The Liar and Related Paradoxes: Fuzzy Truth Value Assignment for Collections of Self-Referential Sentences

K. Vezerides and Ath. Kehagias
vezerid@ac.anatolia.edu.gr kehagiat@gen.auth.gr
http://users.auth.gr/ ˜kehagiat

September 24, 2003

Abstract

We study self-referential sentences of the type related to the *Liar paradox*. In particular, we consider the problem of assigning consistent fuzzy truth values to *collections of self-referential sentences*. We show that the problem can be reduced to the solution of a system of nonlinear equations. Furthermore, we prove that, under mild conditions, such a system always has a solution (i.e. a consistent truth value assignment) and that, for a particular implementation of logical "and", "or" and "negation", the "mid-point" solution is always consistent. Next we turn to computational issues and present several truth-value assignment algorithms; we argue that these algorithms can be understood as generalized sequential reasoning. In an Appendix we present a large number of examples of self-referential collections (including the Liar and the Strengthened Liar), we formulate the corresponding truth value equations and solve them analytically and/ or numerically.

Keywords. Self-reference, liar paradox, truth, fuzzy logic, nonlinear equations, root finding algorithms.

1 Introduction

Self referential sentences are sentences which talk about themselves. Sentences of this type often generate logical *paradoxes*, the study of which goes back to the ancient Greeks. Many approaches have been proposed to neutralize the paradoxes. In this paper we study the problem from the point of fuzzy logic.

The prototypical example of a self-referential sentence is the *Liar Sentence*: Epimenides the Cretan has supposedly uttered the following sentence:

"All Cretans are liars."

The Liar sentence generates a paradox. To see this consider whether the sentence is true or false. Following a somewhat loose chain of reasoning, let us assume "All Cretans are liars" to mean that everything a Cretan says is not true. Since Epimenides is a Cretan, his statement that "All Cretans are liars" is not true; if we take "not true" to mean "false", then the *opposite* of "All Cretans are liars" must be true; and if we take this opposite to be "All Cretans are truth-tellers", then what Epimenides says must be true, i.e. it is true that "All Cretans are liars"; but then it must be the case that what Epimenides says is false. It seems that we have entered a vicious circle, concluding first that Epimenides' statement is false, then that it is true, then again that it is false and so on ad infinitum.

The above reasoning is not rigorous, since, for example, "All Cretans are liars" does not necessarily mean that all Cretans *always* utter false statements; similarly the negation of "All Cretans are liars" is "*Some* Cretans are truth-tellers" and so on.

But the paradox also appears under more exact reasoning. Consider the following sentence:

"This sentence is false."

In this case, if we assume the sentence to be true, then what it says must hold, i.e. the sentence must be false. But then its opposite must be true, i.e. it must be true that "This sentence is true". But then it is true that "This sentence is false" and we have again entered an oscillation between two conclusions: first that the sentence is false, then that it is true. Notice that we would enter a similar oscillation if we started by assuming the sentence to be false; then we would conclude that the sentence is true, which would mean that the sentence is false etc. A similar effect can be obtained using two sentences. The following pair is the so-called *inconsistent dualist*:

$$A_1 = \text{“Sentence } A_2 \text{ is true”}$$
$$A_2 = \text{“Sentence } A_1 \text{ is false”}.$$

If A_1 is true, then A_2 is also true; but then A_1 must be false and so A_2 must be false and so on ad infinitum.

Note that self-reference does not *necessarily* lead to paradox. The following pair is the so-called *consistent dualist*:

$$A_1 = \text{“Sentence } A_2 \text{ is true”}$$
$$A_2 = \text{“Sentence } A_1 \text{ is true”}.$$

If A_1 is true, then A_2 is also true, which confirms that A_1 is true. In short, accepting that A_1 and A_2 are both true is perfectly consistent. However, note that we could equally well assume that A_1 is false, which would mean that A_2 is also false, which confirms that A_1 is false. In short, we can also accept that A_1 and A_2 are both false. Is this a problem? We will consider some possible answers in the sequel.

We have already mentioned that self-referential sentences have been studied extensively, and from several different points of view. In this section we will only discuss some work which is directly related to the current paper; a more extensive discussion of the literature will be presented in Section 5.

The application of fuzzy logic to the Liar paradox goes back to a paper by Zadeh [51]; in summary, he resolves the paradox by assigning to the Liar sentence a truth value of 1/2. Following Zadeh's paper, several authors have analyzed self-reference using fuzzy logic (for more details see Section 5).

The current paper has been heavily motivated by the work of Grim and his collaborators [10, 21, 22, 41]. Grim considers collections of self-referential sentences and models the *fuzzy reasoning process* as a *dynamical system*. A method is presented to map each self-referential collection to a dynamical system which represents the reasoning process. Each sentence of the self-referential collection has a *time-evolving fuzzy truth value* which corresponds to a *state variable* of the dynamical system. Grim presents several examples of self-referential collections and studies the properties of the corresponding dynamical systems. One of the main points of [10] is that self-referential collections can generate *oscillating* or *chaotic* dynamical behavior.

Grim's formulation is an essential starting point for the current paper; but the issues we address are rather different. Our main interest is in obtaining *consistent truth value assignments*, similarly to [51] and unlike Grim who concentrates on oscillatory behavior. A more detailed understanding of our approach can be obtained by the following outline of the paper.

In Section 2 we present the logical framework which will be used for the study of self-referential sentences. In this we follow very closely Grim's formulation. In Section 3, we reduce the problem of consistent truth value assignment to the solution of a system of *algebraic*[1] nonlinear equations. Such a system will, in general, possess more than one solution. It is a rather remarkable fact that, for a very broad family of self-referential collections, the corresponding equations possess *at least* one solution, i.e. a consistent fuzzy truth value assignment is *always* possible, under some mild continuity conditions; this is the subject of our Proposition 3. In Section 4 we turn to the *computation* of the solutions. We consider this as a separate, *algorithm-dependent* sub-problem. In other words, the same system of algebraic equations can be solved by many different algorithms. We consider several such algorithms and, returning to Grim's point of view, we study the dynamics of each algorithm. In a certain sense, each such algorithm can be understood as a particular reasoning style, and "human-style", sequential reasoning of the form "if A_1 then A_2, if A_2 then not-A_1 etc." is one among many options. In Section 5 we take a brief look at the literature on self-referential sentences, paradoxes and related topics and relate it to our approach. In Section 6 we summarize and discuss our results. Finally, in the Appendix we present several specific examples of self-referential collections, formulate the corresponding equations and solve them analytically and/or numerically, by using the previously presented algorithms; we also compare the behavior of the algorithms.

2 The Logic Framework

The general object of our interest is a finite collection of M sentences which talk about each other's truth value (i.e. a self-referential collection); in particular we explore the extend to which the internal structure of such a system determines the truth values of the sentences. This question will be answered mainly in Section 3.2. In this section we introduce the logical framework which is necessary to address the problem. This framework is quite similar to the one used in [10].

We start with a finite set $\mathbf{V}_1$, the set of *1st-level elementary sentences* (also called *variables*):

$$\mathbf{V}_1 = \{A_1, A_2, ..., A_M\}.$$

From the 1st level variables we *recursively* build $\mathbf{S}_1$, the set of *1st level sentences* (also called logical *formulas*):

$$\begin{array}{c}\text{If } A_m \in \mathbf{V}_1 \text{ then } A_m \in \mathbf{S}_1 \\ \text{If } B_1, B_2 \in \mathbf{S}_1 \text{ then } B_1 \vee B_2, B_1 \wedge B_2, B_1' \in \mathbf{S}_1,\end{array} \tag{1}$$

where $\vee, \wedge, '$ are the logical operators *or, and, negation* [2]. Note that $\mathbf{V}_1 \subseteq \mathbf{S}_1$.

Next we build $\mathbf{V}_2$, the set of *2nd level elementary sentences*:

$$\mathbf{V}_2 = \{\text{“Tr}\,(B) = b\text{”} : B \in \mathbf{S}_1, b \in [0,1]\}$$

where Tr(...) is shorthand for "The truth value of ..."; in other words, "$\mathrm{Tr}(B) = b$" means "The truth value of B is b". Finally, we recursively build $\mathbf{S}_2$, the set *2nd level sentences*:

$$\begin{array}{c}\text{If } C \in \mathbf{V}_2 \text{ then } C \in \mathbf{S}_2 \\ \text{If } D_1, D_2 \in \mathbf{S}_2 \text{ then } D_1 \vee D_2, D_1 \wedge D_2, D_1' \in \mathbf{S}_2.\end{array} \tag{2}$$

[1]I.e. static, *not* time-evolving.

[2]We take the informal point of view that precedence of operators, grouping of terms etc. are well understood from the context and do not require special explanation. Similalrly, we treat the use of parentheses in a completely informal manner.

where $\vee, \wedge, '$ again stand for or, and, negation [3]. Note that $\mathbf{V}_2 \subseteq \mathbf{S}_2$.

We will occasionally use the terms *elementary truth value assessments* for the elements of $\mathbf{V}_2$ and *truth value assessments* for the elements of $\mathbf{S}_2$. Also, we define a special subset of $\mathbf{V}_2$ and the corresponding subset of $\mathbf{S}_2$ as follows. The elementary *Boolean* truth value assessments are denoted by $\widetilde{\mathbf{V}}_2$ and defined by

$$\widetilde{\mathbf{V}}_2 = \{\text{“Tr}\,(B) = b\text{”} : B \in \mathbf{S}_1, b \in \{0,1\}\}\,.$$

The *Boolean* truth value assessments are denoted by $\widetilde{\mathbf{S}}_2$ and defined recursively as follows

$$\text{If } C \in \widetilde{\mathbf{V}}_2 \text{ then } C \in \widetilde{\mathbf{S}}_2$$

$$\text{If } D_1, D_2 \in \widetilde{\mathbf{S}}_2 \text{ then } D_1 \vee D_2, D_1 \wedge D_2, D_1' \in \widetilde{\mathbf{S}}_2.$$

Again we have $\widetilde{\mathbf{V}}_2 \subseteq \widetilde{\mathbf{S}}_2$.

Obviously, we can keep building up the hierarchy of sentences, defining $\mathbf{V}_n$ in terms of $\mathbf{S}_{n-1}$ and $\mathbf{S}_n$ in terms of $\mathbf{V}_n$; but going up to $\mathbf{V}_2, \mathbf{S}_2$ will be sufficient for the purposes of this paper. Let us conclude this section by giving some examples of elements from $\mathbf{V}_1, \mathbf{S}_1, \mathbf{V}_2, \mathbf{S}_2$.

$$\begin{aligned}
&\mathbf{V}_1 : A_1, A_2, ..., A_M.\\
&\mathbf{S}_1 : A_1 \vee A_3, A_2', (A_2 \wedge A_4) \vee A_5' \text{ ... etc.}\\
&\mathbf{V}_2 : \text{“Tr}\,(A_1) = 1\text{”}, \text{ “Tr}\,(A_7') = 0\text{”}, \text{ “Tr}\,((A_2 \vee A_3) \wedge A_1) = 0.3\text{” ... etc.}\\
&\mathbf{S}_2 : \left[\text{“Tr}\,(A_1') = 0\text{”} \wedge \text{ “Tr}\,((A_1 \vee A_4) \wedge A_2) = 0.3\text{”}\right] \vee \text{“Tr}\,(A_3) = 0.8\text{” ... etc.}
\end{aligned}$$

3 Truth Value Assignment

Our main goal is to assign truth values to sentences of the 2nd level (i.e. to elements of $\mathbf{S}_2$). As it turns out we will also need, as an intermediate step, to assign truth values to sentences of the 1st level (i.e. to elements of $\mathbf{S}_1$). In this section we present two ways of achieving this. Our main interest is in the second method, presented in Section 3.2, the so-called *implicit truth* value assignment, which makes use of the self-referential nature of a particular collection. However, for purposes of comparison, in Section 3.1 we will review the “classical”, *explicit* method for truth value assignment to *non*-self-referential sentences.

3.1 Explicit Truth Value Assignment

The following explicit method of assigning truth values to elements of $\mathbf{S}_1$ is well known. We start by assigning an arbitrary truth value to every element of $\mathbf{V}_1$ (1st level variable). This is equivalent to selecting a function $x : \mathbf{V}_1 \to [0,1]$, i.e. $\forall A_m \in \mathbf{V}_1$ we have $\mathrm{Tr}(A_m) = x(A_m)$; for the sake of brevity we will henceforth use the simpler notation

$$\forall A_m \in \mathbf{V}_1 : \mathrm{Tr}\,(A_m) = x_m.$$

Next, take any $B \in \mathbf{S}_1$; it is a logical formula with variables $A_1, ..., A_M$. If we replace every occurrence of A_m with x_m then we obtain a numerical formula containing the variables $x_1, ..., x_M$ and the operators

[3] From the mathematical point of view, the symbols $\vee, \wedge, '$ denote different operators in (1) and in (2): in the first case they operate on elements of $\mathbf{S}_1$ while in the second case they operate on elements of $\mathbf{S}_2$; but their logical interpretation is the same in both cases.

$\vee, \wedge, '$. To obtain the truth value of B we choose a particular *numerical implementation of* $\vee, \wedge, '$ and perform the numerical calculations . I.e.

$$B = F_B\left(A_1, ..., A_m\right)$$
$$\mathrm{Tr}\left(B\right) = f_B\left(\mathrm{Tr}\left(A_1\right), ..., \mathrm{Tr}\left(A_M\right)\right)$$

or, more concisely,

$$\mathrm{Tr}\left(B\right) = f_B\left(x_1, ..., x_M\right).$$

Here F_B is a *logical* formula ($F_B : \mathbf{V}_1 \to \mathbf{S}_1$) and f_B is the corresponding *numerical* formula ($f_B : [0,1]^M \to [0,1]$) obtained by replacing A_m with x_m and now understanding the symbols $\vee, \wedge, '$ as *numerical* operators (specific examples of the procedure appear in the Appendix). The implementation of the logical operators $\vee, \wedge, '$ by numerical operators, namely *t-conorms*, *t-norms* and *negations*, has been studied extensively by fuzzy logicians [17, 28]. Several typical implementations are presented in Table 1.

Family	$x \wedge y$	$x \vee y$	x'
Standard	$\min(x,y)$	$\max(x,y)$	$1-x$
Algebraic	xy	$x+y-xy$	$1-x$
Bounded	$\max(0, x+y-1)$	$\min(1, x+y)$	$1-x$
Drastic	$\begin{pmatrix} x & \text{when } y=1 \\ y & \text{when } x=1 \\ 0 & \text{else} \end{pmatrix}$	$\begin{pmatrix} x & \text{when } y=0 \\ y & \text{when } x=0 \\ 1 & \text{else} \end{pmatrix}$	$1-x$

Table 1

In the above manner, the truth value assignment originally defined on $\mathbf{V}_1$ (i.e. $\mathrm{Tr}(A_m) = x_m$, $m = 1, 2, ..., M$) has been extended to $\mathbf{S}_1$. Now we can use the truth values of 1st level sentences to assign truth values to 2nd level elementary sentences as follows. Given a $C \in \mathbf{V}_2$, which has the form

$$C = \text{“}\mathrm{Tr}\left(B\right) = b\text{”}$$

with $B \in \mathbf{S}_1$, $b \in [0,1]$, we *define*

$$\mathrm{Tr}\left(C\right) = 1 - \left|\mathrm{Tr}\left(B\right) - b\right|. \tag{3}$$

$\mathrm{Tr}(B)$ in (3) has already been defined, since $B \in \mathbf{S}_1$. Note that, according to (3), the maximum truth value of C is 1 and it is achieved when $\mathrm{Tr}(B) = b$; the latter is exactly what C says. More generally, the truth value of C is a decreasing function of the absolute difference between $\mathrm{Tr}(B)$ and b. This certainly appears reasonable[4]. In this manner we can compute the truth value of every $C \in \mathbf{V}_2$. Finally, we can extend truth values from $\mathbf{V}_2$ to $\mathbf{S}_2$ in exactly the same manner as we extended truth values from $\mathbf{V}_1$ to $\mathbf{S}_1$.

Hence starting with a truth value assignment $x_1, ..., x_M$ on $\mathbf{V}_1$ we have obtained a truth value assignment for every $D \in \mathbf{S}_2$, namely

$$\mathrm{Tr}\left(D\right) = f_D\left(x_1, ..., x_M\right)$$

where $f_D : [0,1]^M \to [0,1]$. This *explicit* assignment of truth values does not involve any self-reference or circularity: starting with the initial specification of the truth values of $A_1, ..., A_M$ as $x_1, ..., x_M$, the truth values of all (1st and 2nd level) sentences are obtained as functions of $x_1, ..., x_M$.

[4]For further justification of (3) see [10]. Note however, that a number of other functions could be used; a simple example is $\mathrm{Tr}(C) = 1 - \left(\mathrm{Tr}\left(B\right) - b\right)^2$.

3.2 Implicit Truth Value Assignment

Now suppose that we have a collection of M sentences, which talk about the truth values of each other. We will show a method by which the information each sentence conveys about the truth value of the other sentences (possibly also about its own truth value) can be used to determine (up to a point) the truth values in a consistent manner.

The previously presented framework gives a formal description of sentences which talk about the truth values of other sentences. In particular, members of $\mathbf{S}_2$ talk about the truth values of members of $\mathbf{S}_1 \supseteq \mathbf{V}_1$. Our approach can be summarized as follows.

Suppose we are given M self-referential sentences and consider the set $\mathbf{S}_2$ which is generated from M elementary sentences. We can pick sentences D_1,..., $D_M \in \mathbf{S}_2$ which have the same structure as the original self-referential sentences (examples of the procedure appear in the Appendix). The only difference is that the M self-referential sentences talk about each other, while D_1,..., D_M talk about some elementary, unspecified sentences A_1, ..., A_M. However, since A_1, ..., A_M are unspecified, we can identify A_m with D_m (for $m \in \{1, ..., M\}$). Intuitively, this means that D_m says something about the truth values of A_1,..., A_M, i.e. about the truth values of D_1,..., D_M. This is exactly the situation which we were trying to model in the first place. While philosophical objections can be raised about this type of self-reference, the situation is quite straightforward from the computational point of view (as will be seen in the following) and allows the determination of consistent truth values.

Let us now give the mathematical details. As mentioned at the end of Section 3.1, the truth value of every 2nd level sentence $D \in \mathbf{S}_2$ (for fixed M and a specific choice of t-norm, t-conorm and negation) is a numerical function $f_D(x_1, ..., x_M)$, the independent variables $x_1, ..., x_M$ being the truth values of $A_1, ..., A_M$. To obtain specific truth values by the procedure of Section 3.1, it is necessary to specify $x_1, ..., x_M$. Choose a function $\Phi : \{1, 2, ..., M\} \to \mathbf{S}_2$. $\Phi(1)$, $\Phi(2)$, ..., $\Phi(M)$ are 2nd level sentences[5] which can also be denoted as D_1,..., D_M. Now (for $m \in \{1, ..., M\}$) D_m is a logical formula $F_m(A_1, ..., A_M)$. In other words, we have (for $m = 1, 2, ..., M$):

$$\begin{aligned} D_1 &= F_1(A_1, ..., A_M) \\ D_2 &= F_2(A_1, ..., A_M) \\ &\;... \\ D_M &= F_M(A_1, ..., A_M). \end{aligned}$$

Let us form the system of logical equations

$$\begin{aligned} A_1 = D_1 &= F_1(A_1, ..., A_M) \\ A_2 = D_2 &= F_2(A_1, ..., A_M) \\ &\;... \\ A_M = D_M &= F_M(A_1, ..., A_M). \end{aligned} \tag{4}$$

The "natural" interpretation of (4) is that A_m *says* (or is, or means) D_m. (4) implies that:

$$\begin{aligned} \mathrm{Tr}(A_1) = \mathrm{Tr}(D_1) &= f_1(\mathrm{Tr}(A_1), ..., \mathrm{Tr}(A_M)) \\ \mathrm{Tr}(A_2) = \mathrm{Tr}(D_2) &= f_2(\mathrm{Tr}(A_1), ..., \mathrm{Tr}(A_M)) \\ &\;... \\ \mathrm{Tr}(A_M) = \mathrm{Tr}(D_M) &= f_M(\mathrm{Tr}(A_1), ..., \mathrm{Tr}(A_M)) \end{aligned} \tag{5}$$

[5] Obviously, in a particular situation we will choose these sentences which have the same structure as the self-referntial sentences in which we are interested – see the Appendix.

where $f_m : [0,1]^M \rightarrow [0,1]$ is the numerical formula obtained (by the procedure of Section 3.1) from F_m. A simpler way to write (5) is

$$
\begin{aligned}
x_1 &= f_1(x_1, ..., x_M) \\
x_2 &= f_2(x_1, ..., x_M) \\
&... \\
x_M &= f_M(x_1, ..., x_M).
\end{aligned}
\tag{6}
$$

(6) is a system of M numerical equations in M unknowns; we will refer to it as the system of *truth value equations*. Note that in general the truth value equations will be nonlinear.

Depending on the particular Φ used, (6) may have none, one or more than one solutions in $[0,1]^M$. Hence, by specifying a particular Φ, we obtain a *set* of possible *consistent* truth value assignments for the 1st level elementary sentences. In other words, every solution of (6) is a consistent truth value assignment. At this point we must consider the possibility that the set of solutions is empty, i.e. that there is no consistent truth value assignment. However, as we will soon see, under mild conditions there always exists at least one consistent assignment. Assuming that (6) has one or more solutions, we can choose one of these to assign truth values to the 1st level elementary sentences; next, using exactly the same construction as in Section 3.1, we can assign truth values to 1st level sentences, then to 2nd level elementary sentences and finally to 2nd level sentences. In particular, it is easy to check that at the end of the procedure the 2nd level sentences $D_1, ..., D_M$ will receive the truth values originally specified by the solution of (6) – hence the truth value assignment is, indeed, consistent.

Let us now show that, under mild conditions, every Φ function specifies *at least* one consistent truth value assignment. This is the subject of Proposition 3. However, we first need two auxiliary propositions.

Proposition 1 *If the implementations of $\vee, \wedge, '$ are, respectively, a continuous t-conorm, a continuous t-norm and a continuous negation, then $f_1, f_2, ..., f_M$ in (6) are continuous functions of $(x_1, x_2, ..., x_M)$.*

Proof. We give a sketch of the proof (we omit the complete proof for the sake of brevity; the basic idea is clear enough.). Suppose that $\vee, \wedge, '$ are a continuous t-conorm, t-norm and negation. Take some $m \in \{1, 2, ..., M\}$. Recall that $f_m(x_1, x_2, ..., x_M) = \mathrm{Tr}(D_m)$, where $D_m \in \mathbf{S}_2$. Now, take any $B \in \mathbf{S}_1$; then $\mathrm{Tr}(B) = y(x_1, x_2, ..., x_M)$ and y is a finite combination of $\vee, \wedge, '$ and $x_1, ..., x_M$, which is clearly a continuous function of the vector[6] $(x_1, x_2, ..., x_M)^T$. Furthermore, let $C =$"$\mathrm{Tr}(B) = b$"; then

$$
\mathrm{Tr}(C) = 1 - |\mathrm{Tr}(B) - b| = 1 - |y(x_1, x_2, ..., x_M) - b|
$$

which is also a continuous function of $(x_1, x_2, ..., x_M)^T$. Since this is true for every $B \in \mathbf{S}_1$ and every $b \in [0,1]$, we conclude that $\mathrm{Tr}(C)$ is a continuous function of $x = (x_1, x_2, ..., x_M)^T$ for *every* $C \in \mathbf{V}_2$. Finally, since $D_m \in \mathbf{S}_2$, and $\mathrm{Tr}(D_m)$ is a finite combination of $\vee, \wedge, '$ and a finite number of terms $\mathrm{Tr}(C_1)$, $\mathrm{Tr}(C_2)$, ..., $\mathrm{Tr}(C_L)$ (where $C_1, C_2, ..., C_L \in \mathbf{V}_2$) it follows that $\mathrm{Tr}(D_m)$ is a continuous function of $(x_1, x_2, ..., x_M)^T$. ■

Proposition 2 *Suppose that X is a nonempty, compact, convex set in R^M. If the function $f : X \rightarrow X$ is continuous, then there exists at least one fixed point $\overline{x} \in X$ satisfying*

$$
\overline{x} = f(\overline{x}).
$$

[6]We use column vectors; the superscript T indicates the transpose.

Proof. This the well-known *Brouwer's Fixed Point Theorem.* Its proof can be found in a number of standard texts, for instance in [3, pp.323-329]. ■

Now we can easily prove the existence of consistent truth value assignments.

Proposition 3 *If the implementations of* $\vee, \wedge, '$ *are, respectively, a continuous t-conorm, a continuous t-norm and a continuous negation, then (6) has at least one solution* $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_M)^T \in [0,1]^M$.

Proof. We define the vector function $f(x_1, x_2, ..., x_M)$ as follows:

$$f(x_1, x_2, ..., x_M) = (f_1(x_1, x_2, ..., x_M), f_2(x_1, x_2, ..., x_M), ..., f_M(x_1, x_2, ..., x_M))^T$$

where (for $m \in \{1, 2, ..., M\}$) $f_m(x_1, x_2, ..., x_M)$ is the function appearing in (6). Since $f_m(x_1, x_2, ..., x_M)$ computes a truth value, we have $f_m : [0,1]^M \to [0,1]$ and hence $f : [0,1]^M \to [0,1]^M$. Furthermore, by Proposition 1 each f_m is a continuous function and so f is also a continuous function. Now we can apply Proposition 2 with $X = [0,1]^M$. ■

When we use Boolean truth value assessments, we can prove an additional result about consistent truth value assignments.

Proposition 4 *Suppose that in (4)* $D_1, D_2, ..., D_M \in \widetilde{\mathbf{S}}_2$ *and the implementations of* $\vee, \wedge, '$ *are, respectively, max, min and the standard negation. Then (6) admits the solution* $(1/2, 1/2, ..., 1/2)^T$.

Proof. Take any $m \in \{1, 2, ..., M\}$; then $F_m(A_1, A_2, ..., A_M)$ is a combination (through $\vee, \wedge, '$) of a finite number of elements $C_1, C_2, ..., C_L \in \widetilde{\mathbf{V}}_2$. Take any C_l (with $l \in \{1, 2, ..., L\}$); it has the form

$$C_l = \text{“}\mathrm{Tr}(B_l) = b_l\text{”} \tag{7}$$

where $B_l \in \mathbf{S}_2$ and $b_l \in \{0, 1\}$. The corresponding numerical term will have the form

$$\mathrm{Tr}(C_l) = 1 - |\mathrm{Tr}(B_l) - b_l| . \tag{8}$$

or

$$z_l = 1 - |y_l - b_l| \tag{9}$$

where $y_l =$Tr(B_l) and $z_l =$Tr(C_l). Now, y_l will be a finite combination of x_1, ..., x_M through max, min and negation operators, hence when $x_1 = x_2 = ... = x_M = 1/2$ we also get $y_l = 1/2$. Then, for $b_l \in \{0, 1\}$ we also get from (9) that $z_l = 1/2$.

Hence every term appearing in $f_m(1/2, 1/2, ..., 1/2)$ (the numerical version of $F_m(A_1, A_2, ..., A_M)$) will be equal to 1/2. Since these terms will be combined with max, min and negation operators it follows that $f_m(1/2, 1/2, ..., 1/2) = 1/2$ and this satisfies the m-th truth value equation:

$$x_m = \frac{1}{2} = f_m(1/2, 1/2, ..., 1/2) . \tag{10}$$

Since (10) holds for every $m \in \{1, 2, .., M\}$, it follows that (6) admits the solution $(1/2, 1/2, ..., 1/2)^T$. ■

3.3 Discussion

We have seen that the assignment of consistent truth values to a collection of self-referential propositions can be reduced to solving the system of (algebraic) truth value equations. Furthermore, Proposition 3 shows that, subject to some mild continuity conditions, the truth value equations admit at least one solution.

Proposition 3 indicates that the Liar and related self referential collections cease to be paradoxical in the context of fuzzy logic. From the mathematical point of view the situation is rather straightforward. A system of truth value equations may have no solution in $\{0,1\}^M$; by expanding to $[0,1]^M$ the set in which we seek solutions we can guarantee that the system *always* has at least one solution. Of course, this result is achieved at the price of admitting fuzzy solutions, i.e. truth values which fall short of certainty.

Next, let us briefly compare explicit and implicit truth value assignment. Explicit assignment is straightforward: starting with the initial (and more or less arbitrary) choice of truth values for the 1st level variables, the truth values of 1st and 2nd level sentences are uniquely determined. If initial truth values were restricted to be either 0 or 1, then the initial choice would essentially be a choice of axioms, i.e. the assertion of certain propositions (or their negations); choosing truth values in $[0,1]$ can be seen as a "*generalized axiomatization*". Generally, in implicit truth value assignment the truth values of 1st and 2nd level sentences are not uniquely determined; rather a number of possible consistent truth values are obtained. There are of course cases (examples will be presented in the Appendix) where there is a single consistent truth value assignment. One is tempted to remark that in such cases the collection of self-referential sentences is equipped with an *implicit axiomatization*.

Let us now return to the system of truth value equations. We have established that it always has a solution; but many additional questions can be asked about it. Let us list a few such questions[7].

1. Under what conditions does (6) possess a unique solution?
2. Are some solutions "better" than other? For instance, solutions in the interior of the hypercube $[0,1]^M$ are "fuzzier" than solutions on the boundary, which in turn are fuzzier than solutions on the vertices. What are existence and uniqueness conditions for vertex or boundary solutions?
3. What is the structure of the set of solutions? For instance:
   (a) are there conditions under which it is a vector space?
   (b) what is its dimension (vector space dimension, Hausdorff dimension etc.)?
   (c) is it equipped with a "natural" order? is it a lattice?
4. Assuming that some of the above properties are established, are they invariant under different implementations of $\vee, \wedge, '$ and/ or the function $\mathrm{Tr}(\cdot)$?

One can also ask computationally oriented questions; the most obvious one is: "how to solve the truth value equations?". There is a large number of root finding algorithms that can be used to this end and we will discuss some of them in the next section. However, we believe it is useful to distinguish between properties of the truth value equations (such as the properties listed above) and algorithm properties. We will further discuss this distinction in Section 4.4 [8].

[7]We make no attempt to *answer* these questions in the current paper; they will be the subject of future research.

[8]Note that in our formulation the numbers of unknowns and equations are both equal to M. This is a natural consequence of the formulation, since both the unknowns and equations are corresponded to the original M self-referential sentences. From the mathematical point of view, since every variable is associated to an equation of the form $x_m =$

4 Computational Issues

We now discuss a number of algorithms which can be used to solve the truth value equations.

4.1 Root Finding

Probably the most popular method for solving systems of nonlinear equations is the iterative Newton-Raphson algorithm. A description and analysis can be found in many numerical analysis textbooks (e.g. [18]). Using

$$x = (x_1, ..., x_M)^T ,$$

let us rewrite the truth value equations (6) as follows. Define (for $m \in \{1, 2, ..., M\}$):

$$h_m (x) = x_m - f_m (x_1, ..., x_M)$$

and

$$h (x) = (h_1 (x), ..., h_M (x))^T .$$

Then the truth value equations (6) are equivalent to

$$h (x) = 0.$$

Let us define a time varying vector

$$x (t) = (x_1 (t), ..., x_M (t))^T ;$$

then the Newton-Raphson method consists in the iteration ($t = 0, 1, ...$):

$$x (t + 1) = x (t) - [G (x (t))]^{-1} \cdot h (x (t)) \tag{11}$$

with

$$G (x) = \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & .. & \frac{\partial h_1}{\partial x_M} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & ... & \frac{\partial h_2}{\partial x_M} \\ ... & ... & ... & ... \\ \frac{\partial h_M}{\partial x_1} & \frac{\partial h_M}{\partial x_2} & ... & \frac{\partial h_M}{\partial x_M} \end{pmatrix} .$$

Depending on the choice of t-norms and t-conorms, the partial derivatives of h_1,..., h_M may be undefined at some points of $[0, 1]^M$; however at such points $G (x)$ can be approximated by a differentiable function.

It is usually stated in textbooks that the Newton-Raphson algorithm will converge to a solution $\overline{x}$ if it starts with $x (0)$ "suficiently close to $\overline{x}$". But this statement needs some clarifications.

$f_m (x_1, ..., x_M)$, the number of variables *must* equal the number of equations. Of course, the m-th function f_m will generally *not* depend on all of $x_1, ..., x_M$.

One could also consider more complicated situations, for instance a collection of M sentences which can be separated into two groups of sizes M_1 and M_2(with $M_1 + M_2 = M$) having the following property: the sentences of the first group talk only about this group, while the sentences of the second group talk about both groups. This is a self-referential collection which contains a *closed* self-referential sub-collection (namely the first group). The sub-collection corresponds to a smaller system of truth value equations; the solutions obtained from this sub-system of M_1 equations will in general be different from the ones obtained from the original system of M equations.

Another possibility is to consider a collection of N sentences for which truth values are given and fixed and then a second collection of M sentences which talk about themselves and the first collection of sentences. This still results in a set of M equations which involve M unknown truth values $x_1, x_2, ..., x_M$ and N fixed truth values (call them *parameters*) $z_1, z_2, ..., z_N$. Our Proposition 3 still applies and guarantees the existence of a consistent truth value solution $(x_1, x_2, ..., x_M)^T$ for every choice of $z_1, z_2, ..., z_N$.

A *fixed point* of (11) is a point $\overline{x}$ which has the following property: if for some t we have $x(t) = \overline{x}$, then for $s = 1, 2, ...$ we have $x(t+s) = x(t)$. For this to hold it is necessary and sufficient that

$$[G(\overline{x})]^{-1} \cdot h(\overline{x}) = 0. \tag{12}$$

It is clear from (12) that every solution $\overline{x}$ of (6) is a fixed point of (11) but the converse is not necessarily true, i.e. (11) may have fixed points which are not solutions of (6). Furthermore, a fixed point is called *stable* if

$$\lim_{t\to\infty} x(t) = \overline{x}.$$

for every choice of $x(0)$ "sufficiently close" to $\overline{x}$; otherwise it is called *unstable*. The iteration of (11) can, by definition, converge only to stable fixed points, i.e. some solutions of (6) cannot be obtained because they are unstable fixed points.

There is an additional possibility: (11) may converge to a solution of (6) which lies outside of $[0,1]^M$. This problem can be addressed by the following modification of (11):

1. if, for some t and m, (11) gives $x_m(t) > 1$, then set $x_m(t) = 1$;
2. if, for some t and m, (11) gives $x_m(t) < 0$, then set $x_m(t) = 0$.

In conclusion, the Newton-Raphson algorithm may fail to find a consistent truth value assignment for several reasons:

1. the algorithm may fail to converge;
2. it may converge to a value which does not solve (6) or lies outside of $[0,1]^M$;
3. it may miss a solution which is an unstable fixed point.

Furthermore, there is no obvious way in which to obtain *all* the solutions of (6) using (11)[9].

However, as a practical matter, the numerical experiments of the Appendix indicate that (11) generally converges to a consistent truth value assignment, i.e. a solution of (6) belonging to $[0,1]^M$.

4.2 Inconsistency Minimization

Root finding can also be formulated (in a standard manner) as a minimization problem. Let us apply this approach to truth value assignment. We write the following *inconsistency* function

$$J(x_1, ..., x_M) = \sum_{m=1}^{M} (x_m - f_m(x_1, ..., x_M))^2. \tag{13}$$

Note that $J(x_1, ..., x_M)$ is a reasonable measure of the *inaccuracy* in satisfying the truth value equations. For every choice of $(x_1, ..., x_M)^T$, J takes a nonnegative value; every solution of (6) yields a global minimum of J. An intuitive interpretation of (13) goes as follows: $(x_m - f_m(x_1, ..., x_M))^2$ is the inconsistency of the m-th sentence and $\sum_{m=1}^{M} (x_m - f_m(x_1, ..., x_M))^2$ is the *total* inconsistency of the self-referential collection. From Proposition 3 we know that there is at least one consistent truth value assignment, i.e. a truth value assignment of zero inconsistency.

[9] In fact this remark holds not only for Newton-Raphson, but for all numerical root finding algorithms of which we are aware.

There is a vast number of minimization algorithms; indeed the "standard" sequential reasoning of classical Aristotelian logic can be understood as a simplistic inconsistency minimization algorithm which attempts to select the optimizing value of one variable at a time.

In the experiments reported in the Appendix we will use another simple algorithm, namely *steepest descent minimization*. This consists of the following iteration (for $t = 1, 2, ...$):

$$x(t+1) = x(t) - k \cdot \frac{\partial J}{\partial x}. \tag{14}$$

The term $\frac{\partial J}{\partial x}$ in (14) is the gradient of J with respect to x; at points where J is non-differentiable, similarly to the Newton-Raphson case, we can use an appropriate approximation by a differentiable function.

For small enough k every step of (14) yields a decrease of J. Hence (14) will at least converge to the neighborhood of a local minimum of J; however this is not enough to guarantee that (6) is satisfied. Furthermore, there is no guarantee that $x(t)$ as given by (14) will always belong to $[0,1]^M$. However, note that in place of steepest descent one could use a more sophisticated, *constrained* optimization algorithm to minimize some appropriate function of $(x_1, ..., x_M)^T$ under the constraints of zero inconsistency and staying in $[0,1]^M$.

Let us also note that the "inconsistency minimization" approach is somewhat related to connectionist cognitive models and could conceivably be used by actual human reasoners. However, we are not particularly concerned about the "psychological plausibility" of our approach, neither do we propose it as a description of actual human reasoning.

4.3 A Control-Theoretic Algorithm

Finally, inspired from control theory, we propose a root-finding algorithm which can be summarized by the following equation:

$$x(t+1) = x(t) - k \cdot (x(t) - f(x_1(t), ..., x_M(t))) \tag{15}$$

The motivation for (15) can be explained as follows. Choose any $m \in \{1, 2, ..., M\}$. From (15) we see that the difference $x_m(t+1) - x_m(t)$ is equal to $-k \cdot (x_m(t) - f_m(x_1(t), ..., x_M(t)))$. The following informal convergence argument holds provided that k is small and, consequently, the change between $x_m(t)$ and $x_m(t+1)$ is also small. In that case we can simply write $-k \cdot (x_m - f_m(x_1, ..., x_M))$ and argue as follows. If $x_m(t)$ is greater than $f_m(x_1(t), ..., x_M(t))$, then $x_m(t+1)$ becomes smaller than $x_m(t)$ and (assuming k to be sufficiently small) the difference $x_m - f_m(x_1, ..., x_M)$ gets closer to zero. Similarly, if $x_m(t)$ is smaller than $f_m(x_1(t), ..., x_M(t))$, then $x_m(t+1)$ becomes larger than $x_m(t)$ increases and the difference $x_m - f_m(x_1, ..., x_M)$ again gets closer to zero. Finally, if $(x_m - f_m(x_1, ..., x_M))$ equals zero, then x_m remains unchanged. Hence, unlike Newton-Raphson and Steepest Descent, the fixed points of (15) are exactly the solutions of the truth value equations. However, it is not guaranteed that every fixed point is stable.

Eq.(15) is very similar to schemes used for the control of *servomechanisms*; standard control theoretic methods can be used to investigate the behavior of $x(t)$, i.e. prove rigorously existence of stable fixed points, convergence, boundedness etc. (for example, these issues can be investigated by constructing a *Lyapunov function* of (15)).

Similarly to steepest descent, the control theoretic algorithm has a certain degreee of psychological plausibility. Namely, we can imagine a human reasoner who adjusts truth values in small increments, according to the discrepancy from the values predicted by the truth value equations.

4.4 Dynamical Systems and Reasoning

As is the case for every iterative algorithm, the three algorithms presented above can be viewed as *dynamical systems*. Hence our approach is somewhat similar to Grim's, since he also has modelled reasoning about self-referential sentences as a dynamical system. There is of course an important difference: Grim appears to be more interested in oscillatory and chaotic behavior than in convergence to consistent truth values.

While we find the chaotic behavior of Grim's dynamical systems quite interesting, we believe it is a rather secondary aspect of self-referential collections. As we have already mentioned, we believe it is a property of a particular algorithm rather than of the self-referential collection itself. Of course, the algorithms proposed by Grim are somewhat special, in the sense that they are inspired by human reasoning. However, the analogy should not be drawn too far. It is rather unlikely that a human, when reasoning about self-referential sentences, simulates chaotic dynamical systems with infinite precision arithmetic. We do not bring this issue up to criticize Grim's approach[10]. Our point is that both Grim's dynamical systems and the algorithms we have presented in Sections 4.1 – 4.3 can be seen as "generalized reasoning systems". In other words, despite our previous remarks about psychological plausibility, the connection of such systems to human reasoning is rather slender; however, they may reveal interesting aspects of the *mathematical* structure of self-referential collections.

5 Bibliographic Remarks

The literature on self-referential sentences is extensive and originates from many disciplines and points of view. Here we only present a small part of this literature: papers which we have found related to our investigation.

There is a vast philosophical literature on self-referential sentences and the related topic of *truth*. This literature goes back to the ancient Greeks. Some pointers to the early literature can be found in the books [2, 23, 24, 25]. These books also cover recent developments. Regarding the modern literature, we must list Tarski's fundamental paper on the *truth predicate* [44]. Two other interesting early papers are [33, 34]. Important developments in the 70's were the concept of *truth gaps* [48, 49] and Kripke's theory of truth [19]. Some important papers from the 80's and 90's are [9, 11, 32, 43].

Preliminary concepts of *multi-valued logic* [36] have been used in connection to the Liar paradox already in the Middle Ages. The notion of truth gaps is also closely related to multi-valued logic, as is Skyrms' resolution of the Liar [40]. The treatment of the Liar by the methods of fuzzy logic was a natural development. We have already mentioned Zadeh's paper [51] which is, as far as we know, the first work dealing with the Liar in the context of fuzzy logic. We have also mentioned that Grim's work [10, 21, 22, 41] has been a major motivation for the current paper. In the fuzzy logic literature there is a considerable amount of work on self-referential sentences and the concept of truth (for example [13, 14]). These approaches fall within the mathematical logic tradition and are quite different from ours. But some logicians have also addressed computational issues which are related to our concerns. For example the concept of *fuzzy satisfiability* is addressed in [4, 16, 27, 50]; on the same topic see [42].

We believe that ideas from the areas of *nonmonotonic reasoning* and *belief revision* could yield fruitful insights on the Liar and related paradoxes, but we are not aware of any work in this direction. However the AGU axioms for belief revision (see for example [1, 7] and the book [8]) appear to us very relevant to the study of logical paradoxes. Many of these issues are explored in [12]; further interesting recent work includes [6, 20]. Also, self-reference is explicitly treated in [29, 30].

We consider our approach to self-referential sentences only marginally related to psychology and

[10] Neither has Grim claimed that his model describes actual human reasoning.

cognitive science. However the so-called *coherence* theory (or theories) of truth, a topic at the interface between psychology and philosophy, appears related to self-reference. Probably the best known coherence theory of truth is the one expounded by Thagard in [45] and further elaborated in [46, 47] and several other publications. Thagard has also applied his theory to reasoning about a set of sentences, each pair of which may *cohere* or *incohere*. This theory can also be applied to self-referential sentences (though we are not aware of work exploring this connection). Thagard has given a computational formulation of his theory as a *constraint satisfaction* problem [46]; this formulation is rather similar to the inconsistency minimization approach we have presented in Section 4.2. The relationship of coherence to *belief networks* is discussed [47]. It is interesting to note that Thagard has also presented a *connectionist* formulation of the constraint satisfaction problem, which is related to several connectionist models of *nonmonotonic reasoning* [26, 31, 35]. Finally, from our point of view, a particularly interesting paper is [38], which proposes a fuzzy measure of coherence. The constraint satisfaction approach has also been applied in the context of *cognitive dissonance* [5, 39]. It would be interesting to apply the theory of cognitive dissonance to reasoning about self-referential propositions, but as far as we know this has not been done until now.

Last but not least, the first chapter of Hofstadter's book [15] has a large collection of self-referential sentences. Some of these are of a quite different flavor from the Liar and are not easily formalized; however, we recommend the book to the reader for a more general view on the problem of self-reference.

6 Conclusion

In this paper we have presented a fuzzy-logic formulation which can describe a large family of collections of self-referential sentences; in particular it can accommodate the Liar, the inconsistent dualist, the consistent dualist and the *strengthened* Liar (see the Appendix, Section A.7). In our formulation, subject to some mild continuity conditions, every member of this family admits at least one consistent assignment of fuzzy truth values. Hence the main contribution of this paper is to expand Zadeh's analysis and to thus show that the Liar and related self referential collections cease to be paradoxical in the context of fuzzy logic by the simple expedient of expanding the set of possible solutions from $\{0,1\}^M$ to $[0,1]^M$. If we accept the price of admitting fuzzy solutions, i.e. truth values which fall short of certainty, then we can at one stroke resolve a large number of potential paradoxes.

Furthermore, we have presented several computational approaches to the problem of finding consistent truth values. In addition to the rather standard approaches of nonlinear equation solving by Newton-Raphson and by minimization we have presented an algorithm inspired by control theory which, as demonstrated by a number of numerical experiments, appears to strike a good balance between speed of execution, convergence and psychological plausibility (see also the Appendix, Section A.8).

Many issues remain open; in the following paragraphs we list several groups of questions which we would like to investigate in the future.

First, there are mathematical questions regarding the *solutions of the truth value equations*. Examples of such questions include existence and uniqueness of boundary and vertex solutions and invariance of the solutions under different implementations of the logical operators and the truth function $\mathrm{Tr}(\cdot)$. Also, there are questions regarding the *algorithmic* behavior, for instance the stability of a solution under a particular algorithm. The complete resolution of these questions appears to be a difficult problem. Perhaps a fruitful first step will be the study of a restricted family of self-referential collections (for example the study of self-referential *Boolean* truth value assessments).

A second group of questions concerns *optimization issues*. There is scope for experimentation with alternative implementations of the $\mathrm{Tr}(\cdot)$ function, alternative inconsistency functions (for example using absolute values rather than squares) and so on. We have also mentioned that truth value assignment could be formulated as *constrained* optimization, i.e. as the minimization of some fuzziness criterion subject to zero inconsistency.

However, we believe that the most interesting variations of the optimization problem are the ones which attempt to address the issue of *bounded rationality.* Whatever the attractions of infinite valued logic may be, we find rather implausible that humans reason with a continuum of truth values. A (perhaps small) step to increase the relevance of our approach to actual human reasoning would be to minimize inconsistency using a finite (and small) number of truth values. This is an intermdeiate step between the classical approach of searching for solutions in $\{0,1\}^M$ and the fuzzy logic approach of searching in $[0,1]^M$. While the formulation of the problem is identical to the one we have presented here, a major difference is that now there is no guarantee for the existence of a zero inconsistency solution. Hence a number of mathematical questions arise, regarding the properties of the minima of the inconsistency function; from the computational point of view, the use of a finite set of truth values may require the use of entirely different optimization algorithms.

Let us also note that there are other possibilities for addressing the issue of bounded rationality, for example the use of interval-valued fuzzy sets, the description of truth values in terms of tolerance classes etc. These issues require further research.

Finally, we are very interested in the *coherence* approach to self-referential sentences. Since self-referential sentences make claims about each other's truth values it is rather straightforward to setup a network with one node per sentence and connections which are either reinforcing or inhbiting (depending on what each sentence says about each other). We have performed preliminary experiments based on a Boltzmann machine formulation of the problem; we use a variation of Schoch's fuzzy coherence measure [38]. These results will be reported in a future publication.

References

[1] C. Alchourron, P. Gardenfors and D. Mackinson. "On the logic of theory change: partial meet contraction and revision functions". *J. of Symb. Logic,* vol.50, pp.510-530, 1985.

[2] J. Barwise and J. Etchemendy. *The Liar.* Oxford Univ. Press 1987.

[3] K. Binmore. *Fun and Games*, Heath and Company, 1992.

[4] D. Butnariu, E.P. Klement and S. Zafrany. "On triangular norm-based propositional fuzzy logics". *Fuzzy Sets and Systems,* vol.69, pp.241-255, 1995.

[5] L. Festinger. *A theory of cognitive dissonance.* 1957, Stanford Univ. Press.

[6] A. Fuhrmann. "Theory contraction through base contraction". *J. of Phil. Logic,* vol.20, pp.175-203, 1991.

[7] P. Gardenfors. "The dynamics of belief systems: foundations vs. coherence theories". *Rev. International de Philosophie,* vol.172, pp.24-46, 1990.

[8] P. Gardenfors. *Belief Revision.* Cambridge Univ. Press, 1992.

[9] L. Goldstein. " 'This statement is not true' is not true". *Analysis,* vol.52, pp.1-5, 1992.

[10] P. Grim. "Self-reference and chaos in fuzzy logic'". *IEEE Trans. on FS,* vol.1, pp.237-253, 1993.

[11] A. Gupta. "Truth and paradox". *J. of Phil. Logic,* vol.11, pp.1-60, 1982.

[12] A. Gupta and N. Belnap. *The Revision Theory of Truth*. MIT Press, 1993.

[13] P. Hajek and J. Shepherdson. "A note on the notion of truth in fuzzy logic". *An. Of Pure and Appl. Logic*, vol.109, pp.65-69, 2001.

[14] P. Hajek, J. Paris and J. Shepherdson. "The liar paradox and fuzzy logic". *Preprint*, 1998.

[15] D. Hofstadter. *Metamagical Themas*. Basic Books, 1985.

[16] E.P. Klement and M. Navara. "A survey of different triangular norm-based fuzzy logics". *Fuzzy Sets and Systems,* vol.101, pp.241-251, 1999.

[17] G.I. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

[18] N.V. Kopchenova and I.A. Maron. *Computational Mathematics*. MIR, 1975.

[19] S. Kripke. "Outline of a theory of truth". *J. of Philosophy,* vol.72, pp.690-716, 1975.

[20] H.E. Kyburg. "Rational belief". *Brain and Behav. Sci.,* vol.65, pp.231-273, 1983.

[21] G. Mar and P. Grim. "Pattern and chaos: new images in the semantics of paradox". *Nous,* vol.25, pp.659-693, 1991.

[22] G. Mar and P. St. Denis. "What the liar taught Achilles". *J. of Phil. Logic,* vol.28, pp.29-46, 1999.

[23] R.L. Martin. *The paradox of the Liar*, Ridgeview, 1970.

[24] R.L. Martin. *Recent Essays on Truth and the Liar Paradox.* Oxford Univ. Press, 1984.

[25] V. McGee. *Truth, vagueness and paradox*. Hackett, 1991.

[26] H. Narazaki and A.L. Ralescu. "A connectionist approach for rule-based inference using an improved relaxation method". *IEEE Trans. on NN,* vol.3, pp.741-751, 1992.

[27] M. Navara. "Satisfiability in fuzzy logics". *Preprint*, 2000.

[28] H.T. Nguyen and E.A. Walker. *A First Course in Fuzzy Logic*, 2nd Edition, Chapman and Hall, 1999.

[29] D. Perlis. "Languages with self reference I: foundations". *Art. Int.*, vol.25, pp.301-322, 1985.

[30] D. Perlis. "Languages with self reference II: knowledge, belief and modality". *Art. Int.*, vol.34, pp.179-212, 1988.

[31] G. Pinkas. "Symmetric neural networks and propositional logic satisfiability". *Neural Comp.,* vol.3, pp.282-291, 1991.

[32] G. Priest. "The logic of paradox revisited". *J. of Phil. Logic,* vol.13, pp.153-179, 1984.

[33] A. Prior. "Epimenides the Cretan". *J. of Symb. Logic,* vol.23, pp.261-266, 1958.

[34] A. Prior. "On a family of paradoxes". *Notre Dame J. of Formal Logic,* vol.2, pp.16-32, 1961.

[35] P.S. Raghuvanshi and S. Kumar. "Bipolar radial basis function inferencing networks". *Neurocomputing,* vol.14, pp.195-204, 1997.

[36] N. Rescher. *Multi-valued Logic.* McGraw-Hill, 1969.

[37] A. Rieger. "The liar, the strengthened liar and bivalence". *Erkenntnis,* vol.54, pp.195-203, 2001.

[38] D. Schoch. "A fuzzy measure for explanatory coherence". *Synthese,* vol.122, pp.291-311, 2000.

[39] T.R. Schultz and M.R. Lepper. "Cognitive dissonance reduction as constraint satisfaction". *Psych. Rev.,* vol.103, pp.219-240, 1996.

[40] B. Skyrms. "Return of the liar: three-valued logic and the concept of truth". *Amer. Philosoph. Quart.,* vol.7, pp.153-161, 1970.

[41] P. St. Denis and P. Grim. "Fractal images of formal systems". *J. of Phil. Logic,* vol.26, pp.181-222, 1997.

[42] S. Sudarsky. "Fuzzy satisfiability". *Preprint*, 1995.

[43] J. Tappenden. "The liar and sorites paradoxes: toward a unified treatment". *J. of Philosophy,* vol.90, pp.551-577, 1993.

[44] A. Tarski. "Der Wahrheitsbegriff in den formalisierten Sprachen". *Studien Philosophica*, vol. 1, pp.261-405, 1935.

[45] P. Thagard. "Explanatory Coherence". *Beh. Brain Sci.,* vol.12, pp.435-467, 1989.

[46] P. Thagard and K. Verbeugt. "Coherence as constraint satisfaction". *Cog. Sci.,* vol.22, pp.1-24, 1998.

[47] P. Thagard. "Probabilistic networks and explanatory coherence". *Preprint*, 1998.

[48] B. van Fraassen. "Singular terms, truth-value gaps and free logic". *J. of Philosophy,* vol.63, pp.481-495, 1966.

[49] B. van Fraassen. "Presupposition, implication and self-reference". *J. of Philosophy,* vol.65, pp.136-152, 1968.

[50] S.M. Wang, B.S. Wang and G.J. Wang. "A triangular norm-based propositional fuzzy logic". *Fuzzy Sets and Systems,* vol.136, pp.55-70, 2003.

[51] L.A. Zadeh. "Liar's paradox and truth-qualification principle". ERL Memo M79/34, 1979.

A Appendix: Examples and Experiments

In this appendix we present several examples of our formulation of self-referential collections. Whenever possible, we solve the truth value equations analytically; in addition we apply the three algorithms of Section 4 to numerically compute truth values.

A.1 Example 1: The Liar

We present our first example very briefly since it has already been addressed by Zadeh and several other authors. The Liar sentence is

$$A = \text{“}A \text{ is false”}.$$

According to our previous remarks, we define

$$C = \text{“Tr}(A) = 0\text{”}$$

and then set

$$A = C$$

which implies

$$\begin{aligned} \mathrm{Tr}(A) &= \mathrm{Tr}(C) \Rightarrow \\ \mathrm{Tr}(A) &= 1 - |\mathrm{Tr}(A) - 0| \Rightarrow \\ x &= 1 - |x - 0| \Rightarrow \\ x &= 1 - x \end{aligned} \tag{16}$$

where we have set $x = \mathrm{Tr}(A)$. (16) is the truth value equation and it has the unique solution $x = 1/2$.

A.2 Example 2: The Inconsistent Dualist

Our next example is the inconsistent dualist:

$$A_1 : \text{“}A_2 \text{ is true”} \tag{17}$$

$$A_2 : \text{“}A_1 \text{ is false”}. \tag{18}$$

In this case the 1st level elementary sentences are $\mathbf{V}_1 = \{A_1, A_2\}$. We will use two 2nd level elementary sentences

$$\begin{aligned} C_1 &: \text{“Tr}(A_2) = 1\text{”} \\ C_2 &: \text{“Tr}(A_1) = 0\text{”}. \end{aligned}$$

Note that C_1 and C_2 are Boolean elementary truth value assessments, i.e. $C_1, C_2 \in \widetilde{\mathbf{V}}_2 \subseteq \widetilde{\mathbf{S}}_2$. The translation of (17), (18) consists in mapping A_1 to C_1 and A_2 to C_2; then we have the logical equations

$$\begin{aligned} A_1 &= C_1 \\ A_2 &= C_2 \end{aligned}$$

from which follows

$$\begin{aligned} \mathrm{Tr}(A_1) &= \mathrm{Tr}(C_1) = 1 - |\mathrm{Tr}(A_2) - 1| \\ \mathrm{Tr}(A_2) &= \mathrm{Tr}(C_2) = 1 - |\mathrm{Tr}(A_1) - 0|. \end{aligned}$$

Substituting $\mathrm{Tr}(A_m)$ with x_m $(m = 1, 2)$ we get

$$x_1 = 1 - |x_2 - 1|$$
$$x_2 = 1 - |x_1 - 0|$$

and, since $x_1, x_2 \in [0, 1]$, we finally get the truth value equations

$$x_1 = x_2 \tag{19}$$
$$x_2 = 1 - x_1. \tag{20}$$

Obviously, eqs. (19)–(20) have the unique solution $\overline{x} = (\overline{x}_1, \overline{x}_2) = (1/2, 1/2)$.

The inconsistency function is

$$J(x_1, x_2) = (x_1 - x_2)^2 + (x_1 + x_2 - 1)^2$$

and has partial derivatives

$$\frac{\partial J}{\partial x_1} = \frac{\partial}{\partial x_1}\left((x_1 - x_2)^2 + (x_1 + x_2 - 1)^2\right) = 4x_1 - 2$$
$$\frac{\partial J}{\partial x_2} = \frac{\partial}{\partial x_2}\left((x_1 - x_2)^2 + (x_1 + x_2 - 1)^2\right) = 4x_2 - 2.$$

Setting the partial derivatives equal to zero we obtain

$$\frac{\partial J}{\partial x_1} = 4x_1 - 2 = 0 \tag{21}$$
$$\frac{\partial J}{\partial x_2} = 4x_2 - 2 = 0 \tag{22}$$

and, obviously, eqs. (21)–(22) have the unique solution $\overline{x} = (\overline{x}_1, \overline{x}_2) = (1/2, 1/2)$ which gives the unique truth value assignment of zero inconsistency.

Furthermore, using the steepest descent algorithm we obtain the the dynamical system

$$x_1(t+1) = x_1(t) - k \cdot (4x_1(t) - 2) \tag{23}$$
$$x_2(t+1) = x_2(t) - k \cdot (4x_2(t) - 2). \tag{24}$$

This is a linear system which can be written in matrix notation as

$$x(t+1) = Ax(t) + b$$

with

$$A = \begin{pmatrix} 1-4k & 0 \\ 0 & 1-4k \end{pmatrix}, \qquad b = \begin{pmatrix} 2k \\ 2k \end{pmatrix}$$

A has double eigenvalue: $\lambda_1 = \lambda_2 = 1 - 4k$. We have (for $i = 1, 2$)

$$|\lambda_i| = 1 - 4k < 1$$

for sufficiently small k. Hence (23)–(24) is stable. The unique fixed point $\overline{x}$ is obtained by solving

$$x = Ax + b.$$

The solution is $\overline{x} = (1/2, 1/2)$. Because of stability we have for every $x(0)$ that

$$\lim_{t\to\infty} x(t) = \overline{x}.$$

This result is verified by numerical simulation. Using $k = 0.1$ and starting with random initial conditions we have performed several simulations of (23)–(24). A typical run is illustrated in Figure 1. Specifically, Fig.1.a illustrates the evolution of the truth values $x_1(t)$ (solid line) and $x_2(t)$ (dash-dotted line) as computed by the steepest descent minimization algorithm (23)–(24) with $k = 0.1$; Fig.1.b illustrates the corresponding evolution of inconsistency.

Fig. 1.a

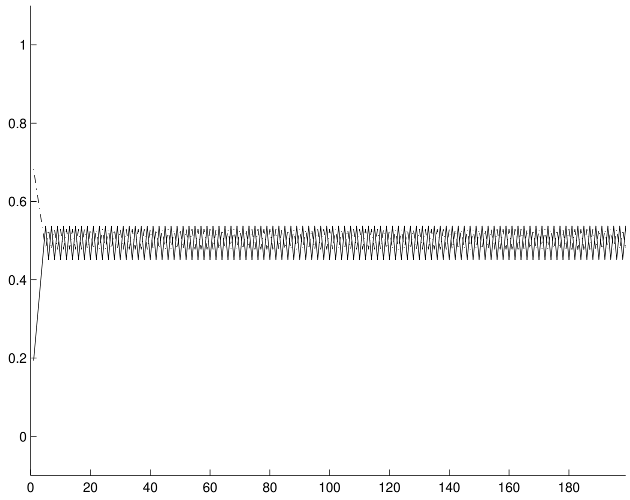
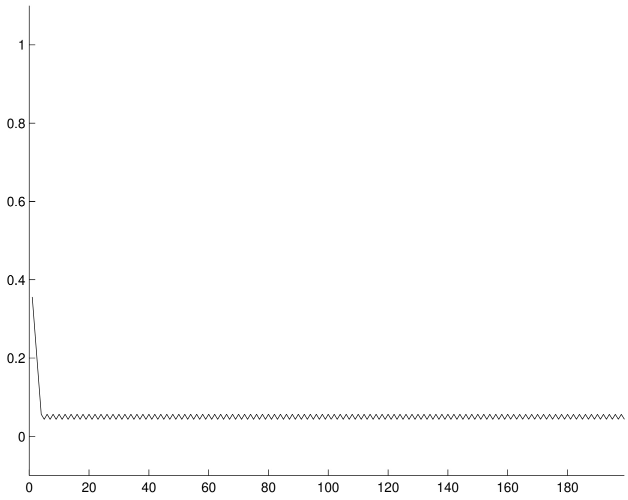


Fig. 1.b



We observe that the algorithm gets to the neighborhood of the optimal truth values $\overline{x}$ in a few steps and thereafter oscillates around $\overline{x}$; note also that at equlibrium the algorithm yields an inconsistency value larger than zero. This behavior is related to the step size k. Using smaller step size we can decrease the amplitude of the oscillation and bring the inconsistency closer to zero, at the expense of slower convergence; a typical simulation with $k = 0.01$ appears in Figure 2. Note that inconsistency is now a lot closer to zero.

Fig. 2.a

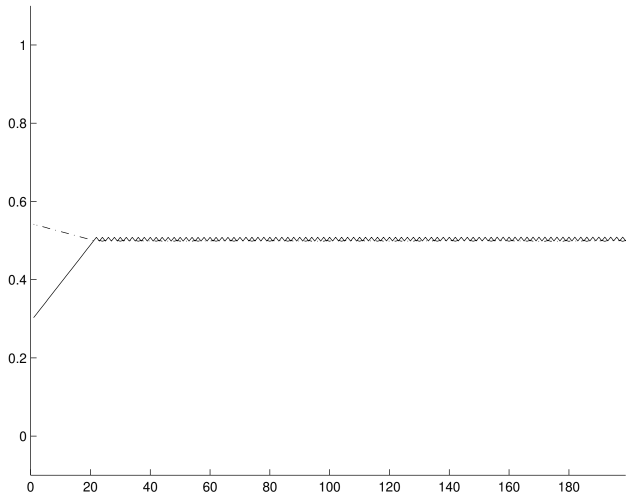
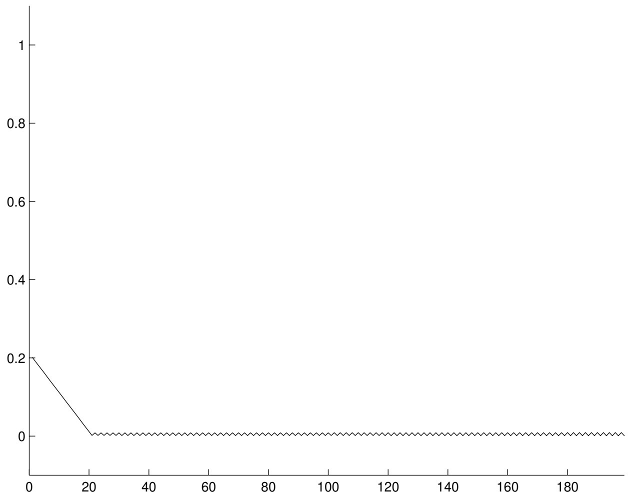


Fig. 2.b



Finally, using the control algorithm, we obtain the dynamical system

$$x_1(t+1) = x_1(t) - k \cdot (x_1(t) - x_2(t)) \tag{25}$$
$$x_2(t+1) = x_2(t) - k \cdot (x_2(t) - 1 + x_1(t)). \tag{26}$$

This is a linear system which can be written in matrix notation as

$$x(t+1) = Ax(t) + b$$

with

$$A = \begin{pmatrix} 1-k & k \\ -k & 1-k \end{pmatrix}, \qquad b = \begin{pmatrix} 0 \\ k \end{pmatrix}$$

A has eigenvalues: $\lambda_1 = -k + 1 + ik$, $\lambda_2 = -k + 1 - ik$. We have (for $i = 1, 2$)

$$|\lambda_i| = \sqrt{1 - 2k + 2k^2} < 1$$

for sufficiently small k. Hence (25)–(26) is stable. The unique fixed point $\overline{x}$ is obtained by solving

$$x = Ax + b.$$

The solution is $\overline{x} = (1/2, 1/2)$. Because of stability we have for every $x(0)$ that

$$\lim_{t\to\infty} x(t) = \overline{x}.$$

This result is verified by numerical simulation. Using $k = 0.1$ and starting with random initial conditions we have performed several simulations of (25)–(26). A typical run is illustrated in Figure 3. Again, Fig.3.a illustrates the evolution of the truth values $x_1(t)$ (solid line) and $x_2(t)$ (dash-dotted line) and Fig.3.b illustrates the corresponding inconsistency (as given by (13)).

Fig. 3.a

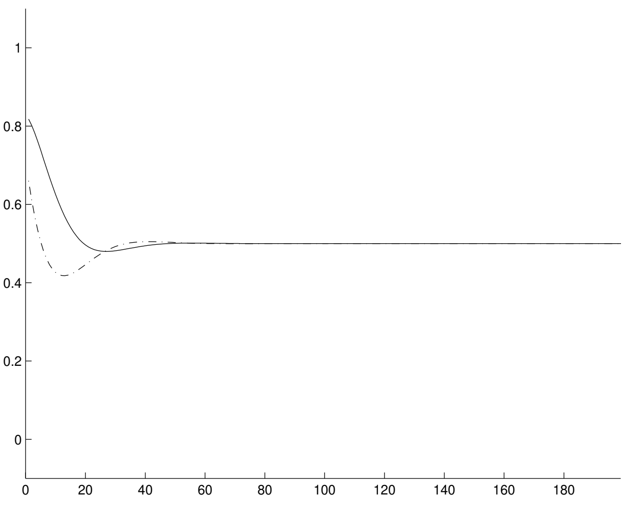
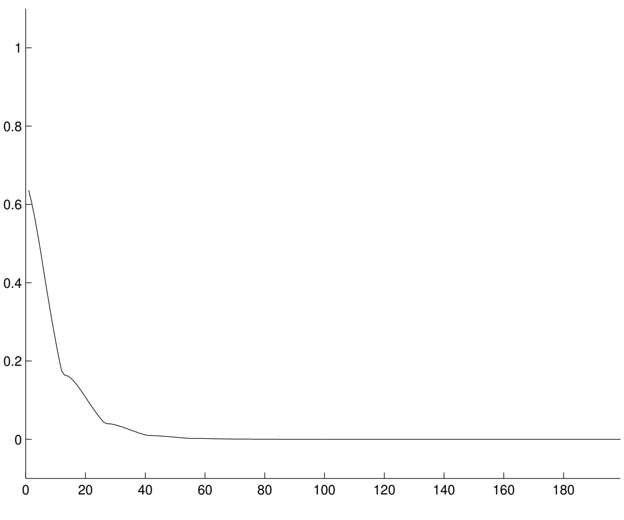


Fig. 3.b



We observe that the control algorithm locates the solution of the truth value equations with higher accuracy than the steepest descent method and almost as fast.

A.3 Example 3: The Consistent Dualist

Our next example is the consistent dualist:

$$A_1 : \text{“}A_2 \text{ is true”}$$
$$A_2 : \text{“}A_1 \text{ is true”}.$$

Again the 1st level elementary sentences are $\mathbf{V}_1 = \{A_1, A_2\}$; the 2nd level elementary sentences are

$$C_1 : \text{“Tr}(A_2) = 1\text{”}$$
$$C_2 : \text{“Tr}(A_1) = 1\text{”}$$

which are Boolean elementary truth value assessments, i.e. $C_1, C_2 \in \widetilde{\mathbf{V}}_2 \subseteq \widetilde{\mathbf{S}}_2$. Setting

$$A_1 = C_1$$
$$A_2 = C_2$$

we obtain

$$\mathrm{Tr}(A_1) = \mathrm{Tr}(C_1) = 1 - |\mathrm{Tr}(A_2) - 1|$$
$$\mathrm{Tr}(A_2) = \mathrm{Tr}(C_2) = 1 - |\mathrm{Tr}(A_1) - 1|$$

and then the truth value equations

$$x_1 = 1 - |x_2 - 1|$$
$$x_2 = 1 - |x_1 - 1|$$

which can be written in simple form as

$$x_1 = x_2$$
$$x_2 = x_1.$$

Any vector of the form $\overline{x} = (\beta, \beta)$ ($\beta \in [0,1]$) is a solution; i.e. there is an infinite number of consistent truth value assignments including complete truth ($\mathrm{Tr}(A_1) = \mathrm{Tr}(A_2) = 1$) and complete falsity ($\mathrm{Tr}(A_1) = \mathrm{Tr}(A_2) = 0$); in accordance to Proposition 4, $(1/2, 1/2)$ is also a solution.

The inconsistency function is

$$J = (x_1 - x_2)^2 + (x_1 - x_2)^2$$

and has partial derivatives

$$\frac{\partial J}{\partial x_1} = 4x_1 - 4x_2$$
$$\frac{\partial J}{\partial x_2} = 4x_2 - 4x_1.$$

Setting the partial derivatives equal to zero we obtain

$$\frac{\partial J}{\partial x_1} = 4x_1 - 4x_2 = 0 \tag{27}$$
$$\frac{\partial J}{\partial x_2} = 4x_2 - 4x_1 = 0 \tag{28}$$

and it is immediate that (27)–(28) has the family of solutions $x = (\beta, \beta)$, each of which gives zero inconsistency. Furthermore, using the steepest descent algorithm we obtain the linear dynamical system

$$x_1(t+1) = x_1(t) - k \cdot (4x_1(t) - 4x_2(t)) \tag{29}$$
$$x_2(t+1) = x_2(t) - k \cdot (4x_2(t) - 4x_1(t)) \tag{30}$$

which can be written in matrix notation as

$$x(t+1) = Ax(t)$$

with

$$A = \begin{pmatrix} 1-4k & 4k \\ 4k & 1-4k \end{pmatrix}$$

A has eigenvalues: $\lambda_1 = 1, \lambda_2 = 1-8k$. We have

$$|\lambda_1| = 1 \text{ and } |\lambda_2| \sqrt{1-8k} < 1$$

hence (29)–(30) will generally be convergent, but will have oscillatory behavior for certain initial conditions. The fixed points are obtained by solving

$$x = Ax.$$

The family of solutions is $x = (\beta, \beta)$. These results are verified by numerical simulation. Using $k = 0.1$ and starting with random initial conditions we have performed several simulations of (29)–(30). Three typical runs are illustrated in Figures 4, 5 and 6; the left panels indicate the evolution of the truth values and the right panels illustrate the corresponding inconsistency. We observe that the actual equilibrium reached depends on the initial conditions.

Fig. 4.a

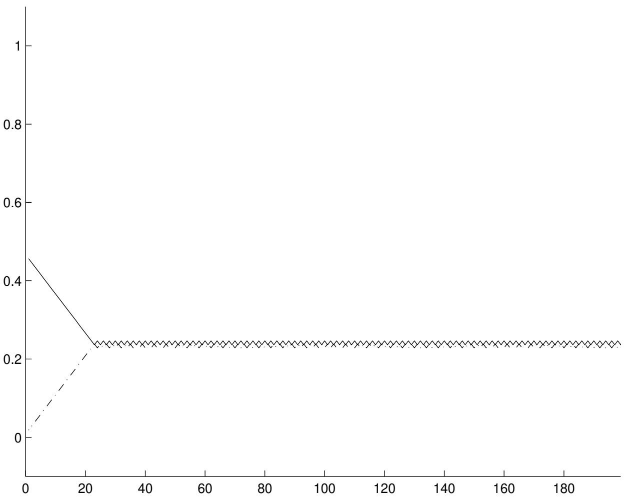


Fig. 4.b

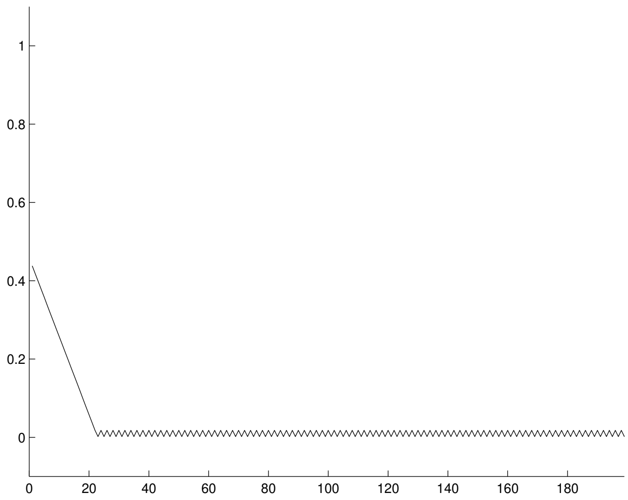


Fig. 5.a

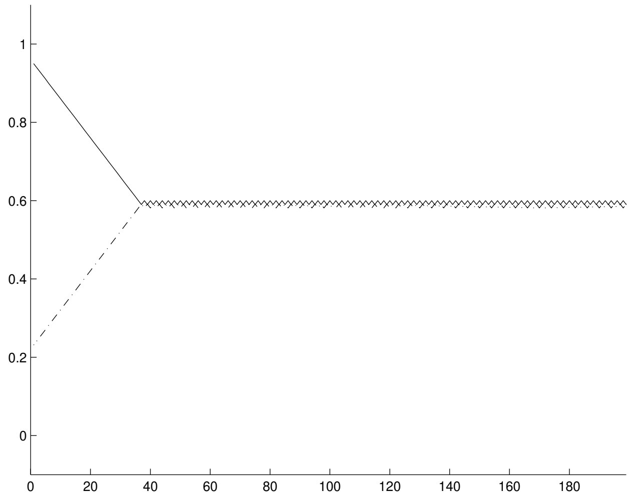


Fig. 5.b

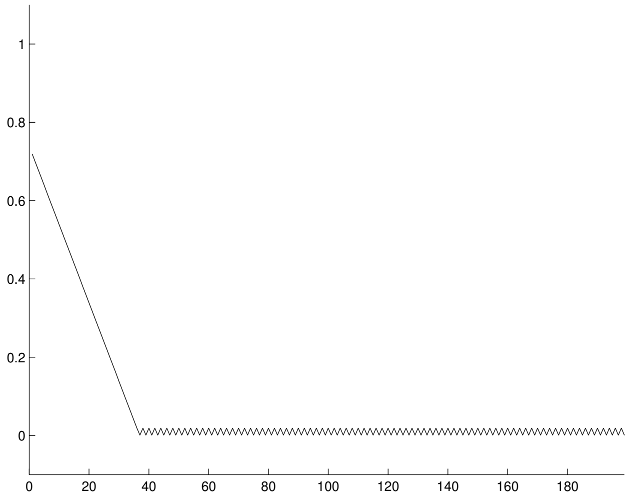


Fig. 6.a

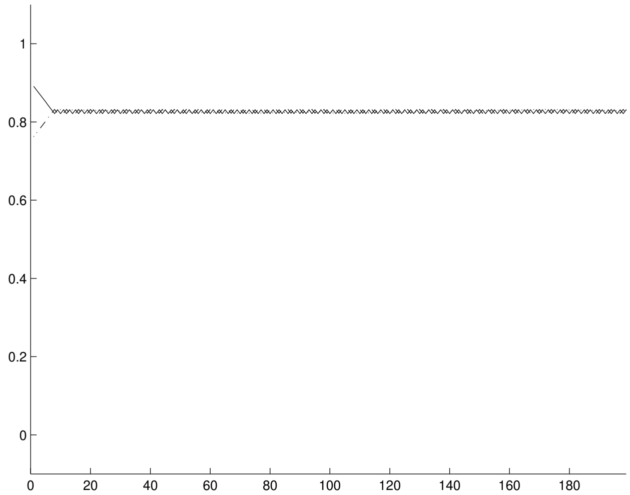
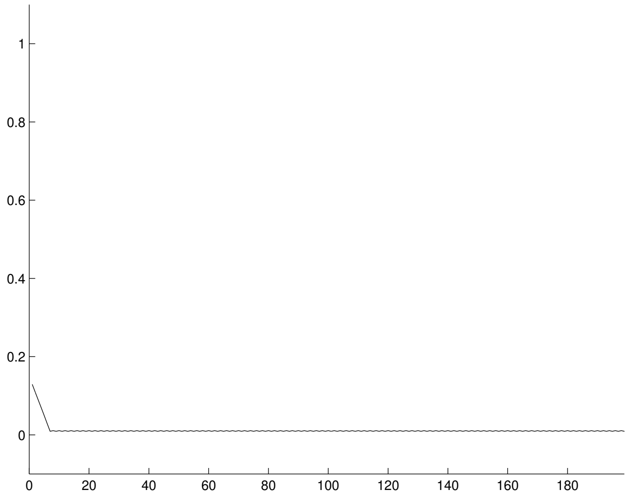


Fig. 6.b



Finally, using the control algorithm, we obtain the dynamical system

$$x_1(t+1) = x_1(t) - k \cdot (x_1(t) - x_2(t)) \tag{31}$$

$$x_2(t+1) = x_2(t) - k \cdot (x_2(t) - x_1(t)). \tag{32}$$

This is a linear system which can be written in matrix notation as

$$x(t+1) = Ax(t)$$

with

$$A = \begin{pmatrix} 1-k & k \\ k & 1-k \end{pmatrix}$$

A has eigenvalues: $\lambda_1 = 1$, $\lambda_2 = 1 - 2k$. We have for sufficiently small k

$$|\lambda_1| = 1 \text{ and } |\lambda_2| \sqrt{1-2k} < 1$$

hence (31)–(32) can have oscillatory behavior (for certain initial conditions). Also, the fixed points of are obtained by solving

$$x = Ax.$$

The family of solutions is, as expected, $x = (\beta, \beta)$. The results are verified by numerical simulation. Using $k = 0.1$ and starting with random initial conditions we have performed several simulations of (31)–(32). Typical runs are illustrated in Figures 7, 8, 9 (again, the left panels indicate the evolution of the truth values and the right panels illustrate the corresponding inconsistency).

Fig. 7.a

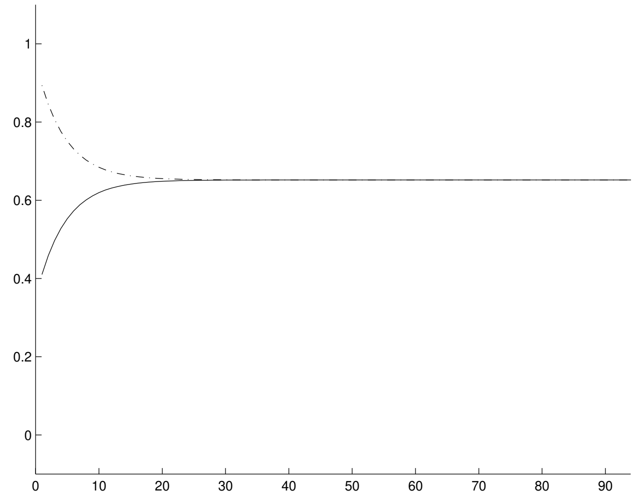


Fig. 7.b

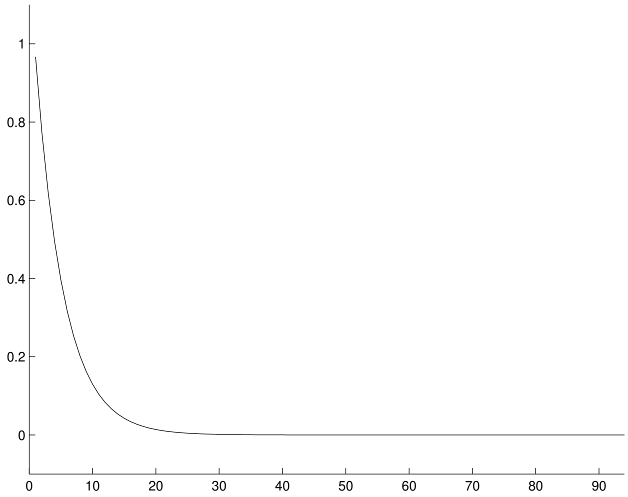


Fig. 8.a

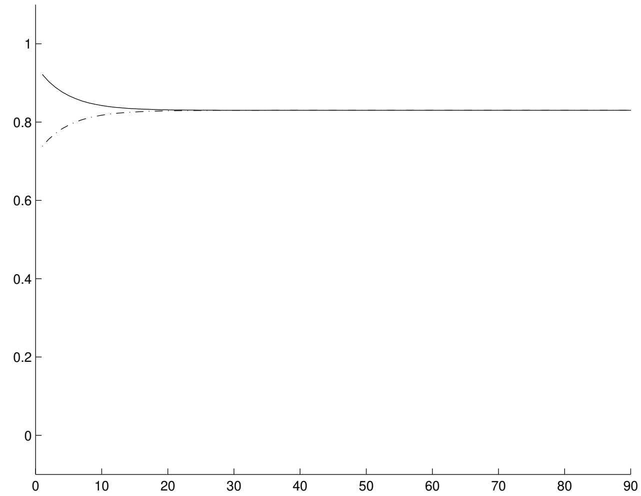


Fig. 8.b

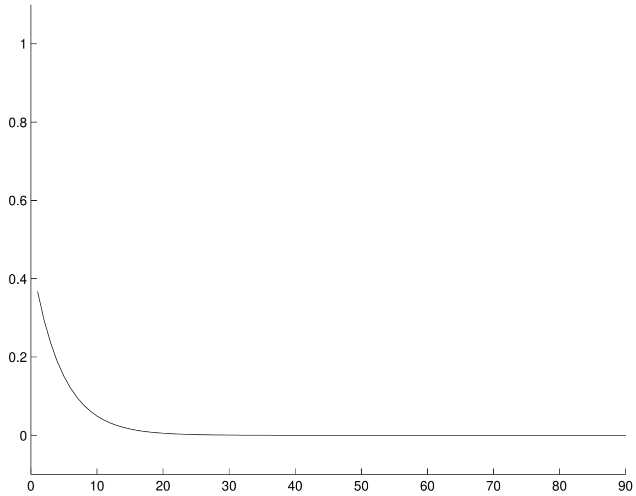


Fig. 9.a

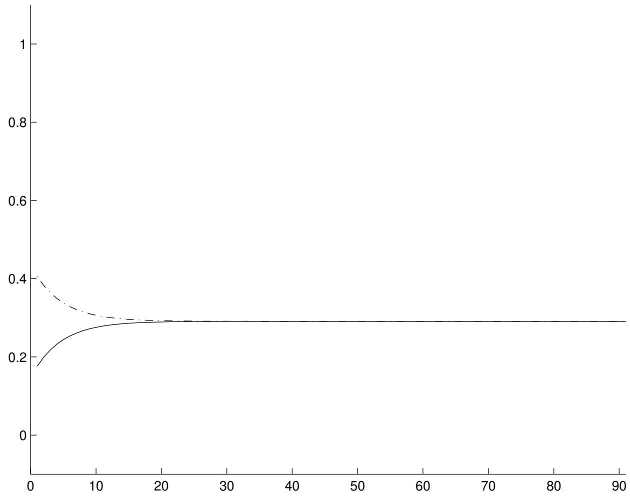
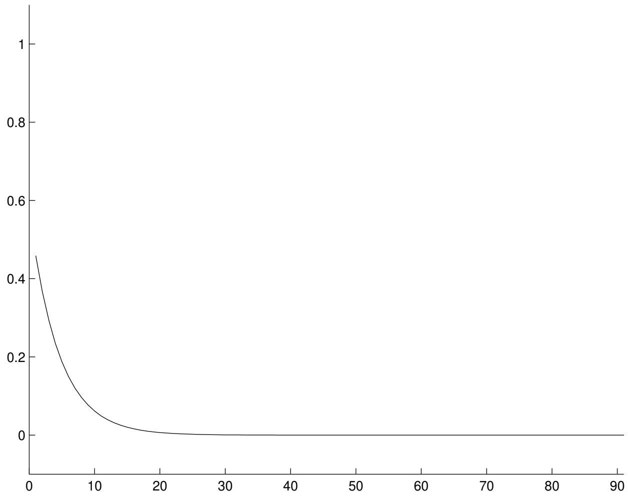


Fig. 9.b



A.4 Example 4

Our third example is somewhat more complicated but still involves only $\widetilde{\mathbf{V}}_2$ sentences:

$$A_1 : \text{"}A_2 \text{ is true and } A_3 \text{ is false "} \tag{33}$$

$$A_2 : \text{"}A_1 \text{ is true and } A_3 \text{ is false "} \tag{34}$$

$$A_3 : \text{"}A_1 \text{ is false".} \tag{35}$$

(33)–(35) translates to

$$A_1 = D_1 = C_1 \wedge C_2$$
$$A_2 = D_2 = C_3 \wedge C_2$$
$$A_3 = D_3 = C_4$$

where

$$C_1 : \text{"Tr}\,(A_2) = 1\text{"}$$
$$C_2 : \text{"Tr}\,(A_3) = 0\text{"}$$
$$C_3 : \text{"Tr}\,(A_1) = 1\text{"}$$
$$C_4 : \text{"Tr}\,(A_1) = 0\text{".}$$

We will consider two different implemementations of $\wedge$.

A.4.1 $\wedge$ Implemented by Minimum

If we implement $\wedge$ by the min t-norm, the truth value equations become

$$x_1 = \min\left[x_2, (1 - x_3)\right] \tag{36}$$

$$x_2 = \min\left[x_1, (1 - x_3)\right] \tag{37}$$

$$x_3 = 1 - x_1 \tag{38}$$

and the inconsistency function is

$$J = \left(x_1 - \min\left[x_2, (1 - x_3)\right]\right)^2 + \left(x_2 - \min\left[x_1, (1 - x_3)\right]\right)^2 + \left(x_3 - (1 - x_1)\right)^2 . \tag{39}$$

The truth value equations can be solved analytically. From (38) we obtain

$$x_1 = 1 - x_3$$

and then (36) – (37) become

$$x_1 = \min [x_2, x_1], \qquad x_2 = \min [x_1, x_1]$$

from which follows that

$$x_1 = x_2, \qquad x_3 = 1 - x_1.$$

In other words, the general solution of (36) – (38) is

$$x = (\beta, \beta, 1 - \beta)$$

with $\beta \in [0, 1]$. Note that this includes the extremal solutions $(1, 1, 0)$ and $(0, 0, 1)$ as well as the mid-point solution $(1/2, 1/2, 1/2)$. Also, using (38) the inconsistency function becomes

$$J = (x_1 - \min [x_2, x_1])^2 + (x_2 - \min [x_1, x_1])^2 + (x_3 - (1 - x_1))^2$$

which attains the minimum value of 0 for every $x = (\beta, \beta, 1 - \beta)$ with $\beta \in [0, 1]$. For the steepest descent algorithm the dynamical system is

$$x (t + 1) = x (t) - k \cdot \frac{\partial J}{\partial x}. \tag{40}$$

We cannot write equations for the gradient explicitly, because the expressions $\min [x_2, 1 - x_3]$ and $\min [x_1, 1 - x_3]$ are not everywhere differentiable. However, using an approximate numerical differentiation we can simulate (40). In Figures 10 and 11 we present results for two typical simulations with $k = 0.01$.

Fig. 10.a

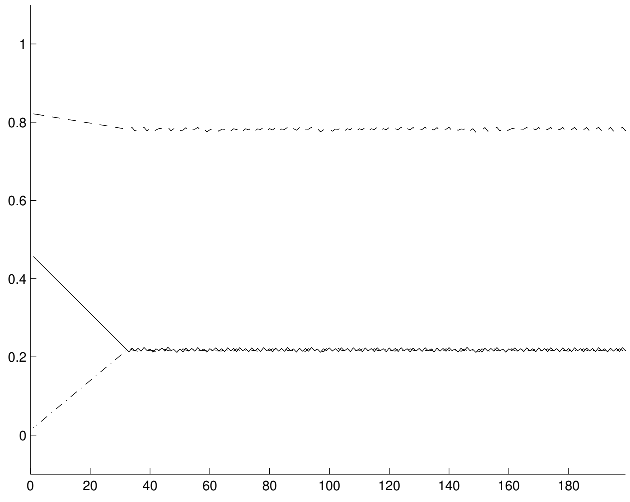


Fig. 11.b

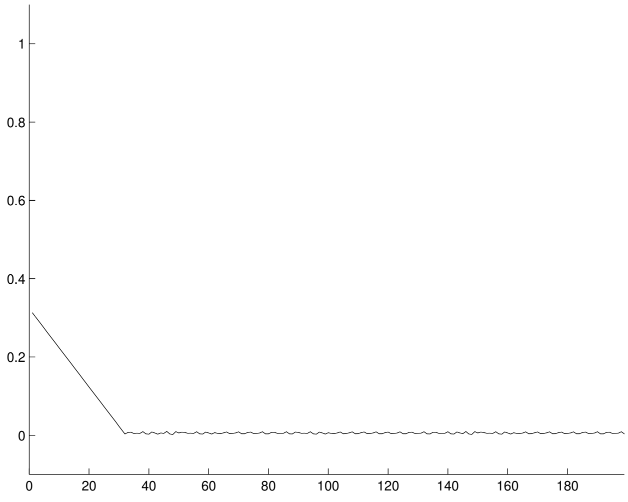


Fig. 11.a

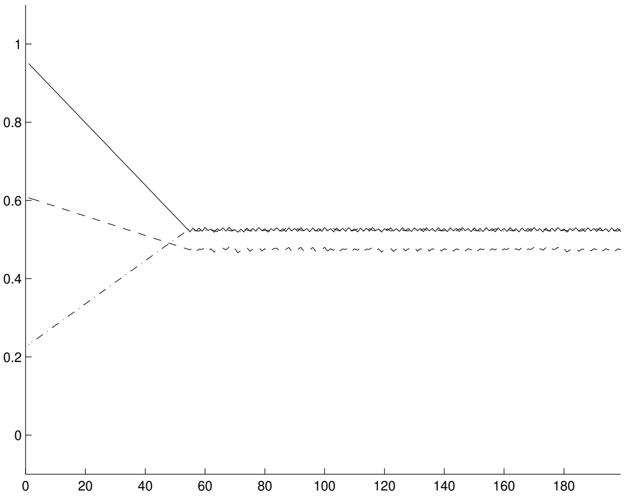
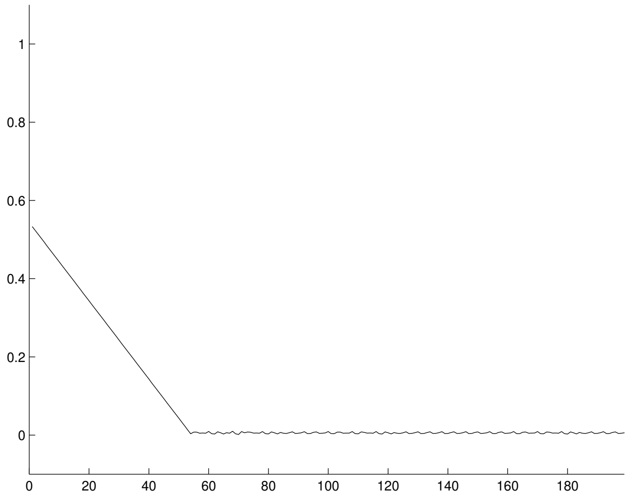


Fig. 11.b



Let us also present the results of the control theoretic algorithm; the dynamical system is

$$
\begin{aligned}
x_1(t+1) &= x_1(t) - k \cdot (x_2 - \min[x_2(t), 1 - x_3(t)]) \\
x_2(t+1) &= x_2(t) - k \cdot (x_2 - \min[x_1(t), 1 - x_3(t)]) \\
x_3(t+1) &= x_3(t) - k \cdot (x_3 - (1 - x_1(t))) .
\end{aligned}
$$

This system is non-linear. In Figures 12 and 13 we present results for two typical simulations with $k = 0.1$.

Fig.12.a

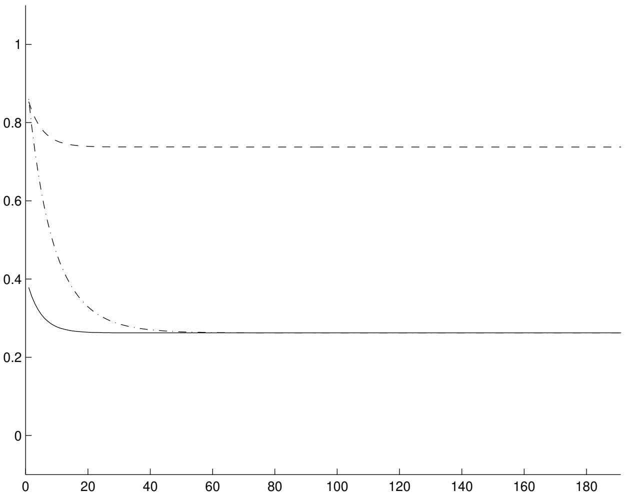


Fig.12.b

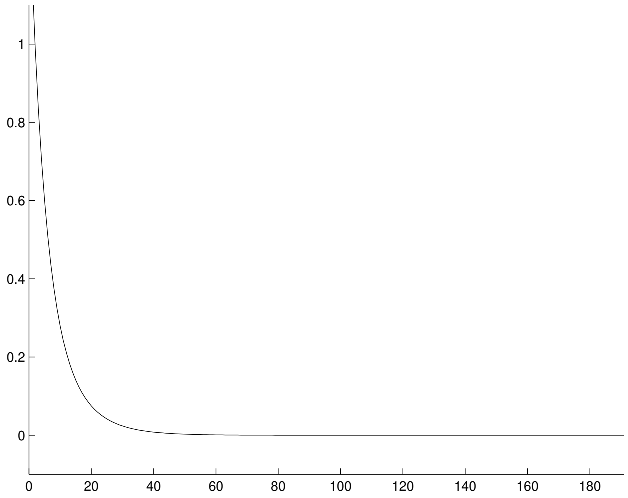


Fig.13.a

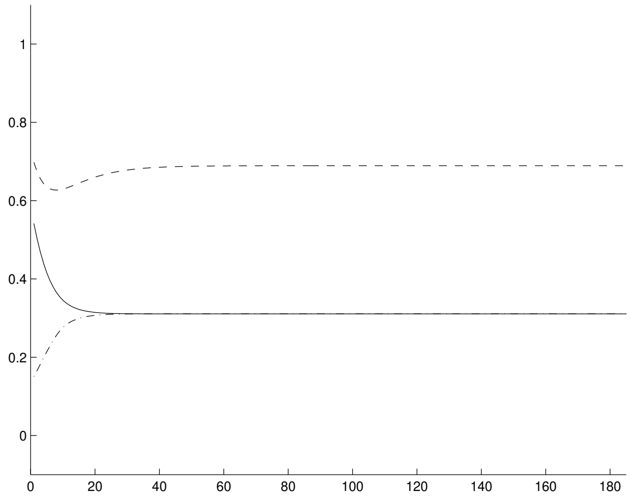
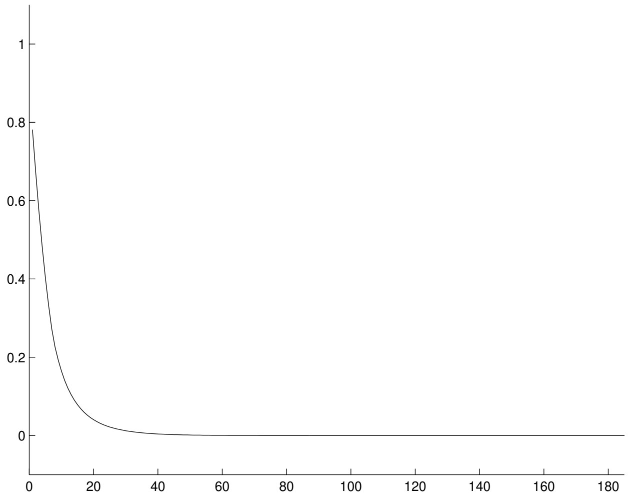


Fig.13.b



A.4.2 $\wedge$ Implemented by Product

Implementing $\wedge$ by the product t-norm we obtain the truth value equations

$$\begin{aligned} x_1 &= x_2 \cdot (1 - x_3) \\ x_2 &= x_1 \cdot (1 - x_3) \\ x_3 &= 1 - x_1 \end{aligned}$$

We can still use $x_1 = 1 - x_3$ to simplify the truth value equations to

$$x_1 = x_2 \cdot x_1, \qquad x_2 = x_1^2, \qquad x_3 = 1 - x_1$$

from which we obtain

$$x_1 = x_1^3, \qquad x_2 = x_1^2, \qquad x_3 = 1 - x_1$$

and finally

$$x_1 \cdot \left(1 - x_1^2\right) = 0, \qquad x_2 = x_1^2, \qquad x_3 = 1 - x_1.$$

This has the solutions

$$(0,0,1)\,, \qquad (1,1,0)\,, \qquad (-1,1,0)\,;$$

the last solution, however, is inadmissible as a truth value assignment. Hence we see that for the same self-referential collection, the product implementation of $\wedge$ yields a subset of the solutions obtained through the min implementation.

The inconsistency function is

$$J = (x_1 - x_2 \cdot (1 - x_3))^2 + (x_2 - x_1 \cdot (1 - x_3))^2 + (x_3 - (1 - x_1))^2$$

Because the product operator is everywhere differentiable, we can write explicitly the gradient equations. We have

$$\begin{aligned} \frac{\partial J}{\partial x_1} &= -4x_1 + 4x_2 + 4x_1x_3 - 4x_2x_3 + 2x_2x_3^2 \\ \frac{\partial J}{\partial x_2} &= 6x_1 - 4x_2 + 2x_3 + 4x_2x_3 - 4x_1x_3 + 2x_1x_3^2 - 2 \\ \frac{\partial J}{\partial x_3} &= 2x_1 + 2x_3 + 4x_1x_2 - 2x_2^2 + 2x_2^2x_3 - 2x_1^2 + 2x_1^2x_3 - 2. \end{aligned}$$

Using the above expressions and the steepest descent algorithm we obtain results of the type presented in Figure 14. Note that in the product implementation the steepest descent algorithm requires a considerably larger number of steps to converge than for the min implementation.

Fig. 14.a

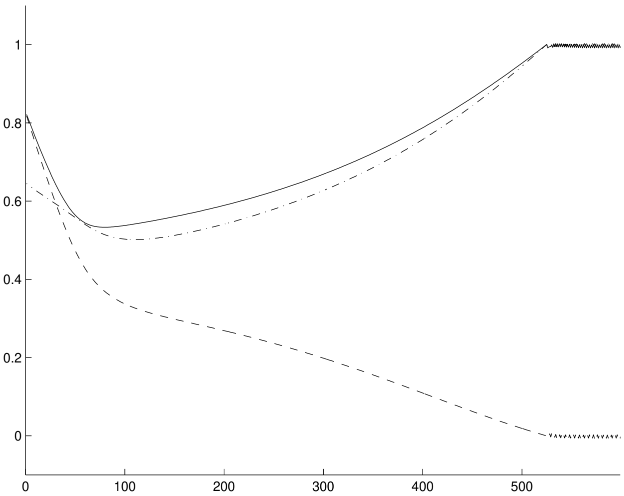
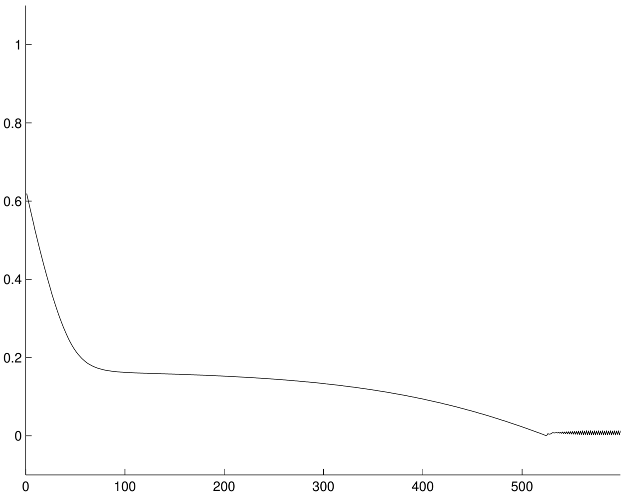


Fig. 14.b



For the the control theoretic algorithm the dynamical system is

$$\begin{aligned} x_1(t+1) &= x_1(t) - k \cdot (x_1(t) - x_2(t) \cdot (1 - x_3(t))) \\ x_2(t+1) &= x_2(t) - k \cdot (x_2(t) - x_1(t) \cdot (1 - x_3(t))) \\ x_3(t+1) &= x_3(t) - k \cdot (1 - x_3(t)). \end{aligned}$$

In Figure 15 we present a typical simulation result. Note the fast convergence to equilibirum.

Fig. 15.a

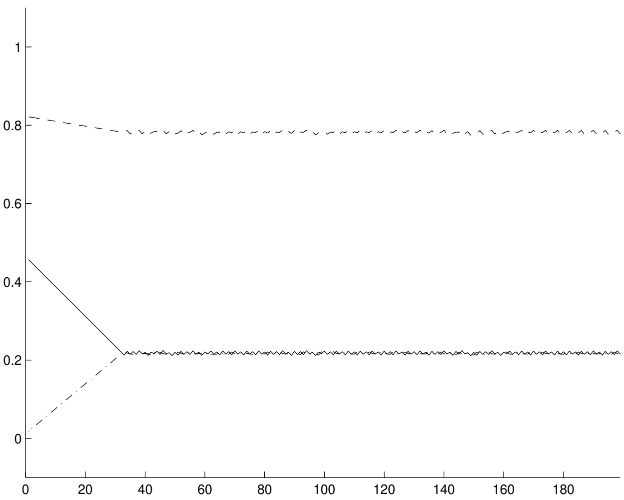
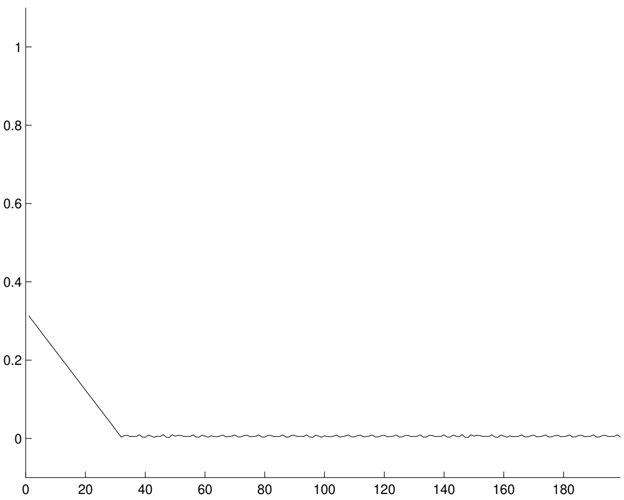


Fig. 15.b



A.5 Example 5

Let us now look at an example similar to the previous one, but involving sentences outside of $\widetilde{\mathbf{S}}_2$:

$$A_1 : \text{“The truth value of } A_2 \text{ is 0.90 and the truth value of } A_3 \text{ is 0.20”} \tag{41}$$

$$A_2 : \text{“The truth value of } A_1 \text{ is 0.80 and the truth value of } A_3 \text{ is 0.30”} \tag{42}$$

$$A_3 : \text{“The truth value of } A_1 \text{ is 0.10”.} \tag{43}$$

(41)-(43) translates to

$$\begin{aligned} A_1 &= D_1 = C_1 \wedge C_2 \\ A_2 &= D_2 = C_3 \wedge C_4 \\ A_3 &= D_3 = C_5 \end{aligned}$$

where

$$\begin{aligned} C_1 &: \text{“Tr}\left(A_2\right) = 0.90\text{”} \\ C_2 &: \text{“Tr}\left(A_3\right) = 0.20\text{”} \\ C_3 &: \text{“Tr}\left(A_1\right) = 0.80\text{”} \\ C_4 &: \text{“Tr}\left(A_3\right) = 0.30\text{”} \\ C_5 &: \text{“Tr}\left(A_1\right) = 0.10\text{”.} \end{aligned}$$

We will consider two different implemementations of $\wedge$.

A.5.1 $\wedge$ Implemented by Minimum

If we implement $\wedge$ with the min operator the truth value equations become

$$\begin{aligned} x_1 &= \min\left[1 - |x_2 - 0.90|, 1 - |x_3 - 0.20|\right] \\ x_2 &= \min\left[1 - |x_2 - 0.80|, 1 - |x_3 - 0.30|\right] \\ x_3 &= 1 - |x_1 - 0.10|. \end{aligned}$$

These equations cannot be further reduced and while in principle they can be solved analytically by distinguishing cases, this requires an inordinate amount of work. Instead, we will solve them numerically, using both Newton-Raphson and the control algorithm. Similarly, we will use steepest descent to minimize the inconsistency function

$$J\left(x_1, x_2, x_3\right) = \sum_{m=1}^{3} J_m\left(x_1, x_2, x_3\right)$$

where

$$\begin{aligned} J_1\left(x_1, x_2, x_3\right) &= \left(x_1 - \min\left[1 - |x_2 - 0.90|, 1 - |x_3 - 0.20|\right]\right)^2 \\ J_2\left(x_1, x_2, x_3\right) &= \left(x_2 - \min\left[1 - |x_2 - 0.80|, 1 - |x_3 - 0.30|\right]\right)^2 \\ J_3\left(x_1, x_2, x_3\right) &= \left(x_3 - 1 - |x_1 - 0.10|\right)^2; \end{aligned}$$

the dynamical system corresponding to the steepest descent algorithm is

$$
\begin{aligned}
x_1(t+1) &= x_1(t) - k \cdot \frac{\partial J}{\partial x_1} \\
x_2(t+1) &= x_2(t) - k \cdot \frac{\partial J}{\partial x_2} \\
x_3(t+1) &= x_3(t) - k \cdot \frac{\partial J}{\partial x_3}.
\end{aligned}
$$

For a numerical implementation the partial derivatives can be approximated numerically. Finally, for the control theoretic algorithm the dynamical system is

$$
\begin{aligned}
x_1(t+1) &= x_1(t) - k \cdot (x_1(t) - \min[1 - |x_2(t) - 0.90|, 1 - |x_3(t) - 0.20|]) \\
x_2(t+1) &= x_2(t) - k \cdot (x_2(t) - \min[1 - |x_1(t) - 0.90|, 1 - |x_3(t) - 0.20|]) \\
x_3(t+1) &= x_3(t) - k \cdot (x_3(t) - 1 + |x_3(t) - 0.20|).
\end{aligned}
$$

In Figure 16 we present some simulation results for the Newton-Raphson algorithm, in Figure 17 we present results for the steepest descent algorithm and in Figure 18 for the control theoretic algorithm. Both Newton-Raphson and the control algorithm discover the same solution, namely $\overline{x} = (0.95, 0.85, 0.15)^T$. Repeated simulations (not presented here) give always the same solution, so it is possible that this is the *unique* consistent truth value assignment for this problem. The steepest descent algorithm gets trapped at a local minimum, namely $\overline{x} = (0.56, 0.71, 0.59)^T$, which does *not* yield zero inconsistency. This is not accidental; we have noticed after repeated simulations (not presented here) that in general the steepest descent algorithm in this case does not yield zero inconsistency, except when started quite close to the true solution. Note the very fast convergence of Newton-Raphson: it reaches equilibrium in 7 steps, as compared to the approximately 130 steps required by the control algorithm.

Fig. 16.a

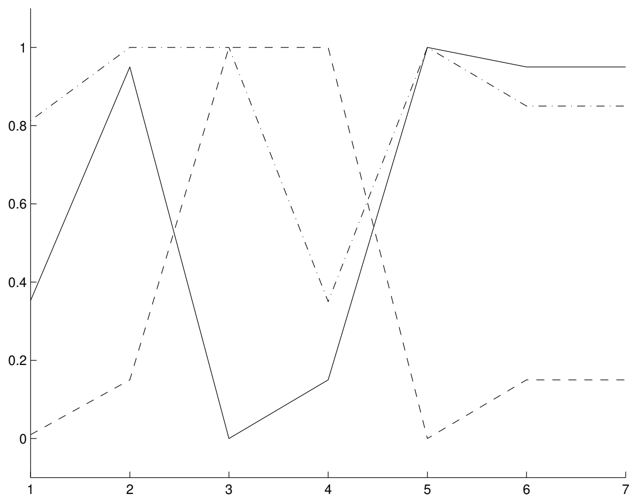


Fig. 16.b

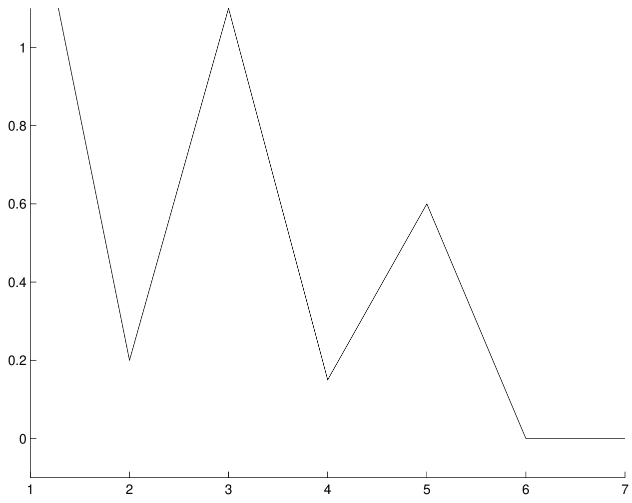


Fig. 17.a

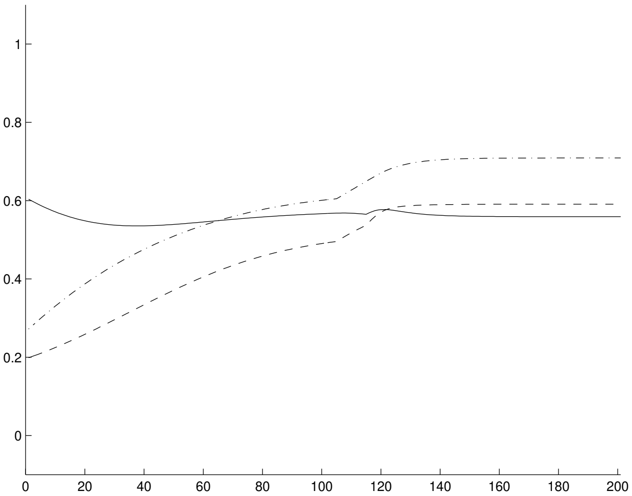


Fig. 17.b

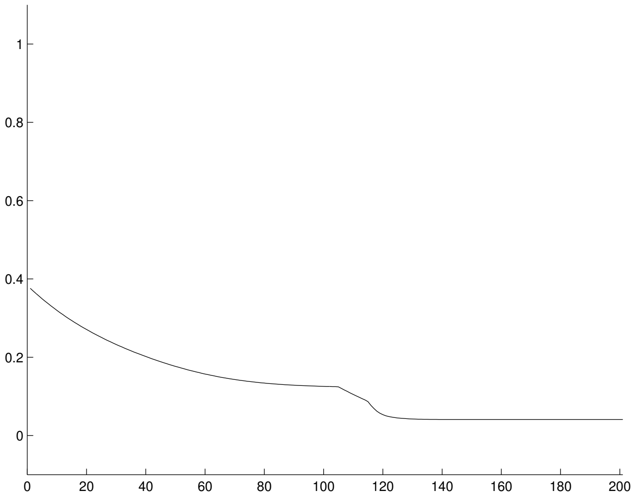


Fig. 18.a

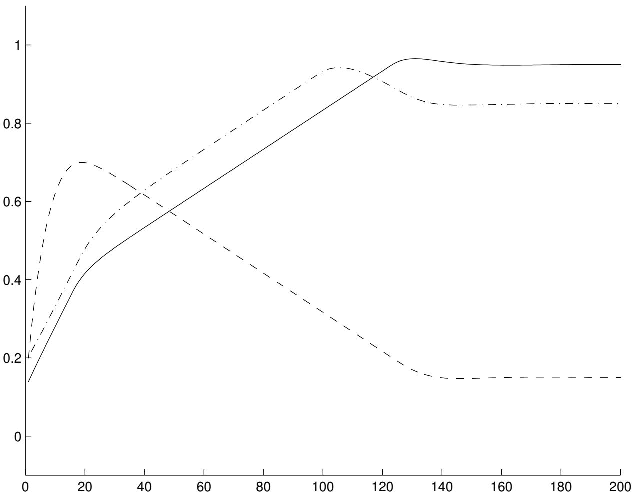
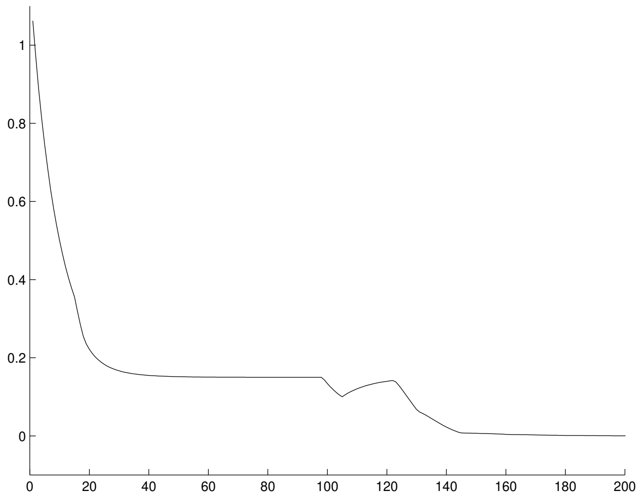


Fig. 18.b



A.5.2 $\wedge$ Implemented by Product

The situation is similar when we implement $\wedge$ by product.. The truth value equations become

$$
\begin{aligned}
x_1 &= (1 - |x_2 - 0.90|) \cdot (1 - |x_3 - 0.20|) \\
x_2 &= (1 - |x_2 - 0.80|) \cdot (1 - |x_3 - 0.30|) \\
x_3 &= 1 - |x_1 - 0.10|
\end{aligned}
$$

We omit the details which are very similar to previous cases and present the results of numerical simulation.

In Figure 19 we present some simulation results for the Newton-Raphson algorithm, in Figure 20 we present results for the steepest descent algorithm and in Figure 21 for the control theoretic algorithm. The Newton-Raphson and steepest descent algorithms reach the solution $\overline{x} = (0.6784, 0.7715, 0.4216)^T$, while the control algorithm reaches the solution $\widetilde{x} = (0.0473, 0.0872, 0.9473)^T$. Both of these are consistent truth value assignments; other simulations (not presented here) have yielded additional solutions. Note that in this case the steepest descent algorithm reaches zero inconsistency. Otherwise, the remarks previously made for the min implementation also hold for the product implementation.

Fig. 19.a

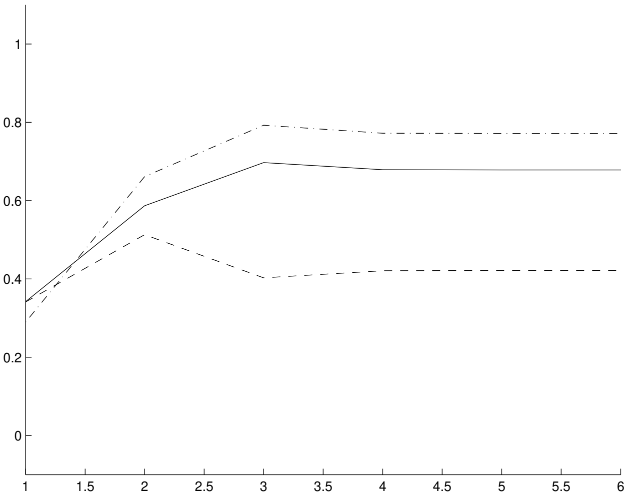


Fig. 19.b

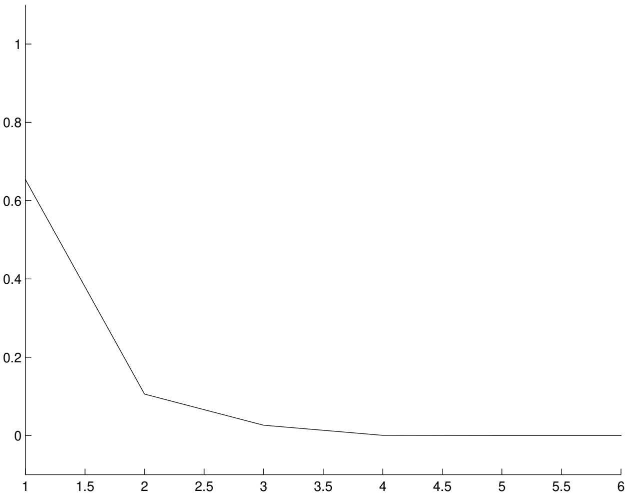


Fig. 20.a

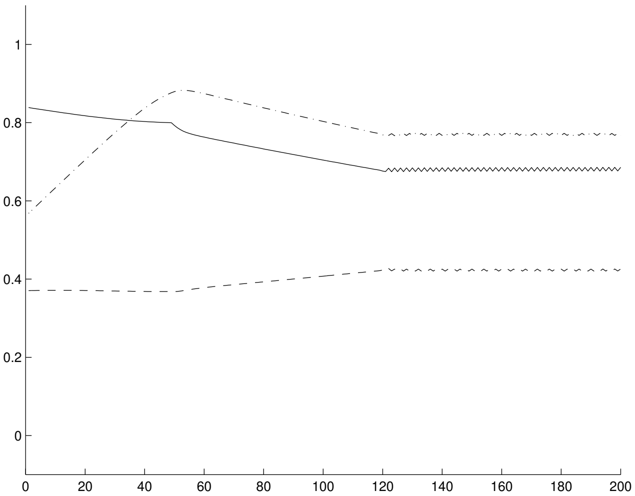


Fig. 20.b

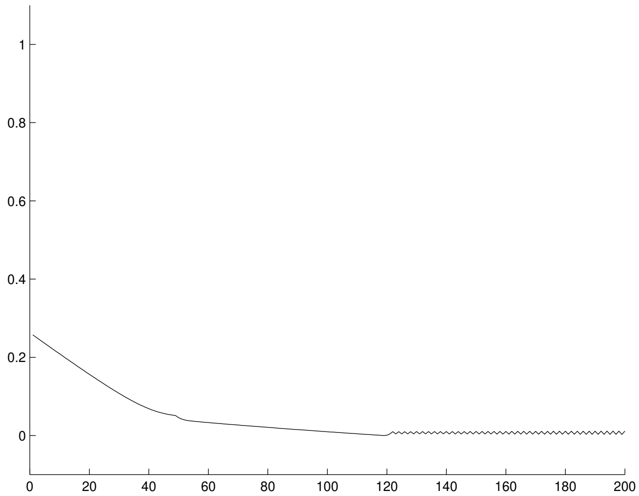


Fig. 21.a

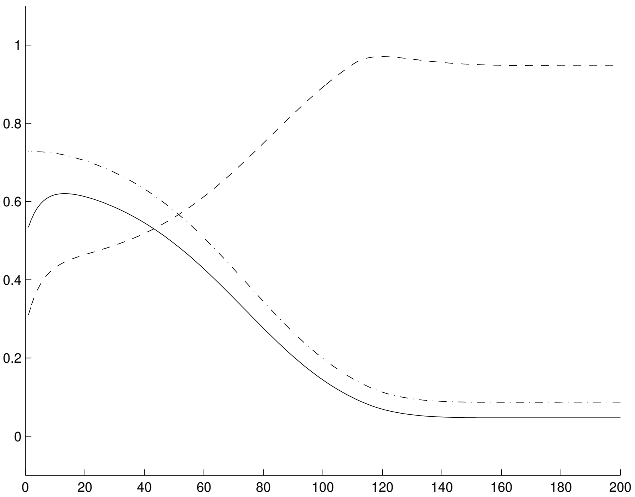
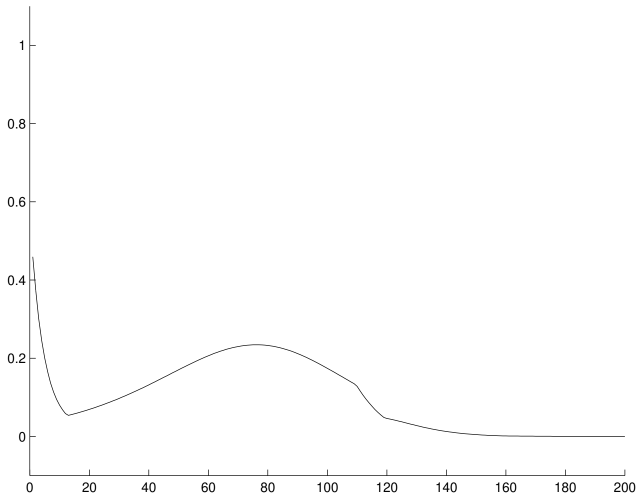


Fig. 21.b



A.6 Example 6

Here is a more complicated example:

A_1 : ("A_1 has truth value 0.75" and "A_2 has truth value 0.35") or "A_4 has truth value 1.00"
A_2 : ("A_1 or A_3 has truth value 1.00") and "A_4 has truth value 0.10"
A_3 : "A_2 has truth value 0.00" and "A_3 has truth value 0.35"
A_4 : "The opposite of A_1 has truth value 0.25"

This translates to

$$\begin{aligned}
A_1 &= (C_1 \wedge C_2) \vee C_3 \\
A_2 &= C_4 \wedge C_5 \\
A_3 &= C_6 \wedge C_7 \\
A_4 &= C_8
\end{aligned}$$

where

C_1 : "The truth value of A_1 is 0.75"
C_2 : "The truth value of A_2 is 0.35"
C_3 : "The truth value of A_4 is 1.00"
C_4 : "The truth value of $A_1 \vee A_3$ is 1.00"
C_5 : "The truth value of A_4 is 0.10"
C_6 : "The truth value of A_2 is 0.00"
C_7 : "The truth value of A_3 is 0.35"
C_8 : "The truth value of A_1' is 0.25".

Note that C_4 belongs to $\mathbf{S}_2$ proper, i.e. it is *not* an elementary truth value assessment.

We consider two alternative systems of truth value equations; in the first case we implement $\vee$, $\wedge$ by the standard implementation; in the second case we implement $\vee$, $\wedge$ by the algebraic implementation.

A.6.1 $\wedge$ Implemented by Minimum, $\vee$ Implemented by Maximum

We present the truth value equations:

$$x_1 = \max\left[\min\left(1 - |x_1 - 0.75|, 1 - |x_2 - 0.35|\right), 1 - |x_4 - 1.00|\right] \tag{44}$$
$$x_2 = \min\left[1 - |\max(x_1, x_3) - 1.00|, 1 - |x_4 - 0.10|\right] \tag{45}$$
$$x_3 = \min\left[1 - |x_2 - 0.00|, 1 - |x_3 - 0.35|\right] \tag{46}$$
$$x_4 = 1 - |1 - x_1 - 0.25|. \tag{47}$$

We omit further details and directly present the results of numerical simulation.

In Figures 22, 23 we present some simulation results for the Newton-Raphson algorithm, with various initial conditions. Note that in both figures the algorithm fails to locate a consistent truth value assignment; this is true for most of the simulations we have run, i.e. usually Newton-Raphson fails to solve the truth value equations (44)–(47).

Fig. 22.a

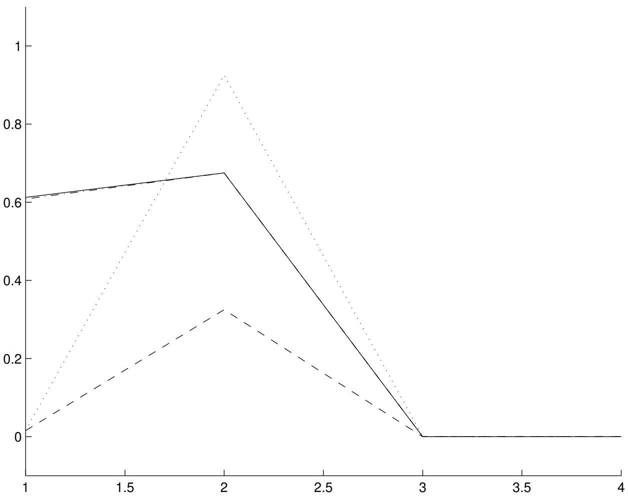


Fig. 22.b

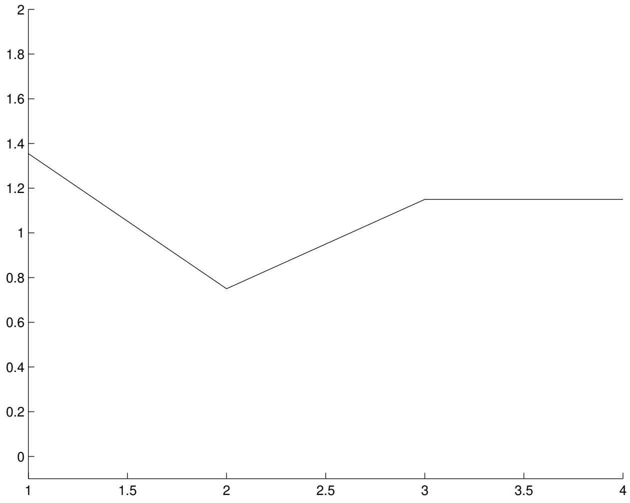


Fig. 23.a

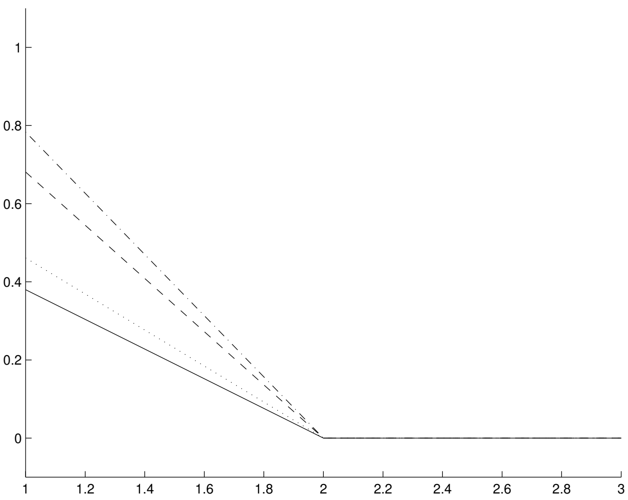
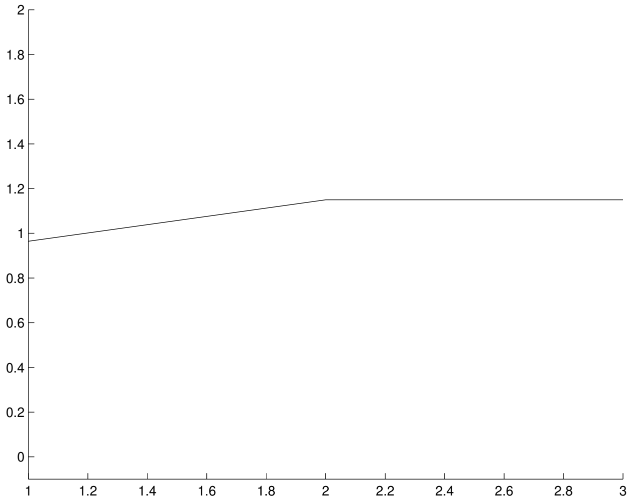


Fig. 23.b



In Figures 24, 25 we present some simulation results for the steepest descent algorithm, with various initial conditions. In Figure 24 the algorithm converges to $\overline{x}$=$(0.875, 0.225, 0.675, 0.875)^T$ which yields zero total inconsistency; this is not the case in Figure 25, where the algorithm converges to a local minimum and the truth value equations are not satisfied. This has occurred in several simualtions, i.e. the steepest descent algorithm does not reliably solve the truth value equations (44)–(47).

Fig. 24.a

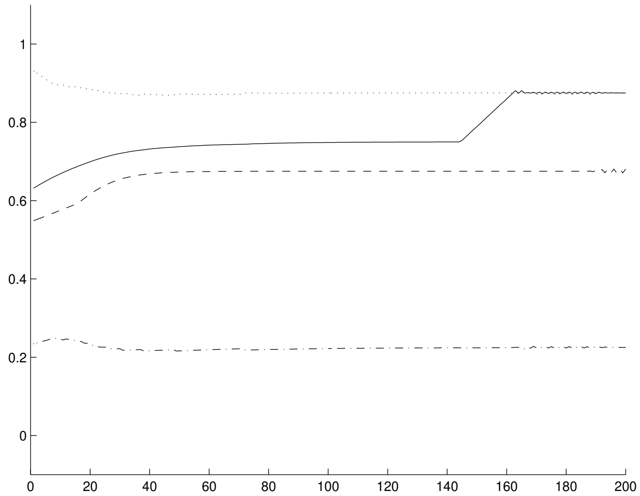


Fig. 24.b

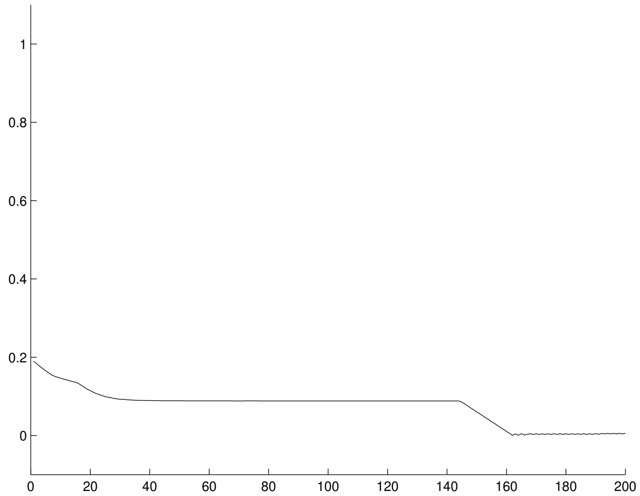


Fig. 25.a

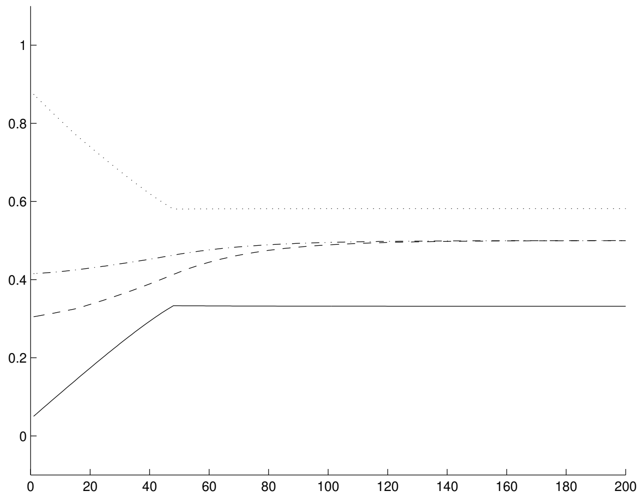
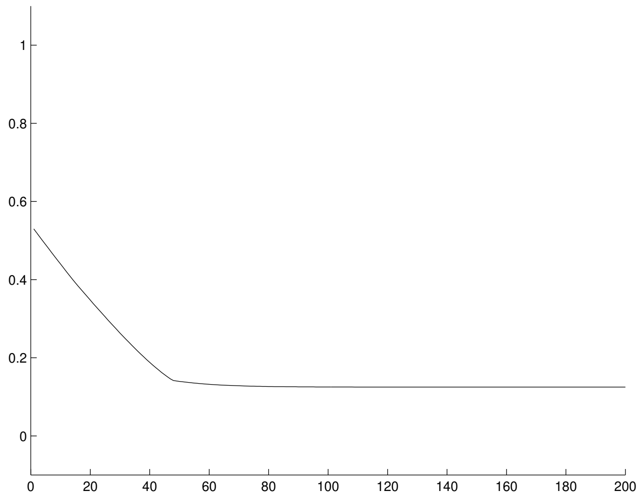


Fig. 25.b



In Figures 26, 27 we present some simulation results for the control theoretic algorithm, with various initial conditions. The algorithm always converges to $\overline{x}$=$(0.875, 0.225, 0.675, 0.875)^T$ which yields zero total inconsistency. Indeed, having run a large number of additional simulations we have noticed that the control algorithm *always* converges; furthermore the equilibrium is always the above mentioned $\overline{x}$, which perhaps indicates that it is the unique solution of the truth value equations (44)–(47).

Fig. 26.a

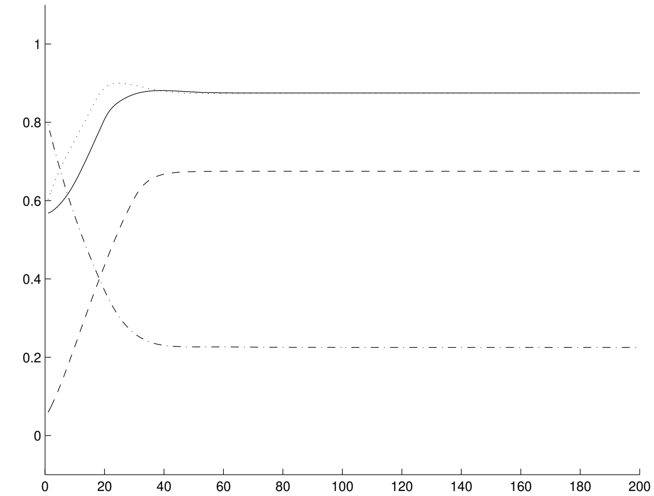


Fig. 26.b

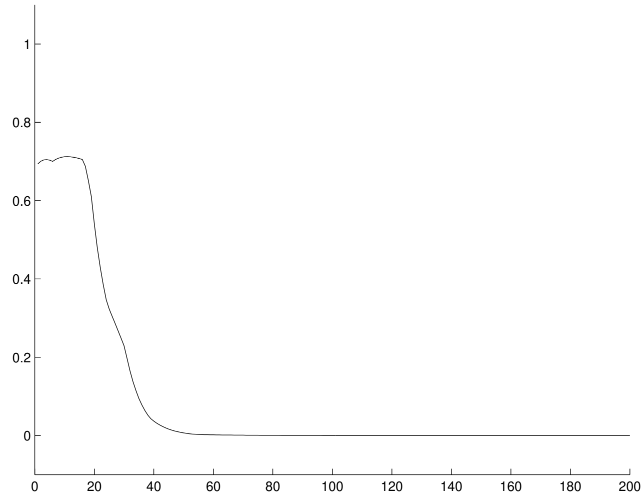


Fig. 27.a

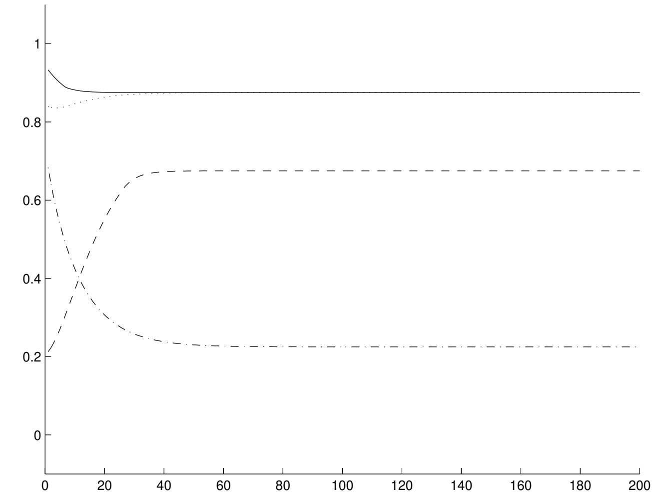
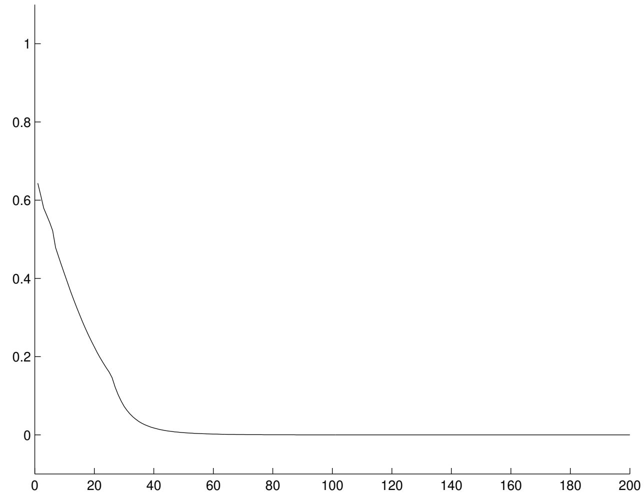


Fig. 27.b



A.6.2 $\wedge$ Implemented by Product, $\vee$ Implemented by Extended Sum

We present the truth value equations:

$$\begin{aligned}
x_1 &= (1-|x_1-0.75|)\cdot(1-|x_2-0.35|)+(1-|x_4-1.00|) \\
&\quad -(1-|x_1-0.75|)\cdot(1-|x_2-0.35|)\cdot(1-|x_4-1.00|) && (48) \\
x_2 &= (1-|x_1+x_3-x_1x_3-1.00|)\cdot(1-|x_4-0.10|) && (49) \\
x_3 &= (1-|x_2-0.00|)\cdot(1-|x_3-0.35|) && (50) \\
x_4 &= 1-|1-x_1-0.25|\,. && (51)
\end{aligned}$$

We omit further details and directly present the results of numerical simulation. In Figures 28 and 29 we present some simulation results for the Newton-Raphson algorithm, with various initial conditions. The algorithm always finds the solution $\overline{x}=(0.9507,0.2942,0.5586,0.7993)^T$. In Figures 30 and 31 we present some simulation results for the steepest descent algorithm, with various initial conditions. Note that in Figure 31 the algorithm converges to a local minimum (nonzero) of the total inconsistency.

Fig. 28.a

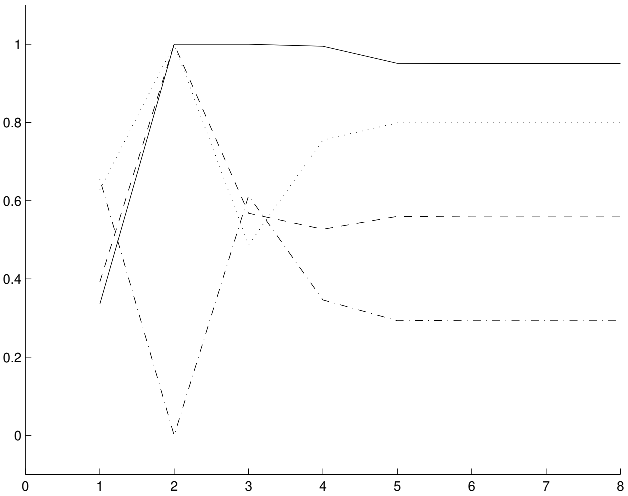


Fig. 28.b

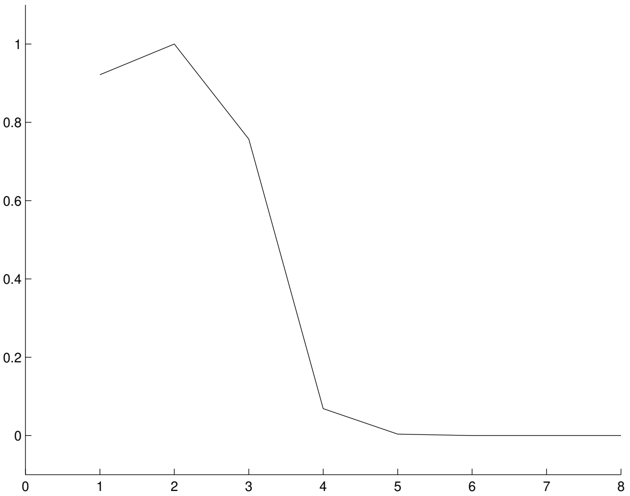


Fig. 29.a

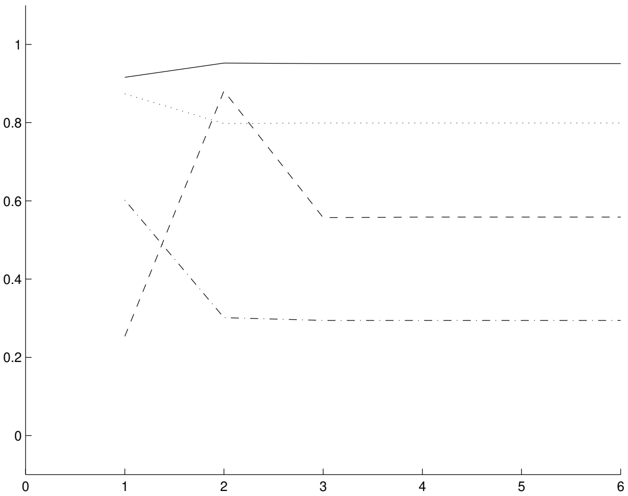


Fig. 29.b

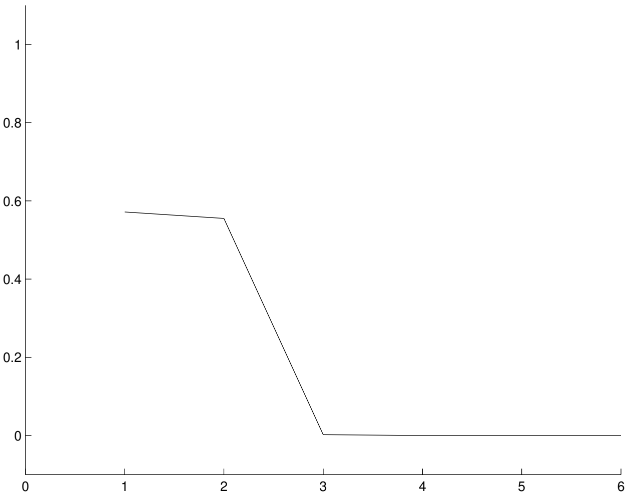


Fig. 30.a

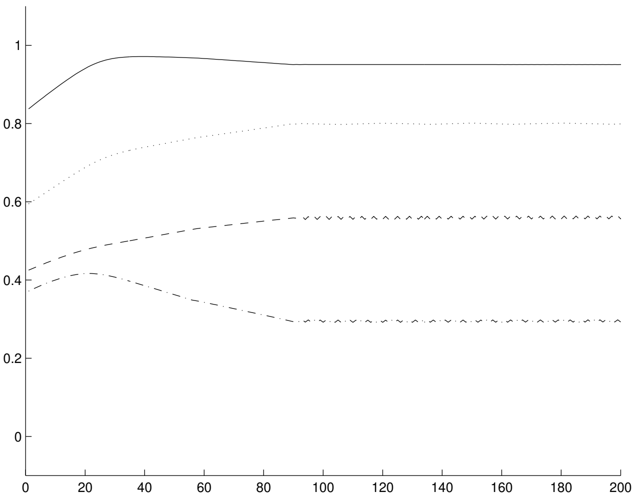


Fig. 30.b

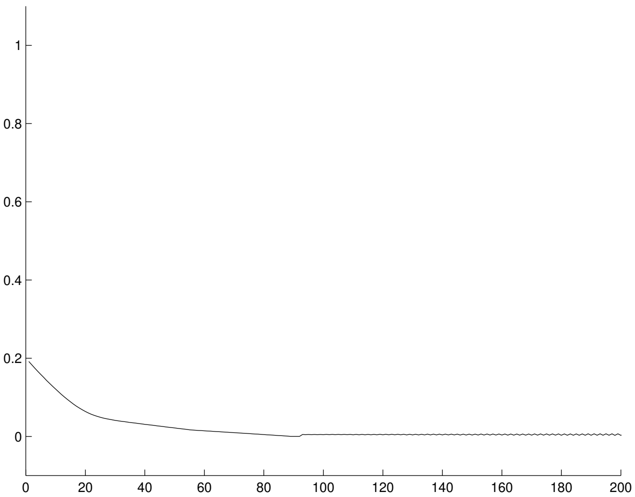


Fig. 31.a

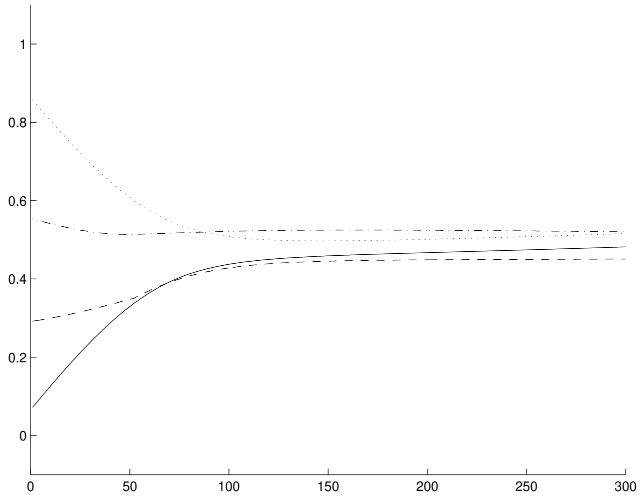
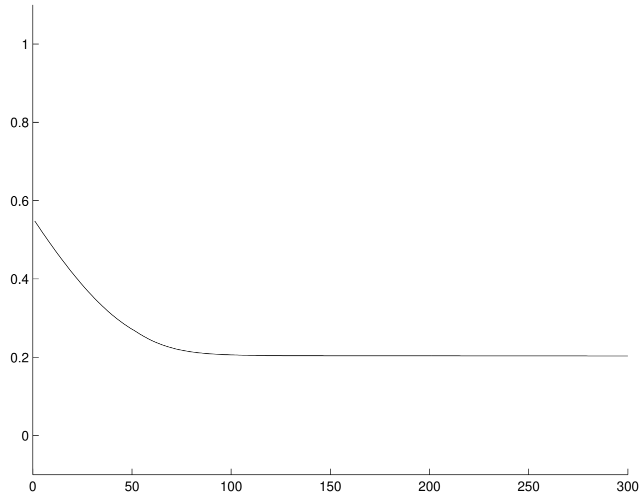


Fig. 31.b



Finally, in Figures 32 and 33 we present some simulation results for the control theoretic algorithm, with various initial conditions. The algorithm always converges to the following solution of the truth value equations (48)–(51): $\overline{x} = (0.9507, 0.2942, 0.5586, 0.7993)^T$, rendering likely that this is the unique solution.

Fig. 32.a

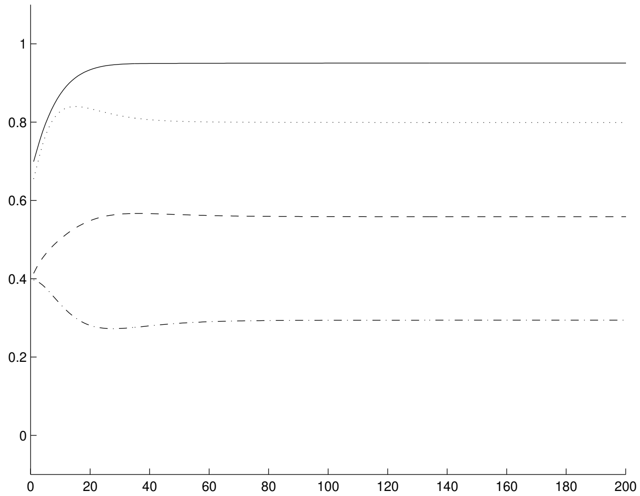


Fig. 32.b

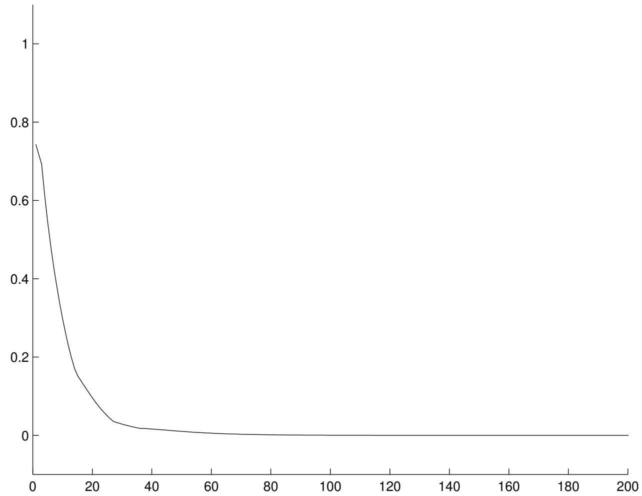


Fig. 33.a

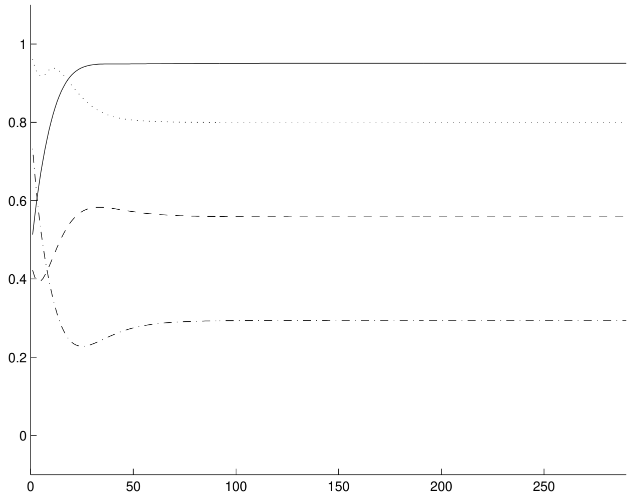
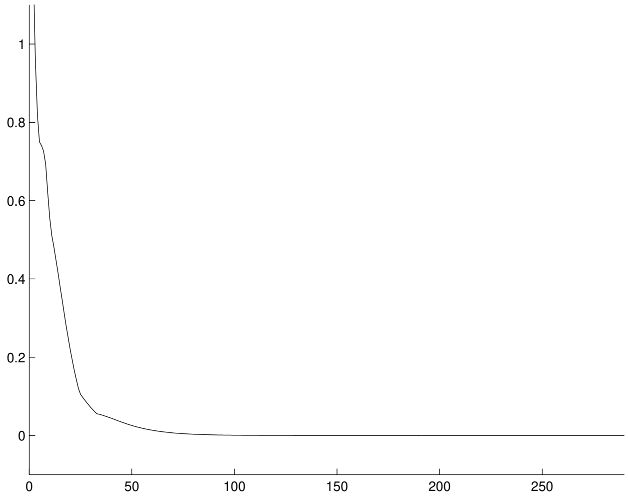


Fig. 33.b



A.7 Example 7: The Strengthened Liar

This final example does not fall, strictly speaking, within the framework we have presented. However it can be treated by a small, straightforward extension of our approach.

The example is the so-called "Strengthened Liar", a self-referential sentence which has been often used as a test of proposed solutions to the Liar paradox [37]. The Strengthened Liar sentence is

$$A = \text{“}A \text{ is not true”}. \tag{52}$$

To treat this and similar sentences in the fuzzy context we must translate it in terms of a membership function for the property of being not true. To this end, consider the sentence

$$C = \text{“The truth value of } A \text{ is not } a\text{”}. \tag{53}$$

A possible truth value assignment for (53) is

$$\mathrm{Tr}\,(C) = \begin{cases} 1 & \text{when } \mathrm{Tr}\,(A) \neq a \\ 0 & \text{else} \end{cases} .$$

However, this is too strict. Consider the case when $a = 1$ and $\mathrm{Tr}(A) = 0.99$. Do we really want to assign $\mathrm{Tr}(C) = 1$? How about the case $\mathrm{Tr}(A) = 0.99999$? A more reasonable truth value assignment is

$$\mathrm{Tr}\,(C) = |\mathrm{Tr}\,(A) - a| \tag{54}$$

which takes the maximum value of 1 when $|\mathrm{Tr}\,(A) - a| = 1$, i.e. in the cases

1. $\mathrm{Tr}(A) = 1$ and $a = 0$;
2. $\mathrm{Tr}(A) = 0$ and $a = 1$.

Let us accept (54) and set $A = C$, i.e.

$$A = \text{“}\mathrm{Tr}\,(A) \neq a\text{”}. \tag{55}$$

(55) is more general than (52); to obtain (52) we set $a = 1$:

$$A = \text{“}\mathrm{Tr}\,(A) \neq 1\text{”}.$$

Hence, setting $x =$Tr(A), we must solve the truth value equation

$$x = |x - 1| = 1 - x$$

which has the unique solution $x = 1/2$.

Obviously this approach can be extended to treat situations which involve statements of the form

"The truth value of B is not b"

for any $B \in \mathbf{S}_1$. Hence our framework can be extended defining $\mathbf{V}_2$, the set of 2nd level elementary sentences to include both sentences of the form

$$\mathrm{Tr}\,(B) = b$$

and

$$\mathrm{Tr}\,(B) \neq b.$$

The definition of $\mathbf{S}_2$ remains unchanged but, since it depends on the expanded $\mathbf{V}_2$, results to an expansion of $\mathbf{S}_2$ as well. We leave the details to the reader

A.8 Discussion of the Algorithms

Let us close with a short comparison of the three numerical algorithms. We see that the Newton-Raphson algorithm converges very quickly but not always to a solution of the truth value equations. Regarding steepest descent, its convergence is guaranteed but not always to point of zero inconsistency. It appears that the control-theoretic algorithm combines the best properties: practically guaranteed convergence to a point of zero inconsistency and in a relatively small number of iterations. Of course, all of the above remarks refer to the experiments of the previous sections; further numerical experiments (and, perhaps, analytical work) are required to establish their general validity.