INTERVAL UNIONS*

HERMANN SCHICHL , FERENC DOMES , TIAGO MONTANHER , AND KEVIN KOFLER

Abstract. This paper introduces the interval union arithmetic, a new concept which extends the traditional interval arithmetic. Interval unions allow to manipulate sets of disjoint intervals and provide a natural way to represent the extended interval division. Considering interval unions lead to simplifications of the interval Newton method as well as of other algorithms for solving interval linear systems. This paper does not aim at describing the complete theory of interval union analysis, but rather at giving basic definitions and some fundamental properties, as well as showing theoretical and practical usefulness of interval unions in a few selected areas.

Key words. interval union arithmetic, union of intervals, interval union Newton method, interval union linear systems.

AMS subject classifications. 65G30, 65G20, 65G40, 49M15

1. Introduction.

Interval analysis is a branch of numerical analysis that was born in the 1960's. It consists of computing with intervals of reals instead of reals, providing a framework for handling uncertainties and verified computations (see e.g. [2, 20, 22] and [14] for a survey). Interval analysis is a key ingredient for numerical constraint satisfaction (see e.g. [12]) and global optimization (see e.g. [7, 16]). Global optimization solvers like *Gloptlab* [4]and *COCONUT* [26, 27] rely heavily on interval analysis to guarantee rigorous solutions, even non-rigorous solvers like *BARON* [25] and α-*Branch and bound* [1] use rigorous computations in some steps of the search. Applications of interval analysis comprise packing problems [28], robotics [6, 19], localization and map building [10, 11], and the protein folding problem [18].
In practice, interval arithmetic must be implemented using outward rounding in order to assure that the result of an interval calculation always contains the result of the corresponding real valued operation evaluated for each value(s) of the used interval(s). Interval arithmetic has been implemented in almost every programming language which is relevant for scientific computing, see for example *Intlab* [24] for *Matlab*, *Filib++* [21] for *C/C++*, *Interval* [13] for *Fortran* and *MathInterval* [5] for *Java*.
Extended interval arithmetic [7, 14, 23] allows operations on intervals where the bounds can be $\pm\infty$. It gives the possibility of performing interval division even when the denominator interval contains zero. For example, assume that we are interested in rigorous bounds for $\mathbf{x} = \frac{[2,3]}{[-1,1]}$. Applying the division rule presented in [23] gives $[-\infty, -2]$ and $[2, \infty]$. The operation above must be interpreted as follows: The resulting quotient of $\frac{a}{b}$ where $a \in [2, 3]$ and $b \in [-1, 1]\backslash 0$ belongs to the set $[-\infty, -2] \cup [2, \infty]$. This example shows the problem of interval arithmetic both from a theoretical and a computational point of view. For the theory of intervals it is an issue since the result of an elementary operation involving two intervals does not belong necessarily to the set of intervals [1] while for computations it is a problem since the interval division operator requires special treatment.
This paper extends the concept of interval arithmetic to interval unions. An interval union is a set of closed and disjoint intervals where the bounds of the extreme intervals can be $\pm\infty$. During the paper we demonstrate that interval unions generalize intervals

*This research was partially supported through the research grants P25648-N25 and P27376-N25 of the Austrian Science Fund (FWF) and CNPQ-205557/2014-7 of the Brazilian council of research.

[1] unless the interval hull is taken, which often leads to serious overestimation of the true result

and allow among others to represent the result of interval division in a natural way. Some of the theoretical results of interval analysis remain valid when we are dealing with interval unions. That is the case, e.g., for the fundamental theorem of interval arithmetic, and therefore the natural extension of real functions to interval unions is similar to the interval case. On the other hand, some inclusion results like the interval mean value theorem do not hold for interval unions, not even for the univariate case. During the paper it is shown that a large part of the interval union arithmetic can be easily implemented if we have an interval arithmetic library at our disposal.

The paper is organized as follows. In Section 2 we present the basics of interval arithmetic. The section is mainly a revision of the traditional case in the extended context. Section 3 describes the generalization from intervals to interval unions, where the basic interval union operations are defined, isotonicity property shown, the fundamental theorem of interval union arithmetic is proven. In addition, in this section, hull and component-wise operations are also defined.
In Section 4 the interval union Newton method for univariate functions is presented. Similar as for the interval Newton method the aim is to enclose all roots of $f(x) \in R$ subject to $x \in X$ where both, R and X are interval unions. We show that the definition of Newton methods can be made through component-wise operations and compare our new approach with the traditional interval Newton algorithm in a set of 32 problems. Our experiment shows that interval union arithmetic can improve Newton methods significantly in the univariate case.
Finally in Section 5, interval union linear systems are studied and shown that the interval Gaussian elimination and Gauss-Seidel algorithms can be extended from intervals to interval unions. The advantages of replacing interval operations by interval unions in linear systems are demonstrated by performing tests on examples in low dimension.

1.1. Notation. We mostly follow [17] for the notation of interval arithmetic. Throughout this paper $\mathbb{R}^{m\times n}$ denotes the vector space of all $m \times n$ matrices A with real entries A_{ik} $(i = 1, \dots, m,\ k = 1, \dots, n)$, and $\mathbb{R}^n = \mathbb{R}^{n\times 1}$ denotes the vector space of all column vectors v of length n and entries v_i $(i = 1, \dots, n)$. For vectors and matrices, the relations $=, \neq, <, >, \leq, \geq$ and the absolute value $|A|$ of the matrix A are interpreted component-wise.
We write A^T to represent the transpose of a matrix A and A^{-T} is short for $(A^T)^{-1}$. The ith row vector of a matrix A is denoted by $A_{i:}$ and the jth column vector by $A_{:j}$. For the $n\times n$ matrix A, $\mathrm{diag}(A)$ denotes the n-dimensional vector with $\mathrm{diag}(A)_i = A_{ii}$. The number of elements of the index set N is given by $|N|$. Let $I \subseteq \{1, \dots, m\}$ and $J \subseteq \{1, \dots, n\}$ be index sets and let $n_I := |I|$, $n_J := |J|$. For the n-dimensional vector x, x_J denotes the n_J-dimensional vector built from the components of x selected by the index set J. For the $m \times n$ matrix A, the expression $A_{I:}$ denotes the $n_I \times n$ matrix built from the rows of A selected by the index sets I. Similarly, $A_{:J}$ denotes the $m \times n_J$ matrix built from the columns of A selected by the index sets J.

2. Interval Arithmetic. This section presents the basics of interval arithmetic. A comprehensive approach to this topic is given by [22]. We are mainly interested in extended interval arithmetic. i.e, when division by intervals containing 0 is allowed. Good references to extended interval arithmetic are [7] and [15].
Let $\underline{a}, \overline{a} \in \mathbb{R}$ with $\underline{a} \leq \overline{a}$ then $\mathbf{a} = [\underline{a}, \overline{a}]$ denotes a **real interval** with $\inf(\mathbf{a}) = \min(\mathbf{a}) = \underline{a}$ and $\sup(\mathbf{a}) = \max(\mathbf{a}) = \overline{a}$. The **set of nonempty compact real**

intervals is denoted by

$$\mathbb{IR} := \{[\underline{a}, \overline{a}] \mid \underline{a} \leq \overline{a},\ \underline{a}, \overline{a} \in \mathbb{R}\}.$$

We extend the definition of real intervals by permitting the bounds of intervals to be one of the ideal points $-\infty$ and ∞ and define $\overline{\mathbb{IR}}$ as the **set of closed real intervals**. We write

$$\overline{\mathbb{IR}} := \mathbb{IR} \cup \{[-\infty, \overline{a}] \mid \overline{a} \in \mathbb{R}\} \cup \{[\underline{a}, \infty] \mid \underline{a} \in \mathbb{R}\} \cup \{[-\infty, \infty], \emptyset\},$$

defining, $[-\infty, \overline{a}] := \{x \in \mathbb{R} \mid x \leq \overline{a}\}$, $[\underline{a}, \infty] := \{x \in \mathbb{R} \mid x \geq \underline{a}\}$, and $[-\infty, \infty] := \mathbb{R}$. The **width** of the interval $\mathbf{a} \in \overline{\mathbb{IR}} \setminus \{\emptyset\}$ is given by $\mathrm{wid}(\mathbf{a}) := \overline{a} - \underline{a}$, its **mignitude** by

$$\langle \mathbf{a} \rangle := \begin{cases} \min(|\underline{a}|, |\overline{a}|) & \text{if } 0 \notin [\underline{a}, \overline{a}], \\ 0 & \text{otherwise.} \end{cases}$$

and its **magnitude** by $|\mathbf{a}| := \max(|\underline{a}|, |\overline{a}|)$. The **midpoint** of $\mathbf{a} \in \mathbb{IR}$ is $\check{\mathbf{a}} := \mathrm{mid}(\mathbf{a}) := (\underline{a} + \overline{a})/2$ and the **radius** of $\mathbf{a} \in \overline{\mathbb{IR}}$ is $\hat{\mathbf{a}} := \mathrm{rad}(\mathbf{a}) := (\underline{a} - \overline{a})/2$. For $\mathbf{a} \in \overline{\mathbb{IR}}$ there is no natural definition of a midpoint. Moreover, if $\check{\mathbf{a}}$ is well defined then $a \in \mathbf{a} \Leftrightarrow |a - \check{\mathbf{a}}| \leq \hat{\mathbf{a}}$ and we say that $\mathrm{midrad}(\check{\mathbf{a}}, \hat{\mathbf{a}})$ is the midrad representation of interval $\mathbf{a}$. For a set S the smallest box containing S is called the **interval hull** of S and denoted by $\square S$. An interval is called **thin** or **degenerate** if $\mathrm{wid}(\mathbf{a}) = 0$.
The **inclusion relations** are given as

$$\mathbf{a} \subset \mathbf{b} \Longleftrightarrow \underline{b} < \underline{a} \wedge \overline{a} < \overline{b}, \quad \mathbf{a} \subseteq \mathbf{b} \Longleftrightarrow \underline{b} \leq \underline{a} \wedge \overline{a} \leq \overline{b}.$$

An **interval vector** $\mathbf{x} = [\underline{x}, \overline{x}]$ or **box** is the Cartesian product of the closed real intervals $\mathbf{x}_i := [\underline{x}_i, \overline{x}_i] \in \overline{\mathbb{IR}}$. We write $\overline{\mathbb{IR}}^n$ to denote the set of all n-dimensional boxes. We also define the **interval matrix** $\mathbf{A} = [\underline{A}, \overline{A}]$ in a similar way and $\overline{\mathbb{IR}}^{m \times n}$ denotes the set of all $m \times n$ interval matrices. Operations defined for intervals (like width, midpoint, radius, mignitude and magnitude) are defined component-wise when applied to boxes or matrices.
Let $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$. The elementary real operations $\circ \in \{+, -, /, *, \hat{}\}$ are extended to the interval arguments $\mathbf{a}$, $\mathbf{b}$ by defining the result of an elementary interval operation to be the set of real numbers which results from combining any two numbers contained in $\mathbf{a}$ and in $\mathbf{b}$. Formally,

$$\mathbf{a} \circ_{\bullet} \mathbf{b} := \{a \circ b \mid a \in \mathbf{a},\ b \in \mathbf{b} \text{ and } a \circ b \text{ is defined}\}.$$

This leads to operations on $\overline{\mathbb{IR}}$ defined by $\mathbf{a} \circ \mathbf{b} := \square(\mathbf{a} \circ_{\bullet} \mathbf{b})$. The elementary operations are **inclusion isotonic**. That means:

$$\mathbf{a} \subset \mathbf{a}', \mathbf{b} \subset \mathbf{b}' \quad \Rightarrow \quad \mathbf{a} \circ \mathbf{b} \in \mathbf{a}' \circ \mathbf{b}' \quad \text{for all } \circ \in \{+, -, /, *, \hat{}\}.$$

For $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$ we get that

$$\mathbf{a} /_{\bullet} \mathbf{b} := \begin{cases} \mathbf{a} \cdot [1/\overline{b}, 1/\underline{b}] & \text{if } 0 \notin \mathbf{b}, \\ [-\infty, +\infty] & \text{if } 0 \in \mathbf{a} \wedge 0 \in \mathbf{b}, \\ [\overline{a}/\underline{b}, +\infty] & \text{if } \overline{a} < 0 \wedge \underline{b} < \overline{b} = 0, \\ [-\infty, \overline{a}/\overline{b}] \cup [\overline{a}/\underline{b}, +\infty] & \text{if } \overline{a} < 0 \wedge \underline{b} < 0 < \overline{b}, * \\ [-\infty, \overline{a}/\overline{b}] & \text{if } \overline{a} < 0 \wedge 0 = \underline{b} < \overline{b}, \\ [-\infty, \underline{a}/\underline{b}] & \text{if } 0 < \underline{a} \wedge \underline{b} < \overline{b} = 0, \\ [-\infty, \underline{a}/\underline{b}] \cup [\underline{a}/\overline{b}, +\infty] & \text{if } 0 < \underline{a} \wedge \underline{b} < 0 < \overline{b}, * \\ [\underline{a}/\overline{b}, +\infty] & \text{if } 0 < \underline{a} \wedge 0 = \underline{b} < \overline{b}, \\ \emptyset & \text{if } 0 \notin \mathbf{a} \wedge \underline{b} = \overline{b} = 0. \end{cases} \tag{1}$$

As one can see in the cases marked with $*$, the result is not a single interval but the union of two disjoint ones. As shown in [23] the division defined by (1) is inclusion isotonic (also see, [15]).
In some applications the interval definition of subtraction may over-estimate the range of the real computation. For example, since $-\mathbf{a} := 0 - \mathbf{a} = [-\sup(\mathbf{a}), -\inf(\mathbf{a})]$ for $\mathbf{b} := \mathbf{a} - \mathbf{a}$, we only have $0 \in \mathbf{b}$ and

$$\mathbf{b} = \mathbf{a} + (-\mathbf{a}) \neq [0,0] = 0 \text{ if } \inf(a) \neq \sup(a),$$

does not hold. In order to cope with this situation we also define inner subtraction for intervals. If $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$ then

$$\mathbf{a} \ominus \mathbf{b} := \begin{cases} [\inf(\mathbf{a}) - \inf(\mathbf{b}), \sup(\mathbf{a}) - \sup(\mathbf{b})] & \text{if } \operatorname{wid}(\mathbf{a}) \geq \operatorname{wid}(\mathbf{b}) \\ [\sup(\mathbf{a}) - \sup(\mathbf{b}), \inf(\mathbf{a}) - \inf(\mathbf{b})] & \text{otherwise} \end{cases} \tag{2}$$

Inner operations lead to significant improvements on the interval Gauss-Seidel algorithm discussed later in this paper.
Let $\mathbf{x} \in \overline{\mathbb{IR}}^n$ and $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$. We define $\operatorname{rg}\bullet(f(\mathbf{x}))$ to be the set

$$\operatorname{rg}\bullet(f(\mathbf{x})) := \{f(x) \mid x \in \mathbf{x} \cap D\},$$

and call it the **range** of f over the box $\mathbf{x}$. We extend the range to a function on $\overline{\mathbb{IR}}$ by $\operatorname{rg}(f(\mathbf{x})) := \square \operatorname{rg}\bullet(f(\mathbf{x}))$, also called the range of f.
We say that a function $\mathbf{f} : \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}$ is **inclusion isotonic** if $\mathbf{x} \subseteq \mathbf{y} \Rightarrow \mathbf{f}(\mathbf{x}) \subseteq \mathbf{f}(\mathbf{y})$. We already established that elementary interval operations are inclusion isotonic and it is also possible to construct interval functions with the isotonicity property for standard functions like exponential, logarithmic and trigonometric, see for example [24] or [5]. Moreover, it is easy to prove that the composition of inclusion isotonic functions is also inclusion isotonic. Formally we have

Proposition 1. *If $\mathbf{g} : \overline{\mathbb{IR}}^m \to \overline{\mathbb{IR}}$ and $\mathbf{f} : \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}^m$ are inclusion isotonic functions then $g(f(x))$ is inclusion isotonic.*

The interval function $\mathbf{f} : \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}$ is an **interval extension** of a function $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$ if

$$\mathbf{f}(x) = f(x) \quad \text{for } x \in D, \text{ and } f(x) \in \mathbf{f}(\mathbf{x}) \quad \text{for all } x \in \mathbf{x} \subseteq D.$$

If f admits a closed form and can be expressed in terms of elementary operations and standard functions we call the interval function $\mathbf{f}$ given by replacing every real operation with its interval counterpart the **natural extension**. Using these definitions we can formulate the fundamental theorem of interval analysis and prove it as in [20]:

Proposition 2 (Fundamental theorem of interval analysis). *If $\mathbf{f}$ is inclusion isotonic and is an interval extension of $f : \mathbb{R}^n \to \mathbb{R}$ then $\operatorname{rg}(f(\mathbf{x})) \subseteq \mathbf{f}(\mathbf{x})$.*

Interval arithmetic also allows to prove a general version of the mean value theorem for multivariate functions, see [22]:

Proposition 3 (Interval mean value theorem). *Let $F : \mathbb{R}^n \to \mathbb{R}^n$ be a differentiable function defined on a box $\mathbf{x} \subset \mathbb{R}^n$. If $\mathbf{F}$ is an interval extension of F and $\mathbf{J}$ an interval extension of the Jacobian of F both of them satisfying the isotonicity property then for $x, y \in \mathbf{x}$*

$$F(y) \in \mathbf{F}(x) + \mathbf{J}(\mathbf{x})(y - x).$$

Proposition 3 leads to the following Taylor extension, see [22].

Corollary 4 (Taylor expansion). *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function defined in a box $\mathbf{x} \subset \mathbb{R}^n$. If $\mathbf{f}$ is an interval extension of f and $\mathbf{g}$ the interval extension of the gradient of f both of them satisfying the isotonicity property then*

$$\mathbf{f}(\mathbf{x}) \subseteq \mathbf{f}(x) + \mathbf{g}(\mathbf{x})^T(\mathbf{x} - x), \quad x \in \mathbf{x}.$$

We define the set

$$f_{k\bullet}^{-1}(\mathbf{x}, \mathbf{y}) := \{z_k \in \mathbf{x}_k \mid \exists z_1, \dots, z_{k-1}, z_{k+1}, \dots, z_n : z \in D \cap \mathbf{x} \wedge f(z) \in \mathbf{y}\}$$

and call it the kth **partial inverse image** of f on $\mathbf{y}$ and for its interval hull we write

$$f_k^{-1}(\mathbf{x}, \mathbf{y}) := \square f_{k\bullet}^{-1}(\mathbf{x}, \mathbf{y}).$$

3. Interval Unions.

3.1. Motivation.

The well known **interval Newton iteration**

$$\mathbf{x}^{(k+1)} := N(\mathbf{x}^k) \cap \mathbf{x}^k, \quad N(\mathbf{x}) = \check{\mathbf{x}} - \frac{\mathbf{f}(\check{\mathbf{x}})}{\mathbf{f}'(\mathbf{x})}, \quad k = 0, 1, 2, \dots. \tag{3}$$

is the interval variant of Newton's method for finding the roots of a function f in a box $\mathbf{x}$. If (3) is applied to an arbitrary univariate function $f : \mathbb{R} \to \mathbb{R}$ and the starting interval $\mathbf{x}_0$, the interval Newton method splits and contracts $\mathbf{x}_0$ into several intervals enclosing the zeros of f over $\mathbf{x}_0$.
By (1) the division operator applied to two intervals $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$ in the cases marked by a $*$ do not map into $\overline{\mathbb{IR}}$. To solve this issue one can either define $/ : \overline{\mathbb{IR}} \times \overline{\mathbb{IR}} \setminus \{0\} \to \overline{\mathbb{IR}}$ or for the marked cases one could take the interval hull of the two resulting intervals. However, keeping the two disjoint intervals in the marked cases is the reason why (3) works properly if $0 \in \mathbf{f}'(\mathbf{x})$. Therefore, it is obvious to define a structure where the division operator and therefore the interval Newton method is defined in a consistent and natural way. It serves as a motivation to introduce interval unions and define operations similar to the interval versions.

3.2. Definition.

Definition 5. *Throughout this paper, interval unions are denoted by bold calligraphic letters. An* ***interval union*** $\boldsymbol{u}$ *of length $l(\boldsymbol{u}) := k$ is a finite set of k disjoint intervals. Since for all disjoint intervals the natural ordering exists we denote the elements of $\boldsymbol{u}$ by $\mathbf{u}_i$ and write*

$$\boldsymbol{u} = (\mathbf{u}_1, \dots, \mathbf{u}_k) \quad \textit{with} \quad \begin{array}{ll} \mathbf{u}_i \in \overline{\mathbb{IR}} & \forall\, i = 1, \dots, k, \\ \overline{\mathbf{u}}_i < \underline{\mathbf{u}}_{i+1} & \forall\, i = 1, \dots, k-1. \end{array} \tag{4}$$

The set of all interval unions of length $\leq k$ is denoted by $\mathcal{U}_k$ and $\mathcal{U} := \bigcup_{k \geq 0} \mathcal{U}_k$ is the set of all interval unions. In addition to this $\mathcal{U}_0 = \emptyset$ and we identify $\mathcal{U}_1$ with $\overline{\mathbb{IR}}$.

Obviously $\mathcal{U}_k \subseteq \mathcal{U}_m \subseteq \mathcal{U}$ if $k \leq m$.

Definition 6. *Let $\boldsymbol{u} := (\mathbf{u}_1, \dots, \mathbf{u}_k) \in \mathcal{U}$ be an interval union. We will identify $\boldsymbol{u}$ with the subset $\bigcup_{i=1}^k \mathbf{u}_i$ of $\mathbb{R}$ that $\boldsymbol{u}$ represents, so for a real number x we say*

$$x \in \boldsymbol{u} \Leftrightarrow \textit{ there exists a } 1 \leq i \leq k \textit{ such that } x \in \mathbf{u}_i.$$

Similarly, for the interval $\mathbf{x}$

$$\mathbf{x} \subseteq \boldsymbol{u} \Leftrightarrow \text{ there exists a } 1 \leq i \leq k \text{ such that } \mathbf{x} \subseteq \mathbf{u}_i.$$

Finally, for another interval union $\boldsymbol{v}$

$$\boldsymbol{v} \subseteq \boldsymbol{u} \Leftrightarrow \text{ for all } \boldsymbol{v} \in \boldsymbol{v} \text{ there exists a } 1 \leq i \leq k \text{ such that } \boldsymbol{v} \subseteq \mathbf{u}_i.$$

DEFINITION 7. *Let S be a finite set of intervals, the **union creator** $\mathcal{U}(S)$ is defined as the smallest interval union $\boldsymbol{u}$ that satisfies $\mathbf{a} \subseteq \boldsymbol{u}$ for all $\mathbf{a} \in S$.*

LEMMA 8. *Let S be a set of intervals, the union creator is inclusion isotonic:*

$$S \subseteq S' \Longrightarrow \mathcal{U}(S) \subseteq \mathcal{U}(S').$$

Proof. Follows directly from the definition. □

LEMMA 9. *The **interval hull of a union** $\boldsymbol{u} \in \mathcal{U}$ is given by*

$$\square \boldsymbol{u} = [\underline{\mathbf{u}}_1,\ \overline{\mathbf{u}}_{l(\boldsymbol{u})}].$$

Proof. Follows directly from Definition 5. □

DEFINITION 10. *We define $\mathcal{U}_k^n$ and $\mathcal{U}^n$, respectively, as the set of all interval union vectors of dimension n. Similarly, we introduce $\mathcal{U}_k^{n\times m}$ and $\mathcal{U}^{n\times m}$ as the sets of interval union matrices of size $n \times m$ with the usual definition of the operations. We denote interval union matrices by capital bold calligraphic letters like $\boldsymbol{\mathcal{A}}$ or $\boldsymbol{\mathcal{B}}$ and denote interval union vectors by lower case bold calligraphic letters like $\boldsymbol{x}$ or $\boldsymbol{y}$.*
The interval union vector $\boldsymbol{u} \in \mathcal{U}$ regarded as a subset of $\mathbb{R}^n$ is always a finite set of boxes. More specifically, if $\boldsymbol{u}_j$ has length k_j we get the $\prod_{j=1}^n k_j$ disjoint boxes $\prod_{j=1}^n \mathbf{u}_{j,\ell_j}$, $1 \leq \ell_j \leq k_j$. We write for $\mathbf{u} \in \overline{\mathbb{IR}}^n$ that $\mathbf{u} \in \boldsymbol{u}$ iff $\mathbf{u}$ is one of these boxes.

Note that storing this set as an interval union vector requires just $\sum_{j=1}^n k_j$ intervals which is a clear advantage over storing all the individual boxes, especially in higher dimensions.
If $\boldsymbol{u} \in \mathcal{U}_k \setminus \{\emptyset\}$ we define the **magnitude** and **mignitude** of the interval union respectively by

$$|\boldsymbol{u}| := \max(|\mathbf{u}_1|, \ldots, |\mathbf{u}_k|) = \max(|\underline{\mathbf{u}}_1|, |\overline{\mathbf{u}}_k|)$$

and

$$\langle \boldsymbol{u} \rangle := \min(\langle \mathbf{u}_1 \rangle, \ldots, \langle \mathbf{u}_k \rangle).$$

We also define for $\boldsymbol{u} \in \mathcal{U}_k \setminus \{\emptyset\}$ the **maximum**, **minimum** and **maximum width** of interval unions by

$$\max(\boldsymbol{u}) := \overline{\mathbf{u}}_k, \qquad \min(\boldsymbol{u}) := \underline{\mathbf{u}}_1$$

and

$$\max \operatorname{wid}(\boldsymbol{u}) := \max(\operatorname{wid}(\mathbf{u}_1), \ldots, \operatorname{wid}(\mathbf{u}_k))$$

Given the interval union $\boldsymbol{u} \in \mathcal{U}_k$ and a point $x \in \mathbb{R}$ we define the projection of x as follows

$$\operatorname{proj}(x, \boldsymbol{u}) = \begin{cases} x & \text{if} \quad x \in \boldsymbol{u} \\ \overline{\mathbf{u}}_i & \text{if} \quad x \in]\overline{\mathbf{u}}_i, \underline{\mathbf{u}}_{i+1}[\text{ and } x - \overline{\mathbf{u}}_i < \underline{\mathbf{u}}_{i+1} - x, \\ \underline{\mathbf{u}}_{i+1} & \text{if} \quad x \in]\overline{\mathbf{u}}_i, \underline{\mathbf{u}}_{i+1}[\text{ and } x - \overline{\mathbf{u}}_i \geq \underline{\mathbf{u}}_{i+1} - x, \\ \overline{\mathbf{u}}_k & \text{if} \quad x > \overline{\mathbf{u}}_k, \\ \underline{\mathbf{u}}_1 & \text{if} \quad x < \underline{\mathbf{u}}_1. \end{cases}$$

Some functions defined for intervals do not extend naturally to interval unions. For such functions we present different definitions that can be useful in several contexts. Let $\boldsymbol{u} \in \mathcal{U}_k \setminus \{\emptyset\}$ be an interval union, we denote the component-wise mid-point and radius respectively by $\check{\boldsymbol{u}}_c := (\check{\mathbf{u}}_1, \ldots, \check{\mathbf{u}}_k)$ and $\hat{\boldsymbol{u}}_c := (\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_k)$ whenever $-\infty < \underline{\mathbf{u}}_1 \leq \overline{\mathbf{u}}_k < \infty$. We denote the component-wise width and magnitude of $\boldsymbol{u}$ by $\operatorname{wid}(\boldsymbol{u})_c := (\operatorname{wid}(\mathbf{u}_1), \ldots, \operatorname{wid}(\mathbf{u}_k))$ and $|\boldsymbol{u}|_c := (|\mathbf{u}_1|, \ldots, |\mathbf{u}_k|)$ respectively. In some applications we also need to define operations above over the hull of $\boldsymbol{u}$. In such cases we add a subscript h to identify the hull operation. For example the hull mid-point operator and hull width of $\boldsymbol{u}$ are given by $\check{\boldsymbol{u}}_h := \square \boldsymbol{u}$ and $\operatorname{wid}(\boldsymbol{u}) := \operatorname{wid}(\square \boldsymbol{u})$.

3.3. Maximum length and filling gaps. The motivation from Section 3.1 hints a problem which can arise when considering interval unions, since during iterative evaluations the number of intervals inside a union can grow uncontrollably. This can be easily anticipated if considering the task of finding zeros of a function having an infinite number of zeros in the starting box via the interval Newton method. Actually, this problem arises in several other interval methods where intervals unions could prove quite useful. We propose to solve the problem by restricting the maximum length of unions and by defining gap filling strategies.

DEFINITION 11. *Let $\boldsymbol{u} \in \mathcal{U}$ be an interval union and let $\mathbf{u}_i$, $\mathbf{u}_{i+1} \in \boldsymbol{u}$. The open interval $\mathbf{g}_i$ between the intervals $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ is called the ith **gap** of $\boldsymbol{u}$ and is defined as*

$$\mathbf{g}_i = (\overline{\mathbf{u}}_i,\ \underline{\mathbf{u}}_{i+1}). \tag{5}$$

DEFINITION 12. *A **gap collection** $\hat{\mathfrak{v}}$ of length k is a set of k disjoint open real intervals. We will write*

$$\hat{\mathfrak{v}} = \langle \hat{\mathfrak{v}}_1, \ldots, \hat{\mathfrak{v}}_k \rangle \quad \text{with} \quad \begin{array}{ll} \boldsymbol{v}_i =]\underline{v}_i, \overline{v}_i[& \forall\ i = 1, \ldots, k,\ \ \underline{v}_i < \overline{v}_i \in \mathbb{R}, \\ \overline{v}_i \leq \underline{v}_{i+1} & \forall\ i = 1, \ldots, k-1. \end{array}$$

We denote by $\widehat{\mathcal{U}}_k$ the set of all gap collections of size $\leq k$ and by $\widehat{\mathcal{U}} := \bigcup_{i \in \mathbb{N}} \widehat{\mathcal{U}}_i$ the set of all gap collections.
We will again identify $\hat{\mathfrak{v}} \in \widehat{\mathcal{U}}$ with the set $\bigcup_{\hat{\boldsymbol{v}} \in \hat{\mathfrak{v}}} \hat{\boldsymbol{v}} \subseteq \mathbb{R}$ and write $x \in \hat{\mathfrak{v}}$, $\mathbf{x} \subseteq \hat{\mathfrak{v}}$, and $\hat{\mathfrak{w}} \subseteq \hat{\mathfrak{v}}$ for $x \in \mathbb{R}^n$, $\mathbf{x} \in \overline{\mathbb{IR}}$, and $\mathfrak{w} \in \widehat{\mathcal{U}}$.

LEMMA 13. *Let $\boldsymbol{u}$ be an interval union of length k, and let $\hat{\boldsymbol{u}} = \langle \mathbf{g}_1, \ldots, \mathbf{g}_{k-1} \rangle$ be the sequence of all gaps of $\boldsymbol{u}$. Then $\hat{\boldsymbol{u}} \in \widehat{\mathcal{U}}_{k-1}$, i.e., $\operatorname{wid}(\mathbf{g}_i) > 0$ holds for all $\mathbf{g}_i \in \hat{\boldsymbol{u}}$. Therefore, $\boldsymbol{u} \mapsto \hat{\boldsymbol{u}}$ defines a map $\mathcal{U}_k \to \widehat{\mathcal{U}}_{k-1}$.*

Proof. Because of (5), the strict inequality in (4), and since the bounds $\overline{u}_i$ and $\underline{u}_{i+1}$ are real numbers. ∎

LEMMA 14. *Let $\boldsymbol{u} \in \mathcal{U}_k$ and $\mathbf{x} \in \overline{\mathbb{IR}}$.*

1. $\boldsymbol{u} \cup \hat{\boldsymbol{u}} = \square \boldsymbol{u}$.

2. *The mapping* $\hat{}$ *is bijective* $\mathcal{U}_{k,\mathbf{x}} := \{\boldsymbol{u} \in \mathcal{U}_k \mid \square\boldsymbol{u} = \mathbf{x}\} \to \hat{\mathcal{U}}_{k-1,\mathbf{x}} := \{\hat{\mathcal{U}} \in \hat{\mathcal{U}}_{k-1} \mid \hat{\mathcal{U}} \subseteq \mathbf{x}\}$.

Definition 15. *Let* $\boldsymbol{u} \in \mathcal{U}_k \setminus \mathcal{U}_1$ *and* $\boldsymbol{g} \subseteq \hat{\boldsymbol{u}}$ *a set of gaps of* $\boldsymbol{u}$. *We define the gap filling* $\mathcal{F}(\boldsymbol{u},\boldsymbol{g}) \in \mathcal{U}_{k-|\boldsymbol{g}|}$ *as the unique interval union with* $\widehat{\mathcal{F}}(\boldsymbol{u},\boldsymbol{g}) = \hat{\boldsymbol{u}} \setminus \boldsymbol{g}$ *and* $\square\widehat{\mathcal{F}}(\boldsymbol{u},\boldsymbol{g}) = \square\boldsymbol{u}$, *i.e., we fill all the gaps from* $\boldsymbol{g}$ *in* $\boldsymbol{u}$.
We write $\mathcal{F}(\boldsymbol{u},\mathbf{g})$ *for* $\boldsymbol{g} = \{\mathbf{g}\}$ *and* $\mathcal{F}(\boldsymbol{u},\mathbf{g}_1,\dots,\mathbf{g}_\ell)$ *for* $\boldsymbol{g} = \{\mathbf{g}_1,\dots,\mathbf{g}_\ell\}$.

If $\mathbf{g}_i$ is the ith gap of $\boldsymbol{u}$ we get $\mathcal{F}(\boldsymbol{u},\mathbf{g}_i)$ by setting $\overline{u}_i := \overline{u}_{i+1}$ and removing the interval $\mathbf{u}_{i+1}$ from $\boldsymbol{u}$.

Lemma 16. *For* $\boldsymbol{u} \in \mathcal{U}$ *and* $\boldsymbol{g} \subseteq \hat{\boldsymbol{u}}$ *we have*

$$\boldsymbol{u} \subset \mathcal{F}(\boldsymbol{u},\boldsymbol{g}). \tag{6}$$

Proof. If $\boldsymbol{g} = \{\mathbf{g}_i\}$, by (4), $\overline{u}_i < \underline{u}_{i+1}$ therfore $\mathbf{u}_i \cup \mathbf{u}_{i+1} \subset [\underline{u}_i, \overline{u}_{i+1}]$, proving (6). Since $\mathcal{F}(\boldsymbol{u},\boldsymbol{g}) = \mathcal{F}(\boldsymbol{u},\boldsymbol{g}\setminus\{\mathbf{g}\})$ the general case follows by induction on the size of $\boldsymbol{g}$. □

Now we will introduce the concept of gap ordering to determine which gap to fill first. Usually, the width of the gap plays a part in that ordering (sometimes also a relative width with respect to the position of the interval along the real axis), and also the position of the gap might be interesting. Since we do not want to fix this ordering for developing the theory we will just assume that we are given a linear order $\trianglelefteq$ on the set of all open intervals of $\mathbb{R}$ with the property that for arbitrary $\mathbf{x} \in \mathbb{IR}$ every collection of open intervals contained in $\mathbf{x}$ has a maximal element w.r.t. $\trianglelefteq$.

Definition 17. *The index set of the* **n smallest gaps** *of* $\boldsymbol{u}$ *(w.r.t.* $\trianglelefteq$*) is defined by*

$$\mathcal{G}^S_n(\boldsymbol{u}) \subseteq \{1,\dots,k-1\},\ |\mathcal{G}^S| = n,\ \text{such that if } i \in \mathcal{G}^S \text{ then } \mathbf{g}_i \lhd \mathbf{g}_j \text{ for all } j \notin \mathcal{G}^S.$$

Similarly, the index set of the **n largest gaps** *of* $\boldsymbol{u}$ *(w.r.t.* $\trianglelefteq$*) is defined by*

$$\mathcal{G}^L_n(\boldsymbol{u}) \subseteq \{1,\dots,k-1\},\ |\mathcal{G}^L| = n,\ \text{such that if } i \in \mathcal{G}^L \text{ then } \mathbf{g}_i \rhd \mathbf{g}_j \text{ for all } j \notin \mathcal{G}^L.$$

For $r \in \{L,S\}$ *we denote by* $\mathfrak{g}^r_n(\boldsymbol{u}) := \{\mathbf{g}_i \in \hat{\boldsymbol{u}} \mid i \in \mathcal{G}^r_n(\boldsymbol{u})\}$ *the set of smallest respectively largest gaps of* $\boldsymbol{u}$. *For convenience we define* $\mathfrak{g}^r_n(\boldsymbol{u}) = \hat{\boldsymbol{u}}$ *if* $n \geq l(\boldsymbol{u})$ *and* $\mathfrak{g}^r_n(\boldsymbol{u}) = \emptyset$ *if* $n \leq 0$.

Definition 18. *We define the* ***length restriction mapping*** $\Gamma_k : \mathcal{U} \to \mathcal{U}_k$ *by* $\Gamma_k(\boldsymbol{u}) := \mathcal{F}(\boldsymbol{u}, \mathcal{G}^S_{l(\boldsymbol{u})-k}(\boldsymbol{u}))$, *i.e., we fill the* $l(\boldsymbol{u}) - k$ *smallest gaps of* $\boldsymbol{u}$, *and we do not change* $\boldsymbol{u}$ *if* $l(\boldsymbol{u}) \leq k$.

Defining the interval union hull of a set M of real numbers is not straightforward. Unfortunately, there is nothing like the smallest interval union of length k containing M. For bounded sets M we can get something like uniqueness by filling all but the largest gaps in M. If the set is unbounded, e.g., $M =]-\infty,0] \cup \bigcup_{j=-\infty}^{\infty}[2^{2j}, 2^{2j+1}]$, there may be gaps of arbitrary size. In the following definition, we will resolve that problem by fixing a bounded region $\mathbf{x}$ and filling all gaps that are not contained in $\mathbf{x}$. If M is bounded we can always choose $\mathbf{x} = \square M$.

Definition 19. *Fix* $\mathbf{x} \in \mathbb{IR}$ *and* $\mathbb{N} \ni k > 1$, *and let* $M \subseteq \mathbb{R}$ *and* $\overline{M}$ *its topological closure. Then* $M^c := \mathbf{x} \setminus \overline{M}$ *is a countable (possibly finite) union of open intervals. Let* $\widehat{M}^c$ *be the set of these intervals, and* $\hat{\boldsymbol{u}} \in \widehat{\mathcal{U}}_{k-1}$ *the subset of the* $k-1$ *largest elements of* $\widehat{M}^c$. *We define the* ***interval union hull*** $\mathcal{U}_{k,\mathbf{x}}(M)$ *of length* k *of* M *with respect to* $\mathbf{x}$ *as the unique interval union in* $\mathcal{U}_{k,\square M}$ *with* $\widehat{\mathcal{U}_{k,\mathbf{x}}}(M) = \hat{\boldsymbol{u}}$.

3.4. Arithmetic for Interval Unions. In this section, similarly to interval arithmetic, basic set and elementary operations as well as properties like inclusion isotonicity are defined and explained for interval unions. Most of the theory translates nicely from intervals to interval unions, but some properties do not: e.g., due to the lack of convexity it is not possible to prove a mean value theorem for interval unions.

Definition 20. *Let $\mathbf{x} \in \mathbb{IR}$ be an interval, $\boldsymbol{u} := (\mathbf{u}_1, \dots, \mathbf{u}_k)$ and $\boldsymbol{s} := (\mathbf{s}_1, \dots \mathbf{s}_t)$ interval unions. Define the index set J as $J := \{i \in \{1, \dots, k\} \mid \mathbf{u}_i \cap \mathbf{x} \neq \emptyset\}$ and for $J \neq \emptyset$ also define $\underline{J} := \min(J)$ and $\overline{J} := \max(J)$.*
*(i) The **union operation** for $\boldsymbol{u}$ and $\mathbf{x}$ is defined as $\boldsymbol{u} \cup \mathbf{x} := \mathcal{U}(\boldsymbol{u} \cup \{\mathbf{x}\})$. Obviously, we have*

$$(7)\quad \boldsymbol{u} \cup \mathbf{x} = \begin{cases} (\mathbf{u}_1, \dots, \mathbf{u}_i, \mathbf{x}, \mathbf{u}_{i+1}, \dots \mathbf{u}_k) \text{ where } \overline{u}_i < \underline{x} \text{ and } \overline{x} < \underline{u}_{i+1} & \text{if } J = \emptyset \\ (\mathbf{u}_1, \dots, \mathbf{u}_{\underline{J}-1}, [\min(\underline{u}_{\underline{J}}, \underline{x}), \max(\overline{u}_{\overline{J}}, \overline{x})], \mathbf{u}_{\overline{J}+1}, \dots \mathbf{u}_k) & \text{otherwise.} \end{cases}$$

(i') The union operation for $\boldsymbol{u}$ and $\boldsymbol{s}$ is defined by

$$(8)\qquad \boldsymbol{u} \cup \boldsymbol{s} := \boldsymbol{u} \cup \mathbf{s}_1 \cup \dots \cup \mathbf{s}_t.$$

*(ii) The **intersection operation** for $\boldsymbol{u}$ and $\mathbf{x}$ is defined as $\boldsymbol{u} \cap \mathbf{x} := \mathcal{U}(\{\mathbf{u}_1 \cap \mathbf{x}, \dots, \mathbf{u}_k \cap \mathbf{x}\})$. We have*

$$(9)\quad \boldsymbol{u} \cap \mathbf{x} = \begin{cases} \emptyset & \text{if } J = \emptyset \\ ([\max(\underline{u}_j, \underline{x}), \min(\overline{u}_j, \overline{x})]) & \text{if } J = \{j\} \\ ([\max(\underline{u}_{\underline{J}}, \underline{x}), \overline{u}_{\underline{J}}], \mathbf{u}_{\underline{J}+1}, \dots, \mathbf{u}_{\overline{J}-1}, [\underline{u}_{\overline{J}}, \min(\overline{u}_{\overline{J}}, \overline{x})]) & \text{otherwise.} \end{cases}$$

(ii') The intersection operation for $\boldsymbol{u}$ and S is defined by

$$(10)\qquad \boldsymbol{u} \cap \boldsymbol{s} := (\boldsymbol{u} \cap \mathbf{s}_1) \cup \dots \cup (\boldsymbol{u} \cap \mathbf{s}_t).$$

Note that there is a slight ambiguity in the notation, as $\boldsymbol{u} \cup \boldsymbol{s}$ can also denote the union of the two sets of intervals $\boldsymbol{u}$ and $\boldsymbol{s}$. However, there will be no confusion between these two concepts, as the same real set is represented.

Lemma 21. *Let $\mathbf{x} \in \mathbb{IR}$ be an interval, $\boldsymbol{u} := (\mathbf{u}_1, \dots, \mathbf{u}_k)$, $\boldsymbol{s} := (\mathbf{s}_1, \dots \mathbf{s}_t)$ interval unions.*
(i) For the union operation defined by (7) we have $x \in \boldsymbol{u} \cup \mathbf{x}$ iff $x \in \boldsymbol{u}$ or $x \in \mathbf{x}$.
(i') For the union operation defined by (8) we have $x \in \boldsymbol{u} \cup \boldsymbol{s}$ iff $x \in \boldsymbol{u}$ or $x \in \boldsymbol{s}$.
(ii) For the intersection operation defined by (9) we have $x \in \boldsymbol{u} \cap \mathbf{x}$ iff $x \in \boldsymbol{u}$ and $x \in \mathbf{x}$.
(ii') For the intersection operation defined by (10) we have $x \in \boldsymbol{u} \cap \boldsymbol{s}$ iff $x \in \boldsymbol{u}$ and $x \in \boldsymbol{s}$.

Definition 22. *Let $\mathbf{x} \in \mathbb{IR}$ be an interval, $\boldsymbol{u} := (\mathbf{u}_1, \dots, \mathbf{u}_k)$ and $\boldsymbol{s} := (\mathbf{s}_1, \dots \mathbf{s}_t)$ interval unions and let $\circ \in \{+, -, /, *, \hat{}\}$ be an elementary interval operation defined in Section 2.*
*(i) The **elementary interval union operation** corresponding to $\circ$ applied to $\boldsymbol{u}$ and $\mathbf{x}$ is given by*

$$\boldsymbol{u} \circ \mathbf{x} := \mathcal{U}(\{\mathbf{u}_1 \circ_\bullet \mathbf{x}, \dots, \mathbf{u}_k \circ_\bullet \mathbf{x}\})$$

(i') The elementary interval union operation corresponding to $\circ$ applied to $\boldsymbol{u}$ and $\boldsymbol{s}$ is given by

$$\boldsymbol{u} \circ \boldsymbol{s} := \mathcal{U}(\{\boldsymbol{u} \circ \mathbf{s}_1, \dots, \boldsymbol{u} \circ \mathbf{s}_t\})$$

Note that for the interval division operator (1) the above definition gives a natural embedding of the problematic cases into the set of interval unions: for arbitrary $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$ we have

$$\Big(\mathcal{U}(\{\mathbf{a}\})/\mathbf{b}\Big) \in \mathcal{U}.$$

LEMMA 23. *Let $\boldsymbol{u} := (\mathbf{u}_1, \ldots, \mathbf{u}_k)$ and $\boldsymbol{s} := (\mathbf{s}_1, \ldots \mathbf{s}_t)$ be interval unions then the elementary interval union operations $\circ \in \{+, -, /, *, \hat{}\}$ defined by (22) are inclusion isotonic:*

$$\boldsymbol{u} \subseteq \boldsymbol{u}' \text{ and } \boldsymbol{s} \subseteq \boldsymbol{s}' \Longrightarrow \boldsymbol{u} \circ \boldsymbol{s} \subseteq \boldsymbol{u}' \circ \boldsymbol{s}' \text{ for all } \{+, -, /, *, \hat{}\}.$$

Proof. The union creator $\mathcal{U}$ is inclusion isotonic by Lemma 8. Interval operations are inclusion isotonic by Section 2, therefore the composition of them is also inclusion isotonic. ∎

In addition to the usual definition of elementary operations we also introduce component-wise operations that will be useful in the context of interval union linear systems.

DEFINITION 24. *Let $\boldsymbol{u} := (\mathbf{u}_1, \ldots, \mathbf{u}_k)$ and $\boldsymbol{s} := (\mathbf{s}_1, \ldots \mathbf{s}_k)$ be interval unions of the same length and let $\circ \in \{+, -, /, *\}$ then the component-wise interval union operation corresponding to $\circ$ applied to $\boldsymbol{u}$ and $\boldsymbol{s}$ is given by*

$$\boldsymbol{u} \circ_c \boldsymbol{s} := \mathbf{u}_1 \circ \mathbf{s}_1 \cup \ldots \cup \mathbf{u}_k \circ \mathbf{s}_k.$$

In the following we will fix a "cutoff" $\mathbf{x} \in \mathbb{R}$ for filling the gaps as described before in Definition 19. If the function f in the following definition is well-behaved (e.g. piecewise continuous and the pre-image of every interval is bounded), then the result will not depend on $\mathbf{x}$ as long as $\mathbf{x}$ is big enough.

DEFINITION 25. *Let $\boldsymbol{u} \in \mathcal{U}^n$ be an interval union vector and $\boldsymbol{s} \in \mathcal{U}$ an interval union, and let $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$. For fixed $\ell > 1$ we define the range of length ℓ of f over $\boldsymbol{u}$ (w.r.t. $\mathbf{x}$) as*

$$\operatorname{rg}_\ell(f(\boldsymbol{u})) := \mathcal{U}_{\ell,\mathbf{x}}(\{\operatorname{rg}\bullet(f(\mathbf{u})) \mid \mathbf{u} \in \boldsymbol{u}\}) \tag{11}$$

and the kth **partial inverse image** *of length ℓ of f on $\boldsymbol{u}$ and $\boldsymbol{s}$ as*

$$f^{-1}_{\ell,k}(\boldsymbol{u}, \boldsymbol{s}) := \mathcal{U}_{\ell,\mathbf{x}}(\{f_{k\bullet}(\boldsymbol{v}, \mathbf{s}) \mid \boldsymbol{v} \in V, \mathbf{s} \in \boldsymbol{s}\}). \tag{12}$$

As in the interval case, we call a function $\mathbf{f} : \mathcal{U}^n \to \mathcal{U}$ **inclusion isotone** if $\boldsymbol{u}' \subseteq \boldsymbol{u} \Rightarrow \mathbf{f}(\boldsymbol{u}') \subseteq \mathbf{f}(\boldsymbol{u})$. Moreover, we say $\mathbf{f} : \mathcal{U}^n \to \mathcal{U}$ is the interval union extension of a function $f : D \subseteq \mathbb{R}^n \Rightarrow \mathbb{R}$ in $\boldsymbol{u} \in \mathcal{U}^n$ if

$$\mathbf{f}(x) = f(x) \quad \text{for } x \in D \cap \boldsymbol{u}, \text{ and } \quad f(x) \in \mathbf{f}(\boldsymbol{u}) \quad \text{for all } x \in D \cap \boldsymbol{u}.$$

We also refer to interval union extensions only as extensions when there is no possibility of misunderstandings. As in the interval case we can define a natural interval union extension for functions composed by elementary operations and standard function only by replacing real operations by their interval union counterparts. The following proposition states that the fundamental theorem of interval analysis can be naturally extended to interval unions.

PROPOSITION 26. *If $\mathbf{f}$ is inclusion isotonic and the interval union extension of f : $\mathbb{R}^n \to \mathbb{R}$ then $f_{rg}(\boldsymbol{u}) \subseteq \mathbf{f}(\boldsymbol{u})$.*

Proof. immediately from the application of the fundamental theorem of interval analysis to every component $\mathbf{u}_i$ of $\boldsymbol{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_k)$. ∎

On the other hand, due to the lack of convexity when working with interval unions we are not able to prove the interval union mean value theorem. For example, consider $f(x) = x^2$ and the interval union $\boldsymbol{u} = ([-3, -1], [1, 3])$. If we take $x = -2 \in [-3, -1]$ and $y = 2 \in [1, 3]$ then there is no $\xi \in \boldsymbol{u}$ such that $4 = 4 - 8\xi$, and hene the statement fails even for univariate functions.

4. Interval union Newton method.

In this section we consider the problem of rigorously enclosing all solutions of

$$f(x) \in \boldsymbol{z}, \quad x \in \boldsymbol{x} \tag{13}$$

where $f : \mathbb{R} \to \mathbb{R}$ is a differentiable function. In Section 4.1 we review the interval Newton method for the case where $\boldsymbol{z}$ is set to be zero and $\boldsymbol{x}$ is a closed and bounded interval. In Section 4.2 we formulate the interval union Newton operator. Numerical experiments comparing both approaches are presented in Section 4.3.

4.1. Interval Newton method.

Let $\mathbf{x}$ be a bounded interval and $f : \mathbb{R} \to \mathbb{R}$ a differentiable function. We are interested in enclosing all solutions of

$$f(x) = 0, \quad x \in \mathbf{x}. \tag{14}$$

Interval newton methods to solve this problem are based on the interval mean value theorem applied to (14). Formally, if $y \in \mathbf{x}$ such that $f(y) = 0$ then

$$0 = f(y) \in \mathbf{f}(x) + \mathbf{f}'(\mathbf{x})(y - x)$$

for any fixed $x \in \mathbf{x}$. Therefore, the solution set of the problem can be given as

$$\mathcal{S}_x := \{y \in \mathbf{x} \mid \exists f^* \in \mathbf{f}(x) \text{ and } g^* \in \mathbf{f}'(\mathbf{x}) \text{ such that } f^* + g^*(y - x) = 0\} \tag{15}$$

regardless of the choice of x. The usual interval Newton method fixes x as the midpoint of $\mathbf{x}$ and generates a sequence of nested intervals such that

$$\mathbf{x}_0 \supseteq \mathbf{x}_1 \supseteq \ldots \supseteq \mathcal{S}_x,$$

where

$$\mathbf{x}^{(k+1)} = N(\mathbf{x}^k) \cap \mathbf{x}^k, \quad k = 0, 1, 2, \ldots$$

The operator $N(\mathbf{x})$ is called **interval Newton function** and is given by

$$N(\mathbf{x}) = \check{\mathbf{x}} - \frac{\mathbf{f}(\check{\mathbf{x}})}{\mathbf{f}'(\mathbf{x})}. \tag{16}$$

Algorithms based on the interval Newton method can be divided in two groups depending on whether or not they rely on extended division, i.e. splitting intervals after the division into the two unconnected result intervals. Some authors like Moore [20] and Alefeld [2] only apply the interval Newton operator to boxes where $0 \notin \mathbf{f}'(\mathbf{x})$. More sophisticated algorithms like those proposed by Kearfott [15] and Hansen [7] allow division by intervals containing zero and process each box resulting from the division separately.

The simplest interval Newton method with extended division for enclosing all solutions of (14) is given in Algorithm 1. The algorithm takes the interval $\mathbf{x}$ and applies the interval Newton operator to it. If the resulting intervals are not empty or too thin then they are split, an interval to be processed is chosen and the iteration continues. The proof of finiteness and rigorousness of the interval Newton algorithm is given in [15]. For multivariate versions of this algorithm see [7–9].

Algorithm 1 Interval Newton algorithm

Input: The interval $\mathbf{x}_0$, the interval extensions $\mathbf{f}$ and $\mathbf{f}'$ of f and f' respectively and the narrow component tolerance $\epsilon > 0$.

Output: A list of intervals $\mathcal{C}$ with $\mathbf{x} \in \mathcal{C} \Rightarrow \mathrm{wid}(\mathbf{x}) < \epsilon$ and the guarantee that for all $y \in \mathbf{x}_0$ with $f(y) = 0$ there exists at least one interval $\mathbf{x} \in \mathcal{C}$ such that $y \in \mathbf{x}$.

$\mathcal{W} \leftarrow \mathbf{x}_0$;
while $\mathcal{W} \neq \varnothing$ **do**
 $\mathbf{x} \leftarrow$ get_first($\mathcal{W}$);
 $x \leftarrow \check{\mathbf{x}}$;
 $[\mathbf{x}_1, \mathbf{x}_2] \leftarrow \left(x - \frac{\mathbf{f}(x)}{\mathbf{f}'(\mathbf{x})}\right) \cap \mathbf{x}$; ▷ Newton operator
 for $i \leftarrow 1 : 2$ **do**
  if $\mathbf{x}_i \neq \varnothing$ **then** ▷ Elimination test
   if $0 \notin \mathbf{f}(\mathbf{x}_i)$ **then** continue
   else if $\mathrm{wid}(\mathbf{x}_i) < \epsilon$ **then** ▷ Solution test
    $\mathcal{C} \leftarrow \mathbf{x}_i$;
   else
    $\mathcal{W} \leftarrow [\underline{\mathbf{x}}_i, \check{\mathbf{x}}_i]$; $\mathcal{W} \leftarrow [\check{\mathbf{x}}_i, \overline{\mathbf{x}}_i]$;
   end if
  end if
 end for
end while
return $\mathcal{C}$;

The list of intervals $\mathcal{C}$ returned by the algorithm need not to be disjoint. Moreover, it is possible that the algorithm saves an interval $\mathbf{x}$ in $\mathcal{C}$ even when it contains no root of f. The only guarantee we have is that when $y \in \mathbf{x}$ satisfies $f(y) = 0$ then $y \in \mathbf{x}_i \subseteq \mathcal{C}$ for some i.

4.2. Interval union Newton method. Let $\boldsymbol{x}$ and $\boldsymbol{\mathfrak{r}}$ be interval unions with p and q elements respectively. Applying the interval mean value theorem to each pair of intervals in $\boldsymbol{x}$ and $\boldsymbol{\mathfrak{r}}$ gives the solution set of (13)

$$\mathcal{S} := \bigcup_{\substack{1<i<p \\ 1<j<q}} \mathcal{S}_x(\mathbf{x}_i, \mathbf{r}_j)$$

where

$$\mathcal{S}_x(\mathbf{x}, \mathbf{r}) := \{y \in \mathbf{x} \mid \exists r \in \mathbf{r},\ f^* \in \mathbf{f}(x) \text{ and } g^* \in \mathbf{f}'(\mathbf{x}) \text{ such that } f^* + g^*(y - x) = r\}$$

for any fixed $x \in \mathbf{x}$. Therefore, we can solve (13) by applying Algorithm 1 $p \times q$ times. However, the interval union arithmetic provides a more natural approach, without the need of running multiple instances of the same algorithm. Let $\boldsymbol{u}^k = (\mathbf{u}_1^k, \ldots, \mathbf{u}_n^k)$ be an interval union, f a differentiable function and $\mathbf{f}$ and $\mathbf{f}'$ interval union extensions of f and of its derivative f'. The interval union Newton iteration is given by

$$\boldsymbol{u}^{k+1} := (N(\mathbf{u}_1^k) \cap \mathbf{u}_1^k, \ldots, N(\mathbf{u}_n^k) \cap \mathbf{u}_n^k) \tag{17}$$

where $N(\mathbf{x})$ is the interval Newton function. Note that the interval union Newton iteration is rigorous since it is a component-wise application of the interval mean value theorem. Algorithm 2 uses (17) to enclose all solutions of (13). It also needs the auxiliary function *checkAndRemove* which is given in Algorithm 3. In the next section we perform numerical experiments to compare the performance of Algorithm 1 with Algorithm 2.

Algorithm 2 Interval union Newton algorithm

Input: The interval union $\boldsymbol{u}_0$, the interval union extensions $\mathbf{f}$ and $\mathbf{f}'$ of f and $f^{'}$ and the narrow component tolerance $\epsilon > 0$.

Output: The interval union $\mathscr{s} = (\mathbf{x}_i)$ with $\operatorname{wid}(\mathbf{x}_i) < \epsilon$ and the guarantee that for all $y \in \boldsymbol{u}_0$ with $f(y) = 0$ there exist an $\mathbf{x}_i$ such that $y \in \mathbf{x}_i$.

$\boldsymbol{u} \leftarrow \boldsymbol{u}_0$;
while $\boldsymbol{u} \neq \varnothing$ **do**
    $\boldsymbol{u} \leftarrow (N(\mathbf{u}_1) \cap \mathbf{u}_1, \ldots, N(\mathbf{u}_n) \cap \mathbf{u}_n)$; ▷ Newton operator
    $\boldsymbol{x} \leftarrow \varnothing$;
    for $\mathbf{x}_i \in \boldsymbol{u}$ **do**
        if $\mathbf{f}(\mathbf{x}_i) \cap \mathbf{r} \neq \varnothing$ **then** ▷ Elimination test
            if $\operatorname{wid}(\mathbf{x}_i) < \epsilon$ **then** ▷ Solution test
                $S \leftarrow \mathbf{x}_i$;
            else
                $\boldsymbol{x} \leftarrow$ checkAndRemove($\mathbf{x}_i, \epsilon, \mathbf{f}$);
            end if
        end if
    end for
    $\boldsymbol{u} \leftarrow \boldsymbol{x}$;
end while
return S;

Algorithm 3 Check and Remove

Input: The interval $\mathbf{x}$ and the narrow component tolerance ϵ

Output: An interval union $\boldsymbol{u}$ with two elements

$x \leftarrow \check{x}$; $\mathbf{y} \leftarrow [x - \frac{\epsilon}{2}, x + \frac{\epsilon}{2}]$;
if $\mathbf{f}(\mathbf{y}) \cap \mathbf{r} \neq \varnothing$ **then** Save $\mathbf{y}$ as solution of (13)
end if
$\boldsymbol{u} \leftarrow \{[\underline{\mathbf{x}}, \underline{\mathbf{y}}], [\overline{\mathbf{y}}, \overline{\mathbf{x}}]\}$;
return $\boldsymbol{u}$;

4.3. Numerical experiments. We compare interval and interval union Newton methods for univariate functions using a *Java* implementation that is part of *JGloptlab* [5]. We used 32 test functions listed in Table 1 most of them taken from [3]. For each function we consider the natural extensions for both f and f'. In our implementation, we have followed the pseudo-codes of Algorithms 1 and 2 precisely, without any additional acceleration or optimization.

For each function f_i we seek the enclosure of all solutions $f_i(x) = 0$ where $x \in \mathbf{x}$ and $\mathbf{x}$ is a bounded interval. The narrow component tolerance ϵ is set to 10^{-7} and the maximum number of function evaluations is set to 100000. If we are unable to reduce the width of every component of the solution set below ϵ before the maximum number of function evaluations is reached we relax the tolerance parameter by a factor of 10 and restart the process. Table 1 shows the test functions and the Table 2 present the results of the experiment. A supplementary table comparing other aspects of both algorithms can be found in http://www.mat.univie.ac.at/~dferi/research/UnionsTests.pdf.

$f_1 = -\sum_{k=1}^{5} k\sin((k+1)x+k),\ [-100,100],$	$f_2 = 1+x+x^2+x^3+x^4-x^5,\ [-2,2]$
$f_3 = \sin(x) - 2\cos(x^2-1),\ [-100,100],$	$f_4 = 1-\cos(x)+\frac{x^2}{4000},\ [-100,100]$
$f_5 = (x+\sin(x))\exp(-x^2),\ [-100,100],$	$f_6 = x(1-x),\ [-6,6]$
$f_7 = x^4-10x^3+35x^2-50x+24,\ [-100,100],$	$f_8 = \exp(-3x)-\sin^3(x),\ [0,100]$
$f_9 = \sin(x)+\sin(\frac{10x}{3})+\ln(x)-0.84x,\ [1,100],$	$f_{10} = \sin(x),\ [-100,100]$
$f_{11} = 24x^4-142x^3+303x^2-276x+93,\ [-100,100],$	$f_{12} = \sin(\frac{1}{x}),\ [0.02,100]$
$f_{13} = 2x^2-\frac{3}{100}\exp(-200(x-0.0675)^2),\ [1,100],$	$f_{14} = \frac{x^2}{20}-\cos(x)+2,\ [-100,100]$
$f_{15} = \sin(1+x+x^2+x^3+x^4),\ [-20,20],$	$f_{16} = x^2-\cos(18x),\ [-100,100]$
$f_{17} = (x-1)^2(1+10\sin^2(x+1))+1,\ [-100,100],$	$f_{18} = \exp(x^2),\ [-10,10]$
$f_{19} = x^4-12x^3+47x^2-60x-20\exp(-x),\ [-10,10],$	$f_{20} = x^6-15x^4+27x^2+250,\ [-10,10]$
$f_{21} = \sin^2\left(1+\frac{x-1}{4}\right)+\left(\frac{x-1}{4}\right)^2,\ [-100,100],$	$f_{22} = (x-x^2)^2+(x-1)^2,\ [-100,100]$
$f_{23} = \exp(\sin(x))+\cos(x^2),\ [-100,100],$	$f_{24} = \cos(\sin(x^2-1)-1),\ [-20,20]$
$f_{25} = \sin(\cos(\exp(x))),\ [0,10],$	$f_{26} = -\frac{1}{(x-2)^2+3},\ [0,100]$
$f_{27} = \cos(x^2-x^3),\ [-10,10],$	$f_{28} = \sin(\exp(x)),\ [0,10]$
$f_{29} = \cos(\pi(8x^3-1))+\sin(\pi(8x^2-1)),\ [-20,20],$	$f_{30} = \frac{1}{x},\ [-10,10]$
$f_{31} = \tan(x),\ [-10,10],$	$f_{32} = \cot(x),\ [-10,10]$

Table 1: The test functions $f_1 - f_{32}$ and the corresponding initial bounds for the variable x.

The test results are given for both the interval Newton (Algorithm 1 in column `INewton`) and for the interval union Newton (Algorithm 2 in column `IUNewton`). In particular, Table 2 shows for each test function (`func`) the number of boxes possibly containing solutions found (`Sol`), the number of function evaluations needed to enclose all solutions (`FunEv`) and the narrow component tolerance ϵ (`Wid`) used.

It is clear from that the interval union arithmetic significantly increases the efficiency of the Newton method. Table 2 shows that both the number of function evaluations and the number of boxes possibly containing solutions is smaller when using the interval union Newton method. Moreover, the tolerance achieved with the interval union method is, in every case, at least as small as the tolerance achieved with interval

fun	INewton			IUNewton			fun	INewton			IUNewton		
	Sol	FunEv	Wid	Sol	FunEv	Wid		Sol	FunEv	Wid	Sol	FunEv	Wid
f_1	3212	6424	10.0	410	6883	1E-7	f_2	3	164	1E-7	1	39	1E-7
f_3	3454	6916	10.0	6367	82782	1E-7	f_4	2	97	1E-7	1	37	1E-7
f_5	23832	95393	1E-2	3	59629	1E-2	f_6	2	38	1E-7	2	39	1E-7
f_7	14673	38638	0.1	7	367	1E-7	f_8	11521	96463	1	32	1931	1E-7
f_9	2	67	1E-7	2	50	1E-7	f_{10}	778	1569	10.0	63	893	1E-7
f_{11}	8082	22861	0.1	0	227	1E-7	f_{12}	5397	63841	1E-7	15	213	1E-7
f_{13}	0	1	1E-7	0	2	1E-7	f_{14}	0	1	1E-7	0	3	1E-7
f_{15}	15306	31865	1	15712	57924	1E-3	f_{16}	786	10218	1E-7	10	175	1E-7
f_{17}	0	1	1E-7	0	3	1E-7	f_{18}	0	1	1E-7	0	3	1E-7
f_{19}	1150	3319	1	8	339	1E-7	f_{20}	15772	73030	0.1	0	105	1E-7
f_{21}	0	28	1E-7	0	13	1E-7	f_{22}	1	123	1E-7	1	101	1E-7
f_{23}	3071	6340	10.0	3187	43862	1E-7	f_{24}	13362	30544	1	254	3757	1E-7
f_{25}	379	777	1	7011	77237	1E-7	f_{26}	0	1	1E-7	0	3	1E-7
f_{27}	3656	7312	10.0	20093	70984	1E-2	f_{28}	373	776	1	7011	72631	1E-7
f_{29}	15966	32320	1	17992	65801	1E-3	f_{30}	0	2	1E-7	0	1	1E-7
f_{31}	8	131	1E-7	7	117	1E-7	f_{32}	6	91	1E-7	6	109	1E-7

Table 2: Comparison between the interval and the interval union Newton method. The number of solutions obtained with each method is given in `Sol`, the number of function evaluations in `FunEv` and the final tolerance is given in `Wid`.

Newton method.

5. Systems of Interval Union Equations. This section extends the concept of interval linear systems to interval unions. The algorithms used to solve interval linear systems can be naturally adapted to the interval union case with a few modifications. The basic definitions of interval union linear systems are given in 5.1, the Gaussian elimination and the Gauss-Seidel algorithm are discussed in 5.2, finally in 5.3 some examples are given to demonstrate the usefulness of the interval union approach.

5.1. Basics. Let $\boldsymbol{\mathcal{A}} \in \mathcal{U}^{n\times n}$ be an interval union matrix and $\boldsymbol{\mathcal{b}} \in \mathcal{U}^n$ an interval union vector. An interval union linear system of equations is the family of linear systems given by

$$\tilde{A}x = \tilde{b} \quad \text{for all } \tilde{A} \in \boldsymbol{\mathcal{A}} \text{ and } \tilde{b} \in \boldsymbol{\mathcal{b}}. \tag{18}$$

The solution set of interval union linear systems is the union of solution sets from every combination of interval matrices and vectors contained in $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{b}}$, formally we have

DEFINITION 27. *The set $\mathcal{S} := \{x \in \mathbb{R}^n \mid \tilde{A}x = \tilde{b} \quad \text{for all } \tilde{A} \in \boldsymbol{\mathcal{A}} \text{ and } \tilde{b} \in \boldsymbol{\mathcal{b}}\}$ is the* **solution set** *of (18).*

If $\boldsymbol{\mathcal{A}} \in \mathcal{U}_1^{n\times n}$ and $\boldsymbol{\mathcal{b}} \in \mathcal{U}_1^n$ then problem (18) reduces to a typical interval linear system. Finding the interval hull of the solution set is $NP-$Hard for general interval linear systems and therefore it is also $NP-$Hard to find the interval hull of $\mathcal{S}$.
We say that a square interval matrix $\mathbf{A}$ is regular if every matrix $A \in \mathbf{A}$ is non-singular. In the same way, **the interval union matrix $\boldsymbol{\mathcal{A}}$ is regular** if every real matrix $A \in \mathbf{A}$ with $\mathbf{A} \in \boldsymbol{\mathcal{A}}$ is non-singular. The interval matrix $\mathbf{A} \in \mathcal{U}_1^{n\times n}$ is diagonally dominant if

$$\langle \mathbf{a}_{ii} \rangle \geq \sum_{\substack{1<i<p \\ 1<j<q}} |\mathbf{a}_{ij}|, \text{ for all } i = 1, \ldots, n. \tag{19}$$

The interval union matrix $\boldsymbol{\mathcal{A}}$ is **diagonally dominant** if relation (19) remains valid when we replace interval operations with interval union operations.

In general, algorithms for solving interval linear systems of equations benefit greatly from preconditioning. We say that the interval linear system $\mathbf{A}'x = \mathbf{b}'$ is preconditioned if

$$\mathbf{A}' = M\mathbf{A}, \quad \mathbf{b}' = M\mathbf{b}$$

where M is a real matrix. Typically $M = \check{\mathbf{A}}^{-1}$ is chosen, but some authors suggests better strategies for choosing M, see for example [15]. Similarly, algorithms for solving interval union linear systems may also take advantage of preconditioning, however, the choice of the preconditioning matrix is harder than in the interval case. The study of this topic will be addressed in a future work.

5.2. Algorithms. Let $\boldsymbol{\mathcal{A}}$ be an interval union matrix and and $\boldsymbol{b}$ an interval union vector. We present two methods to enclose the solution set $\mathcal{S}$ given by Definition 27. The algorithms discussed here can be easily generalized to the case where $\boldsymbol{\mathcal{A}}$ is not square.

Interval Gaussian elimination, as described in [8, 15], is obtained by just replacing real operations with interval ones in the Gaussian elimination algorithm. The interval version of the algorithm also allows to perform partial or full pivoting using the mignitude for element comparison. As proved in [15], the fundamental theorem of interval arithmetic guarantees that if $\mathbf{x}$ is the interval vector obtained with interval Gaussian elimination then $\mathcal{S} \subseteq \mathbf{x}$. Since the fundamental theorem of interval union arithmetic is already proved, the same conclusion holds if we replace all real operations with interval union counterparts in the Gaussian elimination. Moreover, the definition of the mignitude for interval unions allows the same pivoting strategies as in the interval case.

Consider $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{b}$ of the form

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \boldsymbol{a}_{11} & \boldsymbol{a}_{12} \\ \boldsymbol{a}_{21} & \boldsymbol{a}_{22} \end{pmatrix} \text{ and } \boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix}. \tag{20}$$

The interval union Gaussian elimination with backward substitution and without pivoting gives

$$\boldsymbol{q} = \frac{-\boldsymbol{a}_{21}}{\boldsymbol{a}_{11}}, \; \boldsymbol{x}_2 = \frac{\boldsymbol{b}_2 + \boldsymbol{b}_1\boldsymbol{q}}{\boldsymbol{a}_{22} + \boldsymbol{a}_{12}\boldsymbol{q}}, \quad \boldsymbol{x}_1 = \frac{\boldsymbol{b}_1 - \boldsymbol{a}_{12}\boldsymbol{x}_2}{\boldsymbol{a}_{11}}.$$

It is trivial to generalize the Gaussian elimination to higher dimensions, but the two dimensional case is good enough to show some interesting properties of the Gaussian elimination applied to interval union systems.

Let us first assume that every entry of (20) is an interval instead of an interval union. In this case, if $0 \in \boldsymbol{a}_{11}$ then the interval Gaussian elimination will fail even with extended division. However, as demonstrated on Example 28 below, using interval union arithmetic we may obtain useful bounds for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ even if $0 \in \boldsymbol{a}_{11}$.

Even for systems with $0 \notin \boldsymbol{a}_{11}$ the union Gaussian elimination may give us sharper bounds for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ than the interval Gauss-Seidel algorithm. This is demonstrated on Example 29 below, where by using interval union Gaussian elimination we obtain bounds almost as sharp as solving several interval linear sub-systems separately.

During the interval Newton method, in each iteration, we have to solve an interval linear system of form $\mathbf{A}(\mathbf{x} - x) = \mathbf{b}$ where $\mathbf{x}$ is the box currently processed, $\mathbf{A}$ is the interval matrix given by evaluating the Jacobian of the function f over $\mathbf{x}$ and $\mathbf{b}$ is usually set to $-\mathbf{f}(x)$. The usual approach this system is the interval Gauss-Seidel algorithm that is based on the so called **Gauss-Seidel operator**

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k \cap \mathbf{y}_i, \quad i = 1 \ldots n, \tag{21}$$

where

$$\mathbf{y}_i = \mathbf{b}_i + \frac{\mathbf{r}_i}{\mathbf{a}_{ii}}, \quad \mathbf{r}_i = \mathbf{b}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \mathbf{a}_{ij}(\mathbf{x}_j \ominus \mathbf{x}_j),$$

and $\ominus$ is the interval inner subtraction as defined by (2). The interval Gauss-Seidel algorithm applies equation (21) as long the bounds of the processed box are improved. In practice, we iterate as long as the difference between the largest widths of $\mathbf{x}^{k+1}$ and $\mathbf{x}^k$ is bigger than a given tolerance ϵ, see Algorithm 4.

Algorithm 4 Interval Gauss-Seidel

Input: The interval matrix $\mathbf{A}$, the interval vectors $\mathbf{b}$ and $\mathbf{x}$ and the tolerance $\epsilon \geq 0$
Output: The interval vector $\mathbf{y}$ such that $\mathcal{S} \subseteq \mathbf{y} \subseteq \mathbf{x}$ or a proof that $\mathbf{x} \cap \mathcal{S} = \varnothing$.

1: $\mathbf{y} \leftarrow \mathbf{x}$ and $x \leftarrow \check{\mathbf{x}}$;
2: **while** true **do**
3:     **for** $i = 1, \ldots, n$ **do**
4:         **if** $0 \notin \mathbf{a}_{ii}$ **then**
5:             $\mathbf{r}_i \leftarrow \mathbf{b}_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} \mathbf{a}_{ij}(\mathbf{y}_j \ominus x_j))$;
6:             $\mathbf{y}_i^{'} \leftarrow x_i + \frac{\mathbf{r}_i}{\mathbf{a}_{ii}}$;
7:             $\mathbf{y}_i^{'} \leftarrow \mathbf{y}_i \cap \mathbf{y}_i$;
8:             **if** $\mathbf{y}_i^{'} == \varnothing$ **then**
9:                 return $\varnothing$;
10:             **end if**
11:         **end if**
12:     **end for**
13:     **if** $\max \operatorname{wid}(\mathbf{y}) - \max \operatorname{wid}(\mathbf{y}^{'}) < \epsilon$ **then**
14:         break;
15:     **end if**
16:     $\mathbf{y} \leftarrow \mathbf{y}^{'}$ and $x \leftarrow \check{\mathbf{y}}$;
17: **end while**
18: return $\mathbf{y}$;

Note that Algorithm 4 does not update the variables x_i when $0 \in \mathbf{a}_{ii}$. When this happens several authors (see [7], [15]) suggest a second step of the Gauss-Seidel algorithm which is based on the extended interval division (1). The second step consists of applying equation (21) to all indices i for which $0 \in \mathbf{a}_{ii}$ and then save the largest gap produced by the interval division. Then two boxes that are identical in every entry except for the one with the largest gap are returned.

Based on Algorithm 4 the interval union version of the Gauss-Seidel elimination can be formulated, where the interval union version of the Gauss-Seidel operator (21) is applied to every equation. The interval union Gauss-Seidel procedure differs from Algorithm 4 in steps 1 and 16. They be modified to use the component-wise interval union midpoint instead of the interval midpoint, since this is necessary in order to guarantee that the interval union fundamental theorem holds for $\mathbf{r}_i$.

As a natural consequence, Algorithm 4 with interval unions returns an interval union vector which stores not only the boxes with the largest gap but all gaps. This simple modifications can lead to significant improvements over the interval Newton procedures for multivariate functions.

5.3. Examples. We conclude the section by showing some advantages of using interval union arithmetic to solve interval or interval union linear systems.

EXAMPLE 28. *Let* $\mathbf{A}$ *and* $\mathbf{b}$ *be an interval matrix and an interval vector given by*

$$\mathbf{A} = \begin{pmatrix} [3.5, 4.5] & [1.0, 2.0] \\ [1.0, 2.0] & [-0.5, 0.5] \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} [1.0, 2.0] \\ [1.5, 2.0] \end{pmatrix}.$$

The interval Gaussian elimination will fail to enclose the solution set of $\mathbf{A}x = \mathbf{b}$ even with preconditioning. The function *verifylss* of *Intlab* [24] also fails and return $[-\infty, \infty]^2$ as solution. If intervals are replaced by interval unions in the standard Gaussian elimination, even without preconditioning we obtain the solution

$$x \in \boldsymbol{u} = (\{[-\infty, 0.204082], [0.270531, \infty]\},\ \{[-\infty, -0.217391], [1.28571, \infty]\})^T. \tag{22}$$

Now as (22) suggests (and shown in Figure 1-left) $\mathcal{S}$ may be split into four disjoint sets, and we see that the Gaussian elimination with interval unions provided useful information about $\mathcal{S}$ even though $\mathbf{A}$ is not regular.

EXAMPLE 29. *Now let* $\boldsymbol{\mathcal{A}}$ *be an interval union matrix and* $\mathbf{b}$ *an interval vector given by*

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \{[-5, -3], [4, 5]\} & [0.5, 1.0] \\ [0.5, 1.0] & \{[-3, -2], [2, 3]\} \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} [1.0, 2.0] \\ [1.5, 2.0] \end{pmatrix}.$$

The solution set of $\boldsymbol{\mathcal{A}}x = \mathbf{b}$ is the union of each interval linear system $\mathbf{A}_i x = \mathbf{b}$ for $i = 1, \dots, 4$. Figure 1-right shows the result of applying the interval union Gaussian elimination and the Gauss-Seidel algorithm to $\boldsymbol{\mathcal{A}}x = \mathbf{b}$ as well as the interval hull of each interval linear system. Note again that the Gauss-Seidel procedure overestimates the bounds of the interval hull while Gaussian elimination give us a sharp enclosure of the four sets. The reason for this is that every interval matrix $\mathbf{A} \in \boldsymbol{\mathcal{A}}$ is regular and diagonally dominant. Our final example shows how the multivariate interval Newton method can benefit from interval union analysis.

EXAMPLE 30. *Assume that we want to enclose the solution set of*

$$x_1^2 + x_2^2 - 1 = 0, \quad x_1^2 - x_2 = 0, \quad x := \begin{pmatrix} x_1 & x_2 \end{pmatrix}^T \in ([0, 0.9482],\ [-1.2502, 0])^T.$$

We use the Gauss-Seidel algorithm applied to the interval Newton operator and precondition the Jacobian matrix by the inverse of its midpoint as described by [8]. It gives

$$x \in \mathbf{x}' = ([0, 0.9482],\ [-1.2502, -0.8486])^T$$

and

$$x \in \mathbf{x}'' = ([0, 0.9482],\ [-0.2896, 0.0000])^T.$$

Despite the significant improvement in the resulting box, the result is still not optimal. Applying the interval union Gauss-Seidel algorithm we have

$$x \in \boldsymbol{u} = (\{[0, 0.1933], [0.825, 0.9482]\},\ \{[-1.2502, -0.8486], [-0.2896, 0]\})^T$$

Using the interval Gauss-Seidel algorithm we have achieved a 45% contraction of the search domain. On the other hand, applying the interval union procedure we reduced the bounds of both variables, and achieved a 81% contraction of the search domain.

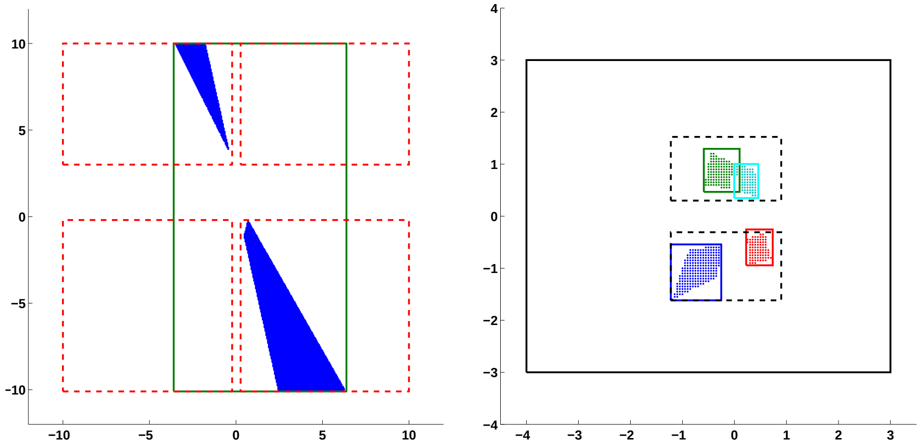

Fig. 1: Left - Solution set of Example 28 in the box $[-10, 10]^2$. The solution obtained by the interval Gauss-Seidel is given in the solid box. The solution obtained by the Gaussian elimination is given by dashed boxes. Right - Solution set of Example 29 in the box $[-10, 10]^2$. Gauss-Seidel solution is given in the outer solid box, Gaussian elimination is represented by dashed boxes. The solution set of each interval system and its interval hull is given by the inner solid boxes.

References.

[1] C. S. Adjiman, I. P. Androulakis, C. D. Maranas, and C. A. Floudas, *A global optimization method αBB for process design*, Computers and Chemical Engineering, 20 (1996), pp. 419–424.

[2] G. Alefeld and J. Herzberger, *Introduction to Interval Computations*, Academic Press, 1984.

[3] A. Baldwin, *Parallel Global Optimization Using Interval Analysis*, PhD thesis, University of South Dakota, Vermillion, SD, USA, 2011.

[4] F. Domes, *GloptLab – a configurable framework for the rigorous global solution of quadratic constraint satisfaction problems*, Optimization Methods and Software, 24 (2009), pp. 727–747, http://www.mat.univie.ac.at/~dferi/research/Gloptlab.pdf.

[5] F. Domes, *JGloptLab – a rigorous global optimization software*. in preparation, 2016, http://www.mat.univie.ac.at/~dferi/publications.html.

[6] C. Grandon, D. Daney, and Y. Papegay, *Combining CP and interval methods for solving the direct kinematic of a parallel robot under uncertainties*. IntCP 06 Workshop, 2006, ftp://ftp-sop.inria.fr/coprin/daney/articles/intcp06.pdf.

[7] E. R. Hansen, *Global Optimization Using Interval Analysis*, Marcel Dekker Inc., New York, 1992.

[8] E. R. Hansen, *A multidimensional interval newton method*, Reliable Computing, 12 (2006), pp. 253–272.

[9] E. R. Hansen and R. I. Greenberg, *An interval newton method*, Applied Mathematics And Computation, 12 (1983), pp. 89–98.

[10] L. Jaulin, *Interval constraints propagation techniques for the simultaneous localization and map building of an underwater robot*, 2006, http://www.mat.univie.ac.at/~neum/glopt/gicolag/talks/jaulin.pdf.

[11] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, *Guaranteed nonlinear estimation using constraint propagation on sets*, International Journal of Control, 74 (1999), pp. 1772–1782, https://www.ensieta.fr/e3i2/Jaulin/observer.pdf.

[12] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag, 2001.

[13] R. B. Kearfott, *Interval arithmetic: A fortran 90 module for an interval data type*, ACM Trans. Math. Software, 22 (1996), pp. 385–392.

[14] R. B. Kearfott, *Interval Computations: Introduction, Uses, and Resources*, Euromath, Bulletin 2 (1996), pp. 95–112.

[15] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, 1996.

[16] R. B. Kearfott, *Interval computations, rigour and non-rigour in deterministic continuous global optimization*, Optimization Methods Software, 26 (2011), pp. 259–279, http://dx.doi.org/10.1080/10556781003636851.

[17] R. B. Kearfott, M. T. Nakao, A. Neumaier, S. M. Rump, S. P. Shary, and P. van Hentenryck, *Standardized notation in interval analysis*, in Proc. XIII Baikal International School-seminar "Optimization methods and their applications", vol. 4, Irkutsk: Institute of Energy Systems, Baikal, 2005, pp. 106–113.

[18] L. Krippahl and P. Barahona, *PSICO: Solving protein structures with constraint programming and optimization*, Constraints, 7 (2002), pp. 317–331, http://ssdi.di.fct.unl.pt/~pb/papers/ludi_constraints.pdf.

[19] J.-P. Merlet, *Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis*, International Journal of Robotics Research, 23 (2004), pp. 221–235, http://www-sop.inria.fr/coprin/equipe/merlet/Papers/IJRR2004.pdf.

[20] R. E. Moore, *Interval analysis*, Prentice-Hall, 1966.

[21] M. Nehmeier, *filib++, expression templates and the coming interval standard*, Reliable Computing, 15 (2011), pp. 312–320.

[22] A. Neumaier, *Interval methods for systems of equations*, vol. 37 of Encyclopedia of Mathematics and its Applications, Cambridge Univ. Press, Cambridge, 1990.

[23] D. Ratz, *Inclusion isotone extended interval arithmetic*, tech. report, Institut für Angewandte Mathematik, Karlsruhe, 1996, http://digbib.ubka.uni-karlsruhe.de/volltexte/67997.

[24] S. M. Rump, *INTLAB – INTerval LABoratory*, 1998 - 2008, http://www.ti3.tu-harburg.de/~rump/intlab/.

[25] N. V. Sahinidis, *BARON 12.1.0: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2013, http://www.gams.com/dd/docs/solvers/baron.pdf.

[26] H. Schichl, *Global optimization in the coconut project*, in Numerical Software with Result Verification, R. Alt, A. Frommer, R. Kearfott, and W. Luther, eds., vol. 2991 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 243–249, doi:10.1007/978-3-540-24738-8_14, http://dx.doi.org/10.1007/978-3-540-24738-8_14.

[27] H. Schichl, M. C. Markót, A. Neumaier, X.-H. Vu, and C. Keil, *The COCONUT Environment*, 2000-2010, http://www.mat.univie.ac.at/coconut-environment. Software.

[28] P. G. Szabó, M. C. Markót, and T. Csendes, *New Approaches to Circle Packing in a Square: With Program Codes*, Optimization and Its Applications, Springer, Dordrecht, 2007.