k, compute $x^{k+1} = x^k - \beta F'(x^k)$ [GA]
- stop if $|F'(x^k)| < \Delta$.

The sigmoidic algorithm is

- choose an initial point x^0 and a threshold Δ ;
- at step k, compute $x^{k+1} = x^k \dfrac{2}{1 + \exp(\mu F'(x^k))}$ [SA]
- stop if $|F'(x^k)| < \Delta$.

We shall show that [SA] is equivalent to a certain [GA] applied to another function than F but with the same critical set, i.e. there exists H(x) such that:

$$x^k - \beta H'(x^k) = x^k \frac{2}{1 + \exp(\mu F'(x^k))} \quad , \text{ for every } k.$$

A sufficient condition is

$$\beta H'(x) = x \tanh\left(\frac{\mu F'(x)}{2}\right) \qquad (\Omega)$$

Indeed, given F, it is always possible to find H (modulo a constant function) such that the set of critical points of H is the union of zero with the set of critical points of F.

Now, for given $\beta > 0$ and $\mu > 0$, if we consider the mapping Ω which associates to any real function F the function H defined by (Ω) and by $H(x^*) = F(x^*)$, one can be interested in how some properties such as fixed points, convexification, are transformed by Ω.

For example, let us take $F(x) = a\,x + b$, with $a > 0$ so that Min $F(x)$ has the positive solution $x^* = 0$. $H=\Omega F$ is defined by: $\beta\, H'(x) = x \tanh(\frac{\mu a}{2})$ and $H(0) = F(0) = b$.

Thus $H(x) = \frac{x^2}{2\,\beta}\tanh(\frac{\mu a}{2}) + b$. The function F has no critical point and is not convex.

The gradient algorithm cannot work properly with $\Delta < a$. On the contrary, H is quadratic, extremely steep ($\beta \ll 1$) and one can expect a good numerical behavior.

Let us consider now $F(x) = |a\,x - b|$ with $a > 0$ and $b > 0$ and $x^* = \frac{b}{a}$.

We have: for $x > \frac{b}{a}$, $\beta\, H'(x) = x \tanh(\frac{\mu a}{2})$

for $x < \frac{b}{a}$, $\beta\, H'(x) = -\,x \tanh(\frac{\mu a}{2})$

and $H(\frac{b}{a}) = F(\frac{b}{a}) = 0$.

Hence,

for $x > \frac{b}{a}$, $H(x) = [\,x^2 - (\frac{b}{a})^2\,]\ \dfrac{\tanh(\frac{\mu a}{2})}{2\,\beta}$

for $x < \frac{b}{a}$, $H(x) = -\,[\,x^2 - (\frac{b}{a})^2\,]\ \dfrac{\tanh(\frac{\mu a}{2})}{2\,\beta}$

The function H(x) is given in Figure 17 for $\beta = 0.01$, $a = 1$, $b = 5$, $\mu = 4.5$.

At last, for $F(x) = \frac{1}{2}(x-a)^2$ with $a \geq 0$, we have $\beta\, H'(x) = x \tanh(\mu(x-a))$. Since H(x) is difficult to integrate directly, let us discuss its properties from the expression of its derivative. Two cases can be considered:

- when $x \approx a$, $x \tanh(\mu(x-a))$ can be approximated by $\mu\, x\,(x-a)$ which is a quadratic map and is positive for $x > a$ and negative in $[0,a]$, which provides exactly the attracting and repelling effects that one would expect from a good algorithm.
- when $x \gg a$ (recall that x^0 and a are assumed positive), the approximation becomes simply $\beta\, H'(x) = x$, which is equivalent to the classic gradient of F(x).

Figure 18 shows a picture of H'(x) and H(x) in the case $\beta = 0.01$, $\mu = 4.5$ and $a = 1.00$.

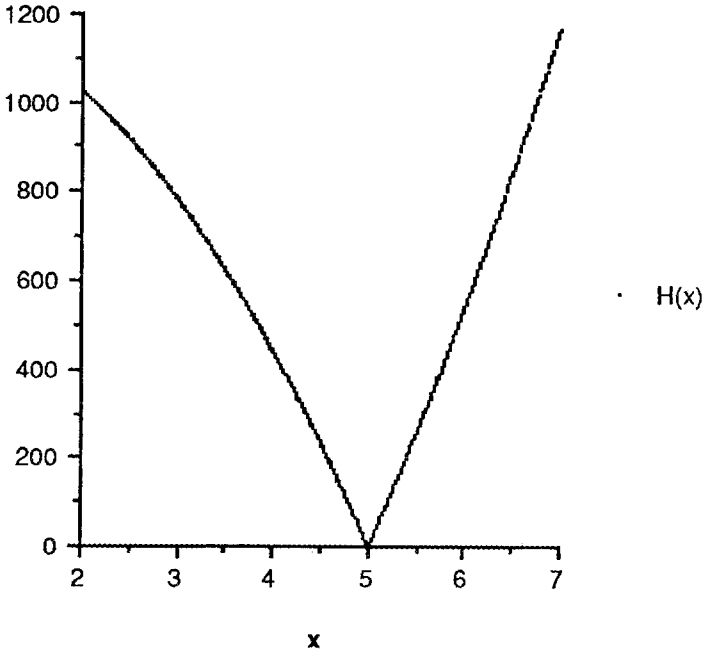


Figure 17: H(x) when F(x) = | x - 5 |, $\mu = 4.5$, and $\beta = 0.01$.

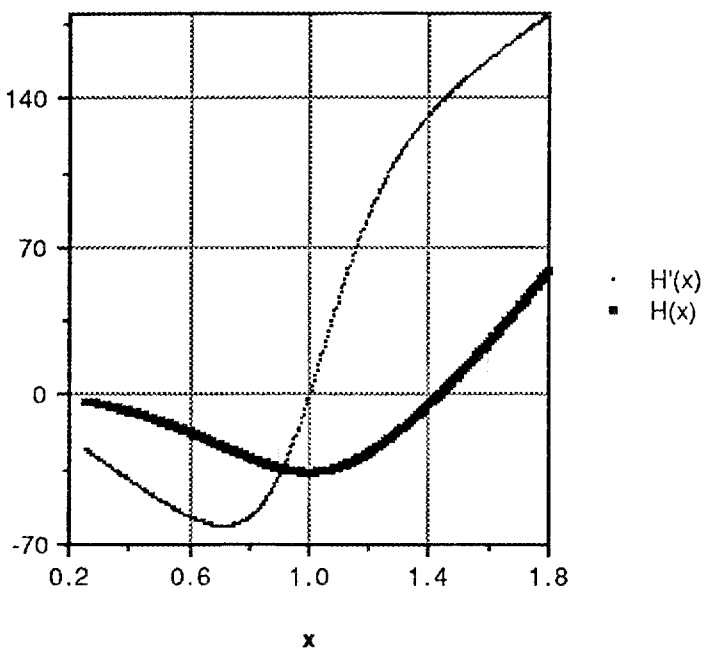


Figure 18: H(x) and H'(x) for $\mu = 4.5$, $\beta = 0.01$, and a = 1.00.

INTERNAL DISTRIBUTION

1. Y. Y. Azmy
2-6. J. C. Culioli
7. F. C. Difilippo
8. G. de Saussure
9. C. W. Glover
10. J. Han
11. D. T. Ingersoll
12. J. P. Jones
13. S. Lenhart
14. G. E. Liepins
15. F. C. Maienschein
16–20. R. C. Mann
21. F. G. Perey
22. R. B. Perez
23. E. M. Oblow
24. F. G. Pin
25–29. V. Protopopescu
30. R. S. Solanki
31. R. T. Santoro
32. C. R. Weisbin
33. A. Zucker
34. J. J. Dorning (consultant)
35. R. Haralick (consultant)
36. Central Research Library
37. Y-12 Technical Library
38–39. Laboratory Records
40. EPMD Reports Office
41. ORNL Patent Office
42. Laboratory Records–RC

EXTERNAL DISTRIBUTION

43. Dr. P. G. Anderson, School of Computer Science and Technology, Rochester Institute of Technology, One Lomb Memorial Drive, P.O. Box 9887, Rochester, New York 14623-0887

44. Professor J. P. Aubin, CEREMADE, Universite de Paris-Dauphine, Place du Marechal Delattre de Tassigny, 75017 Paris, France

45. Dr. J. M. Barry, Australian Nuclear Science & Technology Organisation, Applied Mathematics and Computing Division, Lucas Heights Research Laboratories, New Illawarra Road, Lucas Heights, NSW 2234

46. Dr. Griff L. Bilbro, North Carolina State University, Box 7914, Raleigh, NC 27695-7914

47. Professor Guy Cohen, ENSMP-Centre d'Automatique, 35, rue St. Honore, 77305 Fontainebleau, France

48. N. E. Cotter, Department of Electrical Engineering, The University of Utah, 3262 Merrill Engineering Bldg., Salt Lake City, Utah 84112

49. Dr. David I. Feinstein, Boeing Electronics Company, High Technology Center, P.O. Box 24969, MS 7J-24, Seattle, Washington 98124-6269

50. Professor Walter Freeman, University of California, Berkeley, CA 94720.

51. G. Giandonato, CSELT, Via Reiss Romoli 279, 10148 – Torino, Italy

52. Dr. Daniel Goryn, University of Minnesota, Dept. of Electrical Engineering, EE 4-178, 200 Union Street, Minneapolis, MN 55455

53. Professor D. M. Himmelbau, Department of Chemical Engineering, The University of Texas at Austin, Austin, TX 7812-1062

54. Professor Morris W. Hirsch, Department of Mathematics, University of California, Berkeley, CA 94720

55. Paul Hollis, Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911

56. Roxane Kamen, Code 421, NOSC, San Diego, CA 92152

57. Dr. Arthur Karler, Karler Laboratories, P.O. Box 373, Berkeley, CA 94701

58. George Komen, Sandia National Laboratory, Division 1274, P.O. Box 5800, Albequerque, NM 87185

59. Professor J. F. Magnan, Department of Mathematics, The Florida State University, Tallahassee, FL 32306

60. A. J. Mimms, Douglas Aircraft Company, IMC–35-29, 3855 Lakewood Blvd., Long Beach, CA 90846

61. Joseph Naft, Computer Aided Design Laboratory, The University of Maryland, Engineering Research Center, College Park, MD 20742

62. Dr. Charles M. Patton, Bldg. 3U-L9, Hewlett-Packard Company, 100 NE Circle Blvd., Corvallis, OR 97331

63. Robert F. Richbourg, Department of Geography and Computer Science, United State Military Academy, OAIAE (MADN-B), West Point, NY 10996-1695

64. Dr. Asim Roy, Department of Decision and Information Systems, Arizona State University, 1401 E. Brentrup Drive, Tempe, AZ 85287

65. Dr. L. Michael Santi, Memphis State University, Mechanical Engineering Dept.,Memphis, Tennessee 38152

66. Dr. Fabio Schoen, Università degli Studi di Milano, Dipartimento di Scienze dell'Informazione, Via Moretto da Brescia, 9, 20133 Milano, Italy

67. Professor Richard M. Schori, Oregon State University, Dept. of Mathematics, Corvallis, OR 97331-4605

68. Dr. Steve Serbin, Dept. of Mathematics, The University of Tennessee, Ayres Hall, Knoxville, TN 37996-2100

69. Dr. John Shepanski, Data Systems Laboratory, One Space Park, Redondo Beach, CA 90278

70. Professor William Simon, University of Rochester Medical Center, Dept. of Biophysics, 601 Elmwood Avenue, Rochester, NY 14642

71. Patrick K. Simpson, Artificial Intelligence Center, General Dynamics, P.O. Box 85310, San Diego, CA 92138 MZ 7202K

72. Gene Tagliarini, Clemson University, Department of Computer Science, 435 Nursing Building, Clemson, SC 29634-1906

73. Dr. B. W. Wah, National Science Foundation, 1800 G Street, NW, Washington, DC 20550

74. Dr. James A. White, Honeywell, Inc., 3660 Technology Drive, Minneapolis, MN 55418

75. Dr. George W. Whitson, III, Department of Computer Science, The University of Texas at Tyler, 3900 University Blvd., Tyler, TX 75701

76. Dr. Jack H. Winters, AT&T Bell Laboratories, Research Department, Holmdel, NJ 07733

77. Dr. H. C. Yee, NASA – Ames Research Center, MS: 202A-1, Moffett Field, CA 94035

78. Professor Yechiam Yemini, 450 Computer Science Bldg., Columbia University, New York, NY 10027

79. D. Zwillinger, The MITRE Corporation, Burlington Road, MS: E025, Bedford, MA 01730-0208

80. Office of the Assistant Manager, Energy Research and Development, DOE-ORO, Oak Ridge, TN 37831

81–90. Office of Scientific and Technical Information, US/DOE, P.O. Box 62, Oak Ridge, TN 37831