Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-1 (2019) 574–579

Interval Superposition Arithmetic for Guaranteed Parameter Estimation

Junyan Su * Yanlin Zha * Kai Wang * Mario E. Villanueva * Radoslav Paulen ** Boris Houska *

* School of Information Science and Technology, ShanghaiTech University, Shanghai, China. ({sujy,zhayl,wangk,meduardov, borish}@shanghaitech.edu.cn)
** Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Bratislava, Slovakia. (radoslav.paulen@stuba.sk)

Abstract: The problem of guaranteed parameter estimation (GPE) consists in enclosing the set of all possible parameter values, such that the model predictions match the corresponding measurements within prescribed error bounds. One of the bottlenecks in GPE algorithms, commonly exploiting set inversion, is the construction of enclosures for the image-set of factorable functions. In this paper, we introduce a novel set-based computing method called interval superposition arithmetics (ISA) for the construction of enclosures of such image sets and its use in GPE algorithms. The main benefits of using ISA in the context of GPE lie in the improvement of enclosure accuracy and in the implied reduction of the number of set-membership tests in the set-inversion algorithm.

Keywords: Set Arithmetics, Interval arithmetics, Guaranteed Parameter Estimation.

1. INTRODUCTION

In science and engineering, the behavior of processes and systems is often described using a mathematical model. Mathematical model development often follows three steps: model structure specification, design (and realization) of experiments, and estimation of unknown model parameters (Franceschini and Macchietto, 2008). In the last step, parameters are sought for which the model outputs match the available measurements (Ljung, 1999).

One way of addressing the parameter estimation problem is the use of set-membership estimation (Schweppe, 1968), also called guaranteed parameter estimation (GPE). The GPE problem can be formulated as an identification of the set of all possible model parameter values which are not falsified by the plant measurements, within some prescribed error bounds. A set-inversion algorithm (e.g. SIVIA by Jaulin and Walter, 1993) can be applied to find such set for nonlinear models. Here, the parameter set is successively partitioned into smaller boxes and using exclusion tests some of these boxes are eliminated, until a desired approximation is achieved. Since its advent, GPE has found various applications (see e.g., Marco et al., 2000; Jaulin et al., 2002; Lin and Stadtherr, 2007; Hast et al., 2015; Paulen et al., 2016).

An important computational aspect of SIVIA is that its complexity is proportional to the tightness of the interval enclosures. Thus, considerable effort has been invested into the development of different set-arithmetics to produce tighter enclosures of the image-set of factorable functions, for example Taylor (Makino and Berz, 1996) (TMA) and Chebyshev model arithmetics (CMA) (Battles and Trefethen, 2004; Rajyaguru et al., 2017) (see also Paulen et al., 2016, for their application to GPE).

Here, we propose an attempt to improve GPE algorithms using a novel nonconvex set-arithmetic called Interval Superposition Arithmetic (ISA). This arithmetic operates over Interval Superposition models (ISM), representing a piecewise constant enclosure over a grid of the domain. Unlike a naive application of interval arithmetic (IA) over the grid, the computational and storage complexity of ISA is polynomial. Furthermore, it is able to exploit separable structures in the computational graph of a factorable function. Finally, the remainder bounds in ISA (unlike those in TMA and CMA) are based on globally valid algebraic relations—namely, addition theorems. As a result, ISMs are tighter than Taylor models—at least over large domains.

The rest of the paper is organized as follows, Section 2 reviews GPE and set inversion. Section 3 presents an overview of ISA. An algorithm for intersecting ISMs with an interval—which forms the basis for a set-inversion algorithm—is presented in Section 4. It is important to notice that the intersection algorithm runs in polynomial time, but the complexity of computing an arbitrarily close approximation of the parameter set is exponential. The application of the proposed algorithm to a simple case study is shown in Section 5. Section 6 concludes the paper.

Notation The set of real valued compact interval vectors is denoted by $\mathbb{I}^n = \{[a,b] \subset \mathbb{R}^n \mid a,b \in \mathbb{R}^n,\ a \leq b\}$. Let $I = [a,b] \in \mathbb{I}$ and $c \in \mathbb{R}$, $c + I = I + c$ we have $[a+c, b+c]$. Similarly, $cI = Ic$ denotes $[ca, cb]$ if $c \geq 0$ ($[cb, ca]$ if $c < 0$).

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.06.124

The diameter of I is denoted by $\mathrm{diam}(I) = b - a$. Interval operations are evaluated by IA (Moore et al., 2009), e.g.,

$$[a,b] + [c,d] = [a+b, c+d] \ ,$$
$$[a,b] * [c,d] = [\min\{ac,ad,bc,bd\}, \max\{ac,ad,bc,bd\}]$$
$$\exp([a,b]) = [\exp(a), \exp(b)]$$

2. GUARANTEED PARAMETER ESTIMATION

We consider a system represented by the algebraic model

$$y = f(x) \ . \tag{1}$$

Here, $x \in \mathbb{R}^{n_x}$ denotes unknown parameter while $y \in \mathbb{R}^{n_y}$ the (observed) output variables. The model is described by the, possibly nonlinear, function $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$.

Given $n_m \in \mathbb{N}$ measurements, $y_1^{\mathrm{m}}, \ldots, y_{n_m}^{\mathrm{m}} \in \mathbb{R}^{n_y}$, the GPE paradigm works under the assumption that the true system outputs $y_1^\circ, \ldots, y_N^\circ$ can be observed only within some bounded measurement bounds. Thus, for each $i \in \{1, \ldots, n_m\}$, we have

$$y_i^\circ \in y_i^{\mathrm{m}} + [-\eta_i, \eta_i] =: Y_i \in \mathbb{I}^{n_y} \ , \tag{2}$$

with $\eta_1, \ldots, \eta_{n_m} \geq 0$. The aim of GPE is to compute the set

$$X_{\mathrm{e}} := \{x \in X_0 \mid \forall i \in \{1, \ldots, N\} : f(x) \in Y_i\} \ , \tag{3}$$

i.e., the set of parameters (within some admissible domain $X_0 \in \mathbb{I}^{n_x}$) for which the model outputs are consistent with all the uncertain observations Y_i.

Computing (3) requires intersecting the preimage of Y_i under f, with the initial parameter domain, i.e.,

$$X_{\mathrm{e}} = \left(\bigcap_{i=1}^{n_m} f^{-1}(Y_i) \right) \cap X_0 \ . \tag{4}$$

This problem is intractable, in all but the simplest cases, and thus one has to settle for approximations of this set. State-of-the-art algorithms for set inversion provide inner ($\mathbb{X}_{\mathrm{int}}$) and boundary ($\mathbb{X}_{\mathrm{bnd}}$) subpavings, i.e. lists of non overlapping interval vectors, satisfying

$$\bigcup_{X \in \mathbb{X}_{\mathrm{int}}} X \subseteq X_{\mathrm{e}} \subseteq \left(\bigcup_{X \in \mathbb{X}_{\mathrm{int}}} X \right) \cup \left(\bigcup_{X \in \mathbb{X}_{\mathrm{bnd}}} X \right) \ . \tag{5}$$

In a nutshell, these algorithms work by subdividing the parameter domain X_0 into smaller boxes such that $X_0 = \bigcup_j X_j$. Set arithmetics are then used to construct enclosures of f on X_j, i.e. sets $\overline{Y}_j \subset \mathbb{R}^{n_y}$ satisfying

$$\overline{Y}_j \supseteq \{f(x) \mid x \in X_j\} \ . \tag{6}$$

Using the information provided by the enclosure Y_j, the following set membership tests can be performed to classify the parameter boxes X_j as interior or boundary boxes:

(1) If $\overline{Y}_j \subseteq Y_i$ for all $i \in \{1, \ldots, n_m\}$, $X_j \in \mathbb{X}_{\mathrm{int}}$.
(2) Else, if $Y_i \cap f(X) = \emptyset$ for some $i \in \{1, \ldots, n_m\}$, $X_j \cap X_{\mathrm{e}} = \emptyset$.
(3) Else, $X \in \mathbb{X}_{\mathrm{bnd}}$.

Figure 1 shows the result of the above process for the function $f = x_1^3 + x_2^3$ over $X_0 = [-3,3]^2$, with $Y = [-2,2]$. The set X_0 has been divided into $N = 20$ equidistant pieces along each coordinate, resulting in 400 interval vectors X_j. The plot shows the set $\bigcup_{i=1}^{N^{n_x}} \left(X_j \times \overline{Y}_j\right)$, and its projection onto the (x_1, x_2)-space. The red and blue boxes belong to $\mathbb{X}_{\mathrm{int}}$ and $\mathbb{X}_{\mathrm{bnd}}$, respectively.

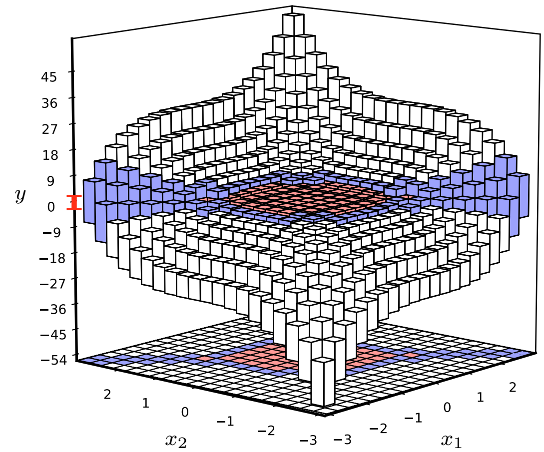

Fig. 1. Graph of an enclosure of $f = x_1^3 + x_2^3$ over $X_0 = [-3,3]^2$ (gridded using $N = 20$ subintervals at each coordinate). The sets $\mathbb{X}_{\mathrm{bnd}}$ (blue) and $\mathbb{X}_{\mathrm{int}}$ (red) were computed using $Y = [-2,2]$.

In practice, the domain X_0 is subdivided iteratively by bisecting boundary boxes, starting with $\mathbb{X}_{\mathrm{bnd}} = X_0$ and $\mathbb{X}_{\mathrm{int}} = \emptyset$. The bounding, set-membership, and bisection operations are repeated until a termination criterion, e.g.

$$\forall X \in \mathbb{X}_{\mathrm{bnd}}, \quad \mathrm{diam}(X) \leq \epsilon \ , \tag{7}$$

is met for a user-defined tolerance $\epsilon > 0$.

One of the bottlenecks of set inversion algorithms is the over-conservatism of existing set-arithmetics, particularly over large domains. Hence we propose to approach this problem within a novel set-arithmetics paradigm.

3. INTERVAL SUPERPOSITION ARITHMETIC

Interval superposition arithmetic is a novel method for enclosing the image of nonlinear factorable functions. It operates by propagating nonconvex sets, called interval superposition models, through computational graph of the function. Unlike Taylor and Chebyshev models—which require derivative information in order to compute their coefficients or remainder bounds—ISA does not rely on local approximation methods. Instead, it uses global algebraic relations—such as addition theorems—as well as partially separable structures within the function.

3.1 Interval superposition models

Consider an interval domain $X = [\underline{x}_1, \overline{x}_1] \times \ldots \times [\underline{x}_{n_x}, \overline{x}_{n_x}]$. Now, take a partition of X into intervals of the form

$$X_i^j = [\underline{x}_i + (j-1)h_i, \underline{x}_i + jh_i] \quad \text{with} \quad h_i = \frac{\overline{x}_i - \underline{x}_i}{N} \ , \tag{8}$$

for all $i \in \{1, \ldots, n_x\}$ and all $j \in \{1, \ldots, N\}$, with N being a user-specified integer. An interval superposition model of a real-valued function $f : \mathbb{R}^{n_x} \to \mathbb{R}$ on X is an interval valued function $\Gamma : X \times \mathbb{I}^{n_x \times N} \times \mathbb{I}^{n_x} \to \mathbb{I}$, given by

$$\Gamma(x, A, X) = \sum_{i=1}^{n_x} \sum_{j=1}^{N} A_i^j \varphi_i^j(x) \ , \tag{9}$$

with

$$\varphi_i^j(x) = \begin{cases} 1 & \text{if } x_i \in X_i^j, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Here, $A_i^j = \left[\underline{A}_j^i, \overline{A}_j^i\right]$ are the components of a matrix

$$A = \begin{pmatrix} A_1^1 & \dots & A_1^N \\ \vdots & \ddots & \vdots \\ A_{n_x}^1 & \dots & A_{n_x}^N \end{pmatrix} \in \mathbb{I}^{n_x \times N} , \quad (11)$$

which, for a fixed X, completely determines the enclosure function of f. Note that ISMs for functions $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are defined by stacking ISMs for each f_i. The matrix A is constructed such that $\Gamma(\cdot, A, X)$ is a piecewise constant enclosure function of f over X, i.e.

$$\forall x \in X, \quad f(x) \in \Gamma(x, A, X) . \quad (12)$$

The name *interval superposition model* is motivated by the structure of the enclosure: At any $x \in X_1^j \times \ldots \times X_{n_x}^j$, the interval $\overline{Y} = \Gamma(x, A, X)$ is given by the Minkowski sum (or superposition) of n_x interval functions $\sum_{j=1}^N A_i^j \varphi_i^j(x)$. The separable structure of ISMs allows for a storage complexity of order $\mathbf{O}(n_x N)$, since only $n_x N$ intervals need to be stored, in the matrix A, to represent the N^{n_x} pieces of the enclosure. In Figure 1, the graph of an ISM is shown over a partition of X (with $N = 20$). Although this set consists of 400 interval vectors (shown in red, white and blue), only 40 intervals are stored in the matrix A.

This separability also allows for the global minima and maxima of $\Gamma(\cdot, A, X)$ over X,

$$\lambda(A) = \sum_{i=1}^{n_x} \underbrace{\min_{j \in \{1,\ldots,N\}} \underline{A}_i^j}_{=:L(A_i)} \text{ and } \mu(A) = \sum_{i=1}^{n_x} \underbrace{\max_{j \in \{1,\ldots,N\}} \overline{A}_i^j}_{=:U(A_i)} ,$$

to be computed with a complexity of order $\mathbf{O}(n_x N)$. The interval $[\lambda(A), \mu(A)]$ denotes the range of ISM.

3.2 Arithmetic rules for interval superposition models

Interval superposition arithmetics propagates ISMs through the computational graph of a factorable function, defined by a finite recursive composition of atom operations from a finite library $\mathcal{L} = \{\exp, \sin, +, *, \ldots\}$.

Consider the functions $g, h : \mathbb{R}^{n_x} \to \mathbb{R}$, and a (possibly bivariate) atom operation α. Let the interval matrices $A, B \in \mathbb{I}^{n_x \times N}$ be the respective parameters for ISMs of g and h over X. In ISA, a univariate composition rule is a map taking A as an input and returning an interval matrix $C \in \mathbb{I}^{n_x \times N}$ parameterizing an ISM such that

$$\forall x \in \mathbb{X}, \quad (\alpha \circ g)(x) \in \Gamma(x, C, X) .$$

Here, $\alpha \circ g$ denotes the composition of α and g.

Bivariate composition rules in ISA are defined analogously, with the map taking both A and B as inputs. Although such maps are specific for each atom operation α, the main steps are outlined in Algorithms 1 and 2 for univariate compositions and bivariate products, respectively. The addition rule in interval superposition arithmetic is simple. An interval superposition model of $g + h$ on X is parameterized by the matrix $C = A + B$, with the sum computed componentwise using interval arithmetics.

***Theorem** 1.* Let $\Gamma(x, A, X)$ and be an ISM of g on X. If the matrix $C \in \mathbb{I}^{n_x \times N}$ is computed using Algorithm 1, then $\Gamma(x, C, X)$ is an ISM of $\alpha \circ g$ on X.

Algorithm 1. Composition rule of interval superposition arithmetic

Input: Matrix $A \in \mathbb{I}^{n_x \times N}$ parameterizing $\Gamma(\cdot, A, X)$ for g and an atom operation α.

Main Steps:

(1) Choose, for all $i \in \{1, \ldots, n_x\}$, central points $a_i \in \mathbb{R}$ satisfying

$$L(A_i) \le a_i \le U(A_i) \quad \text{and set} \quad \omega = \sum_{i=1}^{n_x} a_i .$$

(2) Choose a suitable remainder bound $r_\alpha(A) \ge 0$ such that

$$\left| \sum_{i=1}^{n_x} \alpha(\omega + \delta_i) - (n_x - 1)\alpha(\omega) - \alpha\left(\omega + \sum_{i=1}^{n_x} \delta_i\right) \right| \le r_\alpha(A)$$

for all $\delta \in \mathbb{R}^{n_x}$ with $\forall i \in \{1, \ldots, n_x\}$, $L(A_i) \le a_i + \delta_i \le U(A_i)$.

(3) Compute the interval valued coefficients

$$C_i^j = \alpha\left(\omega - a_i + A_i^j\right) - \frac{n_x - 1}{n_x} \alpha(\omega) .$$

for all $i \in \{1, \ldots, n_x\}$ and all $j \in \{1, \ldots, N\}$, where $\alpha\left(\omega - a_i + A_i^j\right)$ is evaluated in interval arithmetic.

(4) Set $C_k^j \leftarrow C_k^j + r_\alpha(A) \cdot [-1, 1]$ for all $j \in \{1, \ldots, N\}$ with

$$k \in \underset{i \in \{1,\ldots,n_x\}}{\operatorname{argmax}} \sum_{j=1}^N \overline{A}_i^j - \underline{A}_i^j$$

.

Output: Matrix $C \in \mathbb{I}^{n_x \times N}$ parameterizing $\Gamma(\cdot, C, X)$ for $\alpha \circ g$.

Algorithm 2. Product rule of interval superposition arithmetic

Input: Matrices $A, B \in \mathbb{I}^{n_x \times N}$ parameterizing and $\Gamma(\cdot, A, X)$ and $\Gamma(\cdot, B, X)$ for h and g, respectively.

Main Steps:

(1) Compute the central points, $\forall i \in \{1, \ldots, n_x\}$

$$a_i = \frac{\mathrm{U}(A_i) + \mathrm{L}(A_i)}{2} \quad \text{and} \quad b_i = \frac{\mathrm{U}(B_i) + \mathrm{L}(B_i)}{2}$$

then set

$$a = \sum_{i=1}^{n_x} a_i , \quad b = \sum_{i=1}^{n} b_i , \quad c = \sum_{i=1}^{n_x} a_i b_i , \quad \text{and} \quad \omega = \frac{ab - c}{n_x} .$$

(2) Compute $\rho_i(A) = \frac{U(A_i) - L(A_i)}{2}$ and $\rho_i(B) = \frac{U(B_i) - L(B_i)}{2}$ for all $i \in \{1, \ldots, n_x\}$ as well as the associated remainder bound

$$R(A, B) = \left(\sum_{i=1}^{n_x} \rho_i(A)\right)\left(\sum_{i=1}^{n_x} \rho_i(B)\right) - \sum_{i=1}^{n_x} \rho_i(A)\rho_i(B) .$$

(3) Compute, for each $i \in \{1, \ldots, n_x\}$ and all $j \in \{1, \ldots, N\}$

$$C_i^j = \left(A_i^j + a - a_i\right)\left(B_i^j + b - b_i\right) - (a - a_i)(b - b_i) - \omega .$$

(4) Set $C_k^j \leftarrow C_k^j + R(A, B) \cdot [-1, 1]$ for all $j \in \{1, \ldots, N\}$ with

$$k \in \underset{i \in \{1,\ldots,n_x\}}{\operatorname{argmax}} \sum_{j=1}^N \overline{A}_i^j - \underline{A}_i^j$$

.

Output: Matrix $C \in \mathbb{I}^{n_x \times N}$ parameterizing $\Gamma(\cdot, C, X)$, for $g * h$.

Proof. Consider an arbitrary point x in the domain X. Since $\Gamma(x, A, X) = \sum_{i=1}^{n_x} \sum_{j=1}^N A_i^j \varphi_i^j(x)$ is an ISM of g, there exists a sequence $j_1, \ldots j_{n_x} \in \{1, \ldots, N\}$ and points $y_i \in A_i^{j_i}$ satisfying $g(x) = \sum_{i=1}^{n_x} y_i$. Let $\delta_i = y_i - a_i$, with ω defined as in Algorithm 1 one can write

$$\begin{aligned}\alpha(g(x)) = \quad & \alpha\left(\omega + \sum_{i=1}^{n_x}\delta_i\right) \\ = \quad & \sum_{i=1}^{n_x}\left(\alpha(\omega+\delta_i) - \frac{n_x-1}{n_x}\alpha(\omega)\right) \\ & -\underbrace{\left(\sum_{i=1}^{n_x}\alpha(\omega+\delta_i) - (n_x-1)\alpha)\omega - \alpha\left(\omega+\sum_{i=1}^{n_x}\delta_i\right)\right)}_{r_\alpha(A)[-1,1]} .\end{aligned}$$

Since $\delta_i \in A_i^{j_i} - a_i$, we have $\alpha(\omega+\delta_i) \in \alpha\left(\omega - a_i + A_i^{j_i}\right)$ as well as

$$\begin{aligned}\alpha(g(x)) &\in \sum_{i=1}^{n_x}\left(\alpha\left(\omega - a_i + A_i^{j_i}\right) - \frac{n_x-1}{n_x}\alpha(\omega)\right) + r_\alpha(A)[-1,1] \\ &= \sum_{i=1}^{n_x} C_i^{j_i} .\end{aligned}$$

This implies the statement of the theorem. □

Theorem 2. Let $\Gamma(x, A, X)$ and $\Gamma(x, B, X)$ be ISMs of g and h, respectively, on X. If $C \in \mathbb{I}^{n_x \times N}$ is computed using Algorithm 2, then $\Gamma(x, C, X)$ is an ISM of $g * h$ on X.

A proof of Thm. 2 proceeds along the same lines as the proof of Thm. 1 and is omitted for the sake of brevity.

The construction of remainder bounds and central points used in Algorithm 1 relies on globally valid algebraic relations, called addition theorems, of common univariate operations. As an example, for the exponential function, the addition theorems $e^{\omega+\delta_i} = e^{\omega}e^{\delta_i}$ and $e^{\omega+\sum_{i=1}^{n_x}\delta_i} = e^{\omega}\prod_{i=1}^{n_x}e^{\delta_i}$, hold globally over the real numbers. Letting $t_i = e^{\delta_i} - 1$, $r_\alpha(A)$ can be constructed by bounding the left-hand side of the expression in Step 2) of Algorithm 1. This yields the expression

$$e^{\omega}\left|\sum_{i=1}^{n_x} t_i + 1 - \prod_{i=1}^{n_x}(1+t_i)\right| \le e^{\omega}\left(\prod_{i=1}^{n_x}(1+s_i) - \sum_{i=1}^{n_x} s_i - 1\right)$$

with $s_i = \max\left\{e^{U(A_i)-a_i} - 1, 1 - e^{L(A_i)-a_1}\right\}$. Choosing $a_i = \log\left(\frac{1}{2}\left(e^{U(A_i)} + e^{L(A_i)}\right)\right)$, minimizes

$$s_i = \frac{e^{U(A_i)} - e^{L(A_i)}}{e^{U(A_i)} + e^{L(A_i)}} .$$

The technical derivations for the remainder bounds $r_\alpha(A)$ and the central points a_i for other atom operations can be found in (Zha et al., 2016).

The final ingredient for ISA is the construction of a (trivial) ISM for the input variables x_i. As each variable is independent of the rest, the coefficients can be set as $A_k^j = 0$ for all $k \neq i$ and all $j \in \{1, \ldots, N\}$. The ith row of A is then initialized as $A_i^j = X_i^j$ for each $j \in \{1, \ldots, N\}$.

The computational complexity for both composition rules in ISA is of order $\mathbf{O}(n_x N)$. Furthermore, whenever f is separable, i.e. $f(x) = \sum_{j=1}^{n_x} f_j(x_j)$ for some factorable functions $f_1, \ldots, f_{n_x}$, the global approximation error is of order $\mathbf{O}\left(\frac{1}{N}\right)$ over all bounded domains $X \in \mathbb{I}^{n_x}$.

Algorithm 3. Intersection of a superposition model with an interval

---

Input: Parameters A and X of the input model and an interval Y

Main Step:

(1) Sort each A_i to obtain the permutations $\underline{\Pi}$ and $\overline{\Pi}$.
(2) Choose a finite number n_J of intervals $\underline{J}_k = [0, \underline{j}_k]$ with index vectors $\underline{j}_k \in \{1, \ldots, N\}^{n_x}$ such that

$$\forall k \in \{1, \ldots, n_J\}, \quad \sum_{i=1}^{n_x} \underline{A}_i^{\underline{\pi}_i((\underline{j}_k)_i)} \ge \overline{y}$$

(3) Choose a finite number n_J of intervals $\overline{J}_k = [0, \overline{j}_k]$ with index vectors $\overline{j}_k \in \{1, \ldots, N\}^{n_x}$ such that

$$\forall k \in \{1, \ldots, n_J\}, \quad \sum_{i=1}^{n_x} \overline{A}_i^{\overline{\pi}_i((\overline{j}_k)_i)} \le \underline{y}$$

Output: Permutations $\underline{\Pi}, \overline{\Pi}$ and intervals $\underline{J} = (\underline{J}_1, \ldots, \underline{J}_{n_J})$, $\overline{J} = (\overline{J}_1, \ldots, \overline{J}_{n_J})$.

---

4. ISA-BASED SET-INVERSION ALGORITHM

This section proposes a novel search strategy based on ISA for addressing GPE. It has as its core computing the intersection of an ISM with an interval.

Consider an ISM, of the function f over X, parameterized by $A \in \mathbb{I}^{n_x \times N}$. The direct way of computing the intersection between this ISM and $Y = [\underline{y}, \overline{y}]$ is to compute the value of the ISM at each interval $X_{j_1} \times \ldots \times X_{j_N}$ in the partition of X. This requires computing all possible superpositions of coefficients A_i^j. Such approach, while straightforward, is unfortunately not efficient since its computational complexity is $\mathbf{O}\left(N^{n_x}\right)$.

As it turns out, computing an over approximation of the desired intersection can be done by testing only certain selected combinations. The proposed approach, requires sorting the components $A_i^j = [\underline{A}_i^j, \overline{A}_i^j]$ of the rows A_i of the matrix A in both decreasing and increasing orders. The corresponding permutations are denoted by the functions $\overline{\pi}_i, \underline{\pi}_i : \{1, \ldots, N\} \to \{1, \ldots, N\}$ which satisfy

$$\overline{A}_i^{\overline{\pi}_i(1)} \ge \overline{A}_i^{\overline{\pi}_i(2)} \ge \ldots \ge \overline{A}_i^{\overline{\pi}_i(N)}$$

and

$$\underline{A}_i^{\underline{\pi}_i(1)} \le \underline{A}_i^{\underline{\pi}_i(2)} \le \ldots \le \underline{A}_i^{\underline{\pi}_i(N)} .$$

In the following, we use the shorthand $\overline{\Pi} = (\overline{\pi}_1, \ldots, \overline{\pi}_{n_x})$ and $\underline{\Pi} = (\underline{\pi}_1, \ldots, \underline{\pi}_{n_x})$. The main pre-processing step for computing a set inversion is outlined in Algorithm 3.

Theorem 3. Let $\underline{\Pi}, \overline{\Pi}$ and $\underline{J} = (\underline{J}_1, \ldots, \underline{J}_{n_J})$, $\overline{J} = (\overline{J}_1, \ldots, \overline{J}_{n_J})$ be computed by Algorithm 3. Define

$$\underline{\Xi} = \bigcup_{k \in \{1,\ldots,n_J\}} \bigcup_{j \in \underline{J}_k} \Xi_1^{\underline{\pi}_1(j_1)} \times \ldots \times \Xi_{n_x}^{\underline{\pi}_{n_x}(j_{n_x})} \tag{13}$$

and

$$\overline{\Xi} = \bigcup_{k \in \{1,\ldots,n_J\}} \bigcup_{j \in \overline{J}_k} \Xi_1^{\overline{\pi}_1(j_1)} \times \ldots \times \Xi_{n_x}^{\overline{\pi}_{n_x}(j_{n_x})} \tag{14}$$

with $\Xi_i^j = [\underline{x}_i + (j-1)h_i, \underline{x}_i + jh_i]$ and $h_i = \frac{\overline{x}_i - \underline{x}_i}{N}$. Then,

$$X \setminus \left(\underline{\Xi} \cup \overline{\Xi}\right) \supseteq \mathbb{X}_{\text{int}} \cup \mathbb{X}_{\text{bnd}} . \tag{15}$$

Proof. By construction, the function f takes values larger than $\overline{y}$ on all interval boxes $\Xi_1^{\pi_1(j_1)} \times \ldots \times \Xi_{n_x}^{\pi_{n_x}(j_{n_x})}$ for any $j \in \overline{J}_k$. Similarly, f takes smaller values than $\underline{y}$ on all intervals $\Xi_1^{\overline{\pi}_1(j_1)} \times \ldots \times \Xi_{n_x}^{\overline{\pi}_{n_x}(j_{n_x})}$ for any $j \in \underline{J}_k$. Consequently, the union of all of these boxes cannot possibly contain a point of $\mathbb{X}_{\rm int} \cup \mathbb{X}_{\rm bnd}$, which is the statement of the theorem. □

Theorem 3 provides a constructive procedure for finding the desired outer approximation of the set $\mathbb{X}_{\rm int} \cup \mathbb{X}_{\rm bnd}$. Notice that the computational complexity of Algorithm 3 is of order $\mathbf{O}(n_x N \log(N))$, because we need to sort the intervals along all coordinate directions. The associated storage complexity is of order $\mathbf{O}(n_x N)$. Finally, we have to keep in mind, that computing and storing the sets $\underline{\Xi}$ and $\overline{\Xi}$ is expensive in general, as these sets may be composed of an exponentially large amount of sub-intervals. Nevertheless, it is not necessary to store these sets explicitly as long as we store the permutation matrices $\underline{\Pi}$ and $\overline{\Pi}$ as well as the boxes $\underline{J}$ and $\overline{J}$, which uniquely represent the set $X \setminus \left(\underline{\Xi} \cup \overline{\Xi}\right)$.

Notice that there are various heuristics possible for refining the above procedure. However, the corresponding methods are analogous to the implementation in SIVIA and based on state-of-the-art branching techniques. Thus, the proposed technique based on Algorithm 3 can be embedded in an exhaustive search procedure, if one wishes to approximate the set $\mathbb{X}_{\rm int} \cup \mathbb{X}_{\rm bnd}$ with any given accuracy.

5. NUMERICAL EXAMPLES

This section illustrates some of the benefits of ISA as a bounding method for the range of factorable functions, as well as its application to GPE. Algorithms 1, 2, and a set-inversion algorithm based on Algorithm 3 were implemented in the programming language `Julia`. For comparison, a basic SIVIA algorithm was also implemented in `Julia`. The termination for both algorithms was based on (7). All results were obtained on an Intel Xeon CPU X5660 with 2.80GHz and 16GB RAM.

5.1 Bounding a nonlinear function: ISA vs. TMA

Consider the nonlinear factorable function

$$f(x) = e^{\sin(x_1) + \sin(x_2)\cos(x_2)}$$

over the domain $X = [0,1] \times [0, \overline{x}_2]$. Here, $\overline{x}_2 \in [0.1, 20]$ denotes a parameter which controlling the diameter of the domain. In order to measure the quality of an arithmetic, we used the Hausdorff distance between the range of f, $f(X) = f(x) | x \in X$, and an enclosure set $\overline{Y} \supseteq f(X)$. This distance is given by

$$d_{\rm H}(f(X), \overline{Y}) = \max_{y \in \overline{Y}} \min_{x \in f(X)} ||x - y||_\infty \ .$$

Figure 2 shows the overestimation of enclosures in the form of Taylor models of orders 1 and 2 as well as interval superposition models with $N = 1$, $N = 10$, and $N = 100$ as a function of the domain parameter $\overline{x}_2$. Although the Hausdorff distance between $f(X)$ and $\overline{Y}$ does not increase monotonically with $\overline{x}_2$, the rough trend observed on the plot is that the overestimation increases with the size of the domain. Furthermore, the plot shows that interval superposition models outperform Taylor models over large domains. One aspect that is not shown in the figure is that over small domains, e.g. over $[0, 10^{-1}]^2$, enclosures based on Taylor models outperform those constructed using interval superposition arithmetics.

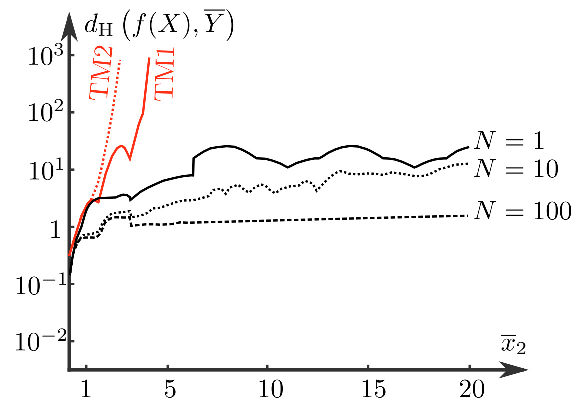

Fig. 2. Overestimation of enclosure sets with respect to the domain size. The plot compares enclosures based on TMs of orders 1 (solid red) and 2 (dotted red) as well as ISMs with $N = 1$ (solid black), $N = 10$ (dotted black), and $N = 100$ (dashed black).

5.2 Guaranteed parameter estimation via ISA

We consider a reaction system modelled by:

$$\begin{aligned} \dot{z_1}(t) &= -(x_1 + x_3) z_1(t) + x_2 z_2(t), \quad & z_1(0) = 1, \\ \dot{z_2}(t) &= x_1 z_1(t) - x_2 z_2(t), \quad & z_2(0) = 0 \ , \end{aligned} \tag{16}$$

with $y(t) = z_2(t)$ (Paulen et al., 2016). The output variable, can be represented as a factorable function

$$y(t) = \mathrm{e}^{\frac{-t\rho}{2}} \left(\mathrm{e}^{\frac{t\sigma}{2}} - \mathrm{e}^{\frac{-t\sigma}{2}} \right) x_1(t)/\sigma, \tag{17}$$

with $\sigma = (x_1^2 + x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3 - 2x_2x_3)^{\frac{1}{2}}$ as well as $\rho = x_1 + x_2 + x_3$. In the following, we fix $x_3 = 0.35$ and consider $n_m = 15$ measurements corresponding to the time instants $t_i = 1, 2, \ldots, 15$. Process measurements were obtained by simulating (16) with $x = (0.6, 0.15, 0.35)^T$, rounding to the second significant digit. Measurement errors of $\pm 10^{-3}$ were added to these values.

The performance of the proposed GPE algorithm using ISA was tested against a standard SIVIA. We use interval superposition models with $N = 2$, 10, 20. Figure 3 shows a summary of the results of the GPE algorithm using ISMs with $N = 2$. The left plot, shows an approximation of the set X_e. The plot shows the inner partition (in red) for $\epsilon = 10^{-5}$ and the boundary partitions for $\epsilon = 10^{-4}$ (light blue) and $\epsilon = 10^{-5}$ (dark blue). The central and right plots show, respectively, a comparison of the number of iterations and CPU time against the tolerance ϵ—for SIVIA (solid red line) and ISM-based set-inversion with $N = 2$ (solid black line), $N = 10$ (dotted black line), and $N = 20$ (dashed black line). In terms of the number of iterations and the number of boundary boxes (not shown), ISM-based set-inversion (for all N) outperforms SIVIA. This is due to the fact that ISA is able to detect and exploit structures in the factorable function to remove redundant boxes. On the contrary, with respect to the CPU time, SIVIA outperforms the proposed algorithm. This can be traced back to the fact that the cost per iteration is larger for ISA. Furthermore, the implementation is still

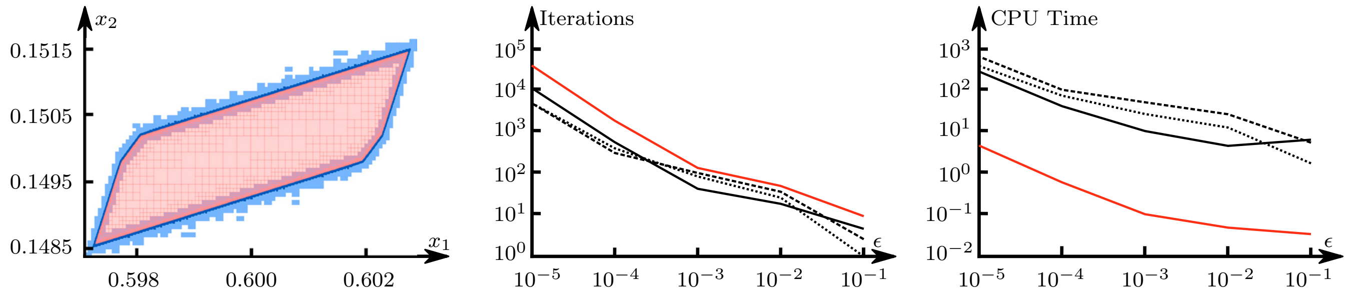

Fig. 3. Results for the GPE problem. Left: Parameter inner partition (in red) for $\epsilon = 10^{-5}$ and the boundary partitions for $\epsilon = 10^{-4}$ (light blue) and $\epsilon - 10^{-5}$ (dark blue). Center: Number of iterations vs. diameter of boundary partition. Right: CPU time vs. diameter of boundary partition. Center and right plots show results for SIVIA (solid red line) and ISM-based set-inversion with $N = 2$ (solid black line), $N = 10$ (dotted black line), and $N = 20$ (dashed black line).

at the prototype stage and requires further refinement in terms of computing the remainder bounds and memory management in the algorithms.

6. CONCLUSION

This paper presented Interval superposition arithmetics, a novel set arithmetic for computing enclosures of the image set of factorable functions and its use in guaranteed parameter estimation. The main advantage of ISA is its polynomial storage and computational complexity. The core routine behind the proposed GPE method is the intersection of an interval superposition model and an interval. Although the proposed intersection routine has a computational complexity of order $\mathbf{O}(n_x N \log(N))$, computing an arbitrarily accurate approximation of the parameter set requires exponential run time. Our numerical examples illustrate the advantages of ISA over other set arithmetics when constructing enclosures for factorable functions—particularly over large domains. We have also shown how the proposed technique can be used to solve a GPE problem. Although the number of iterations is reduced when using ISA, the overall CPU time is larger than SIVIA. This suggests that, although ISA can improve certain aspects of GPE algorithms, there is still much room for improvement. Improved ISA-based algorithms for constructing approximations of inverse image sets in polynomial runtime will be investigated in the future work.

ACKNOWLEDGMENTS

This work was supported by: National Science Foundation China (NSFC), Grant 61473185; ShanghaiTech University, Grant F-0203-14-012; and Slovak Research and Development Agency, project APVV SK-CN-2015-0016 "CN-SK cooperation: Verified Estimation and Control of Chemical Processes". RP also acknowledges: Slovak Research and Development Agency, project APVV 15-0007; and European Commission, grant agreement 790017 (GuEst).

REFERENCES

Battles, Z. and Trefethen, L.N. (2004). An extension of matlab to continuous functions and operators. *SIAM Journal on Scientific Computing*, 25(5), 1743–1770.

Franceschini, G. and Macchietto, S. (2008). Model-based design of experiments for parameter precision: State of the art. *Chem. Eng. Sci.*, 63(19), 4846–4872.

Hast, D., Findeisen, R., and Streif, S. (2015). Detection and isolation of parametric faults in hydraulic pumps using a set-based approach and quantitative-qualitative fault specifications. *Control Engineering Practice*, 40, 61 – 70.

Jaulin, L., Kieffer, M., Walter, E., and Meizel, D. (2002). Guaranteed robust nonlinear estimation with application to robot localization. *Trans. Sys. Man Cyber Part C*, 32(4), 374–381.

Jaulin, L. and Walter, E. (1993). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4), 1053–1064.

Lin, Y. and Stadtherr, M.A. (2007). Validated solutions of initial value problems for parametric ODEs. *Applied Numerical Mathematics*, 57(10), 1145–1162.

Ljung, L. (1999). *System Identification (2nd Ed.): Theory for the User.* Prentice Hall PTR, Upper Saddle River, NJ, USA.

Makino, K. and Berz, M. (1996). *Remainder Differential Algebras and Their Applications*, chapter 5, 63–75. SIAM.

Marco, M.D., Garulli, A., Lacroix, S., and Vicino, A. (2000). A set theoretic approach to the simultaneous localization and map building problem. In *Proc. of the 39th IEEE Conf. on Decision and Control*, 833–838.

Moore, R.E., Kearfott, R.B., and Cloud, M.J. (2009). *Introduction to interval analysis*, volume 110. Siam.

Paulen, R., Villanueva, M.E., and Chachuat, B. (2016). Guaranteed parameter estimation of non-linear dynamic systems using high-order bounding techniques with domain and cpu-time reduction strategies. *IMA Journal of Mathematical Control and Information*, 33(3), 563–587.

Rajyaguru, J., Villanueva, M.E., Houska, B., and Chachuat, B. (2017). Chebyshev model arithmetic for factorable functions. *Journal of Global Optimization*, 68(2), 413–438.

Schweppe, F. (1968). Recursive state estimation: Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1), 22–28. doi: 10.1109/TAC.1968.1098790.

Zha, Y., Villanueva, M.E., and Houska, B. (2016). Interval superposition arithmetic. *arXiv preprint arXiv:1610.05862v2.*