2.3.5 Packages for Symbolic Manipulators

Both interval and symbolic computations use computers to obtain mathematically rigorous statements. Also, since interval computations benefit from *a priori* rearrangement and simplification of expressions, and since algorithms such as interval Newton methods require evaluation of functions and derivatives with different data types, it is natural to consider interval computations within a symbolic computation environment.

Interval packages are available both within the Mathematica [132] and Maple [36] systems[38]. These packages are useful for various interactive explorations, and as interval calculators. For example, the following is a portion of a Mathematica session[39].

(Input):	`f[x_] := x(x-1)`	
(Input):	`fp[t_] = D[f[t],t]`	
(Output):		`-1 + 2 t`
(Input):	`ix = Interval[0,1]`	
(Output):		`Interval[0,1]`
(Input):	`f[ix]`	
(Output):		`Interval[-1, 0]`
(Input):	`fp[ix]`	
(Output):		`Interval[-1, 1]`
(Input):	`f[1]`	
(Output):		`0`
(Input):	`fp[1]`	
(Output):		`1`
(Input):	`Sin[ix]`	
(Output):		`Interval[0, Sin[1]]`

These packages may be useful for illustration and teaching of concepts in nonlinear systems and global optimization, such as in [5].

The Maple package [36] contains capabilities consistent with Corliss' BIAS proposal[40] [37]. An advantage of the Maple package is that the source code is available.

[38] A general reference for Mathematica is [244], while a general reference for Maple is [229].
[39] with format edited for presentation here
[40] Corliss' BIAS proposal has the same name as the Jansson / Knüppel package of [135], but is something different.

In their present state symbolic manipulation packages should be used with some care, since internal simplifications in the algebraic expressions may result in interval values that do not correspond to exhibited algebraic expressions. that is, the packages may rearrange expressions without telling the user, and such rearrangements give different interval extensions. Furthermore, *programs* that include loops, etc. written in these languages execute thousands of times more slowly than corresponding programs in languages such as Fortran 90.

2.3.6 Capabilities in Spreadsheets

Both spreadsheets and logic programming languages have been used in a technique termed *constraint satisfaction* in [89], *constraint logic programming* in [232], *subdefinite calculations* in [12], and *substitution-iteration* in [112], [222], and in Chapter 7 below. In a nutshell, such techniques consider equations such as $x_2 = x_1^3$ not as defining a *direction* of computation, but a *relation*; thus, if x_2 is known to lie in a certain interval, then it can be used to narrow the possible range of values for x_1, just as interval bounds for x_1 can narrow the range of values for x_2. This can be used for solving nonlinear systems of equations, proving infeasibility of inequality constraints, etc.

A spreadsheet enhancement, built upon Microsoft Excel, that supports such interval constraint propagation has been constructed by Hyvönen *et al.* [89] as an add-on function library, based on their C++ package InC++ [88].

A group at the Russian Institute of Artificial Intelligence has developed the notepad-like program UniCalc [12], in which objective functions, constraints of various types, etc. are entered into a screen area, and calculation proceeds in an order similar to that of an interval constraint propagation spreadsheet. A difference is that, in UniCalc, the expressions are completely parsed into component operations, as described with the expanded system on page 39 and in Chapter 7. The package, with a nice user interface, runs on DOS systems[41], and is available commercially. An advantage is that the user can stop the calculations in mid-stream and change certain values, for interactive explorations. The computations in UniCalc are rapid compared to other implementations of such techniques, but a disadvantage of an early version of UniCalc was lack of rigor in the arithmetic[42].

[41] and possibly on other systems now

[42] That is, directed roundings were not used, so that it was possible for a mathematically incorrect (though approximately correct) result to be given.