COMPUTER ALGEBRA SUPPORT FOR THE COMPLETED SET OF INTERVALS

Yilmaz Akyildiz
Mathematics Department, Bosphorus University
81815 Bebek, Istanbul, Turkey.
e-mail: akyildiz@boun.edu.tr

Evgenija D. Popova
Institute of Mathematics & Informatics, Bulgarian Academy of Sci.
"Acad. G. Bonchev" str., block 8, BG-1113 Sofia, Bulgaria
e-mail: epopova@iph.bio.bas.bg

Christian P. Ullrich
Institute for Informatics, University of Basel
Mittlere Str. 142, CH-4056 Basel, Switzerland
e-mail: ullrich@ifi.unibas.ch

Abstract

Being algebraic completion of the conventional interval arithmetic, the arithmetic on extended (proper and improper) intervals possesses group and other algebraic properties suitable for implementation in computer algebra systems. We give an overview of a *Mathematica* package `directed.m` supporting the completed set of intervals and discuss the benefits, provided by this package, for doing numeric and symbolic-algebraic interval computations.

Keywords: Interval Arithmetic, Implementation, Computer Algebra, *Mathematica*

1 Introduction

Several extensions of the classical interval arithmetic [2] have been proposed but the only providing an algebraic completion of interval arithmetic is that leading to the set of generalised (proper and improper) intervals. First developed by H.-J. Ortolf [10], E. Kaucher [5, 6] and E. Gardeñes [4], further investigated by S. Markov [8, 9] and others, generalised interval arithmetic is obtained as an extension of the set of normal (proper) intervals by improper intervals and a corresponding extension of the definitions of the interval arithmetic operations. The corresponding extended interval arithmetic structure possesses group properties with respect to addition and multiplication and a number of other advantages.

Consider the algebraic completion $D = \{[a^-, a^+] \mid a^-, a^+ \in R\} = IR \cup \overline{IR}$ of conventional interval arithmetic, obtained as an extension of the set of normal (proper) intervals $IR = \{[a^-, a^+] \mid a^- \leq a^+,\ a^-, a^+ \in R\}$ by improper intervals $\overline{IR} = \{[a^-, a^+] \mid a^- \geq a^+,\ a^-, a^+ \in R\}$, a corresponding extension of the inclusion order relation

$$A \subseteq B \iff (b^- \leq a^-) \text{ and } (a^+ \leq b^+), \quad \text{for } A, B \in D, \tag{1}$$

interval arithmetic and lattice operations [4, 6].

Dual is an important operator that reverses the end-points of the intervals and expresses an element-to-element symmetry in D. For $A = [a^-, a^+] \in D$ "dual" is defined by

$$\texttt{Dual}[A] = A_- = [a^+, a^-].$$

We utilize functional "$\pm$" notations: a^λ for the interval end-points and A_λ for the intervals, where $\lambda \in \Lambda = \{+,-\}$, so that extended interval formulas can be written in a closed form. The binary variable λ can be expressed as a "product" of two or more binary variables, $\lambda = \mu\nu$, $\mu, \nu \in \Lambda$, defined by $++ = -- = +$, and $+- = -+ = -$.

The extension of the arithmetic and lattice interval operations from the familiar set of normal intervals [2] into D leads to extended interval arithmetic. Corresponding interval arithmetic formulae can be found in [4, 6] (in table form) and [12]–[14] (in functional "$\pm$" notations). The restrictions of the extended interval arithmetic operations in the conventional interval space produce the familiar operations [2] for normal intervals.

The substructures $(D, +, \subseteq)$ and $(D \setminus \mathcal{T}, \times, \subseteq)$, where $\mathcal{T} = \{A \in D \mid A = [0,0] \text{ or } a^-a^+ < 0\}$ are isotone groups [6]. Hence, there exist unique inverse elements ($-A_-$ and $1/B_-$) with respect to the operations $+$ and $\times$ such that

$$A - A_- = 0 \quad \text{and} \quad B/B_- = 1. \tag{2}$$

Dual operator is distributive with respect to the arithmetic operations.

$$(A_1 \circ \ldots \circ A_n)_- = (A_1)_- \circ \ldots \circ (A_n)_-, \qquad A_i \in D,\ i = 1, \ldots, n;\ \circ \in \{+,-,\times,/\} \tag{3}$$

Recently, a full characterization of the distributivity relations on multiplication and addition of generalised intervals has been done [12].

The following functionals characterize extended intervals and most of the interval arithmetic properties. For an interval $A \in D$ define "direction" $\tau : D \to \Lambda$ and for an interval $A \in D \setminus \mathcal{T}$ define "sign" $\sigma : D \setminus \mathcal{T} \to \Lambda$ by

$$\tau(A) = \begin{cases} +, & \text{if } a^- \leq a^+, \\ -, & \text{if } a^- \geq a^+. \end{cases} \qquad \sigma(A) = \begin{cases} +, & \text{if } a^{-\tau(A)} \geq 0, \\ -, & \text{if } a^{\tau(A)} \leq 0. \end{cases}$$

An generalised interval A is called proper, if $\tau(A) = +$ and improper otherwise. With every interval $A \in D$ we can associate a proper interval $A_{\tau(A)} = [a^{-\tau(A)}, a^{\tau(A)}]$ where $a^{-\tau(A)} \leq a^{\tau(A)}$. In particular, σ is well defined over $R \setminus 0$. For $A \in \mathcal{T}$, $\sigma(A) = 0$.

Functional $\chi : D \to [-1, 1]$ is defined by $\chi([0,0]) = -1$ and

$$\chi(A) = \begin{cases} a^-/a^+, & \text{if } |a^-| \leq |a^+|, \\ a^+/a^-, & \text{otherwise.} \end{cases}$$

Functional χ admits the geometric interpretation that A is more symmetric than B iff $\chi(A) \leq \chi(B)$. For $A \in D$ we have $\chi(A_-) = \chi(A)$.

Definition of norm and metric, as well as many topological properties of $(D, +, \times, \subseteq)$ are given in [6]. Some other properties of the extended interval arithmetic can be found in [4, 5]. In order to emphasize that an extended interval can be considered also as a proper interval (in set-theoretical sense) coupled with "direction", sometimes the algebraic extension of the conventional interval arithmetic is called directed interval arithmetic. A technique [8] for projecting intervals from D onto conventional interval space allows an interpretation of the improper intervals and corresponding extended interval arithmetic assertions in terms of the interval arithmetic space of proper intervals. Generalised interval arithmetic seems to be useful and quite promising for a straightforward computation of inner and outer inclusion of functional ranges, for the effective solution of some interval algebraic problems and some problems related to interpolation and identification under uncertainties, control theory etc.

Although extended interval arithmetic has been implemented as a PASCAL-XSC module [11], the most natural environment for its implementation is a powerful and versatile computer algebra system like *Mathematica* [15], where the better algebraic properties of the extended interval arithmetic can be exploited effectively. In the following we present a *Mathematica* package `directed.m` that extends *Mathematica* interval capabilities by providing a new data object representing extended multi-intervals, as well as operations and functions for basic arithmetic on them. We give here an overview of the package design, its usage and demonstrate the advantages of the implemented extension on some examples.

2 Numerical Computations with Extended Intervals

The *Mathematica* package `directed.m` [1] was designed as an experimental demonstrative package intended to provide functionality that can not be obtained by conventional interval arithmetic supported by the *Mathematica* [15] kernel function `Interval` [7]. At a first stage, *Mathematica* interval capabilities were extended by the definitions of a new data object `Directed` and a number of functions handling numerical extended intervals.

Directed[{a, b}]	is the directed interval from a to b.
Directed[{a_1, b_1, Round}, {a_2, b_2, Round}, . . .]	is the set of outwardly rounded extended intervals a_1 to b_1, a_2 to b_2, ... called extended multi-interval.

An optional parameter `Round` is included in the syntax of the `Directed` data object to ensure correct outward rounding of the inexact numbers participating in the extended intervals according to the strict definitions of interval computer operations [3, 5]. Symbols `Infinity` and `Indeterminate`, mathematical constants (`E, Pi`, etc.), exact singletons (e. g. `Sqrt[11], Sin[2], Log[12], ...`) can be involved at the interval end-points and combined with inexact numbers providing that interval operations handle former exactly while the latter are rounded in the corresponding direction.

Intersecting, equally directed elements of an extended multi-interval are merged into single intervals which are put into normal order. To provide correct handling of intervals involving exact singletons and/or mathematical constants numerical approximation takes a substantial part of the corresponding operations/functions, in particular to decide whether interval elements intersect.

```
In[1] := a = Directed[{3, 2}, {0, Pi}, {0., Sqrt[26], Round}]
Out[1] = Directed[{-2.22507 10^(-308), Sqrt[26]}, {3, 2}]
```

Basic arithmetic on numerical extended multi-intervals is automatic, performed in machine or user-specified precision with outwardly rounding. Arithmetic operations are implemented according to the strict definitions of outwardly rounded computer operations [3, 5] providing that the resulting extended interval always encloses (according to the extended order relation) the true result. The outward rounding is performed *a posteriori* rather than as directed rounding in hardware.

```
In[2] := Directed[3., {E, 2}] - 3
Out[2] = Directed[{-3+E, -1},{-4.44089 10^-16, 4.44089 10^-16}]
```

Normal intervals are special case of the single extended intervals and extended multi-intervals generalize Kahan's intervals. Division by a directed interval with 0 in its interior results in two semi-infinite directed intervals.

```
In[3] := 1 / Directed[{3, -2}]
Out[3] = Directed[{-1/2, -Infinity}, {Infinity, 1/3}]
```

Properties (2), implemented as corresponding rewrite rules for the interval operations provide no blowing-up of the interval result if the arguments involve approximate real numbers.

D is a conditionally complete lattice regarding $\subseteq$ with the following lattice operations:

$$\begin{aligned} \inf_{\subseteq}(A,B) &= [\max\{a^-,b^-\},\min\{a^+,b^+\}], \\ \sup_{\subseteq}(A,B) &= [\min\{a^-,b^-\},\max\{a^+,b^+\}]. \end{aligned}$$

The lattice operations convex hull and intersection provide set theoretical functionality for normal intervals but not for improper intervals. Functions `IntHull` and `IntIntersection` perform lattice operations on extended intervals.

IntHull[int_1, ... , int_p] gives convex hull of a number of extended intervals IntIntersection[int_1, ... , int_p] gives intersection of a number of extended intervals

```
In[5] := IntHull[Directed[{2, 3}], Directed[{5, 7}]]
Out[5] = Directed[{2, 7}]
In[6] := IntIntersection[a, Directed[{-11, -Infinity}, {7, 2}]]
Out[6] = Directed[{7, -Infinity}]
```

Mathematica package `directed.m` contains two functions `InclusionQ` and `InclusionEQ` testing the antireflexive, resp. reflexive inclusion relations between extended intervals.

InclusionQ[int_1, int_2] delivers True if every element of the extended multi-interval int_2 is contained (1) in but is not equal to some element of the multi-interval int_1 and False otherwise. InclusionEQ[int_1, int_2] delivers True if every element of the multi-interval int_2 is contained in or is equal to some element of the multi-interval int_1 and False otherwise.

These functions were extended to test the corresponding relation between a sequence of directed intervals and/or numbers.

```
In[7] := InclusionQ[Directed[{-7, -12}, 5, {3, 13}],
       Directed[{3, 5}, {6, 7}]], Directed[{Pi, 4}], Pi]
Out[7] = True
```

Comparison operations <=, <, >=, > are redefined to test an order relation, defined by

$$A \preceq B \iff (a^- \leq b^-) \; and \; (a^+ \leq b^+). \tag{4}$$

int_1 <= int_2 delivers True if every element in the extended multi-interval int_1 is in relation (4) with some element in the multi-interval int_2 and False otherwise.

A sequence of extended multi-intervals can be compared

```
In[8] := Directed[{0.7, 5, Round}, {12, 3}] >
       Directed[{2/3, Cos[3]}, 0.] >= Directed[{-E, -1}]
Out[8] = True
```

Function `IntMemberQ` provides set theoretical functionality for extended intervals.

IntMemberQ[int_1, arg_2] delivers True if the set of values defined by every element of the multi-interval (or numerical value) arg_2 is contained in the set of values defined by some element of the directed multi-interval int_1 and False otherwise.

Extended interval `b` contains 4 and 5 as set theoretical members but not according to the extended inclusion relation.

```
In[9] := IntMemberQ[b=Directed[{7, 4}, {12, Infinity}], 4]
Out[9] = True
In[10] := IntMemberQ[b, 5]
Out[10] = True
```

Several utility functions provide convenient manipulations with extended intervals. The utility functions `Direction, Sign, Chi, First, Second` and `Proper` are specific for manipulation with extended intervals. Function `Proper` delivers the proper projection of an extended multi-interval. Using `Interval` instead will cause double rounding of the end-points.

```
In[11] := {Proper[a], Interval @@ a}
Out[11] = {Directed[{-(2.22507 10^-308), Sqrt[26]}],
      Interval[{-(4.45015 10^-308), Sqrt[26]}]}
```

`Dual` is an important functional which reverses the end-points of an extended interval.

```
In[12] := Dual[b]
Out[12] = Directed[{4, 7}, {Infinity, 12}]
```

Kernel `Min, Max` functions are overloaded to deliver the greatest lower or the least upper bound of an extended multi-interval. The function `Abs` maps `Directed` to a numerical value.

```
In[13] := Abs[Directed[{-11, -7}, {2, 3}]]
Out[13] = 11
```

Sometimes, an inner inclusion of the true interval solution can be very useful giving an estimation of the tightness of the obtained outer interval solution. An inner inclusion interval is an interval which is contained in the true solution interval. Some safety problems also search for a minimum set of the solutions instead of an inclusion. Inner inclusions in conventional interval arithmetic can be obtained only if inwardly rounded interval operations are implemented in addition to the outwardly rounded ones which requires an extension of the set of operation symbols. An important property of the extended interval arithmetic is that inner inclusions can be obtained only by outwardly rounded operations and the corresponding dual of the input interval expression [4]. Roundings $\bigcirc, \Diamond : D \longrightarrow RD$ (where RD is the set of computer representable extended intervals) are defined by $\bigcirc A = [\triangle a^-, \bigtriangledown a^+]$ (inward rounding), and $\Diamond A = [\bigtriangledown a^-, \triangle a^+]$ (outward rounding); $\bigtriangledown$, $\triangle$ are the floating-point directed roundings toward $-\infty$ and $+\infty$, respectively. For $A \in D$ we have [4]

$$\texttt{Dual}[\Diamond\texttt{Dual}[A]] \;=\; \bigcirc A \;\subseteq\; A \;\subseteq\; \Diamond A \;=\; \texttt{Dual}[\bigcirc\texttt{Dual}[A]].$$

If $\circ \in \{+, -, \times, /\}$ is an operation in D, the properties

$$(\bigcirc A) \circledcirc (\bigcirc B) \;\subseteq\; A \circ B \;\subseteq\; (\Diamond A) \mathbin{\Diamond\!\!\!\circ} (\Diamond B)$$
$$\texttt{Dual}[(\Diamond\texttt{Dual}[A]) \mathbin{\Diamond\!\!\!\circ} (\Diamond\texttt{Dual}[B])] \;=\; (\bigcirc A) \circledcirc (\bigcirc B)$$

are extended for rational expressions to facilitate obtaining an inner inclusion. In order to give the user the opportunity for both outward and inward rounding of an extended interval involving inexact numbers, by analogy with the kernel function `N` converting all numbers to `Real` form and `N[expr, n]` performing computations with n-digit precision numbers, we have defined function `R` to produce the approximate real interval including the extended interval argument. Applying function `R` is often more convenient than using the parameter `Round` for rounding inexact intervals.

Example 1 *Find an interval F, such that* $F \subseteq \{(2.3 + b)/c \mid b \in B, c \in C\}$, *wherein $B, C \in D$ are proper intervals.*

In outwardly rounded conventional interval arithmetic we can obtain only

$$(2.3 + B)/C \;\supseteq\; \{(2.3 + b)/c \mid b \in B, c \in C\},$$

while in extended interval arithmetic we can get `F` *satisfying the required property.*

```
F = (2.3 + Dual[R[Dual[B]]]) / Dual[R[Dual[C]]]
```

Symbolic manipulation proved to be an efficient tool for detection and removal of dependency relations between variables and the reduction of the number of occurrences of variables in range computation of interval functions. However, the limited possibilities for reduction and the varying character of expressions mean that we never can be sure to have been producing the best computable form for an expression (if existing) but only a more suitable one. Under certain monotonicity conditions for the function, a theorem [4] for eliminating the dependency problem gives the best computable form for an expression in generalised interval arithmetic.

Example 2 *Compute the exact range of function*

$$f(t) \quad = \quad \frac{t+[1/5,2]}{t-[1/4,7/3]} \qquad \textit{over} \;\; T=[3,36/5].$$

By conventional interval arithmetic we obtain interval $[64/139, 69/5]$ *for the range of* f*. Because* $f(t)$ *is monotonously decreasing on* t *over* T*, monotonously increasing on* t *in the numerator and monotonously decreasing on* t *in the denominator, we can apply the theorem from [4] to eliminate the effect of multi-incidence of variable* t*. According to this theorem*

$$f(T) \quad = \quad \frac{\texttt{Dual}[T]+[1/5,2]}{T-[1/4,7/3]},$$

obtaining thus $[148/139, 15/2]$*, which is the exact range of* f*.*

Function $f(t)$ *can be written in the equivalent form*

$$f(t) \quad = \quad 1+\frac{[1/5,2]+[1/4,7/3]}{t-[1/4,7/3]}$$

but last expression involves one arithmetic operation more than the expression in generalised interval arithmetic and thus one more round-off error when doing approximate computations.

Computer algebra systems provide the best facilities for checking monotonicity conditions. Embedding generalised interval arithmetic in *Mathematica* allows a combined usage of both techniques: symbolic preprocessing and the theorem for eliminating the dependency problem in producing the best computable form for the range of an expression.

3 Symbolic-Algebraic Interval Computations

One major class of calculations made possible by the symbolic computation capabilities of the computer algebra systems is that involving manipulation of algebraic formulas. The algebraic manipulations involve simplifying rational expressions and finding algebraic solutions for several kinds of equations. Due to the many conditionals involved in the interval formulas, implementing symbolic intervals and complete symbolic interval arithmetic is not suitable. We have chosen another strategy: to define in *Mathematica* symbols representing extended intervals and to model the specific algebra of extended intervals by corresponding rewrite rules.

Any symbol (name of variable) can represent an extended interval if its type is explicitly specified as `Directed`. The package `directed.m` treats a symbol `symb` as extended interval if that symbol had been given an explicit assignment

```
symb /: Head[symb] = Directed
```

where the kernel `Head` function identifies the type of the objects. This way, we can use symbols representing extended intervals instead of symbolic data objects `Directed` (e. g. `Directed[{a, b}]`). Symbols without explicit type assignment are considered as degenerate (point) intervals for which all built-in algebraic rules are valid.

There are often many different ways to write the same algebraic expression. From the derived distributive relations [12] between extended intervals we have inferred rules for taking a common symbolic multiplier out of brackets and rules for disclosing brackets on multiplication of a sum by extended interval, where either the sum or the multiplier involves symbolic extended interval. We consider symbolic-numerical expressions being finite interval sums involving two-terms products of a common symbolic multiplier and a coefficient which is either a numerical expression or a numerical extended interval. The rules for taking a common multiplier out of brackets were implemented in the *Mathematica* package by corresponding simplification equations [14]. These simplification equations being associated with the kernel operation addition are applied automatically so that any input expression involving extended intervals is automatically simplified if possible.

```
In[14] := x /: Head[x] = Directed;
In[15] := Directed[{2, 7}] x - x^2 + Directed[{3, 5}] x
Out[15] = -x^2 + x Directed[{5, 12}]
```

Since the validity conditions for the interval distributive relations involve values of `Direction`, `Chi` and/or `Sign` functions, associated with the common symbolic multiplier, often an explicit a priori assignment to these values is required for simplification of an interval expression. Two functions `On/Off[IntervalSimplification]` are defined to switch on/off printing of messages under what conditions an interval subexpression can be simplified. Generating messages when *Mathematica* tries to simplify an expression is switched off by default.

```
In[16] := Directed[{2, 7}] x - Directed[{5, 3}] Dual[x]
Out[16] = Directed[{2, 7}] x - Directed[{5, 3}] Dual[x]

In[17] := On[IntervalSimplification]

In[18] := Directed[{2, 7}] x - Directed[{5, 3}] Dual[x]
  IntervalSimplification::chi:
   "Directed[{2, 7}] x + Directed[{-3, -5}] Dual[x]"
  will be simplified if Sign[x]=0, Direction[x]=1, Chi[x]<=-(1/2).
Out[18] = x Directed[{2, 7}]+Directed[{-3, -5}] Dual[x]

In[19] := x /: Direction[x] = 1;  x /: Sign[x] = 0; x /: Chi[x] = -2/3;
In[20] := In[16]
Out[20] = Directed[{-1, 2}] Dual[x]
```

Distributivity (3) of the `Dual` operator on the interval arithmetic operations is another key point of the knowledge database for symbolic manipulation of interval expressions. Function `ExpandDual[expr]` is defined to do all possible expansions of the `Dual` function around sums, products and powers. Actually this function transforms the `Dual` of a sum into a sum of dual terms, the `Dual` of a product into a product of dual terms and the `Dual` of a power into the power of a dual argument everywhere in an expression. Further research is necessary for the definition of functions `Sign`, `Direction` and `Chi`, so that they automatically determine the corresponding value for an arbitrary symbolic-numerical interval expression. A solution of this problem will allow the definition of a function `IntervalExpand` designed to disclose the brackets around symbolic-numerical interval expressions and related functions allowing to transform interval expressions into other interval expressions.

Note, however, that the most important application of the automatic simplification of interval expressions is not for tight range computation but for the solution of interval equations. While in range computation we can do *a proiri* simplification of the analytic expression according to the common algebraic rules, transforming interval equations can be done only according the specific properties of interval arithmetic, otherwise the initial interval problem will be changed. Due to the algebraic identities (2) many interval algebraic equations can be solved only by elementary algebraic transformations. Modelling equations in most of the real-life practical problems involve multiple occurences of the interval parameters. Applying the theorem for eliminating the dependency problem often leads to interval equations similar to that one in the following example.

Example 3 *Find a positive proper interval (if it exists) which is the algebraic solution to the equation*

$$\frac{1+[15/4,19/6]\,\texttt{t}}{7/5+[1/2,3/5]\,\texttt{Dual[t]}+[1/3,1/2]\,\texttt{Dual[t]}} \quad = \quad [5/3,2].$$

We specify in *Mathematica* that the symbol `t` represents a directed interval and input a symbolic-numerical expression specifying the equation. *Mathematica* automatically simplifies the denominator in the left-hand side of the equation to the expression $7/5+[5/6,11/10]\,\texttt{Dual[t]}$. Applying successive elementary transformations, based on the algebraic identities (2), to the structure of the equation, latter can be reduced to the equivalent one

$$1+[15/4,19/6]\,\texttt{t}+[-5/3,-2](7/5+[5/6,11/10]\,\texttt{Dual[t]}) \quad = \quad 0.$$

Due to validity of corresponding distributive relation, the parentheses in the above equation can be removed multiplying each of the additive terms by the interval $[-5/3,-2]$. By that we obtain the following equivalent equation

$$1+[15/4,19/6]\,\texttt{t}+[-7/3,-14/5]+[-11/6,-5/3]\,\texttt{Dual[t]} \quad = \quad 0,$$

which is automatically simplified to the equation

$$[-4/3,-9/5]+[23/12,3/2]\,t \quad = \quad 0.$$

Now, the sought solution $[16/23,6/5]$ is obtained as dual of the quotient of the negative intercept and the coefficient of `t` in the last equation.

This example shows that the distributive law for extended intervals is an indispensable tool for the reduction interval algebraic equations, with multi-incidence of the unknown variable, to simpler ones. The general normal form of simplified interval algebraic equations is given in [13]. This is helpful for the explicit algebraic solution of some interval equations which are not linear in generall. For example, the interval equation

$$\frac{[7,-11]+[1,5]\times X}{X} \quad = \quad [3,2], \qquad 0\notin X$$

is algebraically equivalent to the equation

$$[1,5]\times X+[-3,-2]\times X \quad = \quad [-7,11], \qquad 0\notin X.$$

However in D, like in conventional interval arithmetic, there are only conditionally valid distributive relations and therefore these equations are not linear. Left-hand side of last equation cannot be further simplified and according to [13] the equation posesses two nonzero algebraic solutions: $X_1=[2,3]$ and $X_2=[-15,-34]$. Automatic simplification of symbolic-numerical interval expressions is also helpful for the reduction of the round-off errors (when rational arithmetic is not used) due to the reduced number of arithmetic operations in the simplified equation.

Many interval algebraic equations having rational function in the left side can be reduced to some basic type pseudo-linear algebraic equation [13] in extended interval arithmetic (this is due to the distributivity relations and properties (2)). A *Mathematica* function `AlgebraicEquationSolve[eqn]` is defined to give all numeric algebraic solutions and/or all parametric algebraic solutions to a pseudo-linear interval equation having normal form and in one variable.

```
In[21] := AlgebraicEquationSolve[Directed[{5, 6}] x + Directed[{-10, -3}] x ==
   Directed[{1, 2}]]
Out[21] = { }
```

```
In[22] := AlgebraicEquationSolve[Directed[{1, 5}] x + Directed[{-3, -2}] x ==
   Directed[{-7, 11}]]
Out[22] = {Directed[{2, 3}], Directed[{-15, -34}], Directed[{-7/2, 11/2}],
    Directed[{7, -11}]}

In[23] := x /: Sign[x] = -1
In[24] := AlgebraicEquationSolve[Directed[{1, 3}] x + Directed[{-2, -3}] x ==
   Directed[{3, 1}]]
Out[24] = {Directed[{-2, -3}]}
```

All nonzero algebraic solutions to the equation $[-4,3] \times X + [6,-2] \times X = [2,1]$ are $\{X_1 = [-\frac{1}{2} + \frac{3}{2}\alpha,\ \alpha],\ \alpha \geq \frac{1}{3}\}$, $\{X_2 = [\beta,\ \frac{1}{3} + \frac{2}{3}\beta],\ \beta \geq 0\}$, $\{X_3 = [\frac{1}{3} + \frac{2}{3}\gamma,\ \gamma],\ \gamma \leq -\frac{1}{2}\}$, $\{X_4 = [\delta,\ -\frac{1}{2} + \frac{3}{2}\delta],\ \delta \leq 0\}$

Several other *Mathematica* functions are defined to help finding algebraic solutions to systems of linear and pseudo-linear interval equations.

4 Conclusion

The algebraic properties of extended interval arithmetic make it a powerful tool for explicit solution of some interval algebraic problems and the best environment for exploiting these properties is a computer algebra system. Calculating with interval variables is a novel approach in combining symbolic and interval computations showing the possibilities for developing interval computer algebra. The implemented facilities for doing symbolic-numerical interval computations allow an easy computation and exploration in the algebra of extended intervals facilitating the solution of many practical problems. Both computer algebra and interval computations benefit from this interaction: computer algebra turns into valuable tool for scientific computing, interval computations expand the methodology tools and get an increased efficiency to serve the applications.

5 Acknowledgements

This work was supported by the TUBITAK DOPROG and the Bulgarian National Science Fund under grant No. I-507/95.

References

[1] Akyildiz, Y.; Popova, E.; Ullrich, C.: *Towards a More Complete Interval Arithmetic in Mathematica*. In V. Keränen, P. Mitic, A. Hietamäki (Eds.): *Innovation in Mathematics*, Proceedings of the Second International Mathematica Symposium. Computational Mechanics Publications, Southampton, UK, 1997, pp. 29–36. (http://www.math.bas.bg/~epopova/directed.html)

[2] Alefeld, G.; Herzberger, J.: *Einführung in die Intervallrechnung.* Bibliographisches Institut Mannheim, 1974. (English translation: *Introduction to Interval Computations.* Academic Press, 1983.)

[3] Durst, E.: *Realisierung einer erweiterten Intervallrechnung mit Überlaufarithmetik.* Diplomarbeit, Universität Karlsruhe, 1975.

[4] Gardeñes, E.; Trepat, A.: *Fundamentals of SIGLA, an Interval Computing System over the Completed Set of Intervals.* Computing, **24**, 1980, pp. 161–179.

[5] Kaucher, E.: *Über metrische und algebraische Eigenschaften einiger beim numerischen Rechnen auftretender Räume.* Dissertation, Universität Karlsruhe, 1973.

[6] Kaucher, E.: *Interval Analysis in the Extended Interval Space IR.* Computing Suppl. **2**, 1980, pp. 33–49.

[7] Keiper, J. B.: *Interval Arithmetic in* Mathematica. Interval Computations, No. 3, 1993, pp. 76–87.

[8] Markov, S. M.: *On Directed Interval Arithmetic and its Applications.* J. UCS, **1**, 7, 1995, pp. 510–521.

[9] Markov, S. M.: *Isomorphic Embeddings of Abstract Interval Systems.* Reliable Computing **3**, 1997, pp. 199–207.

[10] Ortolf, H.-J.: *Eine Verallgemeinerung der Intervallarithmetik.* Geselschaft für Mathematik und Datenverarbeitung, Bonn **11**, 1969, pp. 1–71.

[11] Popova, E. D.: *Extended Interval Arithmetic in IEEE Floating-Point Environment.* Interval Computations **4**, 1994, pp. 100–129.

[12] Popova, E. D.: *Generalized Interval Distributive Relations.* Preprint No.2, Institute of Mathematics and Informatics, BAS, 1997.

[13] Popova, E. D.: *Algebraic Solutions to a Class of Interval Equations.* J. Universal Computer Science **4**, 1, 1998, pp. 48–67. (http://www.iicm.edu/jucs_4_1)

[14] Popova, E.; Ullrich, C.: *Simplification of Symbolic-Numerical Interval Expressions.* In Gloor, O. (Ed.): Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation, ACM Press, 1998, pp. 207–214.

[15] Wolfram, S.: Mathematica *A System for Doing Mathematics by Computer.* Addison-Wesley, 1991 (2nd ed.).