Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

Towards a More Complete Interval Arithmetic in Mathematica

Y. Akyildiz[1], E. Popova[2] & C. Ullrich

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

Abstract

We consider an algebraic completion of the conventional interval arithmetic, called here extended (directed) interval arithmetic, as suitable for embedding in computer algebra systems. An experimental *Mathematica* package for extended interval arithmetic giving some extra functionality is presented. An advanced methodology for doing exact symbolic algebraic transformations on interval expressions and some applications for the efficient solution of interval algebraic problems are outlined along with some open problems and implementation considerations.

1 Introduction

Recently, there has been a considerable two-way traffic between numerical and symbolic computations. The usage of floating-point and interval arithmetic in intermediate computations appeared to result in a dramatic speed-up for some algebraic algorithms providing still guaranteed correct results [2]. Most scientific/engineering problems require a combination of analytic and numerical techniques and often deal with interval input data instead of approximate values.

[1]Mathematics Department, Bosphorus University, 81815 Bebek, Istanbul, Turkey. e-mail: akyildiz@boun.edu.tr

[2]Institute of Mathematics & Computer Science, Bulgarian Academy of Sciences, Acad. G. Bonchev str. block 8, BG-1113 Sofia, Bulgaria. e-mail: epopova@bgearn.acad.bg

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

Answering the requirements for controlling round-off errors and handling uncertain input data, the general-purpose computer algebra systems Reduce, Maple, *Mathematica* supply interval arithmetic [5]. Starting with an experimental demonstrative interval arithmetic package, *Mathematica* [10] has included `Interval` as a kernel function in its 2.2 version. The interval arithmetic in *Mathematica* Version 2.2 is completed with outward rounding and multi-intervals [5, 6]. The interval data object is integrated smoothly within the rest of the system and most of the built-in functions are designed to work together in a unified manner. However, applying the built-in *Mathematica* functions for solving algebraic problems involving `Interval` data yields interval results which have nothing to do with the solution of the corresponding problem. This is caused by the fact that the interval arithmetic structure [1] does not possess group properties. Due to the lack of inverse elements with respect to $+$ and $\times$ operations the solution of the algebraic interval equations $A + X = B$ and $A \times Y = B$ can not be generally expressed in terms of the interval operations even if they actually exist. There is no distributivity between addition and multiplication except for certain special cases. Lattice operations are not closed with respect to the inclusion relation and arithmetic operations for conventional intervals are of little use for the computation of inner inclusions.

Most of these drawbacks have been surmounted by the algebraic completion of the classical interval arithmetic structure extending the set of conventional (proper) intervals $\{[a^-, a^+] \mid a^- \leq a^+; a^-, a^+ \in R\}$ by improper intervals $\{[a^-, a^+] \mid a^- \geq a^+; a^-, a^+ \in R\}$ to the set of extended (proper and improper) intervals $D = \{[a^-, a^+] \mid a^-, a^+ \in R\}$, and corresponding extensions of the inclusion relation ($\subseteq$), interval arithmetic and lattice operations. Studied by several authors [3, 4, 7], the extended interval arithmetic structure possesses group properties with respect to addition and multiplication operations. Lattice operations are closed with respect to the inclusion order relation. Handling of norm and metric are very similar to norm and metric in linear spaces [4]. Dual is an important operator that reverses the end-points of the intervals. The substructures $(D, +, \subseteq)$ and $(D \backslash \mathcal{T}, \times, \subseteq)$, where $\mathcal{T} = \{A \in D \mid A = [0,0] \text{ or } a^-a^+ < 0\}$ are isotone groups; hence, there exist unique inverse elements ($-\mathtt{Dual}(A)$ and $1/\mathtt{Dual}(B)$) with respect to $+$ and $\times$ such that

$$A - \mathrm{Dual}(A) = 0 \quad \text{and} \quad B \times 1/\mathrm{Dual}(B) = 1. \tag{1}$$

Retaining all important properties of interval analysis, the extended interval arithmetic space possesses some other useful properties that make it a suitable environment for embedding of interval algebraic problems and their efficient solution (see e. g. [3, 8]). To emphasize that an extended interval can be considered as pair of a proper interval (in set-theoretical sense) and a "direction" [7], sometimes we shall call extended intervals "directed".

An attractive goal is to utilize the algebraic completeness of the ex-

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

tended interval arithmetic embedding latter in a versatile, powerful computer algebra system like *Mathematica* and investigating how the algebraic and other properties of this arithmetic can be exploited for symbolic-algebraic manipulation of interval expressions, developing of explicit interval algorithms, as well as studying other aspects of symbolic-numerical computations.

2 Directed Intervals in Mathematica

A *Mathematica* package `directed.m` for extended interval arithmetic was designed as an experimental demonstrative package intended to provide functionality that can not be obtained by conventional interval arithmetic [8]. 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.

Designing extended interval arithmetic in *Mathematica* we tried to keep and preserve all the functionality [5], provided by the kernel function `Interval`, since conventional interval arithmetic is a special case of extended interval arithmetic. `Interval` data object supports conventional multi-intervals which in addition to the versatility that provide list data structures and the computer algebra system itself gave us good reasons to implement extended multi-intervals. `Interval` data object and interval kernel functions, however, were not overloaded by the definitions for extended intervals due to several reasons: At the experimental stage of the package it was preferable to define a new data object. Names, definitions, and/or properties of some functions defined for `Interval` were not so appropriate for directed intervals and we used other names and/or implemented other definitions. In order to give the user the opportunity for both outward and inward rounding of an extended interval having inexact numbers as bounds, an optional parameter `Round` was included in the syntax of the data object `Directed`. `Round` specifies outward directed rounding of the approximate real end-points providing that the delivered interval always encloses the input interval. In particular the end-points are calculated with whatever arithmetic is available and then they are adjusted outwardly several Units in the Last Place in an appropriate manner by using the `ulp` function from *Mathematica* package `NumericalMath'IntervalAnalysis'`.

Lattice operations convex hull and intersection provide set-theoretical functionality for normal intervals but not for improper intervals. Merging only intersecting intervals, kernel function `IntervalUnion` does not fulfill the definition of the interval lattice operation convex hull. That is why instead of the kernel functions `IntervalUnion` and `IntervalIntersection` we defined functions `IntHull` and `IntIntersection` to perform lattice operations on extended intervals. Kernel function `IntervalMemberQ` tests only the reflexive inclusion relation between `Interval` data objects. *Mathema-*

tica package `directed.m` provides two functions `InclusionQ` and `InclusionEQ` testing the antireflexive and reflexive inclusion relation, respectively between directed intervals. These functions were extended to test the corresponding relation between a sequence of directed intervals and/or numbers. Several utility functions `Direction, Sign, Chi, First, Second, Proper` are specific for manipulation with directed intervals. 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 `Interval` to `Interval` and `Directed` to a numerical value.

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. Inner inclusions in conventional interval arithmetic can be obtained only if inwardly rounded interval operations are implemented in addition to the outwardly rounded ones requiring an extension of the set of operation symbols. In contrst, inner inclusions in the extended interval arithmetic can be obtained only by outwardly rounded operations and corresponding dual of the input interval expression [3, 8]. Various techniques have been developed (non of them universal) to reduce the interval widening in range computation of interval functions with multiple occurrences of variables. A simple approach, based on the extended interval arithmetic properties, has been proposed in [3] for the reduction of the dependency inflation in rational function evaluation. The following example demonstrates the computation of the exact range of $f(t) = (t+a)/(t+b)$ for $t = [4, 9]$, $a = [5/6, 7/8]$ and $b = [1, 3/2]$ according to this approach and an inner inclusion for the numerical approximation of the exact range. By analogy with the kernel function `N` we have defined function `R` to deliver outwardly rounded directed intervals. Applying function `R` is often more convenient than specifying the parameter `Round` in the input intervals and especially if both inner and outer inclusions are sought.

```
In[1] := <<directed.m
In[2] := {t, a, b} = {Directed[{4., 9.}], Directed[{5/6, 7/8}],
         Directed[{1, 3/2}]};
In[3] := fExact = (t + a)/(Dual[t] + b)
Out[3] = Directed[{29/33, 79/80}]
In[4] := {t, a, b} = Dual[R[Dual[#]]]& /@ {t, a, b} ;
         f_inner = (t + a)/(Dual[t] + b)
Out[4] = Directed[{0.8787...879, 0.9874999...99}]
```

Properties (1), implemented as corresponding rewrite rules for the interval operations do not cause any inflation of the interval result if the arguments involve approximate real numbers.

3 Symbolic-Algebraic Interval Computations

Although to our knowledge *Mathematica* is the only computer algebra system allowing combined usage of exact numbers, mathematical constants and/or exact singletons with approximate real numbers at the interval end-points, providing that former are handled exactly and later rounded correctly by the interval arithmetic operations and functions [5], symbolic interval arithmetic is not supported "because combinatorial explosion of expressions involving `Min` and `Max` functions would quickly render any symbolic result useless" [6]. On the nother side, one of the most important features of computer algebra systems is that they can perform symbolic-algebraic manipulations, as well as numerical calculations.

Based on the distributivity relations between multiplication and addition of extended intervals, a general framework for simplification of symbolic-numerical expressions involving intervals has been recently developed [9]. Symbols representing non-degenerate extended intervals, for which built-in algebraic rules are not valid, are distinguished from symbols representing point intervals or other objects, for which built-in rules are valid, by the explicit assignment

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

The specific algebra of extended intervals and the corresponding conditionally distributive relations are modeled using the mechanism of pattern-matched rewrite rules. The rewrite rules, which specify interval distributive relations, are associated with the kernel addition operation. Then any input symbolic-numerical expression representing a finite interval sum composed of two-terms products of a common symbolic (directed) multiplier and a coefficient, which is either a numerical expression or numerical directed interval, is automatically simplified (if possible) following the general simplification scheme of *Mathematica*. The input expression is taken and those rewrite rules is looked for whose pattern matches part of the expression. That part is then replaced by the replacement text of that rule. Evaluation then proceeds by searching for further matching rules until no more are found.

```
In[5] := x /: Head[x] = Directed;
In[6] := Directed[{2, 7}] x - x^2 + Directed[{3, 5}] x
Out[6] = -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 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 facilitate the use as much as possible. These functions switch on/off printing prompt messages about possible simplifications of any interval subexpression. Generating messages when *Mathematica* tries to simplify an expression is switched off by default.

```
In[7] := Directed[{2, 7}] x - Directed[{5, 3}] Dual[x]
Out[7] = Directed[{2, 7}] x - Directed[{5, 3}] Dual[x]
In[8] := On[IntervalSimplification]
```

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533


```
In[9] := 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[9] = x Directed[{2, 7}]+Directed[{-3, -5}] Dual[x]
In[10] := x /: Direction[x] = 1;  x /: Sign[x] = 0;
       x /: Chi[x] = -2/3;
In[11] := In[9]
Out[11] = Directed[{-1, 2}] Dual[x]
```

Distributivity of the `Dual` operator on 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 `Dual` function around sums, products and powers. 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.

4 Applications to Interval Algebraic Problems

The algebraic completeness of the extended interval arithmetic structure has been used by several authors to develop an algebraic approach to the solution of some "classical" interval problems. In [3] an engineering problem with uncertain data has been reformulated in terms of the extended interval space and explicitly solved there. Some other authors (references and more details can be found in [8]) try to solve some practical problems related to linear algebraic systems involving intervals using properties of the extended interval arithmetic. Four different solution sets to such a system have been defined and studied. The detailed description of each of the solution sets is rather complex and grows exponentially with the dimension of the problem. This is the reason why usually a simpler subset, expressed in terms of intervals, satisfying the definition of the corresponding solution set is looked for. A key role in finding some of these solution sets plays the interval algebraic solution. This solution does not exists in general in the ordinary interval space and the extended interval arithmetic is a natural arithmetic for dealing with interval algebraic equations.

The following example illustrates the application of the algebraic identities (1) and interval distributive relations for the explicit solution of an interval problem reformulated in terms of extended intervals. Find a positive proper interval (if it exists) which is the algebraic solution to the equation

$$\frac{[4,5]-[3,6]\,t}{[5,6]+[3,4]\,t} \;==\; [\frac{1}{8},\frac{2}{3}].$$

Applying successive elementary transformations based on the algebraic identities (1), the structure of the equation is reduced to the equivalent one

$$[4,5]+[-6,-3]\,t+[-\frac{1}{8},-\frac{2}{3}]\,([5,6]+[3,4]\,t) \;==\; 0.$$

The parentheses in the above equation can be removed multiplying each of the additive terms by the interval $[-\frac{1}{8},-\frac{2}{3}]$. By that we obtain the following equivalent equation

$$[4,5]+[-6,-3]\,t+[-\frac{3}{4},-\frac{10}{3}]+[-\frac{1}{2},-2]\,t \;==\; 0,$$

which is automatically simplified to the equation

$$[13/4,5/3]+[-13/2,-5]\,t \;==\; 0.$$

Now, the sought solution $[1/3,1/2]$ is obtained as dual of the quotient of the intercept and the negative coefficient of `t` in the last equation.

The next step towards an enhancement of the knowledge representation database corresponding to the algebra of extended intervals involves investigations of the solutions to the interval algebraic equations generated after all possible simplifications have been carried out. This will allow the definition of a function `IntervalSolve` giving all numerical and/or parametric solutions to certain kinds of interval algebraic equations, as well as the symbolic solutions.

5 Conclusion

Although interval arithmetic is increasingly used in combination with computer algebra and other methods, both approaches — symbolic-algebraic and interval-arithmetic — have been used separately. Performing true symbolic-algebraic computations involving intervals is possible in an algebraically completed set of extended intervals retaining all properties of interval analysis. Providing an excellent environment for experimentation and exploration *Mathematica* allowed us to develop an experimental package for extended interval arithmetic and to apply a novelty for the interval analysis approach for combining symbolic-algebraic and interval computations. However, some open problems still need discussion. For example, is it necessary one and the same interval data object to support extended multi-intervals, having limited usage and high implementation cost, whereas symbolic-algebraic computations are making sense only for single extended

Transactions on Engineering Sciences vol 15, © 1997 WIT Press, www.witpress.com, ISSN 1743-3533

intervals? Further research in the directions outlined throughout the paper will provide the necessary background for a successful integration of the algebra of extended intervals in *Mathematica* and an increased efficiency of interval applications.

References

[1] Alefeld, G.; Herzberger, J.: *Interval Analysis.* Academic Press, 1981.

[2] Collins, G. E.; Krandick, W.: *A Hybrid Method for High Precision Calculation of Polynomial Real Roots.* In Bronstein, M. (Ed.): Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation, ACM Press, 1993, pp. 47–52.

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

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

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

[6] Keiper, J. B.: *Interval Computation.* In *Major New Features in Mathematica Version 2.2.* Technical Report, Wolfram Research, 1993, pp. 20–23.

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

[8] Popova, E. D.; Ullrich, C. P.: *Directed Interval Arithmetic in Mathematica: Implementation and Applications.* Technical Report 96-3, University Basel, Switzerland, 1996, pp. 1-56.

[9] Popova, E. D.; Ullrich, C. P.: *Simplification of Symbolic-Numerical Interval Expressions.* Technical Report 97-1, University Basel, Switzerland, 1997. (http://www.math.acad.bg/~epopova/directed.html and http://www.ifi.unibas.ch/TR/)

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