Journal of Global Optimization **29:** 157–172, 2004.

Improving Interval Analysis Bounds by Translations

EMILIO CARRIZOSA[1], PIERRE HANSEN[2] and FRÉDÉRIC MESSINE[3]

[1]*Universitad de Sevilla, Facultad de Matematicas, Sevilla, Spain (e-mail: ecarrizosa@us.es)*

[2]*GERAD, École des Hautes Études Commerciales for HEC Montréal, Département des Méthodes Quantitatives, Montréal (Québec), Canada (e-mail: pierre.hansen@gerad.ca)*

[3]*Université de Pau et des Pays de l'Adour, Département d'Informatique, Pau, France et le Laboratoire d'Électrotechnique et d'Électronique Industrielle UMR 5828, Équipe EM², Toulouse, France (e-mail: frederic.messine@univ-pau.fr and frederic.messine@leei.enseeiht.fr)*

(Received 22 March 2002; accepted 26 August 2003)

Abstract. We explore how a simple linear change of variable affects the inclusion functions obtained with Interval Analysis methods. Univariate and multivariate polynomial test functions are considered, showing that translation-based methods improve considerably the bounds computed by standard inclusion functions. An Interval Branch-and-Bound method for global optimization is then implemented to compare the different procedures, showing that, although with times higher than those given by Taylor forms, the number of clusters and iterations is strongly reduced.

Mathematics Subject Classifications. 90C26, 68N30

Key words. inclusion functions, interval analysis, interval branch and bound, Taylor forms.

1. Introduction

Designed as a technique for controlling propagation of errors in computing [8], Interval Analysis was soon recognized as a powerful tool for global optimization [3, 5, 6, 10]. Its main use is then in feasibility and optimality tests of Branch-and-Bound methods [8, 10] for solving problems of the form

$$\min\{f(x)\,|\,g_i(x)\geqslant 0 \;\; i=1,2,\ldots,m\}.$$

Indeed, a region X can be discarded as soon as one detects it is either infeasible (because an upper bound for one of the functions g on X is negative), or it cannot contain optimal solutions (because a lower bound for $f(x)$ on X turns out to be worse than the value of an already known feasible solution). Hence, it is of great importance to know, for a given function f, the direct image $f(X)$ of X,

$$f(X):=\{f(x):x\in X\},$$

or, if this is not possible, an enclosure of it. This leads to the concept of *inclusion function*, defined as follows in the univariate case: let $\mathbb{I}$ denote the set of intervals X of the form $X:=[x^L,x^U]$, with $-\infty\leqslant x^L\leqslant x^U\leqslant+\infty$.

Given $f\colon \mathbb{R} \longrightarrow \mathbb{R}$, any $F\colon \mathbb{I} \longrightarrow \mathbb{I}$ containing $f(X)$ is called an inclusion function of F.

A desired property of an inclusion function of f is its isotonicity: F is said to be an *isotone* inclusion function of f if

$$F(Y) \subseteq F(Z) \quad \text{for all } Y, Z \text{ with } Y \subseteq Z.$$

Some examples of (isotone) inclusion functions will be introduced here; the reader is referred to [3, 9, 10] for further details.

If an analytical expression of f is given, one formally replaces the variable x by the corresponding interval variable X, and all the algebraic operations in the definition of f by their corresponding Interval Arithmetic operations, then one obtains the so-called *Natural Extension* of f, denoted throughout the paper by NE.

For sufficiently smooth functions f, it is possible to obtain different inclusion functions from Taylor expansions by constructing enclosures of the remainder. Indeed, f can then be written as

$$f(x) = f(x_0) + \sum_{i=1}^{k-1} \frac{(x-x_0)^i}{i!} f^{(i)}(x_0) + \frac{(x-x_0)^k}{(k)!} f^{(k)}(\xi)$$

for some $\xi \in X$. If $F^{(k)}$ is an inclusion function of $f^{(k)}$, then $f^{(k)}(\xi) \in F^{(k)}(X)$, yielding the *Taylor inclusion function* of order k centered at x_0, $T_k(x_0, X)$, defined as

$$T_k(x_0, X) = f(x_0) + \sum_{i=1}^{k-1} \frac{(X-x_0)^i}{i!} f^{(i)}(x_0) + \frac{(X-x_0)^k}{k!} F^{(k)}(X). \tag{1}$$

The most used interval Taylor inclusion functions are obtained from the *centered expansions* of first or second order,

$$T_1(X) = f(m) + (X-m)F'(X) \tag{2}$$

$$T_2(X) = f(m) + (X-m)f'(m) + \frac{(X-m)^2}{2} F''(X), \tag{3}$$

(m being the midpoint of the interval $X = [x^L, x^U]$), or a non-centered form of $T_1(x_0, X)$, T_1B, due to Baumann [1], consisting of taking, in (1), for $k = 2$, x_0 given by

$$x_0 := \begin{cases} x^U & \text{if } F'^U(X) \leqslant 0, \\ x^L & \text{if } F'^L(X) \geqslant 0, \\ \frac{F'^U(X)x^L - F'^L(X)x^U}{F'^U(X) - F'^L(X)} & \text{otherwise} \end{cases}$$

to maximize the lower bound and

$$x_0 := \begin{cases} x^L & \text{if } F'^U(X) \leqslant 0, \\ x^U & \text{if } F'^L(X) \geqslant 0, \\ \frac{F'^L(X)x^L - F'^U(X)x^U}{F'^L(X) - F'^U(X)} & \text{otherwise} \end{cases}$$

to minimize the upper bound.

These will be the inclusion functions taken as benchmark, with which the inclusion function we propose in Section 2 will be compared.

2. Translation-based Methods for Univariate Polynomial Functions

2.1. PROBLEM SETTING

In this section we address the problem of finding inclusion functions P, yielding sharp enclosures for the range of a real univariate polynomial function p,

$$p(x) = \sum_{k=0}^{n} a_k x^k, \quad \text{with } a_k \in \mathbb{R}, \text{ and } x \in X = [x^L, x^U] \in \mathbb{I}. \tag{4}$$

Throughout this section, n will denote the degree of the polynomial function p considered.

In this case, the Natural Extension NE of this expression of p becomes

$$\text{NE}(X) = \sum_{k=0}^{n} a_k X^k, \quad X = [x^L, x^U] \in \mathbb{I}. \tag{5}$$

Another well-known choice is the Horner scheme H,

$$H(X) = a_0 + X(\cdots(a_{n-2} + X(a_{n-1} + a_n X))\cdots). \tag{6}$$

Observe that these two inclusion functions are not comparable in terms of the enclosures they provide. For instance, for $p(x) = x^2 - x$ and $X = [-1, 1]$, we have

$$\begin{aligned} \text{NE}(X) &= X^2 - X \\ &= [-1, 2] \\ &\subset [-2, 2] = X(X-1) = H(X). \end{aligned}$$

Nevertheless, for the same p and $X = [1, 2]$, we obtain

$$\begin{aligned} \text{NE}(X) &= X^2 - X \\ &= [-1, 3] \\ &\supset [0, 2] = X(X-1) = H(X). \end{aligned}$$

However, an inclusion function sharper than both NE and H is directly obtained from H, by computing the range in a box as the union of ranges in sub-boxes covering the box, [9, 10]:

DEFINITION 1. Given an isotone function $F: \mathbb{I} \longrightarrow \mathbb{I}$, and $c \in \mathbb{R}$, define $F_c: \mathbb{I} \longrightarrow \mathbb{I}$, as

$$F_c(X) = \begin{cases} F([x^L, c]) \cup F([c, x^U]), & \text{if } \; c \in X \\ F(X), & \text{otherwise.} \end{cases}$$

LEMMA 1. *F_c is an isotone function satisfying $F_c(X) \subseteq F(X)$ for all $X \in \mathbb{I}$.*

Proof. First observe that F_c is well defined; indeed, for $c \in X = [x^L, x^U]$, since F is an inclusion function, both $F([x^L, c])$ and $F([c, x^U])$ are closed intervals having $f(c)$ as common point; hence, $F_c(X) \in \mathbb{I}$.

Moreover, F_c is an inclusion function. Indeed, if $c \notin X$, one has $F_c(X) = F(X) \supseteq f(X)$; if $c \in X$, one has

$$f([x^L, c]) \subseteq F([x^L, c])$$
$$f([c, x^U]) \subseteq F([c, x^U]).$$

Hence,

$$f(X) = f([x^L, c]) \cup f([c, x^U]) \subseteq F([x^L, c]) \cup F([c, x^U]) = F_c(X),$$

showing that F_c is also an inclusion function.

In order to see that F_c is isotone, consider $Y, Z \in \mathbb{I}, \; Y \subseteq Z$. Three cases are considered:

1. If $c \notin Z$, then the isotonicity of F implies that $F_c(Y) = F(Y) \subseteq F(Z) = F_c(Z)$.
2. If $c \notin Y$ and $c \in Z$ then either $Y \subseteq [z^L, c]$ or $Y \subseteq [c, z^U]$. In the first case one has

$$F_c(Y) = F(Y) \subseteq F([z^L, c]) \subseteq (F([z^L, c]) \cup F([c, z^U])) = F_c(Z),$$

whereas in the latter case one has

$$F_c(Y) = F(Y) \subseteq F([c, z^U]) \subseteq (F([z^L, c]) \cup F([c, z^U])) = F_c(Z).$$

3. If $c \in Y$ (thus $c \in Z$) then $[y^L, c] \subseteq [z^L, c]$, and $[c, y^U] \subseteq [c, z^U]$.
   Hence, $F_c(Y) = (F([y^L, c]) \cup F([c, y^U])) \subseteq (F([z^L, c]) \cup F([c, z^U])) = F_c(Z)$.

Therefore, F_c is isotone.

Trivially $F_c(X) \subseteq F(X)$ if $c \notin X$. If $c \in X$, then the isotonicity of F implies that $F([x^L, c]) \cup F([c, x^U]) \subseteq F(X)$, thus $F_c(X) \subseteq F(X)$, as asserted. □

Particularized to the inclusion function H, splitting by $c = 0$, one obtains the inclusion function H_0, defined as:

$$H_0(X) := \begin{cases} [\min\{H^L([x^L, 0]), H^L([0, x^U])\}, \max\{H^U([x^L, 0]), H^U([0, x^U])\}], \\ \text{if} \quad 0 \in X, \\ H(X), \quad \text{else.} \end{cases} \tag{7}$$

PROPOSITION 1. *H_0 is an inclusion function which is isotone and satisfies for all $X \in \mathbb{I}$:*

1. $H_0(X) \subseteq H(X)$
2. $H_0(X) \subseteq \mathrm{NE}(X)$.

Proof. H_0 is an isotone inclusion function satisfying $H_0(X) \subseteq H(X)$ by Lemma 1. By defining NE_0 following Definition 1, it suffices to show that $H_0(X) \subseteq \mathrm{NE}_0(X)$ for all $X \in \mathbb{I}$.

We show that $H_0(X) \subseteq \mathrm{NE}(X)$ by induction in the degree k of p. For $k = 0, 1$ the inclusion is straightforward. We assume that the inclusion holds for all polynomial functions of degree smaller than k, and show the result for the polynomial function $p(x) := a_0 + a_1 x + \cdots + a_{k+1} x^{k+1}$, (of degree $k+1$).

If $0 \notin X$, then

$$H_0(X) = H(X) = a_0 + X(H^*(X)) = a_0 + X(H_0^*(X))$$

where $H^*(X)$ (respectively H_0^*) represents the Horner scheme H (respectively H_0) for the polynomial function p^* of degree k, $p^*(x) = a_1 + a_2 x + \cdots + a_{k+1} x^k$.

By the induction assumption, one has

$$H_0^*(X) \subseteq \mathrm{NE}^*(X),$$

where NE^* denotes the natural extension for p^*. Hence,

$$a_0 + X(H_0^*(X)) \subseteq a_0 + X(\mathrm{NE}^*(X)) \subseteq \mathrm{NE}(X).$$

Therefore

$$H_0(X) \subseteq \mathrm{NE}(X) \quad \forall X \text{ with } 0 \notin X.$$

If $0 \in X$, a similar argument shows that

$$H_0([x^L, 0]) \subseteq \mathrm{NE}([x^L, 0])$$

and

$$H_0([0, x^U]) \subseteq \mathrm{NE}([0, x^U]).$$

Hence, $H_0(X) \subseteq \mathrm{NE}_0(X) \subseteq \mathrm{NE}(X)$, and the result holds. □

2.2. TRANSLATION-BASED METHODS

The idea of these methods is to translate the interval X considered into the interval $X_\mu = [x^L + \mu, x^U + \mu]$ by using an expression for $p(x)$ different from (4), and then choosing the value of μ yielding the sharpest enclosure. First, observe that, for any $\mu \in \mathbb{R}$,

$$\begin{aligned} p(x) &= \sum_{j=0}^{n} a_j((x+\mu)-\mu)^j \\ &= \sum_{j=0}^{n} (x+\mu)^j \sum_{k=0}^{n-j} a_{k+j} \binom{k+j}{j} (-\mu)^k \\ &= \sum_{j=0}^{n} f_j(\mu)(x+\mu)^j, \end{aligned} \tag{8}$$

with $f_j(\mu)$ defined as

$$f_j(\mu) = \sum_{k=0}^{n-j} a_{k+j} \binom{k+j}{j} (-\mu)^k. \tag{9}$$

For each inclusion function F previously defined one obtains now, for each $\mu \in \mathbb{R}$, a new translation-based inclusion function $TF(\mu, \cdot)$. For instance, from the Horner scheme H, one obtains TH,

$$TH(\mu, X) = f_0(\mu) + X_\mu(f_1(\mu) + X_\mu(\cdots(f_{n-1}(\mu) + X_\mu f_n(\mu)))), \tag{10}$$

with $X_\mu = X + \mu$.

Furthermore, TH_0 is defined, following Definition 1, as

$$TH_0(\mu, X) = (TH(\mu, X))_0. \tag{11}$$

In the same way, TNE, TNE_0, TT_1, TT_n are defined.

Given an inclusion function F, we obtain for each μ the inclusion function $TF(\mu, \cdot)$. By definition,

$$TF(0, X) = F(X) \tag{12}$$

thus by varying the parameter μ it may be possible to come up with more accurate enclosures. This poses the problem of determining the values of μ yielding the sharpest enclosure.

For this we define, for an inclusion function F, the optimal translation-based inclusion function OTF as

$$OTF(X) = \left[\max_{\mu \in \mathbb{R}} TF^L(\mu, X),\ \min_{\mu \in \mathbb{R}} TF^U(\mu, X)\right], \tag{13}$$

where $TF(\mu, X) = [TF^L(\mu, X), TF^U(\mu, X)]$.

Observe that, by (12),

$$OTF(X) \subseteq F(X) \quad \forall X \in \mathbb{I}.$$

Remark 1. The function OTF is only of interest for theoretical reasons; indeed, the practical determination of OTF amounts to solving two optimization problems which can be non-differentiable and non-convex. Hence, in practice, a few steps of a local-search algorithm will be used, yielding an enclosure possibly less sharp than OTF but with much less computational effort.

PROPOSITION 2.

$$TH_0(\mu, X) \subseteq TNE(\mu, X) = T_n(-\mu, X) \quad \textit{for all } \mu \in \mathbb{R}, X \in \mathbb{I}. \tag{14}$$

Proof. Let $\mu \in \mathbb{R}$ and $X \in \mathbb{I}$. The inclusion $TH_0(\mu, X) \subseteq TNE(\mu, X)$ directly follows from Proposition 1.

Since $p^{(n)}(x) = n!a_n$ for all x, one has that $P^{(n)}(X): [n!a_n, n!a_n]$ is an inclusion function for $p^{(n)}$. By (1), $T_n(-\mu, X)$ can then be written as

$$\begin{aligned}
T_n(-\mu, X) &= p(-\mu) + \sum_{i=1}^{n-1} \frac{(X+\mu)^i}{i!} p^{(i)}(-\mu) + \frac{(X+\mu)^n}{n!} P^{(n)}(X) \\
&= p(-\mu) + \sum_{i=1}^{n-1} \frac{(X+\mu)^i}{i!} \sum_{k=0}^{n-i} \frac{(k+i)!}{k!} a_{k+i}(-\mu)^k + \frac{(X+\mu)^n}{n!} n!a_n \\
&= \sum_{i=0}^{n} (X+\mu)^i \sum_{k=0}^{n-i} \left(\binom{k+i}{i} a_{k+i}(-\mu)^k \right) \\
&= \sum_{i=0}^{n} (X+\mu)^i f_i(\mu) = TNE(\mu, X).
\end{aligned}$$

This shows the result. □

From (14) one directly has

PROPOSITION 3. $OTH_0(X) \subseteq OTNE(X)$ *for all* $X \in \mathbb{I}$.

2.3. NUMERICAL RESULTS

The different inclusion functions previously suggested have been compared according to the bounds they produce. Table 1 summarizes the results obtained for a series of univariate polynomial functions, either taken from the literature, [4, 11, 12], or randomly generated. The first ones are the following:

1. $p_1(x) = \frac{1}{10} - x - \frac{79}{20}x^2 + \frac{71}{10}x^3 + \frac{39}{80}x^4 - \frac{52}{25}x^5 + \frac{1}{6}x^6$, $X = [-2, 11]$, due to Wingo, [11]. There is a misprint in the expression of the function in [11].

Table 1. Results for lower or upper bounds for polynomial functions.

Pb	p_1		p_2	p_3	p_4	p_5	p_6	p_7
	lb	ub	lb	lb	lb	lb	ub	lb
NE	$[-3.34E^5$	$3.12E^5]$	$-3.33E^{10}$	$-1.70E^4$	$-5.00E^2$	$-4.99E^3$	$6.16E^5$	$-4.62E^2$
H	$[-3.73E^5$	$8.68E^4]$	$-2.48E^{10}$	$-1.40E^4$	$-1.23E^3$	$-4.57E^3$	$3.77E^4$	$-1.09E^3$
H_0	$[-3.19E^5$	$1.61E^4]$	$-2.48E^{10}$	$-1.40E^4$	$-4.00E^2$	$-4.57E^3$	$3.77E^4$	$-3.62E^2$
T_1	$[-1.08E^6$	$1.08E^6]$	$-1.51E^{16}$	$-4.67E^4$	$-4.20E^3$	$-1.37E^5$	$2.61E^8$	$-3.7E^3$
T_1B	$[-1.03E^6$	$1.03E^6]$	$-5.15E^{11}$	$-3.71E^4$	$-3.29E^3$	$-1.42E^4$	$4.09E^6$	$-3.07E^3$
T_2	$[-1.19E^6$	$1.59E^6]$	$-9.79E^{11}$	$-2.15E^4$	$-1.50E^3$	$-1.56E^4$	$3.84E^6$	$-1.21E^3$
T_3	$[-1.27E^6$	$1.23E^6]$	$-3.65E^{17}$	$-1.86E^4$	$-3.00E^3$	$-4.63E^4$	$8.22E^8$	$-2.96E^3$
OTH_0	$[-5.46E^4$	$2.35E^2]$	$-1.01E^3$	$-9.96E^2$	$-6.72E^1$	1.92	$-3.11E^1$	$-1.64E^2$
μ	−6.3	−1	−1.08	−5.35	−0.87	−0.17	−0.65	−0.45

2. $p_2(x)=\sum_{i=1}^{50} a_i x^i, x\in[1,2]$, the coefficients are $a_{1,\dots,n}=\{-500,$ 2.5, 1.666666666, 1.25, 1, 0.833333333, 0.714285714, 0.625, 0.555555555, 1, −43.636363636, 0.416666666, 0.384615384, 0.357142857, 0.333333333, 0.3125, 0.294117647, 0.277777777, 0.263157894, 0.25, 0.238095238, 0.227272727, 0.217391304, 0.208333333, 0.2, 0.192307692, 0.185185185, 0.178571428, 0.344827586, 0.666666666, −15.483870970, 0.15625, 0.151515151, 0.147058823, 0.142857142, 0.138888888, 0.135135135, 0.131578947, 0.128205128, 0.125, 0.121951219, 0.119087619, 0.116279069, 0.113636363, 0.111111111, 0.108695652, 0.106382978, 0.208333333, 0.408163265, 0.8}, the Moore function.
3. $p_3(x) = 0.000089248x - 0.0218343x^2 + 0.998266x^3 - 1.6995x^4 + 0.2x^5$, $x\in[0,10]$, the Wilkinson function.
4. $p_4(x)=4x^2-4x^3+x^4, x\in[-5,5]$, the Dixon and Szegö function.
5. $p_5(x)=7x^4-5x^3+4x^2+3x+2$, $X=[0,10]$, generated randomly with integer coefficients in the range $\{-10,\dots,10\}$.
6. $p_6(x)=-5.87x^{13}-2.32x^{12}-1.83x^{11}-16.64x^{10}+7.71x^9+8.71x^8+5.26x^7-5.29x^6-17.69x^5+3.47x^4-12.4x^3-19.35x^2-19.37x+4.34, X=[0.77,3.38]$, generated randomly with real coefficients in the interval $[-20,20]$.
7. $p_7(x)=10x-1.5x^2-3x^3+x^4, x\in[-5,5]$, the Dixon function.

The inclusion functions considered are NE, H, H_0, T_1B and $T_k,\ k=1,2,3$, taking as x_0 the midpoint of the interval X. Moreover, OTH_0 is computed using `fminu` of MatLab to perform a local search, setting μ equal to 0 as starting point, and performing at most 30 iterations. In order to check the computed bounds, an *outwardly rounded interval arithmetic code* must be used [6–8]. Here, we have developed in MatLab the needed operations, i.e., addition and multiplication for computing polynomial functions, with outwardly rounded computations. Hence, the results presented are numerically correct for each inclusion function. It may be possible that the floating computations performing the translation μ produce some numerical errors and that the result differs slightly, or in rare cases largely, from the optimal value. But no negative effect can occur, because $TF(\mu, X)$ is always an inclusion function for all real (or floating) values of μ.

In most cases, either the lower bound (lb) or the upper bound (ub) improved considerably with respect to the other enclosures, the optimal μ for the other bound being close to zero. The bounds, together with the optimal μ for OTH_0 are given in the last two rows of the table. It appears that some surprising improvements of the bounds are obtained for polynomial functions both of low degree (e.g. p_5) and high degree (e.g. p_2).

2.4. EXTENSION TO UNIVARIATE RATIONAL FUNCTIONS

The methodology extends in a straightforward manner to functions r given as the ratio of two polynomial functions p, q. Indeed, if $TH_0^p(\mu_1, X)$ and $TH_0^q(\mu_2, X)$ represent translation-based inclusion functions for p and q according to (11), then one obtains, for each μ_1, μ_2, the inclusion function

$$\frac{TH_0^p(\mu_1, X)}{TH_0^q(\mu_2, X)}.$$

The optimization in the translation parameters yields a new (and sharper) enclosure,

$$OTH_0^{\frac{p}{q}}(X) = \frac{OTH_0^p(X)}{OTH_0^q(X)} \tag{15}$$

$$= \frac{[\max_{\mu_1 \in \mathbb{R}} (TH_0^p)^L(\mu_1, X),\ \min_{\mu_1 \in \mathbb{R}} (TH_0^p)^U(\mu_1, X)]}{[\max_{\mu_2 \in \mathbb{R}} (TH_0^q)^L(\mu_2, X),\ \min_{\mu_2 \in \mathbb{R}} (TH_0^q)^U(\mu_2, X)]}$$

$$\supseteq \left[\max_{\mu \in \mathbb{R}} \left(\frac{TH_0^p(\mu, X)}{TH_0^q(\mu, X)} \right)^L,\ \min_{\mu \in \mathbb{R}} \left(\frac{TH_0^p(\mu, X)}{TH_0^q(\mu, X)} \right)^U \right]. \tag{16}$$

Remark 2. Remark that the latest enclosure, although less sharp, requires the resolution of two instead of four optimization problems. Moreover, if just one out of the two bounds is needed, one has to solve only one instead of four optimization problems.

The improvement in precision of the enclosures obtained in this way is illustrated in Table 2. We have compared $OTH_0^{\frac{p}{q}}$, as defined in (15) as well as the enclosure defined in (16) (the two last lines of the table) with the enclosures NE, $\frac{H_0^p}{H_0^q}$, the first-order Taylor expansion T_1 and the Baumann inclusion function T_1B.

The numerical tests are performed on the rational function $r(x) = \frac{p_1(x)}{p_5(x)}$ over different intervals. Observe that, for large intervals, the standard enclosures cannot exclude zero in the denominator, yielding the trivial interval $[-\infty, +\infty]$ using extended arithmetic.

Table 2. Results for lower or upper bounds for rational functions.

Pb	r over $[0,10]$		r over $[1,5]$		r over $[3,5]$		r over $[3.2,3.6]$	
	lb	ub	lb	ub	lb	ub	lb	ub
NE	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-1.54$	$0.66]$
$\frac{H_0^p}{H_0^q}$	$[-\infty$	$+\infty]$	$[-444.6$	$62.63]$	$[-8.0354$	$-0.00285]$	$[-0.92$	$-0.25]$
T_1	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-1.66$	$0.67]$
T_1B	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-\infty$	$+\infty]$	$[-1.16$	$0.17]$
$OTH_0^{\frac{p}{q}}$	$[-2.088E^4$	$12.33]$	$[-255.31$	$0.2168]$	$[-5.8607$	$-0.0494]$	$[-0.8615$	$-0.2768]$
	$[-\infty$	$+\infty]$	$[-315.12$	$63.01]$	$[-5.9431$	$-0.0494]$	$[-0.8615$	$-0.2768]$

3. The Multivariate Case

In this section, we explore possible extensions of the translation-based method to the case in which the function p under consideration has the form

$$p\colon\, x \in \mathbb{R}^m \longmapsto p(x) := \sum_{i=0}^{n} a_i \left(\prod_{j=1}^{m} x_j^{k_{ij}} \right), \tag{17}$$

where $a_i \in \mathbb{R}$, and $k_{ij} \in \mathbb{N} \cup \{0\}$, and an enclosure for p in the box $X := X_1 \times \dots X_m$ is sought.

The main idea is first to associate with p a series of univariate polynomial functions p^l, but having interval coefficients. For such functions p^l it is easy to extend our translation method in order to obtain an enclosure, yielding then an inclusion function of p.

Define, for each $l = 1, \dots, m$, the polynomial function with interval coefficients p^l,

$$\begin{aligned} p^l\colon\, & (X_1, \dots, X_{l-1}, x_l, X_{l+1}, \dots, X_m) \in \mathbb{I}^{l-1} \times \mathbb{R} \times \mathbb{I}^{m-l} \\ & \longmapsto p^l(X_1, \dots, X_{l-1}, x_l, X_{l+1}, \dots, X_m)\colon \sum_{i=0}^{n} \left(a_i \prod_{j=1, j \neq l}^{m} X_j^{k_{ij}} \right) x_l^{k_{il}}. \end{aligned} \tag{18}$$

Hence, if $P^l(X)$ denotes an enclosure of $p^l(X_1, \dots, X_{l-1}, x_l, X_{l+1}, \dots, X_m)$, then for each nonempty $L \subseteq \{1, 2, \dots, m\}$, the interval $\bigcap_{l \in L} P^l(X)$, is an enclosure for the function p given in (17) over the box X.

We then need to construct an enclosure for a univariate polynomial function $\underline{p}$ with interval coefficients,

$$\underline{p}(x) = \sum_{k=0}^{n} A_k x^k, \tag{19}$$

where $A_k \in \mathbb{I}$, $\forall k \in \{1, \dots, n\}$.

As in (8), one can rewrite $\underline{p}$ as

$$\underline{p}(x) \in \sum_{j=0}^{n} F_j(\mu)(X+\mu)^j \quad \forall x \in X,$$

where, as in (9), $F_j(\mu)$ is defined by

$$F_j(\mu) = \sum_{k=0}^{n-j} A_{k+j} \binom{k+j}{j} (-\mu)^k.$$

We define the translation-based Horner scheme $\underline{TH}$ as

$$\underline{TH}(\mu, X) F_0(\mu) + X_\mu (F_1(\mu) + X_\mu (\cdots (F_{n-1}(\mu) + X_\mu F_n(\mu)))), \tag{20}$$

with $X_\mu = X + \mu$, and we also define $\underline{TH_0}$ directly following Definition 1.
By optimizing the bounds, as in (13), we obtain the enclosure $\underline{OTH_0}$,

$$\underline{OTH_0}(X) := \left[\max_{\mu \in \mathbb{R}} \underline{TH_0}^L(\mu, X),\ \min_{\mu \in \mathbb{R}} \underline{TH_0}^U(\mu, X) \right]. \tag{21}$$

The properties enjoyed by this enclosure are similar to those described in Section 2.

With this, we have at hand a methodology for computing enclosures for multivariate polynomial functions p as defined in (17). The lower bounds obtained are compared in Table 3 with standard bounding procedures. Three bivariate polynomial functions are considered, p_8, p_9, p_{10}, taken from [2, 11],

$$p_8(x_1, x_2) = 2x_1^2 - 1.05x_1^4 + (1/6)x_1^6 - x_1x_2 + x_2^2,$$
$$p_9(x_1, x_2) = (x_1+1)^2 + (x_2-1)^2,$$
$$p_{10}(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4.$$

Table 3. Results for lower bounds for polynomial multivariate functions.

Pb	p_8 over			p_9 over		p_{10} over			
	X_1	X_2	X_3	X_1	X_3	X_1	X_3	X_4	X_5
NE	−39.6	−681.2	−2.05	0	1.0	−459.2	−7.1	−0.777	0.267
T_1	−260.9	−2.3E3	−5.56	−73.5	−2.8	−7.0E5	−29.6	0.117	0.372
T_1B	−175.3	−1.3E4	−4.76	−8.4	1.0	−979.5	−28.8	0.373	0.373
T_2	−274.2	−9.9E3	−3.44	−23.7	−0.3	−5.8E5	−23.9	0.251	0.373
$\underline{TH}^{1,2}(0,\cdot)$	−22.6	−59.0	−1.00	−7.4	−1	−120.2	−5.0	0.268	0.364
$\underline{TH}_0^{1,2}(0,\cdot)$	−22.6	−34.0	−1.00	−3.2	0	−120.2	−5.0	0.268	0.364
$\underline{OTH}_0^{1,2}$	−12.9	−25.0	−0.39	0	1.0	−25.71	−4.2	0.324	0.369

Lower bounds for these polynomial functions are computed on different boxes, $X_1, \ldots, X_5 \in \mathbb{I}^2$,

$$X_1 = [-2.1, 2] \times [0.9, 10]$$
$$X_2 = [0, 5] \times [-5, 5]$$
$$X_3 = [0, 1] \times [-1, 1]$$
$$X_4 = [0.5, 0.6] \times [-1.1, -1]$$
$$X_5 = [0.5, 0.51] \times [-1.01, -1].$$

Different bounding methods are considered, namely, the natural extension NE, first-order and second-order Taylor expansions T_1, T_2, (centered at the midpoint) and Baumann form T_1B, where the three last inclusion functions are the generalization to the multidimensional case of the forms previously defined, [1, 10].

For each bivariate polynomial function $p(x_1, x_2)$, p^1, p^2 are constructed following (18), their corresponding inclusion functions $\underline{TH^{1,2}} := \underline{TH^1} \cap \underline{TH^2}$ following (20), then $\underline{TH_0^{1,2}} := \underline{TH_0^1} \cap \underline{TH_0^2}$, and finally $\underline{OTH_0^{1,2}}$: $\underline{OTH_0^1} \cap \underline{OTH_0^2}$ following (21).

The results are given in Table 3. It appears that $\underline{OTH_0^{1,2}}$ outperforms NE; an exception is p_9, whose expression has its variables separated, and for which the natural extension already produces optimal bounds. For small intervals the translation-based method produces bounds comparable with Taylor forms, whereas for large intervals $\underline{OTH_0^{1,2}}$ is more efficient.

4. Application to Global Optimization

The numerical experiments presented in the previous sections show that translation-based methods may yield much sharper bounds than the standard enclosure procedures. However, since the computation of these forms is much more expensive in CPU-time, it is not clear in advance if, within a Branch-and-Bound procedure, it deserves spending a (much) longer computing time in order to obtain (much) sharper lower bounds.

We have implemented a Branch-and-Bound procedure, based on the Ichida-Fujii method, [5], [10], which encloses all the ε-optimal solutions of the problem. The algorithm, described in what follows, is a two-phase procedure; the first phase seeks the optimal value (up to ε_f), by a Branch-and-Bound procedure with bisection as branching rule and selecting the interval with the lowest lower bound; once the optimal value is found, one starts a new Branch-and-Bound procedure, still branching by bisection, selecting the largest interval, and stopping when all intervals remaining in the list are sufficiently small, namely of length not greater than ε_X. The final list of intervals is such that any ε_f-optimal solution of the problem is contained in the union of these intervals.

Algorithm

Phase 1. 1. Set $X :=$ the interval in which the global minimum is sought.
2. Set $f_{\min} := +\infty$.
3. Set $\mathcal{L} := \{(+\infty, X)\}$.
4. Extract from $\mathcal{L}$ the element with the lowest lower bound.
5. Bisect the interval chosen by its midpoint, yielding V_1, V_2.
6. For j:=1 to 2 do
   (a) Compute $v_j :=$lower bound of f on V_j.
   (b) If $f_{\min} \geqslant v_j$ then
      - Insert (v_j, V_j) in $\mathcal{L}$.
      - Set $f_{\min} := \min(f_{\min}, f(m))$, where m is the midpoint of V_j.
      - If $f_{\min}$ is changed then remove from $\mathcal{L}$ all couples (z, Z) with $z > f_{\min}$.
7. If $f_{\min} < \min_{(z,Z)\in\mathcal{L}} z + \epsilon_f$, then GoTo Phase 2. Else GoTo Step 4.

Phase 2. 1. Extract from $\mathcal{L}$ the largest interval.
2. If the interval chosen has length not greater than ε_X then STOP.
3. Bisect the interval chosen by its midpoint, yielding V_1, V_2.
4. For $j := 1$ to 2 do
   (a) Compute $v_j :=$lower bound of f on V_j.
   (b) If $f_{\min} \geqslant v_j$ then
      - Insert (v_j, V_j) in $\mathcal{L}$.
      - Set $f_{\min} := \min(f_{\min}, f(m))$, where m is the midpoint of V_j.
      - If $f_{\min}$ is changed then remove from $\mathcal{L}$ all couples (z, Z) with $z > f_{\min}$.
5. GoTo 1.

This procedure has been implemented in Fortran 90 on a Digital AlphaServer 8200 5 /625 quadriprocessor, using as bounding procedures NE, $H_0, T_1, T_2, T_1B, T_n, OTH_0, OTNE$. The local optimization used in the translation methods was done with the NAG-subroutine E04ABF, performing at most 30 iterations, seeking for the optimal μ in the interval $[-100, 100]$. All these methods for finding inclusions use an *outwardly rounded interval arithmetic code* developed in [7]. Thus, the computed bounds are correct (no numerical error can occur) and that is why the global optimization based on this principle is said to be *rigorous* [6, 7].

As test functions we have considered first the functions p_1, p_3, p_4, p_6, p_7 of Section 2 together with the Goldstein-Price function, p_{11},

$$p_{11} := 250 + 27x^2 - 15x^4 + x^6.$$

Different intervals and values of ε_f are chosen, whereas ε_X is fixed to 0.00001. The results are summarized in Table 4, where we give the number of iterations needed, Its, the CPU-time in seconds, time(s), and the number of intervals in the final list, Cls.

One can note on these first numerical examples the efficiency of our algorithms; the number of iterations can be strongly reduced (from thousands to 8 in the first example), CPU-times are much better than those obtained with the Natural Extension and of the same order than those produced with Taylor forms. Moreover, there is a dramatic reduction in the number of intervals (clusters) remaining in the final list; thus the well-known clustering problem [2] is avoided.

In order to study if the efficiency of the translation-based method is dependent on the degree of the polynomial function considered, we have generated randomly polynomial functions of degrees $5, 6, 10, 11, 14, 15$ with coefficients uniformly distributed in the set $\{-10, -9, \ldots, 0, \ldots, 10\} \subset \mathbb{N}$. The sample size in all cases is 1000. The interval within which the polynomial functions are optimized is always $[-1, 1], \varepsilon_f = 0.0001$, and $\varepsilon_X = 0.00001$. Table 5 shows for each sample the average number of iterations, Its, the total CPU-times in seconds, time(s), and the average number of clusters, Cls, rounded to the closest integer.

It appears that, compared with those obtained with Taylor forms, the number of clusters and iterations is strongly reduced for all the degrees of polynomial functions tested. However, computing times are higher; hence, there is room for heuristic rules.

5. Conclusion

The purpose of this paper was to show that a linear change of variable can considerably improve the quality of bounds in Interval Arithmetic computations. This leads to the problem of determining the optimal translation, to be obtained via local search. Numerical tests are given for univariate and multivariate polynomial and univariate rational functions, showing a significant improvement of the enclosures. Finally, the different inclusion functions are used in an Interval Branch-and-Bound algorithm, showing that, although at a higher computation cost, translation-based methods reduce strongly the number of clusters and iterations.

Table 4. Results for optimization of univariate polynomial functions.

Pb	Its	p_1 over $[-2, 11]$ $\varepsilon_f = 1.$ time(s)	Cls	Its	p_3 over $[0, 10]$ $\varepsilon_f = 0.01$ time(s)	Cls	Its	p_4 over $[-5, 5]$ $\varepsilon_f = 0.1$ time(s)	Cls	Its	p_6 over $[-100, 0]$ $\varepsilon_f = 0.01$ time(s)	Cls	Its	p_7 over $[-5, 5]$ $\varepsilon_f = 0.01$ time(s)	Cls	Its	p_{11} over $[-30000, 0]$ $\varepsilon_f = 0.01$ time(s)	Cls
NE	9311	4.26	4190	8167	3.00	4727	5460	1.84	2283	–	–	–	1567	0.11	667	1671	0.13	754
H_0	4059	0.90	2187	2721	0.66	1290	2257	0.31	942	3945	1.57	3188	1188	0.06	511	1092	0.08	510
T_1	1167	0.16	1103	628	0.04	517	186	0.01	63	248	0.02	113	184	0.01	135	392	0.02	262
T_2	1217	0.20	1188	584	0.04	511	131	0.01	67	184	0.02	112	179	0.01	143	357	0.02	278
T_1B	1265	0.20	1208	611	0.04	517	159	0.01	58	195	0.02	117	158	0.01	136	317	0.02	271
T_n	1260	0.23	1233	597	0.06	548	75	0.01	5	185	0.03	118	170	0.01	140	341	0.02	265
$OTNE$	10	0.01	1	647	0.57	595	73	0.02	12	19	0.14	1	8	0.01	1	381	0.21	343
OTH_0	8	0.01	1	6	0.01	1	65	0.02	10	16	0.04	1	6	0.01	1	21	0.02	1

Table 5. Results for optimization of randomly generated univariate polynomial functions.

Pb	Its	deg 5 time(s)	Cls	Its	deg 6 time(s)	Cls	Its	deg 10 time(s)	Cls	Its	deg 11 time(s)	Cls	Its	deg 14 time(s)	Cls	Its	deg 15 time(s)	Cls
NE	528	68.01	258	1081	176.59	571	696	169.31	393	1197	289.29	702	1472	461.13	946	1036	343.31	660
H_0	323	25.91	145	580	54.73	263	340	50.96	164	524	69.21	256	531	86.07	282	380	58.04	201
T_1	64	3.98	49	112	7.78	84	77	18.54	59	107	11.88	79	108	15.71	78	84	12.08	63
T_2	61	4.45	51	104	8.94	86	72	21.00	60	97	12.42	79	97	15.27	79	75	13.18	62
T_1B	50	3.25	41	91	6.54	74	66	17.95	55	89	10.78	72	94	14.92	74	71	11.02	62
T_n	58	4.72	49	100	10.63	82	69	22.33	59	93	15.78	77	93	22.82	77	72	17.96	62
$OTNE$	25	13.14	21	42	32.77	34	37	59.34	32	45	67.53	37	52	135.76	44	41	105.28	35
OTH_0	11	6.04	8	16	14.10	10	18	28.67	15	17	27.81	12	22	56.33	17	15	38.81	11

References

1. Baumann, E. (1988), Optimal centered form, *BIT*, 28, 80–87.
2. Du, K. and Kearfott, R.B. (1996), The cluster problem in multivariate global optimization, *Journal of Global Optimization*, 10, 27–32.
3. Hansen, E. (1992), *Global Optimization Using Interval Analysis*, Marcel Dekker, New York.
4. Hansen, P., Jaumard, B. and Lu, S.-H. (1989), Global minimization of univariate functions by sequential polynomial approximation international, *International Journal of Computer Mathematics*, 28, 183–193.
5. Ichida, K. and Fujii, Y. (1979), An interval arithmetic method for global optimization, *Computing*, 23, 85–97.
6. Kearfott, R.B. (1996), *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht, Boston, London.
7. Messine, F. (1997), Méthodes d'optimisation globale basées sur l'analyse d'intervalle pour la résolution de problèmes avec contraintes, PhD Thesis, INPT-ENSEEIHT, Toulouse. Available on the website: www.univ-pau.fr/∼messine
8. Moore, R.E. (1996), *Interval Analysis*, Prentice Hall, Englewood Cliffs, N.J.
9. Ratschek, H. and Rokne, J. (1984), *Computer Methods for the Range of Functions*, Ellis Horwood, Chichester, England.
10. Ratschek, H. and Rokne, J. (1988), *New Computer Methods for Global optimization*, Ellis Horwood, Chichester, England.
11. Visweswaran, V. and Floudas, C.A. (1992), Unconstrained and constrained global optimization of polynomial functions in one variable, *Journal of Global Optimization*, 2(1), 73–100.
12. Wingo, D.R. (1985), Globally minimizing polynomials without evaluating derivatives international, *Journal of Computer Mathematics*, 17, 287–294.