NOLTA, IEICE

Paper

Implementation and improvements of affine arithmetic

Siegfried M. Rump [1 a)] *and Masahide Kashiwagi* [2]

[1] *Institute for Reliable Computing, Hamburg University of Technology, Schwarzenbergstraße 95, Hamburg 21071, Germany, and Visiting Professor at Waseda University, Faculty of Science and Engineering, 3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan*

[2] *Faculty of Science and Engineering, 3–4–1 Okubo, Shinjuku-ku, Tokyo 169–8555, Japan*

[a)] *rump@tuhh.de*

Received October 27, 20XX; Revised December 29, 20XX; Published July 1, 20XX

Abstract: Affine arithmetic is a well-known tool to reduce the wrapping effect of ordinary interval arithmetic. We discuss several improvements both in theory and in terms of practical implementation. In particular details of INTLAB's affine arithmetic toolbox are presented. Computational examples demonstrate advantages and weaknesses of the approach.

Key Words: Affine arithmetic, ordinary interval arithmetic, wrapping effect, overestimation, Min-Range approximation, Chebyshev approximation, INTLAB.

1. Introduction

Ordinary interval arithmetic [18–20, 31] is a convenient tool to estimate the range of a function. It is a convenient way to realize verification methods for proving existence and uniqueness of the solution of linear and nonlinear problems within a computed box. For an overview, see [28]. Interval operations and methods are available in INTLAB [26], the Matlab toolbox for reliable computing. For an introduction to verification methods based on INTLAB see [17].

Higher dimensional intervals are usually realized as n-dimensional boxes, i.e. the Cartesian product of one-dimensional intervals. This allows a very simple and efficient implementation of interval matrix and vector operations [25].

However, as a drawback, there is no "orientation": the boxes are always parallel to the axes. This may cause overestimation, known as the wrapping effect. The most simple example is the 2-dimensional unit square rotated by 45 degrees: The best axis-parallel inclusion increases the radii by a factor $\sqrt{2}$. The wrapping effect is one of the major obstacles when integrating ODEs over a longer time interval.

As an alternative to interval vectors, [2] considers oblique boxes $Q\mathbf{X}$ for orthogonal Q and an interval vector X, where the product is not executed but Q and $\mathbf{X}$ are stored separately.

Another approach involving ellipsoids is defined by [21], see also [15], where the ellipsoid is the image of the unit ball by a triangular matrix; for interesting applications see [23].

DOI: 10.1588/nolta.2.1101

In [1, 5, 7, 8, 10, 12, 30] an affine arithmetic is defined. It is a powerful tool to reduce the wrapping effect. Here interval quantities are represented by $c + \sum \gamma_i \mathcal{E}_i$ for $c, \gamma_i \in \mathbb{R}$ and $\mathcal{E}_i = [-1, 1]$. There is a vast literature on affine arithmetic, see [3] and articles cited over there.

For each interval representation advantages are counterbalanced by increased computational costs for interval operations. In this paper we discuss some improvements of affine arithmetic together with an efficient implementation. We assume the reader to be familiar with basic concepts of ordinary interval arithmetic, see [20, 29]. We use the standard notation for intervals as in [14].

2. Affine arithmetic

The main reason to define affine arithmetic is to attack the dependency problem, which occurs in particular in the estimation of the range of a function. In order to prove inclusion results of our affine representation we need a formal definition of affine quantities. Based on this formalization we briefly repeat known properties of affine arithmetic and then proceed to our improvements.

We define the set of affine quantities $\mathcal{A} := \bigcup\{\mathcal{A}^k : k \in \mathbb{N}_0\}$, where $\mathcal{A}^k := \{\langle c; \gamma\rangle : c \in \mathbb{R}, \gamma \in \mathbb{R}^k\}$ with $\mathbb{R}^0 := \emptyset$. To $C \in \mathcal{A}^k$ we assign an affine function $\psi_C : \mathbb{R}^k \to \mathbb{R}$ with $\psi_C(\varepsilon) := c + \sum_{i=1}^k \gamma_i \varepsilon_i$ for $\varepsilon \in \mathbb{R}^k$. The range of an affine quantity $C := \langle c; \gamma\rangle \in \mathcal{A}^k$ is defined by

$$\operatorname{range}(C) := \{c + \sum_{i=1}^{k} \gamma_i \varepsilon_i : \varepsilon_i \in \mathcal{E}^k\} \qquad \text{where } \mathcal{E} := [-1, 1]. \tag{1}$$

Using power set operations we have $\operatorname{range}(C) = \psi_C(\mathcal{E}^k)$. For $k = 0$ the range is simply the real number $\{c\}$. We call c the midpoint of the affine quantity $C = \langle c; \gamma\rangle$ and γ_ν the error terms: For $k = 1$ the range of $\langle c; \gamma_1\rangle$ is the interval $[c - |\gamma_1|, c + |\gamma_1|]$.

For given $C, D \in \mathcal{A}$ the set $\{(x, y) : x \in C, y \in D\} \subseteq \mathbb{R}^2$ can be visualized. Define, for example, $C = \langle 2; 1, -2, 3, -1\rangle$ and $D = \langle 1; 3, 0, -1, 2\rangle$, then this is the set of all points $(2+\varepsilon_1-2\varepsilon_2+3\varepsilon_3-\varepsilon_4, 1+3\varepsilon_1-\varepsilon_3+2\varepsilon_4) \in \mathbb{R}^2$ for $\varepsilon_i \in \mathcal{E}$, shown in Figure 1. The dependency of the x- and y-coordinates creates the affine shape which is called zonotope in [8]. Note that the Cartesian product $\operatorname{range}(C) \times \operatorname{range}(D)$ is the rectangle $[-5, 9] \times [-5, 7]$.

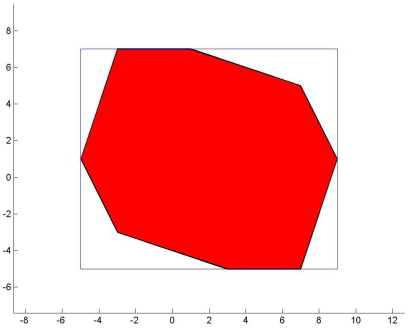

Fig. 1: Cartesian product of two affine quantities.

For $C, D \in \mathcal{A}^k$ with $C = \langle c; \gamma\rangle$, $D = \langle d; \delta\rangle$ addition is defined by

$$C + D := \langle c + d; \gamma + \delta\rangle \in \mathcal{A}^k. \tag{2}$$

It follows

$$\forall \varepsilon \in \mathbb{R}^k : \quad \psi_C(\varepsilon) + \psi_D(\varepsilon) = \psi_{C+D}(\varepsilon) \tag{3}$$

by $\psi_C + \psi_D = \psi_{C+D}$. Note that (3) implies an inclusion property

$$\operatorname{range}(C + D) \subseteq \operatorname{range}(C) + \operatorname{range}(D)$$

as for ordinary interval arithmetic, but not vice versa. The important point is to use the same quantity ε throughout (3). This allows to identify dependencies.

The case $C \in \mathcal{A}^k$ and $D \in \mathcal{A}^\ell$ with $k < \ell$ is resolved via a natural embedding. We set $C' := \langle c; \gamma, z\rangle$ with z denoting $\ell - k$ zeros. Then $\psi_C(\varepsilon) = \psi_{C'}(\varepsilon')$ for all $\varepsilon \in \mathbb{R}^k$ and $\varepsilon' := (\varepsilon, z) \in \mathbb{R}^\ell$. Thus C can safely be replaced by C' without changing the range of C and without jeopardizing the inclusion property (3). The case $k > \ell$ is handled similarly. Based on that we may henceforth assume that two affine quantities have the same number of error terms. Subtraction is defined by

$$C - D := \langle c - d; \gamma - \delta\rangle \in \mathcal{A}^k. \tag{4}$$

It follows

$$\forall \varepsilon \in \mathbb{R}^k : \quad \psi_C(\varepsilon) - \psi_D(\varepsilon) = \psi_{C-D}(\varepsilon), \tag{5}$$

and in particular range$(C - C) = \{0\}$ for any $C \in \mathcal{A}$. Multiplication for $C, D \in \mathcal{A}^k$ is defined by

$$C \cdot D := \langle\, cd\,;\, c\delta + \gamma d, \|\gamma\|_1 \|\delta\|_1 \,\rangle \in \mathcal{A}^{k+1}. \tag{6}$$

It follows

$$\forall \varepsilon \in \mathcal{E}^k\, \exists \varepsilon' \in \mathcal{E} : \quad \psi_C(\varepsilon)\psi_D(\varepsilon) = \psi_{C\cdot D}(\overline{\varepsilon}) \quad \text{for } \ \overline{\varepsilon} := (\varepsilon, \varepsilon') \in \mathcal{E}^{k+1}. \tag{7}$$

Again it is important that the same quantity ε is used throughout (7). Division is based on the reciprocal, which in turn is a special case of a univariate nonlinear function $f : \mathbb{R} \to \mathbb{R}$. In the literature f is represented by a function $F : \mathcal{A} \to \mathcal{A}$ based on some linearization.

We say that the triplet $[\![p, q, \Delta]\!]$ with $p, q, \Delta \in \mathbb{R}$ *represents* $f : \mathcal{D} \subseteq \mathbb{R} \to \mathbb{R}$ *on* $\mathbf{X} \in \mathbb{IR}$, $\mathbf{X} \subseteq \mathcal{D}$, if

$$\forall x \in \mathbf{X} : \quad |px + q - f(x)| \le \Delta. \tag{8}$$

Any function f is represented by

$$[\![0, (m + M)/2, (M - m)/2]\!] \qquad \text{where } \ m := \inf\{f(x) : x \in \mathbf{X}\}, \ M := \sup\{f(x) : x \in \mathbf{X}\}. \tag{9}$$

This is called the *ordinary interval representation.* Practically speaking, m and M are usually difficult to determine unless f has special properties such as monotonicity.

Besides that, two main principles are used to produce a representation $[\![p, q, \Delta]\!]$, the Min-Range and the Chebyshev approximation. The following lemma is well-known; for completeness we sketch part of the proof.

Lemma 1 Let $\mathbf{X} := [a, b] \in \mathbb{IR}$ and twice differentiable $f : \mathbf{X} \to \mathbb{R}$ be given.

For the Min-Range representation suppose f is convex or concave on $\mathbf{X}$ and $f'(x) \neq 0$ on $\mathbf{X}$. Define $p := f'(a)$ if $f'(x)f''(x) \ge 0$ on $\mathbf{X}$, and $p := f'(b)$ otherwise. Then $[\![p, q, \Delta]\!]$ with

$$q := \frac{f(a) + f(b) - p(a + b)}{2} \quad \text{and} \quad \Delta := \left|\frac{f(b) - f(a) - p(b - a)}{2}\right| \tag{10}$$

represents f on $\mathbf{X}$. It is a "Min-Range" representation by $\{px + q + \delta : x \in \mathbf{X}, |\delta| \le \Delta\} = \{f(x) : x \in \mathbf{X}\}$.

For the Chebyshev representation suppose f is convex or concave on $\mathbf{X} = [a, b]$ with $a \neq b$. Define

$$p := \frac{f(b) - f(a)}{b - a}. \tag{11}$$

Furthermore let $\xi \in \mathbf{X}$ be such that $f'(\xi) = p$ and define

$$q := \frac{f(a) + f(\xi) - p(a + \xi)}{2} \quad \text{and} \quad \Delta := \left|\frac{f(\xi) - f(a) - p(\xi - a)}{2}\right|. \tag{12}$$

Then $[\![p, q, \Delta]\!]$ represents f on $\mathbf{X}$.

Proof. For the Min-Range approximation suppose f is convex, i.e. $f''(x) \ge 0$ on $\mathbf{X}$. Then

$$f(a) + f'(a)(x - a) \le f(x) \le f(b) + f'(a)(x - b)$$

for all $x \in \mathbf{X}$, so that

$$f'(a)b - f(b) \le f'(a)x - f(x) \le f'(a)a - f(a).$$

It follows

$$-\Delta = \frac{f(a) - f(b) - f'(a)(a-b)}{2} \le px + q - f(x) \le \frac{f(b) - f(a) - f'(a)(b-a)}{2} = \Delta.$$

For the Chebyshev approximation and convex f we have

$$f(\xi) + p(x - \xi) \le f(x) \quad \text{such that} \quad px - f(x) \le p\xi - f(\xi) \tag{13}$$

for all $x \in \mathbf{X}$. Hence

$$px + q - f(x) \le p\xi - f(\xi) + \frac{f(a) + f(\xi) - p(a+\xi)}{2} = \frac{f(a) - f(\xi) - p(a - \xi)}{2}$$

which, in absolute value, is equal to Δ. The other parts follow similarly. ■

Let $C = \langle c; \gamma \rangle \in \mathcal{A}^k$ be given and suppose $[\![p, q, \Delta]\!]$ represents $f(x) : \mathbb{R} \to \mathbb{R}$ on range(C). Define the function $F : \mathcal{A}^k \to \mathcal{A}^{k+1}$ by

$$F(C) := \langle pc + q; p\gamma, \Delta \rangle. \tag{14}$$

Then

$$\forall \varepsilon \in \mathcal{E}^k \, \exists \varepsilon' \in \mathcal{E} : \quad f(\psi_C(\varepsilon)) = \psi_{F(C)}(\bar{\varepsilon}) \quad \text{for } \bar{\varepsilon} := (\varepsilon, \varepsilon') \in \mathcal{E}^{k+1}. \tag{15}$$

This general principle applies to Min-Range and Chebyshev approximations and to general continuous or differentiable nonlinear functions $f : \mathbb{R} \to \mathbb{R}$, respectively. Often the general formulas in Lemma 1 can be simplified for particular functions. For example, for the reciprocal $f(x) := 1/x$ we obtain the following result. The Min-Range and the Chebyshev approximation on $\mathbf{X} = [1, 2]$ is visualized in Figure 2.

Corollary 1 Let $\mathbf{X} := [a, b]$ with $b > a > 0$ be given. Then $[\![p, q, \Delta]\!]$ represents $f(x) := 1/x$ on $\mathbf{X}$ by Min-Range approximation for

$$p := -1/b^2, \quad q := -\frac{p(a+b)^2}{2a} \quad \text{and} \quad \delta := -\frac{p(a-b)^2}{2a}, \tag{16}$$

and $[\![p, q, \Delta]\!]$ represents $f(x) := 1/x$ on $\mathbf{X}$ by Chebyshev approximation for

$$p := -1/ab, \quad q := -p(\sqrt{a} + \sqrt{b})^2/2 \quad \text{and} \quad \delta := -p(\sqrt{a} - \sqrt{b})^2/2. \tag{17}$$

Similar formulas are derived for $b < 0$.

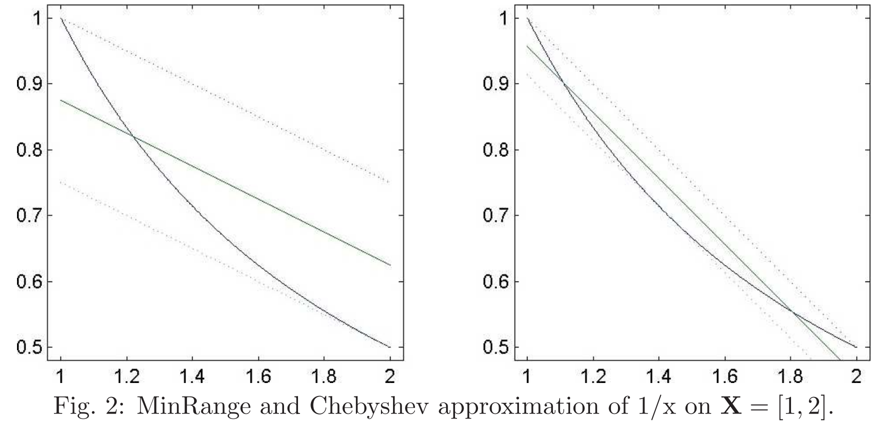

Fig. 2: MinRange and Chebyshev approximation of 1/x on $\mathbf{X} = [1, 2]$.

So far the general definition of affine arithmetic from the literature. Note that for $k = 1$ there is some similarity to the midpoint-radius representation of intervals. The big difference, however, is that using ordinary interval arithmetic $\mathbf{X}-\mathbf{X}\neq[0,0]$ for $\mathrm{rad}(\mathbf{X})\neq 0$, regardless whether intervals are represented in infimum-supremum or midpoint-radius form.

Nevertheless, this leads to a disadvantage of affine arithmetic. For $C=\langle 2;1\rangle$ formula (6) yields

$$C\cdot C=\langle 4;4,1\rangle \quad \text{which means} \quad \mathrm{range}(C\cdot C)=[-1,9]. \tag{18}$$

The same is used in [30]. But $\mathrm{range}(C)=[1,3]$ so that the product in ordinary interval arithmetic is $[1,9]$. It follows that ordinary interval arithmetic computes a perfect and sharp inclusion of $1/C^2$, whereas affine arithmetic stops with division by zero. In that specific example the expected improvement by affine arithmetic turns into a disadvantage.

The same applies to the Chebyshev approximation which is minimizing the maximum error between $f(x)$ and $px+q$ on $\mathbf{X}$, where the Min-Range approximation minimizes the range, that is $\{f(x):x\in\mathbf{X}\}=\{px+q:x\in\mathbf{X}\}$. In the example $1/[1,2]$ in Figure 2 the range by Chebyshev approximation is $[\sqrt{2}-1,1]$ instead of $[0.5,1]$.

Next we present several improvements of traditional affine arithmetic, both theoretically as well as for the practical implementation. In particular the overestimation in the examples just presented are avoided. In our affine arithmetic the result can never be worse than ordinary interval arithmetic.

3. Improvements of affine arithmetic

We first present all results for real operations and address the inevitable rounding errors at the end of this section. The target for INTLAB's affine arithmetic package is to provide improved estimates for the four basic operations and the following standard functions:

$$\begin{aligned}&\mathrm{sqrt},\mathrm{sqr},\\&\exp,\log,\mathrm{log2},\mathrm{log10},\mathrm{power},\\&\sin,\cos,\tan,\cot,\sec,\csc,\\&\mathrm{asin},\mathrm{acos},\mathrm{atan},\mathrm{acot},\mathrm{asec},\mathrm{acsc},\\&\sinh,\cosh,\tanh,\coth,\\&\mathrm{asinh},\mathrm{acosh},\mathrm{atanh},\mathrm{acoth},\\&\mathrm{erf},\mathrm{erfc}.\end{aligned} \tag{19}$$

These functions form a basic set of what is used in numerical analysis. For all of them Min-Range and Chebyshev approximations are provided, most of them optimal.

3.1 The range component

In its original definition, affine arithmetic computes (18), a significant disadvantage even over ordinary interval arithmetic. Fortunately, there is a simple remedy to that.

In [7, Section 3.15] Figueiredo and Stolfi introduce the so-called "mixed IA/AA". Although it seems an important tool, it is not found in Stolfi's implementation [30]. Apparently, other implementations use it [9].

We extend the definition of affine quantities in the following way. The pair $\mathfrak{C}:=\{C,\mathbf{X}\}$ with $C\in\mathcal{A}$ is an affine representation of $\mathbf{X}\in\mathbb{IR}$ if $\mathrm{range}(C)\cap\mathbf{X}\neq\emptyset$. Basic operations on pairs $\mathfrak{C}=\{C,\mathbf{X}\}$ and $\mathfrak{D}=\{D,\mathbf{Y}\}$ are defined by

$$\mathfrak{C}\,\mathrm{op}\,\mathfrak{D}:=\{C\,\mathrm{op}\,D,\mathbf{X}\,\mathrm{op}\,\mathbf{Y}\} \qquad \text{for} \qquad \mathrm{op}\in\{+,-,\times\}. \tag{20}$$

For addition and subtraction and $C,D\in\mathcal{A}^k$ it follows immediately

$$\forall\varepsilon\in\mathcal{E}^k:\quad \psi_C(\varepsilon)\in\mathbf{X},\psi_D(\varepsilon)\in\mathbf{Y}\ \Rightarrow\ \psi_C(\varepsilon)\pm\psi_D(\varepsilon)=\psi_{C\pm D}(\varepsilon)\in\mathbf{X}\pm\mathbf{Y}. \tag{21}$$

For multiplication we have, in addition to (7), the sharper inclusion $\psi_C(\varepsilon)\psi_D(\varepsilon)\in\mathbf{XY}$ for all $\varepsilon\in\mathcal{E}^k$. The advantage is that in addition to the range information produced by affine arithmetic the true range of the individual operation produced by ordinary interval arithmetic is available.

This advantage becomes virulent when applying a nonlinear function $f : \mathbb{R} \to \mathbb{R}$. Let $\{C, \mathbf{X}\}$ with $C = \langle c; \gamma\rangle \in \mathcal{A}^k$ be given and suppose $[\![p, q, \Delta]\!]$ represents $f(x) : \mathbb{R} \to \mathbb{R}$ on range$(C) \cap \mathbf{X}$. Define

$$F(\{C, \mathbf{X}\}) := \{F(C), f(\mathbf{X})\} \tag{22}$$

where $F(C)$ is as in (14) and the power set $f(\mathbf{X}) := \{f(x) : x \in \mathbf{X}\} \subseteq \mathbb{R}$. Then

$$\forall \varepsilon \in \mathcal{E}^k\, \exists \varepsilon' \in \mathcal{E} : \quad \psi_C(\varepsilon) \in \text{range}(C) \cap \mathbf{X} \;\Rightarrow\; f(\psi_C(\varepsilon)) = \psi_{F(C)}(\bar\varepsilon) \quad \text{for } \bar\varepsilon := (\varepsilon, \varepsilon') \in \mathcal{E}^{k+1}. \tag{23}$$

Consider the example $1/(C \cdot C)$ for $C = \langle 2; 1\rangle$ from the end of the last section. Then C is represented by $\mathfrak{C} := \{C, [1, 3]\}$, and

$$\mathfrak{C} \cdot \mathfrak{C} = \{C \cdot C, [1, 9]\} = \{\langle 4; 4, 1\rangle, [1, 9]\}.$$

To compute $1/(C \cdot C)$, formula in (14) requires a representation $[\![p, q, \Delta]\!]$ of $f(x) = 1/x$ on range$(C \cdot C) = [-1, 9]$, which is not possible. The new definition (22) requires a representation $[\![p, q, \Delta]\!]$ only on $\mathbf{X} = [1, 9]$. Following Corollary 1 candidates are, rounded to three figures and ignoring rounding errors, the Min-Range approximation $[\![-0.012, 0.617, 0.395]\!]$ and the Chebyshev approximation $[\![-0.111, 0.889, 0.222]\!]$. Based on one of those $F(C)$ is computed according to (14). In either case the interval part of the result is $1/[1, 9] = [0.111, 1]$.

3.2 Multiplication

Much of the following can be found in [3]. The practical benefit seems limited, we mention it for completeness.

For $C, D \in \mathcal{A}^k$ with $C = \langle c; \gamma\rangle$, $D = \langle d; \delta\rangle$ the additional error term of the product $C \cdot D$ can be improved. Setting $C \cdot D := \langle\, cd\,;\, c\delta + \gamma d, e\,\rangle$, any $e \in \mathbb{R}$ satisfying (7) is a suitable choice. The optimal e is

$$e_{\text{opt}} := \max\{|(\sum_{i=1}^{k} \gamma_i \varepsilon_i)(\sum_{i=1}^{k} \delta_i \varepsilon_i)| : \varepsilon \in \mathcal{E}^k\}. \tag{24}$$

Obviously, $e_{\text{opt}} \le \|\gamma\|_1 \|\delta\|_1$ as in (6). By the symmetry of (24) this means to maximize

$$2(\varepsilon^T\gamma)(\delta^T\varepsilon) = \varepsilon^T[\gamma\delta^T + \delta\gamma^T]\varepsilon =: \varepsilon^T M \varepsilon \qquad \text{subject to } \varepsilon \in \mathcal{E}^k$$

for column vectors $\gamma, \delta \in \mathbb{R}^k$ and the symmetric matrix M. The dependencies can be reduced by collecting common terms

$$\varepsilon^T M \varepsilon = 2\sum_{i=1}^{k} \gamma_i\delta_i\varepsilon_i^2 + 2\sum_{i<j}(\gamma_i\delta_j + \gamma_j\delta_i)\varepsilon_i\varepsilon_j,$$

so that

$$e := \left(\sum_{i=1}^{k} |\gamma_i\delta_i| + \sum_{i<j} |\gamma_i\delta_j + \gamma_j\delta_i|\right) \tag{25}$$

is a suitable choice. Splitting the vector $v \in \mathbb{R}^k$ with $v_i := \gamma_i\delta_i$ into positive and negative parts $v^+, v^- > 0$ with $v = v^+ - v^-$ improves (25) into

$$e := \left(\max\left[\sum_{i=1}^{k} v_i^+, \sum_{i=1}^{k} v_i^-\right] + \sum_{i<j} |\gamma_i\delta_j + \gamma_j\delta_i|\right). \tag{26}$$

Another approach[1] is to use a decomposition of the matrix M. One possibility is the spectral decomposition $M = XDX^T$ with diagonal D and orthogonal X. Normalizing $x := \|\gamma\|_2^{-1}\gamma$ and $y := \|\delta\|_2^{-1}\delta$ the eigenvectors are[2] $x \pm y$ to the eigenvalues $1 \pm x^T y$, so that a little computation yields that

[1] suggested by Marko Lange, Hamburg University of Technology

[2] suggested by Florian Bünger, Hamburg University of Technology

$$e := 0.25\|\gamma\|_2\|\delta\|_2 \max\{\|x+y\|_1^2, \|x-y\|_1^2\} \tag{27}$$

is also a suitable choice. The same method applies to an LDL^T decomposition of M. In either case the decompositions require only $\mathcal{O}(k)$ operations, whereas (25) and (26) need $\mathcal{O}(k^2)$ operations. The matrix M can also be represented as the sum of two rank-one matrices, which leads to an optimization problem in two unknowns. In any case, the computational effort has to be traded against the improvement of the representation.

Practical experience suggests that, in general, (26) is the best possibility. Among many random test cases there was no case where the bound using the spectral decomposition was superior to (26). Since the computational effort is not too large, another possibility is to compute all bounds and use the minimum. The computation of the optimal bound for n error terms requires $\mathcal{O}(n^2)$ operations, see [16]. Table I shows for random test vectors $\gamma, \delta \in \mathbb{R}^{10}$ the minimum, mean, median and maximum ratio of the other proposed bounds and the optimal bound.

Table I: Ratio between the $\|\gamma\|_1\|\delta\|_1$, (25), (26), LDL^T and XDX^T bounds and the optimal bound for 10^6 samples with $k = 10$.

	$\|\gamma\|_1\|\delta\|_1$	(25)	(26)	LDL^T	XDX^T
min	4.00	2.06	2.06	3.91	4.00
mean	7.50	3.82	3.67	6.96	6.10
median	6.69	3.38	3.22	6.15	5.28
max	70.65	36.69	35.58	70.13	60.77

In any case, practical experience suggests that the overall advantage is limited. If there is no correlation between γ and δ, i.e. they have no common error terms, then $\|\gamma\|_1\|\delta\|_1$ is sharp. Finally, for the product $\langle\, cd\,;\, c\delta + \gamma d, e\,\rangle$ only the term e is optimized; but $c\delta + \gamma d$ are dominant unless the midpoints c, d are relatively small compared to the radii γ, δ, respectively.

3.3 The Min-Range approximation

Let $f : \mathbb{R} \to \mathbb{R}$ be represented by $[\![p, q, \Delta]\!]$ on $\mathbf{X}$. The principle of the Min-Range approximation is $\{f(x) : x \in \mathbf{X}\} = \{px + q : x \in \mathbf{X}\}$. If $\mathbf{X}$ contains an extremum of f, then the only and optimal choice is the ordinary interval representation (9).

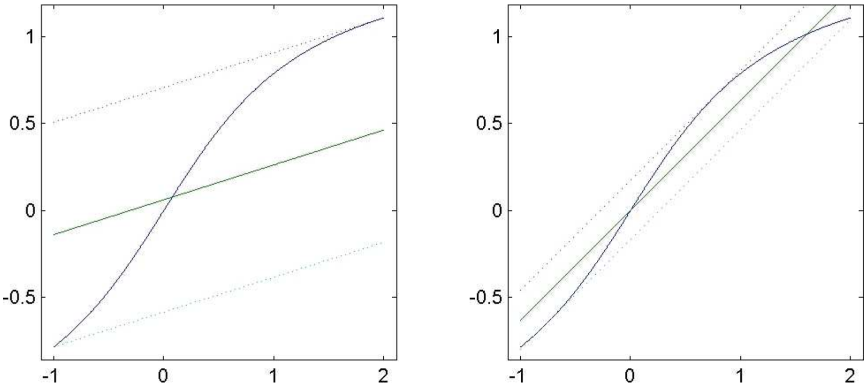

Fig. 3: Improved Min-Range and Chebyshev approximation of atan(x) on $\mathbf{X} = [-1, 2]$.

Suppose f is differentiable and $f'(x) \neq 0$ for all $x \in \mathbf{X}$. For all functions f listed at the beginning of this section the following is true: If f is defined on some $\mathbf{X}$, then there exists at most one $\xi \in \mathbf{X}$ with $f''(\xi) = 0$. With this property we have two classes of functions. For one of them optimal Min-Range approximations are easily computed.

Lemma 2 Let differentiable $f : \mathbf{X} \to \mathbb{R}$ be given and suppose there exists $\theta \in \mathbf{X}$ with $f'(\theta) \leq f'(x)$ for all $x \in \mathbf{X}$. Then f is represented by $[\![p, q, \Delta]\!]$ on $[a, b] = \mathbf{X}$ for

$$p := f'(\theta), \quad q := \frac{f(a) + f(b) - p(a+b)}{2} \quad \text{and} \quad \Delta := \frac{f(b) - f(a) + p(a-b)}{2}. \tag{28}$$

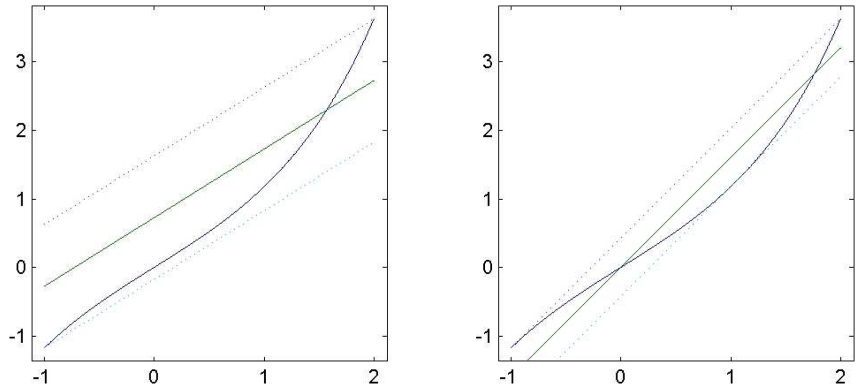

Fig. 4: Improved Min-Range and Chebyshev approximation of sinh(x) on $\mathbf{X} = [-1, 2]$.

If $f'(\theta) \geq f'(x)$ for all $x \in \mathbf{X}$, then f is represented by $[\![p, q, -\Delta]\!]$ on $\mathbf{X}$.

Proof. For all $x \in \mathbf{X}$ there exist $\zeta, \eta \in \mathbf{X}$ with $f(x) = f(a) + f'(\zeta)(x - a) = f(b) + f'(\eta)(x - b)$. First suppose $f'(\theta) \leq f'(x)$ for all $x \in \mathbf{X}$. Using $x - b \leq 0 \leq x - a$ it follows

$$px + q - f(x) \leq px + q - f(a) - p(x - a) = \frac{f(b) - f(a) + p(a - b)}{2} = \Delta,$$

and

$$px + q - f(x) \geq px + q - f(b) - p(x - b) = \frac{f(a) - f(b) + p(a - b)}{2} = -\Delta$$

which is (8). If $f'(\theta) \geq f'(x)$ for all $x \in \mathbf{X}$, we conclude $\Delta \leq px + q - f(x) \leq -\Delta$ for all $x \in \mathbf{X}$. ■

For $f \in \mathfrak{F}_1 := \{\mathrm{atan}, \tanh, \mathrm{asinh}, \mathrm{erf}, -\mathrm{erfc}\}$ and for $f \in \mathfrak{F}_2 := \{\tan, -\cot, \mathrm{asin}, -\mathrm{acos}, \sinh, \mathrm{atanh}\}$ with $a < 0 < b$ the function is neither convex nor concave on $[a, b]$, and the usual Min-Range approximation (10) is not applicable.

For $f \in \mathfrak{F}_1$ we may set $\theta := a$ if $b < -a$ and $\theta := b$ otherwise to satisfy the assumption of the lemma. This is an optimal Min-Range approximation because p is equal to f' in one of the endpoints. The situation is depicted in Figure 3.

For $f \in \mathfrak{F}_2$ we may set $\theta := 1$ to satisfy the assumption of the lemma. This is an improved but, in general, not optimal Min-Range approximation. The situation is depicted in Figure 4. An optimal Min-Range approximation may be computed, however, at the cost of solving a nonlinear equation involving transcendental functions.

In summary we have improved Min-Range approximations for the functions in $\mathfrak{F}_2$, and optimal Min-Range approximations for all other functions in (19).

3.4 The Chebyshev approximation

For functions convex or concave in $\mathbf{X}$ optimal Chebyshev approximations are as in Lemma 1. For functions with an extremum in $\mathbf{X}$ this approximation may be weak, see Figure 5. For the functions in (19) these are, besides $\sin(x)$ and $\cos(x)$, exactly $\mathrm{sqr}(x)$ and $\cosh(x)$.

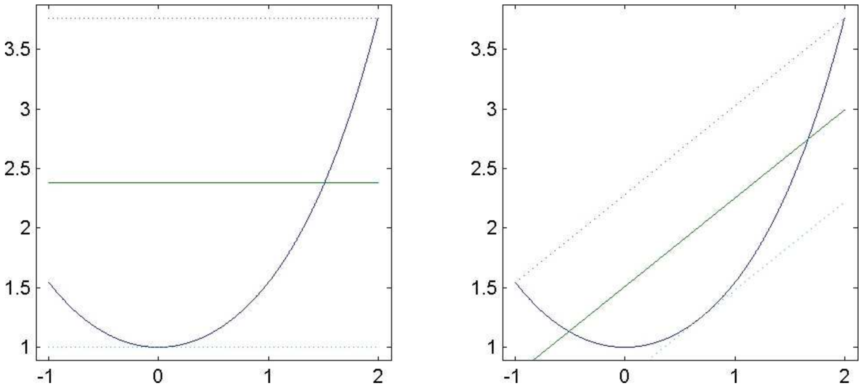

Fig. 5: Min-Range and Chebyshev approximation of cosh(x) on $\mathbf{X} = [-1, 2]$.

It remains to discuss the functions which are neither convex nor concave on $\mathbf{X}$. These are the functions in

$$\mathfrak{F} := \mathfrak{F}_1 \cup \mathfrak{F}_2 = \{\text{atan}, \text{tanh}, \text{asinh}, \text{erf}, \text{erfc}, \tan, \cot, \text{asin}, \text{acos}, \sinh, \text{atanh}\} \quad \text{with } \ a < 0 < b. \tag{29}$$

Note that the Chebyshev approximation by Lemma 1 is also optimal for functions in $f \in \mathfrak{F}$ if $\mathbf{X}$ contains no turning point of f.

Lemma 3 Let twice differentiable $f : \mathbf{X} \to \mathbb{R}$ be given with at most one root of f'' in $\mathbf{X} = [a, b]$. Assume $a \neq b$ and define

$$p := \frac{f(b) - f(a)}{b - a}. \tag{30}$$

If there is exactly one $\xi \in \mathbf{X}$ with $f'(\xi) = p$, then f is represented by $[\![p, q, \Delta]\!]$ on $[a, b] = \mathbf{X}$ with q and Δ as in (12). If $f'(\xi_1) = f'(\xi_2) = p$, then $[\![p, q, \Delta]\!]$ with

$$q := \frac{f(\xi_1) + f(\xi_2) - p(\xi_1 + \xi_2)}{2} \qquad \text{and} \qquad \Delta := \left| \frac{f(\xi_2) - f(\xi_1) - p(\xi_2 - \xi_1)}{2} \right|. \tag{31}$$

represents f on $\mathbf{X}$. If $f(-x) = -f(x)$ on $\mathbf{X}$, then

$$q = 0 \quad \text{and} \quad \Delta := |f(\xi_1) - p\xi_1|. \tag{32}$$

Proof. Define $g(x) := f(x) - f(\xi) - p(x - \xi)$. Since $g'' = f''$ has at most one root in $\mathbf{X}$, g has at most three roots in $\mathbf{X}$ counting multiplicities. By $g(\xi) = g'(\xi) = 0$ there exist at most one $\xi \neq \omega \in \mathbf{X}$ with $g(\omega) = 0$. To establish a contradiction, assume $a < \omega < b$, which means $g(a)g(b) < 0$. If $g(a) > 0$, then

$$f(b) - f(a) = f(b) - f(\xi) + f(\xi) - f(a) < p(b - \xi) - p(a - \xi) = p(b - a),$$

contradicting the definition (30) of p. The case $g(a) < 0$ leads to a contradiction as well, so that $g(x) \geq 0$ or $g(x) \leq 0$ for all $x \in \mathbf{X}$. If there is exactly one $\xi \in \mathbf{X}$ with $f'(\xi) = p$, then f is convex or concave, and we may proceed as from (13) on. Suppose $f'(\xi_1) = f'(\xi_2) = p$. The first part of the proof used only $f'(\xi) = p$. Thus it may be applied to $g_\nu(x) := f(x) - f(\xi_\nu) - p(x - \xi_\nu)$ for $\nu \in \{1, 2\}$, so that $g_1(x) \geq 0$ and $g_2(x) \leq 0$ for all $x \in \mathbf{X}$, or vice versa. With suitable numbering we obtain

$$f(\xi_1) + p(x - \xi_1) \leq f(x) \leq f(\xi_2) + p(x - \xi_2) \qquad \text{for all } \ x \in \mathbf{X}.$$

Hence

$$px + q - f(x) \leq p\xi_1 - f(\xi_1) + q = \frac{f(\xi_2) - f(\xi_1) - p(\xi_2 - \xi_1)}{2} = \Delta,$$

and the remaining cases follow similarly. ■

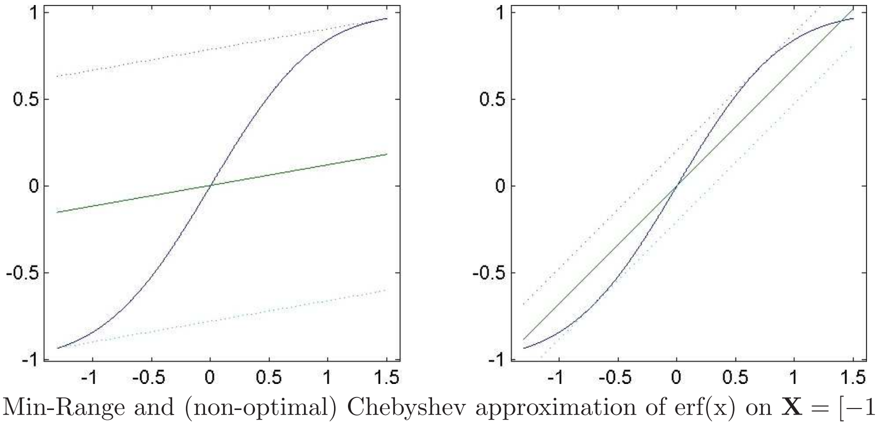

Fig. 6: Min-Range and (non-optimal) Chebyshev approximation of erf(x) on $\mathbf{X} = [-1.3, 1.5]$.

It is well-known [4] that a polynomial approximation of degree n of a nonlinear function is optimal in the Chebyshev sense if and only if the maximum error is attained at least $n+2$ times with alternating signs. The situation is depicted in the left of Figure 6. Hence the previous lemma delivers optimal Chebyshev approximations if there is one $\xi \in \mathbf{X}$ with $f'(\xi) = p$. However, approximations are, in general, not optimal if $f'(\xi_1) = f'(\xi_2) = p$. Optimal approximations might be computed also in that case using, for example, Remez' algorithm; however, this requires a considerable effort.

We mention that also a second order Taylor expansion

$$f(x) \in p(x - \mu) + (f(\mu) + \mathrm{mid}(E)) \pm \mathrm{rad}(E)$$

for $\mu = \mathrm{mid}(\mathbf{X})$ and $E := \mathrm{hull}(0.5f''(X)\mathrm{rad}(X)^2, 0)$ can be used. However, the expansions in Lemma 3 are superior. The same applies to a slope expansions [20].

We finally mention that due to the range component as described in Subsection 3.1, the range of the hyperbolic cosine cannot contain numbers strictly less than 1. Thus, for example, the left bound of the result of $\cosh([-1, 2])$ as in Figure 5 is 1.

3.5 Periodic functions

The tangent and cotangent are periodic, however, each interval of radius π or larger contains a pole. Thus no extra care is necessary. For the sine and cosine extra care is necessary. If the radius of the input interval is 2π or larger, then the ordinary interval approximation $\llbracket 0, 0, 1 \rrbracket$ is the only choice. Otherwise, the interval $\mathbf{X}$ can be transformed by some range reduction method into $[0, 4\pi]$. An accurate and very efficient method is described in [24] allowing to treat huge arguments without using a long arithmetic. This method is implemented in Matlab and for the interval standard functions in INTLAB.

There is a subtle point which needs some attention. The periodic functions under consideration are $\sin, \cos$, for which $f(x) = f(x - 2k\pi)$ and $\tan, \cot$, for which $f(x) = f(x - k\pi)$ for any $x \in \mathbb{R}$ and $k \in \mathbb{Z}$. If $\llbracket p, q, \Delta \rrbracket$ represents one of those function on $\mathbf{X}$, then also on $\mathbf{X} - 2k\pi$ or $\mathbf{X} - k\pi$, respectively. Accordingly, an affine quantity $\langle c, \gamma \rangle$ is to be replaced by $\langle c - 2k\pi, \gamma \rangle$ or $\langle c - k\pi, \gamma \rangle$, respectively. It follows that a possible rounding error in the computation of the transformed midpoint has to be taken care of.

Finally, various case distinctions are necessary, based on which appropriate Min-Range and Chebyshev approximations can be computed using the described methods. This is implemented in INTLAB.

3.6 Rounding errors

Up to now all results were presented in real arithmetic. In a computer implementation the inevitable rounding errors have to be handled. In the INTLAB implementation an $m \times n$ array of affine quantities is represented by a structure with components

$$\begin{array}{lll} \texttt{.mid} & \text{midpoint} & c \in \mathbb{F}^{m\times n} \\ \texttt{.err} & \text{error terms} & \gamma \in \mathbb{F}^{k\times mn} \\ \texttt{.rnderr} & \text{rounding errors} & \varrho \in \mathbb{F}^{1\times mn} \\ \texttt{.range} & \text{range} & \mathbf{X} \in \mathbb{IF}^{m\times n}. \end{array} \tag{33}$$

The structure is optimized for Matlab since access to a column of a (sparse) matrix is easily 100 times faster than to a row. The ij-th component is the affine quantity $\langle c_{ij}; \gamma_{\nu:} \rangle$ with associated range $\mathbf{X}_{ij}$ and rounding error term ϱ_ν, where $\nu := m(i - 1) + j$ denotes the linearized index of the $m \times n$-array. Each affine component comprises of k (correlated) error terms.

Concerning implementation, all operations in the definition of a new affine quantity are executed in interval arithmetic, then the midpoint is stored and the radius added to the error term. For example, INTLAB code for the sum C of two affine arrays A, B of the same size with the same number of error terms is as follows:

```
S = intval(A.mid) + B.mid;          % interval sum of midpoints
C.mid = S.mid;                      % midoint of sum
T = intval(A.err) + B.err;          % interval sum of error terms
C.err = T.mid;                      % midpoint of sum
```

```
   setround(+1)                          % set rounding upwards
   C.rnderr = A.rnderr + B.rnderr + S.rad(:)' + T.rad;
   C.range = A.range + B.range;        % interval sum of ranges
```

Rounding errors are added in rounding upwards so that the stored quantities are upper bounds, `S.rad(:)'` converts `S.rad` into a row vector. It follows that the inclusion property (3) is still satisfied. If A and B have a different number of error terms, a suitable number of zeros is appended. Rounding errors in the computation p, q, Δ for an approximation are treated appropriately.

Each definition of a new affine variable creates new error terms. Consequently, creating an $m \times n$ affine matrix creates mn new error terms. Therefore the component `.rnderr` is stored in a sparse array.

Although rounding errors are relatively small, they may accumulate and influence the quality of the result severely. This happens in particular in iterations, when a variable is used many times. To diminish that effect it is sometimes advisable to put rounding errors into a new error term. In Section 3.8 we show that this is mandatory for the Henon iteration; for other applications as the solution of linear systems it has no significant effect.

3.7 Reduction of error terms

In a larger computation the number of error terms may increase. Basically, there are two possible countermeasures. First, a garbage collection may be performed, that is all error terms which are zero for all active variables can be removed. Second, certain error terms may be collected and put into the rounding error component. The latter may potentially increase the diameter of the computed result.

For garbage collection the complete knowledge of all active variables is mandatory. However, this information is not available in Matlab, so there is no safe possibility to do that. Good strategies for the second possibility as in [13] need also information about all active variables. One might try to remove error terms with relatively small value, for example all γ_i with $|\gamma_i| \leq \varphi \|\gamma\|_\infty$ with a suitable constant φ. However, even absolutely small γ_i may play in important in later computations.

In any case, the number of error terms does not decrease by setting an individual error term to zero and moving its value into the rounding error component, only a garbage collection would do that.

3.8 Computational examples

The presented affine arithmetic is realized in a toolbox in Version 8 of INTLAB [26]. All subsequent examples refer to that implementation.

3.8.1 The dependency problem and the wrapping effect

A new data type `affari` creates affine variables. For example, the INTLAB code

```
   A = infsup(1,3); intDiff = A-A
   B = affari(A); intAff = B-B
```

produces the output

```
intval intDiff =
[   -2.0000,    2.0000]
affari intAff =
[    0.0000,    0.0000]
```

demonstrating how the dependency problem is avoided in affine arithmetic. In higher dimensions, the wrapping effect occurs. The reduction of the wrapping effect can be visualized in a 2-dimensional plot, see Figure 7. Consider

```
A = 0.5*[1 2;-1 1]
close, hold on
x = affari(infsup(-1,1)*ones(2,1));
```

```
plotaffari(x)
for i=1:7
  x = A*x;
  plotaffari(x)
end
```

The spectral radius of the matrix A is about $\sqrt{3}/2 \approx 0.866$, so that the iteration $x^{(k+1)} := Ax^{(k)}$ converges to the zero vector for every initial vector $x^{(0)} \in \mathbb{R}^2$. This can also be verified in affine arithmetic for which the code above produces the graph in Figure 7. Starting from the initial red box $[-1,1]^2$, the iterates depicted in green, blue, cyan, magenta, yellow, white and black, rotate and become smaller in size.

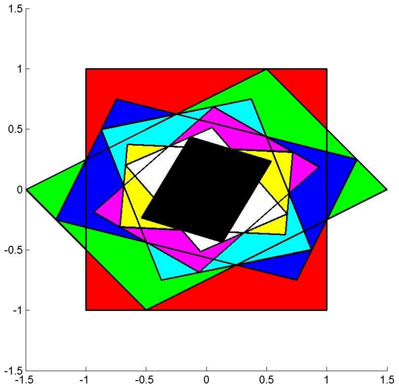

Fig. 7: Two dimensional interval iteration.

3.8.2 Transcendental functions

A similar one-dimensional example involving standard functions is

```
   f = vectorize(inline('sqr(log2(x+1))-x*cos(x)-x*atan(x)+cosh(x)'));
   X = infsup(0,1);
   ResIntval = f(X)
   ResAffari = f(affari(X))
```

with the result

```
intval ResIntval =
[   -0.7854,    2.5431]
affari ResAffari =
[    0.2866,    1.6962]
```

Note that the Chebyshev mode was used. In the Min-Range mode there is no difference between ordinary and affine arithmetic.

3.8.3 Data dependency

The use of affine arithmetic is particularly advantageous if in an expression intervals of small radius occur many times. Consider

$$(x-3)^8 = x^8 - 24\,x^7 + 252\,x^6 - 1512\,x^5 + 5670\,x^4 - 13608\,x^3 + 20412\,x^2 - 17496\,x + 6561$$

and evaluate the right hand side as an inline function f(x) using ordinary interval and affine arithmetic:

```
   x = midrad(4,1e-4);
   yint = f(x)
   affariinit('ApproxMinRange'),  yMinRange = f(affari(x))
   affariinit('ApproxChebyshev'), yChebyshev = f(affari(x))
```

The result is

```
intval yint =
[ -657.8345,  659.8345]
affari yMinRange =
[    0.9445,    1.0627]
affari yChebyshev =
[    0.9779,    1.0257]
```

with a more narrow result in the Chebyshev mode. In other examples, the result of the Min-Range mode may be a subset of the Chebyshev mode, so no approach is generally superior.

After some computations the number of error terms may increase. This creates objects of considerable complexity which is visualized in three dimensions in Figure 8. The graph was produced by the following code:

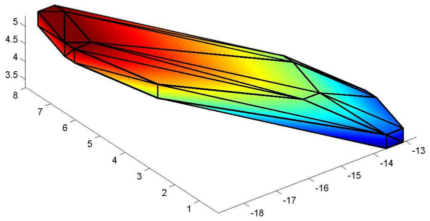

Fig. 8: Affine quantity in 3 dimensions.

```
   A = [ -4  1  2 ;  4  4 -3 ; 1  0 -1 ]
   X = [ infsup(0.5,1.5) ; infsup(-3.2,-2.8) ; infsup(-4.3,-3.7) ]
   close, plotaffari(A*affari(X)-sin(X))
```

3.8.4 Rounding errors

It turns out that often it is superior to put accumulated rounding errors into a separate error term. In INTLAB there are corresponding options which are switched by

`affariinit(’RoundingErrorsToErrorTerm’)` and `affariinit(’RoundingErrorsToRnderr’)`. (34)

The effect can be demonstrated by the following example [12]:

$$f(x) := x^2 - 2x \text{ and } g(x) := x(x+1)\left(\frac{1}{x} - \frac{1}{x+1}\right). \tag{35}$$

Obviously $f(g(x)) = -1$ for $x \in \mathbb{R}\backslash\{0,-1\}$. Following we evaluate $f(g(X))$ for $X = 10000 + [-\varepsilon, \varepsilon]$ in ordinary interval arithmetic and in affine arithmetic using the two options in (34). The radius of the result for ε varying between 1 and 10^{-12} is displayed in Figure 9.

The dotted line depicts the radius using ordinary interval arithmetic, the dashed line the results using the second option in (34), i.e. putting quadratic rounding errors into the `.rnderr` component and using Min-Range (black) and Chebyshev (red) approximation, and the solid line shows the results for putting quadratic rounding errors into an extra error term. Clearly, the latter with Chebyshev approximations is the best option.

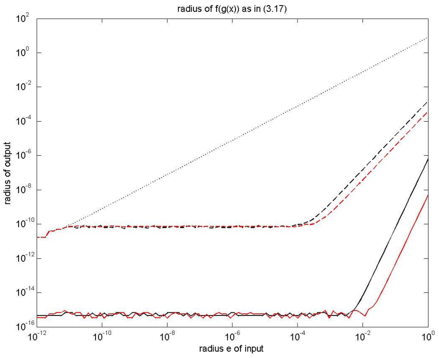

Fig. 9: Radius of f(g(X)) for f,g as in (35) and $X = 10000 + [-\varepsilon, \varepsilon]$.

3.8.5 The Henon map

Moreover, affine arithmetic is advantageous in interval iterations, exactly what should be avoided when using ordinary interval arithmetic. Following [13] we consider the Henon map

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 1 - ax^2 + y \\ bx \end{pmatrix} \tag{36}$$

for $a = 1.057$ and $b = 0.3$. It is known that the mapping is chaotic for $a = 1.06$ and $b = 0.3$. The starting vector is `midrad(0,1e-5)*ones(2,1)`. Ordinary interval arithmetic produces infinite intervals after 46 iterations, whereas for affine arithmetic the radii increase slightly and then decrease. In Figure 10 the radii for 500 iterations are displayed in linear and logarithmic scale. Note that we used the true input values $a = 1.057$ and $b = 0.3$ by INTLAB's built-in verified conversion `a=intval('1.057'); b=intval('0.3');`. Also note that in such type of iterations the radii of ordinary interval arithmetic iterates cannot decrease.

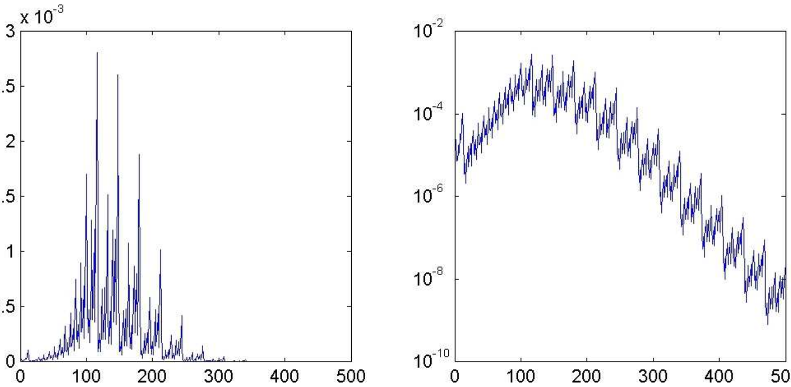

Fig. 10: Radii of Henon's iteration in affine arithmetic in linear and logarithmic scale.

For the Henon it is mandatory to put rounding errors into new error terms, otherwise the radii increase to infinity after 74 iterations.

3.8.6 Butterworth filter

The following iteration [3] occurs in the classical order 2 Butterworth filter:

$$y_{n+2} = \frac{2(c^2-1)y_{n+1} - (c^2 - \sqrt{2}c + 1)y_n + c^2 x_{n+2} - 2c^2 x_{n+1} + c^2 x_n}{c^2 + \sqrt{2}c + 1},$$

where $y_0 = y_1 = 1$ and the inputs x_n satisfy $x_n \in [-1, 1]$. With these assumptions y_n is always bounded. We choose random x uniformly distributed in $[-1, 1]$. As expected, the radii of the iterates grow exponentially using ordinary interval arithmetic. In affine arithmetic they stay at a constant level, also for many more iterations (see Figure 11 for $c = 10$ and 100 iterations). As for the Henon iteration it is mandatory to store quadratic terms in an extra error term.

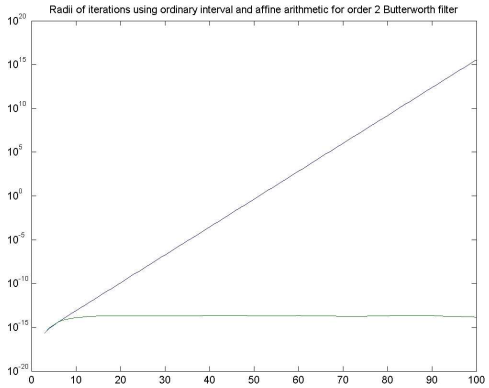

Fig. 11: Radii of Butterworth filter iteration in ordinary and affine arithmetic.

3.8.7 Linear systems

Next we consider systems of linear equations. Define and solve a randomly generated system with tolerances by

```
   n = 20;
   A = randn(n).*midrad(1,1e-8);
   b = randn(n,1); affariinit
   tic, Xint = solvewpp(A,b); Tint = toc
   tic, Xaff = solvewpp(affari(A),b); Taff = toc
   [ Xint(1:4) Xaff(1:4) ]
   med = median(rad(Xint)./rad(Xaff))
```

The linear system is solved by naive Gaussian elimination with interval forward and backward elimination. Usually this is not the method of choice, for an analysis cf. [29], but here we are interested in the wrapping effect. The left and right column display the first four components of the result by ordinary interval and by affine interval arithmetic.

```
Tint =
   0.072799030655320
Taff =
   1.583216680829170
affari ans =
  1.0e+002 *
[  -2.02125659088592,   2.10253898630219] [   0.04059673271195,   0.04059732851923]
[  -2.51697951401905,   2.51383456826459] [  -0.00156756062349,  -0.00156694663514]
[  -2.18514014758802,   2.12576643449358] [  -0.02965666475400,  -0.02965607418252]
[  -0.83833589632967,   0.83397985233406] [  -0.00218306713056,  -0.00218285618838]
med =
    2.021639928709179e+006
```

As can be seen ordinary interval arithmetic produces zero intervals whereas affine arithmetic can verify about 6 decimal places. In turn, interpretation overhead leads to considerably larger computing time. In the median the radii by affine arithmetic are more narrow by some 6 orders of magnitude. As mentioned at the end of last section, in this example there is no difference whether to store rounding errors in extra error terms or not.

3.8.8 Nonlinear functions and global optimization

The next example is the verification of the existence of a local minimum of a nonlinear function. We use Branin's test function [6], which is well-known in global optimization:

$$f(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)\cos(x_1) + s$$

$$\text{with } \; a = 1, b = 5.1/(4\pi^2), c = 5/\pi, r = 6, s = 10, t = 1/(8\pi).$$

Consider the commands

```
X = verifynlss(f,[9;2],'h')     % inclusion of a stationary point
Y = f(hessianinit(X));          % inclusion of all Hessians f''(x) for x in X
LocalMin = isspd(Y.hx)          % verification of positive definiteness
```

The extra parameter `'h'` in `verifynlss` ensures that $f'(x) = 0$ is solved. It is first verified that `X` contains a zero of the nonlinear system $f'(x) = 0$, and second that all Hessian matrices $f''(x)$ for $x \in$ `X` are positive definite, in particular the Hessian at the stationary point. Consequently, `X` contains a strict local minimum of f.

Some verification algorithms for global optimization sometimes use the the so-called *expansion technique* [11]. Assume the interval vector $\mathbf{X}$ contains exactly one local minimum of $f : \mathbb{R}^n \to \mathbb{R}$. If it can be verified that in $\mathbf{Y} \in \mathbb{IR}^n$ with $\mathbf{X} \subseteq \mathbf{Y}$ there is also exactly on local minimum, then there cannot be a minimum in $\mathbf{Y} \backslash \mathbf{X}$. We simulate this by expanding the computed box $\mathbf{X}$ in the previous example:

```
Res = [];
for r=1.05:0.05:1.25
  XX = X+midrad(0,r);
  Yint = f(hessianinit(XX)); IntLocalMin = isspd(Yint.hx);
  Yaff = f(affari(hessianinit(XX))); AffLocalMin = isspd(Yaff.hx);
  Res = [ Res ; r IntLocalMin AffLocalMin ];
end
Res
```

The verification of positive definiteness uses INTLAB's algorithm `isspd` which is based on [27]. Note that only the midpoint and the radius of the matrix are used, thus it works for affine matrices as well. The above code produces

```
Res =
    1.0500    1.0000    1.0000
    1.1000         0    1.0000
    1.1500         0    1.0000
    1.2000         0    1.0000
    1.2500         0         0
```

As can be seen affine arithmetic can verify positive definiteness up to radius 1.2, whereas ordinary interval arithmetic fails for radius 1.1.

3.8.9 Verified Mandelbrot and Julia sets

The final example is the computation of a verified inclusion of a Julia set. We all know the beautiful pictures of Mandelbrot and Julia sets. However, until [22] no picture was known with a rigorous treatment of rounding errors. Given the iteration

$$z_0, c \in \mathbb{C} : \qquad z_{k+1} := z_k^2 + c \quad \text{for } k \geq 1, \tag{37}$$

it is well-known that infinity is a point of attraction if an iterate leaves the complex circle C with radius 2. For each starting point z_0, infinity is a point of attraction or not, and the boundary between the two sets is the Julia set with respect to c. Moreover, the set of $z \in \mathbb{C}$ for which the iteration stays in C for starting point $z_0 = 0$ is the Mandelbrot set.

Suppose c is replaced for $z_0 := 0$ by a rectangle $\mathcal{R}$ and all operations are performed in interval arithmetic, ordinary or affine. If an iterate has empty intersection with C, then all points in $\mathcal{R}$ diverge and no point belongs to the Mandelbrot set. If a certain iterate z_k is finite and is completely enclosed in a previous iterate z_{k_0}, then further iterates cannot leave the union of the iterates $\{z_\kappa : k_0 \leq \kappa \leq k\}$ and the entire rectangle $\mathcal{R}$ belongs to the Mandelbrot set. If neither one is true, the rectangle is bisected. A similar method is used in [22] to identify the Julia set. In Figure 12 the area in black is verified to belong to the Mandelbrot set, the red area definitely does not belong to it, and for the yellow region it could not be decided.

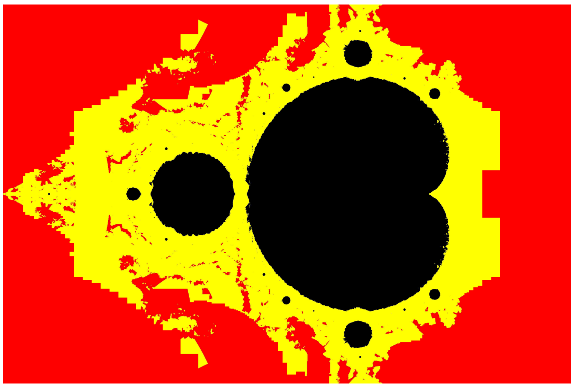

Fig. 12: Mandelbrot set of the iteration $z \mapsto z^2 + c$ for $c \in [-2, 1] \times [-j, j]$ for $j = \sqrt{-1}$.

A well-studied example for the Julia set is $c = -1$. In this case the starting point 0 produces a loop (of length 2) implying that c is in the Mandelbrot set of the iteration, which in turn implies that the Julia set is connected. If therefore the iteration stays in C for all four edges of a rectangle $\mathcal{R} \subset \mathbb{C}$, then this is true for every point in $\mathcal{R}$. Conversely, if for all four edges some iterate is completely outside C, then infinity is a point of attraction for all points in $\mathcal{R}$.

As before we can identify boxes belonging to interior of the Julia set and color it black, as well that they belong to the exterior and color it red. To identify boxes for which neither can be decided, we use the following. If the entire circle C is enclosed in some iterate z_k, then it cannot be decided which points in z_0, if any, are convergent or divergent, and the box is bisected. If this remains true until a certain maximum bisection depth, then the box is colored yellow. The described method improves upon the method used in [22].

In Figure 13 the result of this approach for the Julia set with a maximum bisection depth 11 is displayed. Note that black and red areas are verified to be bounded and divergent, respectively. It follows that the Julia set is enclosed in the yellow area. In some way the chaotic behavior of the dynamics of the iteration is visible.

Using affine arithmetic, one of the three criteria convergent, divergent, unknown, is satisfied for small values of k. In the displayed example the maximum value for proving boundedness was $k = 9$, for proving divergence $k = 16$, and the circle C was included in an iterate for a maximal value of $k = 22$.

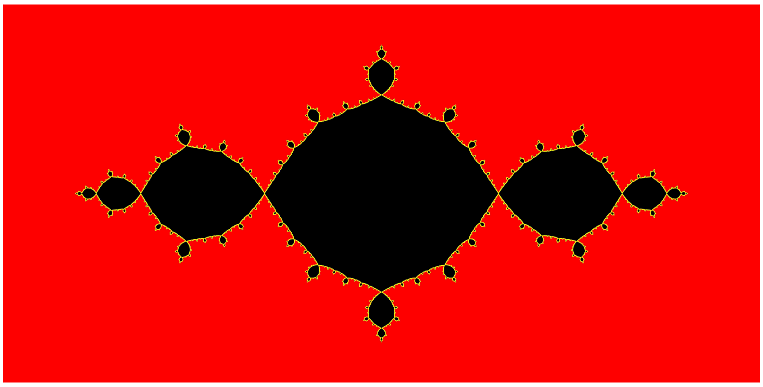

Fig. 13: Julia set of the iteration $z \mapsto z^2 + c$ for $c = -1$.

The affine arithmetic toolbox `affari` is included from INTLAB's Version 8.

4. Acknowledgment

The authors wish to thank Florian Bünger, Luiz Henrique de Figueiredo, Eric Goubault, Christian Jansson, Marko Lange, Kai Torben Ohlhus and Sylvie Putot for their fruitful and constructive comments. Moreover, our thanks to two anonymous referees for their thorough reading and constructive comments.

References

[1] M.V.A. Andrade, J.L.D. Comba, and J. Stolfi. Affine Arithmetic. Presented at INTERVAL'94, 1994.

[2] O. Beaumont. Solving interval linear systems with oblique boxes. In *SCAN'98*, 1998.

[3] O. Bouissou, E. Goubault, J. Goubault-Larrecq, and S. Putot. A generalization of p-boxes to affine arithmetic. *Computing*, 94(2-4):180–201, 2012.

[4] L. Collatz and W. Krabs. *Approximationstheorie. Tschebyscheffsche Approximation mit Anwendungen.* Teubner, Stuttgart, 1973.

[5] J. L. D. Comba and J. Stolfi. Affine arithmetic and its applications to computer graphics. Proc. SIBGRAPI'93 - VI Simposio Brasileiro de Computacao Grafica e Processamento de Imagens (Recife, Brazil), 9–18, 1993.

[6] L.C.W. Dixon and G.P. Szegö (eds.). *Towards Global Optimization 2.* North-Holland, Amsterdam, 1978.

[7] L.H. de Figueiredo and J. Stolfi. *Self-Validated Numerical Methods and Applications.* Brazilian Mathematics Colloquium monograph, IMPA, 1997.

[8] L.H. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37(1-4):147–158, 2004.

[9] E. Goubault. private communication, 2014.

[10] E.R. Hansen. A generalized interval arithmetic. In K. Nickel, editor, *Interval Mathematics*, volume 29, pages 7–18. Springer, 1975.

[11] C. Jansson. On Self-Validating Methods for Optimization Problems. In J. Herzberger, editor, *Topics in Validated Computations — Studies in Computational Mathematics 5*, pages 381–438, North-Holland, Amsterdam, 1994.

[12] M. Kashiwagi. About affine arithmetic (in japanese).

[13] M. Kashiwagi. An algorithm to reduce the number of dummy variables in affine arithmetic. In *SCAN conference*, Novosibirsk, 2012.

[14] R.B. Kearfott, M.T. Nakao, A. Neumaier, S.M. Rump, S.P. Shary, and P. van Hentenfyck. Standardized notation in interval analysis. In *Interval analysis*, volume 4, pages 106–113, 2005.

[15] V. Kreinovich, A. Neumaier, and G. Xiang. Towards a Combination of Interval and Ellipsoid Uncertainty. *Vych. Techn. (Computational Technologies)*, 13:5–16, 2008.

[16] S. Miyajima, T. Miyata, and M. Kashiwagi. On the Best Multiplication of the Affine Arithmetic. *Institute of Electronics, Information and Communication Engineers*, J86-A(2):150–159, 2003.

[17] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction To Interval Analysis.* Cambridge Uni Press (CUP), 2009.

[18] R.E. Moore. *Interval Arithmetic and Automatic Error Analysis in Digital Computing.* Dissertation, Stanford University, 1963.

[19] R.E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1966.

[20] A. Neumaier. *Interval Methods for Systems of Equations.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1990.

[21] A. Neumaier. The wrapping effect, ellipsoid arithmetic, stability and confidence regions. *Computing Supplementum*, 9:175–190, 1993.

[22] J.B.S. Oliveira and L.H. de Figueiredo. Images of Julia sets you can trust, SCAN conference Lyon, 2002. `http://www.tecgraf.puc-rio.br/~lhf/ftp/doc/oral/scan2002.pdf`.

[23] A. Ovseevich and F. Chernousko. On optimal ellipsoids approximating reachable sets. *Problems of Control and Information Theory*, 16:125–134, 1987.

[24] M. Payne and R. Hanek. Radian Reduction for Trigonometric Functions. *SIGNUM Newsletter*, 18:19–24, 1983.

[25] S.M. Rump. Fast and parallel interval arithmetic. *BIT Numerical Mathematics*, 39(3):539–560, 1999.

[26] S.M. Rump. INTLAB - INTerval LABoratory. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[27] S.M. Rump. Verification of Positive Definiteness. *BIT Numerical Mathematics*, 46:433–452, 2006.

[28] S.M. Rump. Verification methods: Rigorous results using floating-point arithmetic. *Acta Numerica*, 19:287–449, 2010.

[29] S.M. Rump. Error estimation of floating-point summation and dot product. *BIT Numerical Mathematics*, 52(1):201–220, 2012.

[30] J. Stolfi. C-library for affine arithmetic. `http://www.ic.unicamp.br/~stolfi/EXPORT/software/c/Index.html/libaa`.

[31] T. Sunaga. Geometry of Numerals. Master's thesis, University of Tokyo, February 1956.