Numerical Algorithms **37:** 213–224, 2004.

An improved interval linearization for solving nonlinear problems

Lubomir V. Kolev

Department of Theoretical Electrotechnics, Faculty of Automatica, Technical University of Sofia, 1756 Sofia, Bulgaria
E-mail: lkolev@tu-sofia.bg

Let $f(x)$ denote a system of n nonlinear functions in m variables, $m \geqslant n$. Recently, a linearization of $f(x)$ in a box $\boldsymbol{x}$ has been suggested in the form $\boldsymbol{L}(x) = Ax + \boldsymbol{b}$ where A is a real $n \times m$ matrix and $\boldsymbol{b}$ is an interval n-dimensional vector. Here, an improved linearization $\boldsymbol{L}(x, y) = Ax + By + b$, $x \in \boldsymbol{x}$, $y \in \boldsymbol{y}$ is proposed where y is a p-dimensional vector belonging to the interval vector $\boldsymbol{y}$ while A and B are real matrices of appropriate dimensions and b is a real vector. The new linearization can be employed in solving various nonlinear problems: global solution of nonlinear systems, bounding the solution set of underdetermined systems of equations or systems of equalities and inequalities, global optimization. Numerical examples illustrating the superiority of $\boldsymbol{L}(x, y) = Ax + By + b$ over $\boldsymbol{L}(x) = Ax + \boldsymbol{b}$ have been solved for the case where the problem is the global solution of a system of nonlinear equations ($n = m$).

Keywords: nonlinear problems, global solution, interval methods, interval linearization

AMS subject classification: 65H10, 65H20, 58C30, 90C30

1. Introduction

Let $f(x)$ denote a system of n nonlinear functions in m variables, $m \geqslant n$. Recently [7–11], a linearisation of $f(x)$ in a box (interval vector) $\boldsymbol{x}$ has been suggested in the form

$$\boldsymbol{L}(x) = Ax + \boldsymbol{b}, \quad x \in \boldsymbol{x}, \tag{1}$$

where A is a real $n \times m$ matrix while $\boldsymbol{b}$ is an interval n-dimensional vector. The new form (1) is an alternative to the traditional linear approximation

$$\boldsymbol{L}(\boldsymbol{x}) = \boldsymbol{A}(\boldsymbol{x} - z) + f(z), \tag{2}$$

where $z \in \boldsymbol{x}$, $\boldsymbol{A}$ is an interval $n \times m$ matrix while $f(z)$ is a real vector. The elements $\boldsymbol{a}_{ij}$ of $\boldsymbol{A}$ can be determined either as interval derivations or, what is better, as interval slopes [6,13,16]. The alternative form (1) has been suggested as an effort to provide a tighter enclosure of $f(x)$ in $\boldsymbol{x}$ as compared to (2). Numerical evidence seems to indicate that the use of (1) rather than (2) in implementing interval methods for solving

systems of nonlinear equations [7–9,11,14] or global optimization [10,14] leads to algorithms of improved numerical efficiency.

In this paper, a better version of (1) is suggested in the form

$$\boldsymbol{L}(x, y) = Ax + By + b, \quad x \in \boldsymbol{x},\ y \in \boldsymbol{y}, \tag{3}$$

where b is a real vector, y is a p-dimensional vector belonging to the interval vector $\boldsymbol{y}$ and A and B are real matrices of appropriate dimensions. As will be shown in the following, the form (3) can be obtained using a combination of Hansen's generalized interval arithmetic (in the simplified form presented in [11]) and the affine arithmetic [3].

It is interesting to note that for two special cases the interval linearisation (1) has been reinvented in two recent papers [1,2] dealing with specific applications ([2] and [1] are written independently from one another and from [8,9,11]). The paper [2] treats the problem of intersection of parametrically defined surfaces and introduces interval planes (linear interval estimations) for the corresponding parametric surface patches while the paper [1] is concerned with the problem of parametric identification. Two approaches are suggested in [2] for the construction of the linear interval estimations used. The second is based on evaluation of the parametrically defined functions using affine arithmetics. Thus, for the special case considered (ignoring notational differences) form (3) is initially obtained. Surprisingly enough, it is however ultimately reduced to form (2).

The present paper is organized as follows. The new form (3) is introduced in section 2 by way of a simple example. In section 3.1, it is shown how (3) can be generated automatically using a simplification of Hansen's generalized interval arithmetic (GIA). Useful properties of this form of GIA are established in section 3.2. Numerical examples illustrating the computational advantages of the new linear form (3) are provided in section 4. The paper ends up in section 5 with concluding remarks.

2. The new linearisation

To elucidate the rationale behind the introduction of the interval linearisation (3), the following simple example will be considered.

Example 1. The problem is to find all solutions of the following system of equations [5]:

$$x_1^2 + x_2^2 - 1 = 0, \tag{4a}$$

$$2x_1^2 - x_2 - 1 = 0, \tag{4b}$$

where

$$x_1 \in [-1, 2], \qquad x_2 \in [-2, 2.2]. \tag{4c}$$

If system (4) is to be solved by a first-order method, we need a suitable linearisation of the system for any current box $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2) \subseteq \boldsymbol{x}_0 = (\boldsymbol{x}_{10}, \boldsymbol{x}_{20})$ where the intervals $\boldsymbol{x}_{10}$ and $\boldsymbol{x}_{20}$ are given in (4c).

First, we shall consider the linear interval form (1). Since system (4) is in separable form the linearisation (1) will be obtained using the method suggested in [8, section 2.1]. System (4) will be written as

$$f_1(x_1, x_2) = f_{11}(x_1) + f_{12}(x_2) - 1 = 0, \tag{5a}$$
$$f_2(x_1, x_2) = f_{21}(x_1) + f_{22}(x_2) - 1 = 0. \tag{5b}$$

The nonlinear functions f_{11}, f_{12} and f_{21} are linearised as

$$\boldsymbol{L}_{ij}(x_j) = a_{ij}x_j + \boldsymbol{b}_{ij}, \quad x_j \in \boldsymbol{x} \tag{6}$$

(the real coefficients a_{ij} and the intervals $\boldsymbol{b}_{ij}$ are determined using procedure 1 of [8]). Hence

$$\boldsymbol{L}_1 = a_{11}x_1 + a_{12}x_2 + \boldsymbol{b}_1, \tag{7a}$$
$$\boldsymbol{L}_2 = a_{21}x_1 + a_{22}x_2 + \boldsymbol{b}_2, \tag{7b}$$

where

$$\boldsymbol{b}_1 = \boldsymbol{b}_{11} + \boldsymbol{b}_{12} - 1, \tag{8a}$$
$$\boldsymbol{b}_2 = \boldsymbol{b}_{21} - 1. \tag{8b}$$

Written in vector form, (7) is the interval linearisation (1) for system (4) in a current box $\boldsymbol{x}$. Using the iterative method of [8], the new iterate $\boldsymbol{x}'$ is obtained as

$$\boldsymbol{x}' = \boldsymbol{u} \cap \boldsymbol{x} \tag{9}$$

with

$$\boldsymbol{u} = -A^{-1}\boldsymbol{b}, \tag{10}$$

where the components of A and $\boldsymbol{b}$ are given by (7) and (8).

Now we shall introduce form (3) for the example considered. Unlike the previous approach, we take into account that some of the intervals $\boldsymbol{b}_{ij}$ are not interdependent. Indeed, we note that

$$f_{21}(x_1) = 2f_{11}(x_1) \tag{11}$$

and hence

$$\boldsymbol{b}_{21} = 2\boldsymbol{b}_{11}. \tag{12}$$

Thus, (8) can be written as

$$\boldsymbol{b}_1 = \boldsymbol{b}_{11} + \boldsymbol{b}_{12} - 1, \tag{13a}$$
$$\boldsymbol{b}_2 = 2\boldsymbol{b}_{11} - 1, \tag{13b}$$

or equivalently in vector form as

$$\boldsymbol{b} = By - e, \quad y \in \boldsymbol{y}, \tag{14a}$$

where

$$B = \begin{bmatrix} 1 & 1 \\ 2 & 0 \end{bmatrix}, \tag{14b}$$

$$\boldsymbol{y} = (\boldsymbol{b}_{11}, \boldsymbol{b}_{12})^{\mathrm{T}}, \qquad y = (y_1, y_2)^{\mathrm{T}}, \quad y \in \boldsymbol{y}, \tag{14c}$$

$$e = (1, 1)^{\mathrm{T}}. \tag{14d}$$

Using the above notations, the linear approximation of (4) can be written as

$$\boldsymbol{L}(x, y) = Ax + By - e, \quad x \in \boldsymbol{x}, \ y \in \boldsymbol{y} \tag{15}$$

which is of the form (3).

Now we shall show that the use of (15) rather than (7) in the computational scheme of the method from [8] leads to an improvement in the numerical efficiency of the method. With this in mind, we first note that on account of (14a) now formula (10) is to be written as

$$\boldsymbol{u}' = A^{-1}By + A^{-1}e, \quad y \in \boldsymbol{y}. \tag{16}$$

Let w_1 and w_2 denote the width of $\boldsymbol{u}$ in (10) and $\boldsymbol{u}'$ in (16), respectively. We shall show that

$$w_2 \leqslant w_1. \tag{17}$$

Indeed, let $C = -A^{-1}$. Then, as is easily seen from (10), (7), (8) and (12)

$$w_1 = |C|w(b) = |C||B|w(\boldsymbol{y}) \tag{18}$$

while from (16)

$$w_2 = |CB|w(\boldsymbol{y}). \tag{19}$$

Thus, comparison of (18) and (19) indicates (17) since

$$|CB| \leqslant |C|\,|B|. \tag{20}$$

Let A1 and A2 denote an algorithm implementing the method of [8] using linearisation (1) and (3), respectively. The theoretical conclusion that algorithm A2 is superior to algorithm A1 has been confirmed by the numerical results concerning the global solution of (4). Table 1 lists data on the number of iterations N_i needed to achieve a desired accuracy ε (the width of the box containing a solution) as well as the number of solutions n_s.

It is seen that the new algorithm A2 needs less iterations than algorithm A1 to solve (4) globally. Moreover, for this example the use of the new form (3) leads to no clustering. Indeed, it is known [5] that system (4) has 3 distinct solutions and this is the number of solutions arrived at by algorithm A2. At the same time, algorithm A1 shows some clustering effect since this algorithm leads to $n_s > 3$.

Table 1
Results for system (4) using algorithms A1 and A2.

	A1		A2	
ε	N_i	n_s	N_i	n_s
10^{-4}	44	5	23	3
10^{-5}	48	4	28	3
10^{-6}	52	4	30	3

3. Automatic generation of the linearisation

In this section, an approach will be presented for generating the interval linearisation (3) for the general case of bounded factorable functions. It is, essentially, a slight modification of the affine arithmetic (AA) approach from [3]. The latter modification is introduced using a simplified version of Hansen's generalized interval arithmetic [4] suggested in [11].

3.1. Use of a modified affine arithmetic (MAA)

In this subsection, $f(x)$ is a bounded multivariate factorable function $f : D \in \mathbb{R}^n \to \mathbb{R}$. To maintain completeness we start with two known facts from [3] (definition 1 and linear combination of generalized intervals).

Definition 1. A generalized (G) interval $\widetilde{X}$ of length m is defined as follows

$$\widetilde{X} = x_0 + \sum_{i=1}^{m} x_i \xi_i, \tag{21a}$$

where x_i, $i = 0, 1, \ldots, m$, are real numbers while ξ_i are unit symmetrical intervals, i.e.,

$$\zeta_i = [-1, 1]. \tag{21b}$$

Let

$$\widetilde{Y} = y_0 + \sum_{i=1}^{m'} y_i \xi_i \tag{22}$$

be a G-interval of length m'. To simplify presentation, we assume that $m' = m$, where m is the length of $\widetilde{X}$ (otherwise, we add zero components either in $\widetilde{X}$ or $\widetilde{Y}$ depending on whether m is smaller or larger than m').

Linear combination. Let $\widetilde{X}$ and $\widetilde{Y}$ be two G-intervals of length m given by (21) and (22). Also, let $\alpha, \beta \in \mathbb{R}$. Then the linear combination of $\widetilde{X}$ and $\widetilde{Y}$, denoted $\alpha\widetilde{X} + \beta\widetilde{Y}$, is another G-interval $\widetilde{Z}$ of the same length m if his elements z_i are computed as follows:

$$z_i = \alpha x_i + \beta y_i, \quad i = 0, 1, \ldots, m. \tag{23}$$

As a corollary we have the definitions of addition of two G-intervals ($\alpha = \beta = 1$) and subtraction of two G-intervals ($\alpha = 1, \beta = -1$).

Now we shall define the operations of multiplication and division of G-intervals (in a way different from [3]). Unlike the linear combination, the operations of multiplication and division of G-intervals result in a G-interval of increased length.

Multiplication. The product $\widetilde{X}\widetilde{Y}$ of two G-intervals $\widetilde{X}$ and $\widetilde{Y}$ of length m is a G-interval $\widetilde{Z}$ of length $m+1$ if the components z_i of $\widetilde{Z}$ are computed as follows:

$$u = \sum_{i=1}^{m} |x_i|, \qquad v = \sum_{i=1}^{m} |y_i|, \qquad c = 0.5 \sum_{i=1}^{m} x_i y_i, \tag{24a}$$

$$z_0 = x_0 y_0 + c, \qquad z_i = x_0 y_i + y_0 x_i, \quad i = 1, \ldots, m, \tag{24b}$$

$$z_{m+1} = uv - 0.5 \sum_{i=1}^{m} |x_i y_i|. \tag{24c}$$

The above rules are a straightforward simplification of the more complex multiplication rules in [11]. The simplification is due to the use of the normalized form (21) of the G-intervals.

It is to be noted that the multiplication (24) leads to smaller overestimation as compared with the multiplication used in [3] because of the "correction" introduced by the additional term c.

To define the operation of division, we have to consider the operation reciprocal $1/\widetilde{Y}$ of a G-interval. But before we need some definitions. Following [3,4], we shall say that the G-interval $\widetilde{X}$ is reduced to the corresponding (ordinary) interval $\boldsymbol{x} = [\underline{x}, \overline{x}]$ if the summation operations in (21a) are carried out. By abuse of language, we shall also say that $\widetilde{X}$ does not contain zero (is positive or negative) if the corresponding reduced interval $\boldsymbol{x}$ does not contain zero (is positive or negative).

Reciprocal. Let $\widetilde{Y}$ be a G-interval of length m that does not contain zero. Then the reciprocal $\widetilde{Z} = 1/\widetilde{Y}$ is another G-interval of length $m+1$ if its components z_i are computed as follows:

$$s = -\frac{1}{\underline{y}\,\overline{y}}, \quad y_1 = -\sqrt{-\frac{1}{s}}, \; y_2 = -y_1, \tag{25a}$$

$$y_s = \begin{cases} y_2, & \text{if } \underline{y} > 0, \\ y_1, & \text{if } \overline{y} < 0, \end{cases} \tag{25b}$$

$$\underline{f} = \frac{1}{y_s} - s y_s, \qquad \overline{f} = \frac{1}{\underline{y}} - s\underline{y}, \tag{25c}$$

$$f_0 = 0.5\big(\underline{f} + \overline{f}\big), \qquad r_f = \overline{f} - f_0, \tag{25d}$$

$$z_0 = s y_0 + f_0, \qquad z_i = s y_i, \quad i = 1, \ldots, m, \tag{25e}$$

$$z_{m+1} = r_f, \tag{25f}$$

where $\underline{y}$ and $\overline{y}$ are the endpoints of the reduced interval $\boldsymbol{y}$.

The above formulae follow directly from the general approach for enclosing univariate functions [7–9,11] by a linear interval form.

The division rule given below is an adaptation of the division suggested in [4] for the special case of generalized intervals in the form (21). It is based on the expression ($0 \notin \widetilde{Y}$)

$$\frac{\widetilde{X}}{\widetilde{Y}} = \frac{x_0}{y_0} + \frac{\sum_{i=1}^{m} (y_0 x_i - x_0 y_i)\xi_i}{y_0(y_0 + \sum_{i=1}^{m} y_j \xi_j)} = c + \frac{1}{\widetilde{Y}}\left[\sum_{i=1}^{m} (x_i - c y_i)\xi_i\right]. \tag{26}$$

Division. Let $\widetilde{X}$ and $\widetilde{Y}$ be G-intervals of length m and $0 \notin \widetilde{Y}$. Then the division $\widetilde{X}/\widetilde{Y}$ is a G-interval $\widetilde{Z}$ of length $m+2$ whose components z_i are computed as follows:

$$\widetilde{Q} = \frac{1}{\widetilde{Y}}, \tag{27a}$$

$$c = \frac{x_0}{y_0}, \qquad p_0 = 0, \qquad p_i = x_i - c y_i, \quad i = 1, \ldots, m, \tag{27b}$$

$$\widetilde{P} = \sum_{i=1}^{m} p_i \xi_i, \tag{27c}$$

$$\widetilde{V} = \widetilde{Q} \cdot \widetilde{P}, \tag{27d}$$

$$z_0 = c + v_0, \qquad z_i = v_i, \qquad i = 1, \ldots, m+2. \tag{27e}$$

It is seen that the division increases the length of the resulting interval $\widetilde{Z}$ by two because of the reciprocal (27a) and multiplication (27d), each operation adding one more element to the initial m elements of $\widetilde{X}$ or $\widetilde{Y}$.

The above operations are sufficient to generate form (3) for multivariate rational functions. For the general case of arbitrary factorable functions we make use of the algorithm presented in [11]. We only need to keep in mind that whenever a G-interval $\widetilde{X}$ of length m is transformed by a univariate function $y = g(x), \quad g : D \subset R \to R$, into a G-interval $\widetilde{Y}$, the resulting interval $\widetilde{Y}$ has a length $m+1$ (as was shown in the case of the univariate reciprocal function).

3.2. Properties of the MAA

In this subsection, we consider some properties of the MAA presented in the previous section. These properties may be useful in practice as will be shown in the next section.

As in [3] we have the following property.

Property 1.

$$\widetilde{X} - \widetilde{X} = 0. \tag{28}$$

Unlike [3] the following property only holds for the new division rule.

Property 2.

$$\frac{\widetilde{X}}{\widetilde{\widetilde{X}}} = 1. \tag{29}$$

Let A, B and C be G-intervals of length m (once again m is the same for all intervals only for simplicity) where for simplicity of notation the tilda sign has been dropped.

Property 3 (Subdistributivity). In the general case

$$A(B + C) \subseteq AB + AC. \tag{30}$$

However, we have three important special cases when

$$A(B + C) = AB + AC. \tag{31}$$

Property 3.1. If

$$b_i c_i = 0, \quad i = 1, \ldots, m, \tag{32}$$

where b_i and c_i are the components of B and C, respectively, then (31) holds.

In other words, (31) is valid if B and C do not share common ξ_i. It should be stressed that although property 3.1 is a special case, it covers many practically important situations. Indeed, according to this property (31) hold if B and C are independent G-intervals. Equality (31) always holds for arbitrary ordinary intervals (written, however, in form (21)). Moreover, it remains valid (both for G and ordinary intervals) when $B = A$, i.e., if (32) is valid, then

$$B(B + C) = BB + BC. \tag{33}$$

Property 3.2. If

$$b_i c_i = 1, \quad i = 1, \ldots, m, \tag{34}$$

then (31) and (33) hold.

This property expresses the fact that (31) and (33) are valid if b_i and c_i have the same sign.

Property 3.3. The equalities (31) and (33) are valid if the set of indices from 1 to m is subdivided into subsets where (32) or (34) hold.

The above properties can be readily proved checking whether

$$|b_i + c_i| < |b_i| + |c_i| \quad \text{or} \tag{35a}$$

$$|b_i + c_i| = |b_i| + |c_i|. \tag{35b}$$

4. Numerical examples

The computational efficiency of the new form (3) will be tested numerically only in the context of solving nonlinear systems of equations. Two "difficult" examples will be considered in this section. The numerical data have been obtained on a 1.7 GHz Pentium PC.

Example 2. The system to be solved is

$$\begin{aligned}
&\big((4x_3+3x_6)x_3+2x_5\big)x_3+x_4=0,\\
&\big((4x_2+3x_6)x_2+2x_5\big)x_2+x_4=0,\\
&\big((4x_1+3x_6)x_1+2x_5\big)x_1+x_4=0,\\
&x_4+x_5+x_6+1=0,\\
&\big(\big((x_2+x_6)x_2+x_5\big)x_2+x_4\big)x_2+\big(\big((x_3+x_6)x_3+x_5\big)x_3+x_4\big)x_3=0,\\
&\big(\big((x_1+x_6)x_1+x_5\big)x_1+x_4\big)x_1+\big(\big((x_2+x_6)x_2+x_5\big)x_2+x_4\big)x_2=0
\end{aligned} \tag{36a}$$

in the initial box $\boldsymbol{x}^0$ with components

$$\begin{aligned}
&\boldsymbol{x}_1=[0.0333\quad 0.2173], &&\boldsymbol{x}_2=[0.4000\quad 0.6000], &&\boldsymbol{x}_3=[0.7826\quad 0.9666],\\
&\boldsymbol{x}_4=[-0.3071\ -0.1071], &&\boldsymbol{x}_5=[1.1071\quad 1.3071], &&\boldsymbol{x}_6=[-2.1000\ -1.9000]
\end{aligned} \tag{36b}$$

(this is an example suggested by W. Walster). Table 2 lists the numerical results obtained (t stands for time in seconds; the meaning of the remaining symbols used is the same as in table 1).

Table 2
Results for system (36) using algorithms A1 and A2.

	A1			A2		
ε	N_i	$t(s)$	n_s	N_i	$t(s)$	n_s
10^{-5}	2166	8.02	5	917	3.46	5

Remark 1. The number of solutions n_s reported in table 2 is before "sieving" the solutions. After incorporation of a subprogram intended to identify as one single solution boxes that are adjacent and pass the criterion for containing a solution, then both algorithms yielded $n_s = 1$.

Example 3. In this example the system is [15]

$$\alpha_k(x)=\mathrm{e}^{x_5(g_{1k}-g'_{3k}x_7-g'_{5k}x_8)}-1+\frac{g_{4k}x_2-g_{5k}}{d_1},\quad k=1,\ldots,4,$$

$$\begin{aligned} d_1 &= (1 - x_1x_2)x_3, \\ \beta_k(x) &= \mathrm{e}^{x_6(g'_{1k} - g'_{3k}x_7 + g'_{4k}x_9)} - 1 + \frac{-g_{5k}x_1 + g_{4k}}{d_2}, \quad k = 1, \ldots, 4, \\ d_2 &= (1 - x_1x_2)x_4, \\ \gamma(x) &= x_1x_3 - x_2x_4, \end{aligned} \tag{37}$$

where $g'_{1k} = g_{1k} - g_{2k}$, $g'_{3k} = g_{3k}10^{-3}$, etc. (the numerical constants g_{ij} are given in [15]).

This example is instructive in the following sense. Consider the expressions

$$p_{\alpha k} = x_5(g_{1k} - g'_{3k}x_7 - g'_{5k}x_8) \tag{38a}$$

and

$$p_{\beta k} = x_6(g_{1k} - g_{2k} - g'_{3k}x_7 + g'_{4k}x_9) \tag{38b}$$

in $\alpha_k(x)$ and $\beta_k(x)$, respectively. The "natural" way to find the G-interval corresponding, e.g., to (38a) is to let

$$\widetilde{Y}_{\alpha k} = g_{1k} - g'_{3k}\widetilde{X}_7 - g'_{5k}\widetilde{X}_8 \tag{39}$$

and then to multiply $\widetilde{X}_5$ and $\widetilde{Y}_{\alpha k}$. This would be the best approach if we were to determine each G-interval

$$\widetilde{P}_{\alpha k} = a_{k0} + a_{k5}\widetilde{X}_5 + a_{k7}\widetilde{X}_7 - a_{k8}\widetilde{X}_8 + a_{k,9+k}\widetilde{X}_{9+k} \tag{40}$$

individually, independently one from another. Since expressions (38) are, however, part of a system of equations, this would lead to the generation of 8 additional variables $\widetilde{X}_{9+k}$, $k = 1, \ldots, 8$.

There is a better approach to determining the G-intervals corresponding to all terms $p_{\alpha k}$, $p_{\beta k}$, $k = 1, \ldots, 4$. It is based on property 3.1. Indeed, $P_{\alpha k}$ and $P_{\beta k}$ can be written equivalently as

$$\widetilde{P}_{\alpha k} = g_{1k}\widetilde{X}_5 - g'_{3k}\big(\widetilde{X}_5\widetilde{X}_7\big) - g'_{5k}\big(\widetilde{X}_5\widetilde{X}_8\big), \tag{41a}$$

$$\widetilde{P}_{\beta k} = g'_{1k}\widetilde{X}_6 - g'_{3k}\big(\widetilde{X}_6\widetilde{X}_7\big) + g'_{4k}\big(\widetilde{X}_6\widetilde{X}_9\big) \tag{41b}$$

which engenders only 4 additional variables $\widetilde{X}_{10}$ to $\widetilde{X}_{13}$ owing to the 4 products in (41).

As in [8] the initial box was chosen to be a hypercube centered at the unique solution x^s and having a side of width 0.1.

The numerical data for this example, when the better presentation (41) has been used, are given in table 3.

The results reported in tables 2 and 3 indicate that the use of the new interval linearisation (3) in first-order interval methods for global solution of nonlinear systems can reduce (in some cases substantially) the execution time of the method used.

Table 3
Results for system (37) using algorithms A1 and A2.

	A1			A2		
ε	N_i	$t(s)$	n_s	N_i	$t(s)$	n_s
10^{-5}	137	1.1	1	6	0.05	1

5. Conclusion

An improved interval linearisation (3) has been suggested for enclosing a system of n nonlinear functions in m variables with $m \geqslant n$ in a given box. It has been shown theoretically that the new linearisation provides a tighter enclosure as compared to the known linear interval form (1). Form (3) can be obtained automatically using the modified affine arithmetic presented in section 3. Several properties of this arithmetic permit in certain cases a better enclosure of the set of nonlinear functions considered.

The numerical examples solved so far are related to the problem of global solution of nonlinear systems of equations. The results are quite encouraging and it is hoped that the improved linearisation (3) can be successfully used in designing new first-order methods for solving a large class of nonlinear problems such as bounding the solution set of underdetermined nonlinear systems or of systems of equalities and inequalities, global optimization. These expectations are based on the experience gained with the application of the form (1) for solving various nonlinear problems [14].

The specific form of the interval linearisation (3) permits the development of algorithms for solving specific nonlinear problems based on the use of linear programming (LP) techniques. Such an approach has already been experimented with form (1) in [12,14]. Currently, an LP algorithm implementing the incorporation of form (3) in a method for global solution of nonlinear systems is been developed and the preliminary results are quite promising. As in the case of the previous form (1), the most efficient use of the new form (3) seems to be a balanced combination with other techniques including constraint propagation [9,14]. Another possibility for improvement is the incorporation of sparse matrix techniques in large-scale sparse problems. Much work, however, is still to be done.

References

[1] I. Braems, M. Kieffer and E. Walter, Set estimation, computation of volumes and data safety, in: *Scientific Computing, Validated Numerics*, eds. W. Kramer and J. Wolff von Gudenberg (Kluwer Academic/Plenum, New York, 2001) pp. 267–278.

[2] K. Bühler and W. Barth, A new intersection algorithm for parametric surfaces based on linear interval estimations, in: *Scientific Computing, Validated Numerics*, eds. W. Kramer and J. Wolff von Gudenberg (Kluwer Academic/Plenum, New York, 2001) pp. 179–190.

[3] J.L.D. Comba and J. Stolfi, Affine arithmetic and its applications to computer graphics, in: *Proc. of VI SIBGRAPHI (Brasilian Symposium on Computer Graphics and Image Processing)*, 1990, pp. 9–18.

[4] E.R. Hansen, *A Generalised Interval Arithmetic*, Lecture Notes in Computer Science (Springer, New York, 1989) pp. 75–82.

[5] E. Hansen, *Global Optimization Using Interval Analysis* (Marcel Dekker, New York, 1992).
[6] L. Kolev, Use of interval slopes for the irrational part of factorable functions, Reliable Comput. 3 (1997) 83–93.
[7] L. Kolev, An efficient interval method for global analysis of nonlinear resistive circuits, Internat. J. Circuit Theory Appl. 26 (1998) 81–92.
[8] L. Kolev, A new method for global solution of systems of nonlinear equations, Reliable Comput. 4 (1998) 1–21.
[9] L. Kolev, An improved method for global solution of nonlinear systems, Reliable Comput. 5(2) (1999) 103–111.
[10] L. Kolev, A new interval approach to global optimization, Comput. Math. Math. Phys. 39(5) (1999) 727–737.
[11] L. Kolev, Automatic computation of a linear interval enclosure, Reliable Comput. 7 (2001) 17–28.
[12] L. Kolev and V. Mladenov, A linear programming implementation of an interval method for global nonlinear DC analysis, in: *IEEE Internat. Conf. Electronics, Circuits and Systems. Surfing the Waves of Science and Technology*, Vol. 1, Instituto Superior Tecnico Lisboa, Portugal, 7–10 September 1998, pp. 75–78.
[13] R. Krawczuk and A. Neumaier, Interval slopes for rational functions and associated centred forms, SIAM J. Numer. Anal. 22 (1985) 604–616.
[14] I. Nenov and D. Fylstra, Interval methods for accelerated global search in the microsoft exel solver, in: *Extended Abstracts of Validated Computing 2002*, ed. R.B. Kearfott, Toronto, Canada, 23–25 May 2002, pp. 145–147.
[15] H. Ratschek and J. Rokne, Experiment using interval analysis for solving a circuit design problem, J. Global Optim. 3 (1993) 546–551.
[16] S. Zuhe and M. A. Wolfe, On interval enclosures using slope arithmetic, Appl. Math. Comput. 39 (1990) 89–105.