Improving interval enclosures

Arnold Neumaier

Fakultät für Mathematik, Universität Wien
Nordbergstr. 15, A-1090 Wien, Austria
email: Arnold.Neumaier@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~neum`

July 18, 2015

Abstract. This paper serves as background information for the Vienna proposal for interval standardization, explaining what is needed in practice to make competent use of the interval arithmetic provided by an implementation of the standard-to-be.

Discussed are methods to improve the quality of interval enclosures of the range of a function over a box, considerations of possible hardware support facilitating the implementation of such methods, and the results of a simple interval challenge that I had posed to the reliable computing mailing list on November 26, 2008.

Also given is an example of a bound constrained global optimization problem in 4 variables that has a 2-dimensional continuum of global minimizers. This makes standard branch and bound codes extremely slow, and therefore may serve as a useful degenerate test problem.

(The 2015 version differs from earlier versions by more details in the section on centered forms.)

Keywords: bicentered form, directed rounding, fractional multilinear expressions, global optimization, implied equations, interval arithmetic, monotonicity, optimal enclosure, overestimation, quaternion, range enclosure, ratsin, ratcos, rearrangement, uncertain rotation

2000 MSC Classification: primary 65G40

Contents

1 **Introduction** 2

2 **Monotonicity** 3

3 **Other improvement techniques** 8

4 **On hardware for intervals and directed rounding** 13

5 **A simple interval challenge** 14

6 **Public solutions to the challenge** 18

1 Introduction

Interval analysis is the theory and practice of rigorously working on a computer with certain (exactly known) and uncertain (i.e., possibly not exactly known) real numbers, represented as intervals; see, e.g., the textbooks [25, 26, 29, 28]. It involves the appropriate use of standard floating-point calculations (in round to nearest mode), directed floating-point calculations (in rounding modes up or down), and interval arithmetic, combined in a way that gives reasonable enclosures of the results with an acceptable cost.

Important current applications of interval analysis include but are not restricted to the following areas (only sample references are given):

- global optimization (used in lots of different applications) [31, 40],
- solving equations with multiple solutions (e.g., in chemical engineering [9], computational geometry [7, 35], robotics [24]),
- quantified constraint satisfaction [49] (e.g, finding safe work spaces for robots [23, 37]),
- design under uncertainty [34, 38],
- computer-assisted proofs [33],
- verified linear and semidefinite programming [14, 16].

Many applications (e.g., all global problems) need good bounds for very wide intervals, some (e.g., rounding error control) need high accuracy for narrow intervals.

Interval arithmetic is the collection of operations with intervals underlying interval analysis, and does not comprise directed floating-point calculations per se; however (see Section 4), the two are closely related and often are used together.

The successful application of interval arithmetic appears to be somewhat difficult for non-expert users. Indeed, the proper use of interval arithmetic requires attention to details beyond those of usual numerical computations since otherwise the results may be useless, either being invalid (i.e., not enclosing the true results) or overly pessimistic. Thus, interval arithmetic cannot be regarded in isolation from interval analysis as the body of theory and techniques for making good use of interval arithmetic.

Tools such as monotonicity arguments, centered forms, subexpression analysis, or transformations to equivalent form, more sophisticated techniques such as Makino's reduction techniques, and even branch and bound techniques may play a role in improving range enclosures. Even if a given functional expression or algorithm cannot be handled directly by such techniques, these techniques often apply to suitable subexpressions.

The purpose of this paper is to provide background information for the Vienna proposal for interval standardization [36], explaining what is needed in practice to make competent use of the interval arithmetic provided by an implementation of the standard-to-be.

Discussed are methods to improve the quality of interval enclosures of the range of a function over a box, considerations of possible hardware support facilitating the implementation of such methods, and the results of a simple interval challenge that I had posed to the reliable computing mailing list on November 26, 2008.

Also given is an example of a bound constrained global optimization problem in 4 variables that has a 2-dimensional continuum of global minimizers. This makes standard branch and bound codes extremely slow, and therefore may serve as a useful degenerate test problem.

Our notation follows the recommendations in Kearfott et al. [15] for standardized notation in interval analysis. In particular, boldface letters such as $\mathbf{x}$ denote intervals or interval vectors; $\underline{x}$ and $\overline{x}$ denote their lower and upper bound, respectively.

Acknowledgments. Contributions and comments to the challenge by Nate Hayes, Lubomir Kolev, Mihaly Markot, Jean-Pierre Merlet, Ramon Moore, John Pryce, and Dan Zuras are gratefully acknowledged.

2 Monotonicity

Monotonicity is one of the most powerful properties that enable skilled users to improve the enclosures obtainable by naive interval evaluation.

Exploiting partial monotonicity. Monotonicity arguments are traditionally used in interval Newton techniques to speed up the branch-and-bound procedure in which the interval Newton method is embedded. They are also traditional in the analysis of uncertain linear problems; see, e.g., Barth and Nuding [2] and Neumaier & Pownuk [38, Theorem 5.1]. However, a monotonicity analysis can be done for many algebraic expressions.

2.1 Theorem. (Ranges for partial monotone functions)
Let $e(x, y, z)$ be an expression in the components of x, y, z. Suppose that $e(x, y, z)$ can be evaluated for all $x, y, z \in \mathbf{x}$ and that the resulting function is monotone increasing in y and monotone decreasing in z. Then the range of the function f defined by $f(x) := e(x, x, x)$ for $x \in \mathbf{x}$ is contained in the interval

$$\mathbf{f} := [\inf_{x\in\mathbf{x}} e(x, \underline{x}, \overline{x}), \sup_{x\in\mathbf{x}} e(x, \overline{x}, \underline{x})].$$

If $e(x, y, z)$ is independent of x then $\mathbf{f}$ is the exact range.

Proof. This follows from the fact that a function monotone increasing (decreasing) in some variable t attains its minimum over an interval at the lower (upper) bound and its maximum at the upper (lower) bound of the interval. □

For the example $e(x, y, z) = x * y - z$ and $\mathbf{x} = [0, 1]$, we get the enclosure $[-1, 1]$ of the range $[-0.25, 0]$, showing that, in general, the theorem need not give the exact range.

To apply the theorem to a given expression $f(x)$, one splits the occurrences of each component x_i into three groups, two (renamed y_i and z_i) where a monotonicity direction can be ascertained, and one (left as x_i) in which one cannot do this. This gives an expression $e(x, y, z)$ of the required form such that $f(x) := e(x, x, x)$. Note that $e(x, y, z)$ need not depend explicitly on all components of x, y, z. Note also that the bounds of $\mathbf{f}$ are bounds of ranges of simpler functions, which can often again be rearranged and/or subjected to further monotonicity arguments.

Important special cases of the monotonicity theorem include the optimal enclosure of the linear interpolation expression

$$\text{linearInt}(x, y, t) := (1 - t)x + ty, \quad t \in [0, 1], \tag{1}$$

discussed in Neumaier [35] and the optimal enclosures

$$\mathbf{f} := \Big[\frac{\underline{x} - \overline{y}}{\underline{x} + \overline{y}}, \frac{\overline{x} - \underline{y}}{\overline{x} + \underline{y}}\Big], \tag{2}$$

$$\mathbf{g} := \Big[\frac{\underline{y}}{\overline{x} + \underline{y}}, \frac{\overline{y}}{\underline{x} + \overline{y}}\Big] \tag{3}$$

of the functions $f(x, y) := (x - y)/(x + y)$ and $g(x, y) := y/(x + y)$ over $x \in \mathbf{x}$ and $y \in \mathbf{y}$ with $\underline{x}, \underline{y} \geq 0$, not both zero. Note that for nonzero y, the rearrangement as

$f(x,y) = 1 - 2/(1 + x/y)$ and $g(x,y) := 1/(1 + x/y)$ also gives optimal results, though at a slightly higher cost.

In practice, the function whose range is wanted must first be brought into the above form by analyzing which part of it is monotone increasing and which part of it is monotone decreasing. This may also involve the introduction of intermediate variables. For example, the function

$$f(x,y,z) := x/\sqrt{x^2 + y^2 + z^2} \tag{4}$$

can be optimally enclosed at $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ by monotonicity arguments using $\mathbf{w} = \mathbf{y}^2 + \mathbf{z}^2$ and the enclosure

$$\mathbf{f} = \Big[\underline{x}/\sqrt{\underline{x}^2 + \overline{w}}\,,\ \overline{x}/\sqrt{\overline{x}^2 + \underline{w}}\Big]. \tag{5}$$

Similarly, the function $f(x,y) := x^2/(x^2 + y^2)$ considered by Pryce & Corliss [47] can be optimally handled by computing $\mathbf{u} := \mathbf{x}^2$, $\mathbf{v} := \mathbf{y}^2$, and

$$\mathbf{f} = [\underline{u}/(\underline{u} + \overline{v}), \overline{u}/(\overline{u} + \underline{v})]. \tag{6}$$

A more complicated instance is the following. To enclose the function

$$f(a,q,e) := (a(1 - q) + e\sqrt{kq})/(1 + q) \tag{7}$$

for $a \in \mathbf{a}$, $q \in \mathbf{q}$, and $e \in \mathbf{e}$, with a fixed constant $k > 0$, and intervals $\mathbf{a} > 0$, $1 \in \mathbf{q} > 0$, and $\mathbf{e} = [-1, 1]$, one uses that a and e appear linearly, hence can be restricted to their endpoints. Simple analysis shows that under the given conditions, a simple sufficient condition for monotone decrease in q is $\overline{q} \le 1 + 16\underline{a}^2/k$, and that the two extremes are attained at $a = \overline{a}$, but opposite ends for e and q. If the sufficient condition holds, we get the optimal enclosure

$$\mathbf{f} = \Big[\,(\overline{a}(1 - \overline{q}) - \sqrt{k\overline{q}})/(1 + \overline{q})\,,\,(\overline{a}(1 - \underline{q}) + \sqrt{k\underline{q}})/(1 + \underline{q})\,\Big].$$

Infinite endpoints. In the theory of interval analysis, intervals are usually considered to be bounded. The consideration of unbounded intervals, which have one or two infinite endpoints, is necessary in several applications including constrained global optimization. Since operations with infinity do not necessarily make sense, monotonicity considerations need extra attention when applied to unbounded intervals. In particular, expressions such as $(x - 1)/(x + 1)$ for $x \in [0, \infty]$ or $x^2 - x$ for $x \in [1, \infty]$ show that optimal bounds at infinity often require an explicit consideration of the limiting behavior rather than a blind use of the endpoints.

Fractional multilinear expressions. Frequently, an expression is affine or rational linear in some of its components. The following recipe generalizes the usual formula for calculating the interval hull of a product or quotient to more general rational functions such as $f(x) = (x_1x_2 - 1)/(x_1 + x_2 + 1)$ that are fractional linear in each variable.

2.2 Theorem. (Fractional multilinear ranges)
Let $e(x)$ be an expression in the components of x that, for some subset K of indices, is fractional linear (i.e., equivalent to the form $(at+b)/(ct+d)$ with a,b,c,d independent of t) in the components $t = x_k$ $(k \in K)$ of x. If $e(x)$ is defined for all $x \in \mathbf{x}$ then the range of $e(x)$ for $x \in \mathbf{x}$ is contained in the convex hull of the ranges on the subboxes of $\mathbf{x}$ obtained by specializing in all possible ways the components x_k $(k \in K)$ to one of their endpoints. If K contains all indices then $\mathbf{f}$ is the exact range.

Proof. This follows from the fact that a fractional linear function is monotone in each interval where it is defined, since functions monotone in a variable t attain their minimum and maximum at the endpoints of the interval under consideration. □

In practice, the ranges on the subboxes will be computed by other interval techniques. Frequently, these calculations share the results of some subexpressions, whose ranges need then to be computed only once. A similar remark also applies to the other techniques for improving range enclosures.

Convex combinations. An example of a fractional multilinear expression of more than occasional relevance is the general convex combination

$$f_n(t,x) := \sum_{k=1}^{n} t_k x_k \Big/ \sum_{k=1}^{n} t_k,$$

where all t_k are nonnegative, and not all of them vanish. For $n = 2$, this can be reduced to linearInt since

$$f_2(t,x) = x_1 + \frac{t_2}{t_1+t_2}(x_2 - x_1),$$

and we may get an optimal enclosure since $\frac{t_2}{t_1+t_2}$ can be optimally enclosed using (3). This is used in Section 6 in Pryce's treatment of the challenge problem.

For general $n > 2$, Theorem 2.2 would require an exponential number of endpoint combinations to be evaluated. However, closer examination reveals that optimal bounds for the range over $t \in \mathbf{t}$ and $x \in \mathbf{x}$ can be obtained with an algorithm of nearly linear complexity in n. First of all, since all t_k are nonnegative, the infimum of the range is attained at $x = \underline{x}$, and the supremum at $x = \overline{x}$; thus it is enough to consider the case where $\mathbf{x}$ is a real number.

We therefore consider the problem of finding the minimum of $f(t,x)$ over $t \in \mathbf{t}$ for fixed x. Using a permutation computable with $O(n \log n)$ operations, we may also assume that the x_k are in increasing order, $x_1 \le \ldots \le x_n$. Now $\partial f/\partial t_j = \sum t_k(x_j - x_k)/(\sum t_k)^2$ is increasing in j, nonpositive for $j = 1$, and nonnegative for $j = n$. This implies that there exists a positive integer $l < n$ such that the minimum is attained for the vector t^l with $t^l_j = \overline{t}_j$ for $j \le l$ and $t^l_j = \underline{t}_j$ for $j > l$. By making optimal use of common subexpressions, it is possible to compute all $f(t^l,x)$ with $O(n)$ operations; thus one gets

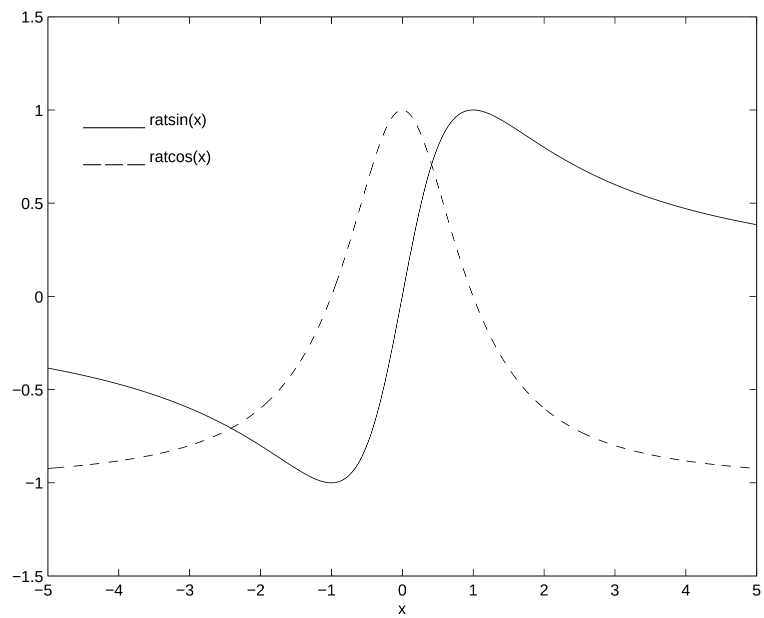
$$\min_{t \in \mathbf{t}} f(t,x) = \min_{l=1:n-1} f(t^l,x)$$

with a total complexity of $O(n \log n)$. The maximum can be found in an analogous way.

Optimal enclosures for unary operations. Univariate functions are piecewise monotone, hence their ranges can be found analytically. For univariate functions arising frequently, one can find the extrema and their values in advance, and then gets optimal enclosures by taking the hull of the endpoints and the interior extremal values; one also needs to take care of correct rounding by looking at the direction of monotonicity. We treat as examples the functions

$$\operatorname{ratsin}(x) := 2x/(1+x^2), \quad \operatorname{ratcos}(x) := (x^2-1)/(x^2+1). \tag{8}$$

They satisfy $\operatorname{ratsin}(x)^2 + \operatorname{ratcos}(x)^2 = 1$ and provide a rational parameterization of the circle (except for the point $(0,1)$ which arises only in the limit $x \to \pm\infty$). An optimal enclosure of $\operatorname{ratcos}(x)$ for $x \in \mathbf{x}$ is simply obtained using (2) with $\mathbf{x}^2$ in place of $\mathbf{x}$ and 1 in place of $\mathbf{y}$.



The function ratsin is a little more complex. It has its extrema at $x = \pm 1$, with values ± 1, and is monotone in each of $[-\infty, -1]$, $[-1, 1]$, and $[1, \infty]$. As a result, one gets eight distinct cases for the optimal enclosure $\mathbf{f}$ of $\operatorname{ratsin}(x)$ for $x \in \mathbf{x}$. Using $f^l := \operatorname{ratsin}(\underline{x})$ and $f^u := \operatorname{ratsin}(\overline{x})$, we get in exact arithmetic:

```
if x̲ < −1,
   if x̄ ≤ −1,  f = [f^u, f^l];    % decreasing
   elseif x̄ < 0,  f = [−1, max(f^l, f^u)];
   elseif x̄ < 1,  f = [−1, f^u];
   else  f = [−1, 1];
   end;
elseif x̄ > 1,     % and already x̲ ≥ −1
   if x̲ < 0,  f = [f^l, 1];
   elseif x̲ < 1,  f = [min(f^l, f^u), 1];
   else  f = [f^u, f^l];    % decreasing
   end;
else     % now x ⊆ [−1, 1]
   f = [f^l, f^u];    % increasing
end;
```

Adding the necessary directed rounding for finite precision calculations is straightforward.

In some cases, a similar prior analysis can be done also for functions of two or more variables. In particular, homogeneous rational functions of two variables x and y can be reduced to the univariate case by considering the reduced variable $z = x/y$, which can then be handled analytically by an analysis of its extrema. (Care is needed in this reduction to get correct results when $\mathbf{y}$ contains zero and for unbounded intervals $\mathbf{x}$ or $\mathbf{y}$; a separate analysis should be performed to ensure that what is programmed really corresponds to the precise mathematical meaning of the original expression.)

3 Other improvement techniques

If monotonicity arguments do not apply, there are a number of other techniques that can be used to improve range enclosures. They are discusssed in this section.

Rearrangement of expressions. Suggested already in the ground-breaking book by Moore [25], one way to improve the naive interval enclosure is by rearranging expressions using the subdistributive law

$$\mathbf{f}(\mathbf{g}+\mathbf{h}) \subseteq \mathbf{fg}+\mathbf{fh}$$

or other algebraic transformations, such as that of $x \in [0,1]^n$, $f(x) = 1/\prod_{i=1}^n (1 - x_i + x_i^2)$ into $f(x) = 1/\prod_{i=1}^n (0.75 + (x_i - 0.5)^2)$ considered in Neumaier [29, p. 38], saving an exponential amount of work in a branch and bound method for minimizing $f(x)$.

The optimal handling of general univariate quadratic expressions with interval coefficients is discussed in Alolyan [1] and Domes & Neumaier [6]. For multivariate polynomials, Moore [27] presents a number of rearrangement techniques. The frequently useful representation of a univariate or multivariate polynomial in Bezier form and its interval evaluation is discussed, e.g., in Garloff & Smith [8] and Neumaier [35].

Using centered forms. Centered forms are traditionally used for the enclosure of functions over fairly narrow intervals since they enjoy the quadratic approximation property, which guarantees an asymptotically small overestimation for sufficiently narrow intervals. This makes centered forms an indispensible part of most branch and bound algorithms for problemswith continuous variables.

That an algorithm for enclosing ranges has the quadratic approximation property means that if the argument width goes to zero like $O(\varepsilon)$, then the overestimation is $O(\varepsilon^2)$ rather than the typical $O(\varepsilon)$. The most general and precise result about the quadratic approximation property is in Neumaier [32, Theorem 8.1], and applies to traditional centered forms (and can be easily adapted to the bicentered forms discussed below), linear Taylor models, and to enclosures computed by affine arithmetic or G-intervals.

One way of computing centered forms is by using slopes. If $f(x)$ is an arithmetic expression in the components of x, a **slope** for f (Krawczyk & Neumaier [21]) is a row vector expression $f[z, x]$ such that

$$f(x) = f(z) + f[z, x](x - z) \quad \text{for all } x, z. \tag{9}$$

Inserting for x the box $\mathbf{x}$ and for z an element from $\mathbf{x}$ (often the midpoint or a suitable corner) results in an enclosure with the quadratic approximation property. For the computation of slopes, see [21], Neumaier [29, Sections 2.2 and 2.3], Shen & Wolfe [56], Kolev [17], and Schnurr [54, 55]. For wider intervals, the fact that the slope itself is evaluated naively may result in unnecessarily wide enclosures. To improve upon this, Stolfi & de Figueiredo [58] discuss **affine arithmetic** as a recursive way of computing centered forms, often with less overestimation. Related work on so-called **G-intervals**, together with applications to the solution of nonlinear systems, has been done by Kolev [18, 19, 20].

Let me give explicitly the slope for a particular example, that of the **Rayleigh quotient**

$$f(x) := \frac{x^T A x}{x^T x}. \tag{10}$$

of a (thin) symmetric matrix A. One easily checks that (9) holds with

$$f[x, z] = \frac{(A - f(z)I)(x + z)}{x^T x}.$$

This can be immediately used for interval evaluation if the denominators are computed as $\sum_i x_i^2$ to avoid overestimation when intervals are inserted.

Here I want to draw attention to the bicentered form introduced in Neumaier [29, p.59], which has some nice additional properties but seems to have been little used in practice.

3.1 Theorem. (Bicentered form)
If $\mathbf{f}^l, \mathbf{s}^l, \mathbf{f}^u$ and $\mathbf{s}^u$ are enclosures of $f(x^l), f[x^l, x], f(x^u)$ and $f[x^u, x]$, respectively, for two particular points $x^l, x^u \in \mathbf{x}$ and all $x \in \mathbf{x}$ then the intervals

$$\mathbf{f}_{\text{inner}} := [\sup \mathbf{f}^l, \inf \mathbf{f}^u], \tag{11}$$

$$\mathbf{f}_{\text{outer}} := \Big[\inf(\mathbf{f}^l + \mathbf{s}^l(\mathbf{x} - x^l)), \sup(\mathbf{f}^u + \mathbf{s}^u(\mathbf{x} - x^u))\Big] \tag{12}$$

satisfy

$$\mathbf{f}_{\text{inner}} \subseteq \mathbf{f} := \{f(x) \mid x \in \mathbf{x}\} \subseteq \mathbf{f}_{\text{outer}}.$$

Proof. The validitiy of the inner approximation follows from the fact that if $\mathbf{f}_{\text{inner}}$ is nonempty then

$$\underline{f} \le f(x^l) \le \sup \mathbf{f}^l \le \inf \mathbf{f}^u \le f(x^u) \le \overline{f}.$$

The validity of the outer approximation follows from the defining equation (9) of the slope, applied to $z = x^l$ and $z = x^u$ since

$$\inf \Big(\mathbf{f}^l + \mathbf{s}^l(\mathbf{x} - x^l)\Big) \le \inf_{x \in \mathbf{x}} \Big(f(x^l) + f[x^l, x](x - x^l)\Big) = \inf_{x \in \mathbf{x}} f(x),$$

$$\sup \Big(\mathbf{f}^u + \mathbf{s}^u(\mathbf{x} - x^u)\Big) \ge \sup_{x \in \mathbf{x}} \Big(f(x^u) + f[x^u, x](x - x^u)\Big) = \sup_{x \in \mathbf{x}} f(x).$$

□

The outer approximation (12) is the bicentered form from [29, p.59]. Suitable values for x^l and x^u can be found by the endpoints minimizing and maximizing a linear Taylor expansion at the midpoint of $\mathbf{x}$, a choice suggested by POWNUK [46] to get a good inner approximation (11) of the range that frequently is optimal. Alternatively, a more sophisticated local search may be employed to find good x^l and x^u.

With exact arithmetic and Pownuk's choice for the centers, *the theorem gives the exact range, and proves optimality*, whenever f is strictly monotone in each variable and the slope enclosures are good enough. Indeed, in that case, the components of $\mathbf{s}^l$ and $\mathbf{s}^u$ have constant sign due to monotonicity, and the signs are such that $\inf \mathbf{s}^l(\mathbf{x} - x^l) = 0$ and $\sup \mathbf{s}^u(\mathbf{x} - x^u) = 0$. Since in exact arithmetic, $\mathbf{f}^l = f(x^l)$ and $\mathbf{f}^u = f(x^u)$ are thin, we find $\mathbf{f}_{\text{inner}} = \mathbf{f}_{\text{outer}}$.

Note that the bicentered form may even be optimal for expressions that are not monotone. This is shown by the univariate example $f(x) = 4x^3 - 12x$, which has the slope $f[z, x] = 4(z^2 + xz + x^2) - 12 = (2x + z)^2 + 3(z^2 - 4)$. In the second form of the slope, the interval evaluation is optimal, and produces a nonnegative slope $f[z, \mathbf{x}]$ when z is an endpoint of $\mathbf{x}$ with $|z| \ge 2$. Thus Theorem 3.1 gives the exact range not only for intervals in the ranges $[-\infty, -1]$, $[-1, 1]$, and $[1, \infty]$ where f is monotone but also for intervals $\mathbf{x}$ containing $[-2, 2]$, although these intervals contain the local maximum and a local minimum of f at $x = \pm 1$.

Since no monotonicity assumption is needed, the bicentered form provides stronger results than monotonicity arguments, although at a significantly higher cost (typically about six times the cost of an interval evaluation). However, in all but fractional multilinear cases and some other simple expressions, checking monotonicity usually requires the prior enclosure of partial derivatives and checking their sign. If the signs are not constant, one may use the optimal centered form of BAUMANN [3] that uses the interval derivatives in the best possible way to get an enclosure. However, there is little computational advantage compared to bicentered forms, and the latter's wider applicability (and typically higher sharpness) makes them the method of choice.

Using implied equations and inequalities. In a number of cases of practical interest, subexpressions in a longer calculation are not independent, and there may be interesting relations between them that can be exploited to get better enclosures. For example, in a context where $u = w^2 + x^2$ and $v = y^2 + z^2$ are relevant subexpressions, an interesting enclosure of the expression $2d/u$, where $d = b(xy - wz) + c(xz + wy)$, may be found as follows: We use the product formula

$$(w^2 + x^2)(y^2 + z^2) = (xy - wz)^2 + (xz + wy)^2 \tag{13}$$

(due to Gauss) and the Cauchy-Schwarz inequality to get $|d| \le \sqrt{uv(b^2 + c^2)}$, and obtain with $k := 4(\mathrm{mag}(\mathbf{b})^2 + \mathrm{mag}(\mathbf{c})^2)$ (where $\mathrm{mag}(\mathbf{x}) = \max(-\underline{x}, \overline{x})$) the enclosure

$$2d/u \in \Big[-\sqrt{kq}\,, \sqrt{kq}\,\Big]. \tag{14}$$

as a function of $q = v/u$. This enclosure will be useful in solving the interval challenge below.

In general, implied equations such as the Gaussian product formula (13) can be found automatically and rigorously by symbolic algebra, using Gröbner basis techniques (BUCHBERGER & WINKLER [4]); cf. also HUYER & NEUMAIER [12]. Similarly, implied inequalities such as the Cauchy-Schwarz inequality can be found automatically and rigorously by SOS (sum of squares) techniques; see, e.g., PARILLO [41] and SCHICHL & NEUMAIER [53] for general theory, and DOMES [5] for a rigorous implementation of some SOS techniques.

Though these methods are slow, they need to be applied only once to an expression, and provide – when successful – important information that may improve later many computed enclosures.

Using Kaucher arithmetic. In a number of cases (see NEUMAIER [35] for a fuller discussion), the results obtainable from a monotonicity analysis can also be obtained by using modal coercion theorems together with Kaucher interval arithmetic. For example, the modal procedure described in the recent patent by HAYES [11] gives an alternative optimal enclosure for (4).

In practice, the use of the coercion theorems for range enclosures is restricted to cases where the expression to be enclosed is totally monotonic, so that the Kaucher interval calculations simply reduce to two separate calculations involving the bounds and real arithmetic with directed rounding. These cases are covered by Therorem 2.1.

Combining centered forms and monotonicity. Functions of the form $f(x,y) = f_0(x) + F(x)^T y$ are always monotone in y, hence y can be reduced to bounds. Which bounds need to be considered can be found out by looking at the signs of the components of $F(\mathbf{x})$. With y fixed at a bound, one can use half a bicentered form in x for each bound to get good enclosures for the range.

Popova et al. [42, 43, 44, 45] (see also Tonon [59]) use Kaucher interval arithmetic for bounding ranges of monotone rational functions. In [43, 44] range enclosure is combined with a general-purpose technique for guaranteed enclosure for the solution of linear systems whose coefficients are rational functions of uncertain parameters,

$$A(p)x = b(p), \quad p \in \mathbf{p} \in \mathbb{IR}^s, \tag{15}$$

where $A(p) \in \mathbb{R}^{n\times n}, b(p) \in \mathbb{R}^n$ are rational in p. While [43, 44] use Kaucher arithmetic, the authors point out that more powerful (general) range enclosing techniques can be used instead. Indeed, the technique works as well with monotonicity arguments or centered forms, and is then more generally applicable since no total monotonicity is required. The basic idea is to use preconditioning to bring the linear system into a system of equations in fixed-point form,

$$x_i = \frac{\alpha_i(p)}{\beta_i(p)} + \frac{r_i(p)^T x}{\gamma_i(p)} \quad (i = 1, \dots, n) \tag{16}$$

with denominators $\beta_i(p), \gamma_i(p)$ that do not vanish for $p \in \mathbf{p}$, and small residual coefficient vectors $r_i(p)$. If, for all $p \in \mathbf{p}$ and all x in some trial box $\mathbf{x}^0$, the ith right-hand side can be enclosed in an interval $\mathbf{x}_i^1$ contained in the interior of $\mathbf{x}_i^0$, standard fixed point theorems [29] imply that the solution set of (15) is contained in $\mathbf{x}^1$, and further iteration (with $\mathbf{x}^0$ replaced by $\mathbf{x}^1$, etc.) may improve the bounds further. The range enclosure of the right-hand side of (16) can be done by the methods described in the present paper. In particular, if the dependence of the coefficients on the parameters is linear, the right-hand side is multilinear or fractional multilinear (depending on how the transformation to fixed-point form is done), and the above theorems apply. (As shown in Jansson [13], the affine-linear dependence also allows a representation in which each interval parameter appears only once.) For a nonlinear parameter dependence, the bicentered form from Theorem 3.1 may be used.

Inner enclosures for the hull can by found either using Kaucher interval arithmetic by the methods of Popova et al. [43, 44], or by specializing the parameters to values guessed from a linearized monotonicity analysis, and then proceeding as in the proof of Theorem 3.1. This is the monotonicity method of Pownuk [46]; for an analysis, see the discussion in Neumaier & Pownuk [38, Section 5 and Example 7.1], where good inner and outer bounds are computed for a linear finite element problem with 81 variables and 101 uncertain parameters.

The technique extends in principle to nonlinear systems of equations but is there less easy to use since these may have several zeros even in generic situations.

Using branch and bound techniques. Range enclosures can always be found by globally minimizing and maximizing the given function over the given box. Thus general techniques from rigorous global optimization [31, 40] may be applied. These techniques are usually much more expensive than any of the previously discussed methods, but they give quite accurate enclosures even for wide intervals and in the absence of any special properties. A somewhat cheaper version often applicable when the input widths are not very large is the range reduction technique by Makino [22, pp. 128–130], generalized and made precise in Neumaier [32, Theorem 9.1].

Optimal enclosure of expressions in the Vienna proposal. This paper calls attention to the fact that subexpressions frequently used in an application can be treated better by special analysis rather than by the naive approach. An interval arithmetic standard should enable the efficient use of these techniques in general rather than require a handful of specific cases.

There are many possible nonstandard expressions that would benefit from an optimal implementation using special analysis. The Vienna proposal for interval standardization [36] does not require the implementation of any of these but suggests some expressions that are practically useful and amenable to a special treatment, namely

- a number of unary operations such as (8) or $(e^x - 1 - x)/x^2$,
- the interval enclosure of arbitrary constant expressions without any interval argument,
- the linear interpolation operation.

The implementation of linear interpolation is recommended since it is a basic operation with a direct geometric meaning. Everyone is familiar with it, it is used in many contexts, and I had met the operation already several times before in applications. There are no other operations like that. Moreover, it is encountered in diverse applications, and it is a nuisance that the naive interval extension often contains points outside the convex hull of the endpoints. Also, it is quite basic in that it enables one to get good enclosures for polynomials and splines given in control polygon form; see the discussion in Hayes [10] and Neumaier [35], where algorithms for optimal interpolation operations are described (by Hayes in terms of modal interval arithmetic, by me in terms of monotonicity and directed rounding only).

4 On hardware for intervals and directed rounding

For more complicated expressions, one often ends up in a mix of interval calculations and real calculations involving directed rounding.

Other applications of directed rounding arise when explicit use is made within interval computations of monotonicity behavior of certain formulas, which allows to compute lower and upper bounds separately using directed rounding. Apart from the algorithms discussed in this paper that exploit partial monotonicity, many other kinds of rigorous computations require such a mix, for example,

- the optimal solution of linear systems with an interval M-matrix by Barth & Nuding [2],
- algorithms by Rohn [51] (see also [29, Chapter 7]) for the hull of interval linear systems, drastically reducing the number of endpoint combinations to be considered,
- the Hansen-Bliek method [30] for solving linear systems with interval coefficients,
- optimal a posteriori bounds in linear programming [14, 16, 39],
- Makino's range reduction technique, mentioned towards the end of Section 3.

Thus, implementations of interval arithmetic should be such that mixed interval calculations and real calculations involving directed rounding can be done efficiently.

One way to achieve this in hardware is to have three floating-point pipelines, each one dedicated to one particular rounding mode (up, down, nearest), with the directed modes following Part 7 of the Vienna proposal for interval standardization [36]. Then it will pay to adapt the algorithms (or the compilers) to optimally use the directed rounding pipelines.

Alternatively, one could implement three different pipelines for floating-point operations in fixed, selectable rounding modes. These can then also be used in ordinary floating-point calculations to parallelize computations, and in interval calculations to do operations in different rounding modes in different pipelines.

If lower bounds of intervals are represented by their negatives (as in Remark 2 in Section 2.3 of the Vienna proposal [36]), the rounding up mode would suffice for directed calculations, and one could make do with two pipelines in place of three.

This would completely remove the penalty for rounding mode switches; an optimizing compiler can choose the rounding modes of each pipeline to maximize total throughput with a minimal number of switches. For example, during interval computations interspersed by approximate floating-point computation (frequent in global optimization), one pipeline might be set to round up, one to round down, and one to round nearest.

In view of the many situations mentioned where one wants to do some directed rounding calculations within some routine using interval arithmetic, the directed pipelines should be available for both interval and directed calculations.

5 A simple interval challenge

In this section, a nontrivial model problem is introduced to which the general techniques discussed above are applied in the next section.

The following test problem, suggested by Nate Hayes (personal communication) and arising from a problem in computer vision, was made public on November 26, 2008 on the Reliable Computing mailing list [50]. Its goal was to expose insights and tools experts have to squeeze the best out of a very limited budget for one range enclosure. It doesn't say anything about the performance of existing techniques or software for range enclosures in general.

Consider the expression

$$f := \frac{a(w^2 + x^2 - y^2 - z^2) + 2b(xy - wz) + 2c(xz + wy)}{w^2 + x^2 + y^2 + z^2}. \tag{17}$$

When a, b, c are thin or very narrow intervals, this enclosure problem is handled well as that of enclosing a Rayleigh quotient (10), which can be efficiently done by centered and bicentered forms, as discussed in Section 3. However, this gives somewhat poor results when a, b, c are fairly wide intervals.

Challenge. Find a cheap and good enclosure for (17) given the following bounds on the variables a, b, c, w, x, y, z:

$$a \in [7, 9], \quad b \in [-1, 1], \quad c \in [-1, 1], \tag{18}$$

$$w \in [-0.9, -0.6], \quad x \in [-0.1, 0.2], \quad y \in [0.3, 0.7], \quad z \in [-0.2, 0.1]. \tag{19}$$

The algorithm used should give a valid enclosure for all boxes whose bounds have the same sign distribution as the bounds shown, irrespective of the values of the bounds. But the work count and the quality of the resulting enclosure are tested only for the particular box defined by (18) and (19). (For other boxes, see the remarks at the end of Section 6.) The computation together with general theoretical results must constitute a proof that the result is a rigorously valid enclosure of the range.

The cost is measured as the number of effective interval operations, defined for simplicity (and in view of potential hardware realizations) as follows:

- Any unary or binary operation involving a (standard or nonstandard) interval, including taking the hull or the intersection of two intervals, is counted as an interval operation.
- A purely real operation and a real compare (used in a branching statement) are counted each as half an interval operation.
- Changing a sign and checking for a sign is not counted.
- Switches of rounding modes are not counted.
- Algorithms using more than 200 effective interval operations do not qualify.

The remainder of this section discusses some properties of the challenge problem. Interesting solutions provided by those who responded to the challenge are given in Section 6.

Origin of the problem. It is well-known that arbitrary 3-dimensional rotations can be expressed in the form

$$Q[r] = \mathbf{1} + 2\begin{pmatrix} -r_2^2 - r_3^2 & r_1r_2 - r_0r_3 & r_1r_3 + r_0r_2 \\ r_1r_2 + r_0r_3 & -r_1^2 - r_3^2 & r_2r_3 - r_0r_1 \\ r_1r_3 - r_0r_2 & r_2r_3 + r_0r_1 & -r_1^2 - r_2^2 \end{pmatrix}, \quad r_0 = \sqrt{1 - |r|^2}, \tag{20}$$

where $\mathbf{1}$ denotes the identity matrix and $r \in \mathbb{R}^3$ satisfies

$$|r| := \sqrt{r_1^2 + r_2^2 + r_3^2} \leq 1.$$

If $r = 0$ then $Q[r]$ is the identity; otherwise, $Q[r]$ describes a rotation around the axis through the vector r by the angle $\alpha = 2 \arcsin |r|$. Alternatively, we may write (20) in the homogeneous *quaternion parameterization*

$$Q[r_0, r] = \mathbf{1} + \frac{2}{r_0^2 + r_1^2 + r_2^2 + r_3^2}\begin{pmatrix} -r_2^2 - r_3^2 & r_1r_2 - r_0r_3 & r_1r_3 + r_0r_2 \\ r_1r_2 + r_0r_3 & -r_1^2 - r_3^2 & r_2r_3 - r_0r_1 \\ r_1r_3 - r_0r_2 & r_2r_3 + r_0r_1 & -r_1^2 - r_2^2 \end{pmatrix} \tag{21}$$

with independent $r_0 \in \mathbb{R}$, $r \in \mathbb{R}^3$, not both zero. $Q[r_0, r]$ satisfies

$$Q[r_0, r] = Q[\lambda r_0, \lambda r] \quad \text{for all } \lambda \neq 0 \tag{22}$$

and reduces to $Q[r]$ if the arbitrary scale is chosen such that $r_0^2 + r_1^2 + r_2^2 + r_3^2 = 1$ and $r_0 \geq 0$. Because of (22), parallel vectors (r_0, r) in the quaternion parameterization give the same rotation. This shows that the 3-dimensional rotation group has the topology of a 3-dimensional projective space.

In computational geometry, the quaternion parameterization is preferable to the more frequently discussed (and more elementary) parameterization by Euler angles, since it does not need expensive trigonometric functions, its parameters have a geometric meaning independent of the coordinate system used, and it has significantly better interpolation properties (Shoemake [57], Ramamoorthi & Barr [48]). Note that the projective identification mentioned above has to be taken into account when constructing smooth motions joining two close rotations $Q[r]$ with nearly opposite r of length close to 1.

The first component of the vector $v := Q[r_0, r]u$ obtained by rotating the vector $u \in \mathbb{R}^3$ by the rotation $= Q[r_0, r]$ takes the form (17) when

$$u = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \quad r_0 = w, \quad r = \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Symmetries. The challenge problem has a lot of structural symmetry, broken only by the bound constraints specified. The symmetry group of the problem is isomorphic to the direct product of $\mathbb{R}^\times \times (\mathbb{Z}/2\mathbb{Z})^4$, where $\mathbb{R}^\times$ denotes the multiplicative group of nonzero reals, and is generated by the scalings $(w,x,y,z) \to (tw,tx,ty,tz)$ for nonzero t and the transposition products

$$(wx)(bc)(z-z),\ (yz)(bc)(w-w),\ (wy)(xz)(a-a)(b-b),$$

and

$$(w-w)(x-x)(b-b)(c-c)\ \text{ or }\ (y-y)(z-z)(b-b)(c-c).$$

Here (wx) swaps w and x, while $(z-z)$ denotes a sign change of z, etc.

Exact solution. For comparison, the exact solution of the challenge problem was obtained for reference purposes by Mihaly Markot and Dan Zuras (personal communication), using a combination of a branch and bound method for verifying the correct active sets at the minimum and the maximum with the solution of a quadratic equations for the residual problem in which all activities were fixed.

The minimum is exactly attained at

$$a=9; b=1; c=1; w=-0.6; x=10.62-\sqrt{113.6744}; y=0.7; z=-0.2;$$

and has the value

$$\begin{aligned} f_{\min} &= (270-\sqrt{284186})/89 \\ &= -2.9560785011851257870357909731159456912301156279187038495?u1. \end{aligned}$$

The maximum is exactly attained at

$$a=9; b=1; c=-1; w=-0.9; x=0.2; y=0.3; z=\sqrt{130.9}-11.4;$$

and has the value

$$\begin{aligned} f_{\max} &= (7*\sqrt{13090}-48)/94 \\ &= 8.0093698421059609250021007563731486914427953942424830543?u1. \end{aligned}$$

The optimal width, i.e., the width of the true range, is

$$w^* := f_{\max} - f_{\min} = 10.9654483432910867120378917294890943826729110221611869 0?u1.$$

(By Section 6.4 of the Vienna proposal [36] for the standardization of interval arithmetic, the ending ?u1 denotes the fact that the true result is between the stated result and the number 1 ulp larger in absolute value, i.e., that it is a correctly truncated result.)

6 Public solutions to the challenge

We now discuss a number of solutions to the simple interval challenge, illustrating the general principles discussed above. (Upon request, programs doing the computations described below, encoded in INTLAB, the Matlab interval toolbox by RUMP [52], may be obtained from the author.)

Naive interval evaluation. The simple interval evaluation of f gives the enclosure $[-7.4889, 19.2889]$, which has 244% of the optimal width. Using savings due to precomputing $u = w^2 + x^2$ and $v = y^2 + z^2$ and evaluating

$$f = (a(u - v) + 2(b(xy - wz) + c(xz + wy)))/(u + v),$$

this takes 21 interval operations.

Using rearrangement. Ramon Moore provided a pure rearrangement evaluation, evaluating

$$u = w^2 + x^2, \quad v = y^2 + z^2,$$
$$r = 1 - 2/(u/v + 1), \quad s = b(xy - wz) + c(xz + wy),$$
$$f = ar + 2s/(u + v)$$

with 24 operations, giving the enclosure $[-5.8080, 11.3655]$, which has 157% of the optimal width. The operation count may be reduced to 23 effective operations by using (2) to compute the same enclosure for r. Note that the rearrangements $s = (bx+cw)y+(cx-bw)z$ or $s = (by + cz)x + (cy - bz)w$ of the expression for s lead to inferior results.

Using monotonicity. A partial monotonicity approach is successful based upon the representation

$$u = w^2 + x^2, \quad v = y^2 + z^2,$$
$$d = b(xy - wz) + c(xz + wy),$$
$$f = (a(u - v) + 2d)/(u + v).$$

Since d is multilinear in the variables, we can compute an enclosure for d according to Theorem 2.2 by specializing x and z to their four endpoint combinations and taking the hull of the results. Similarly, since f is fractional linear, we can compute an enclosure for f by specializing u and v to their four endpoint combinations and taking the hull of the results. Taking care of common subexpressions in the resulting formulas, this leads to 54 interval operations and 12 real operations, hence 60 effective interval operations, giving the enclosure $[-3.6517, 9.2223]$, which has 117.4% of the optimal width.

Using linear interpolation. John Pryce reported an approach based upon the rearrangement

$$W = w^2, \quad X = x^2, \quad Y = y^2, \quad Z = z^2,$$

$$p = (W + X - Y - Z)/(W + X + Y + Z),$$
$$r = 2(xy - wz)/(W + X + Y + Z), \quad s = 2(xz + wy)/(W + X + Y + Z),$$
$$f = ap + br + cs.$$

Moore's recipe gives optimal bounds for p:

$$u = W + X, \quad v = Y + Z, \quad p = 1 - 2/(1 + u/v).$$

To get good bounds on r, s, Pryce suggested the formulas

$$u' = X + Y, \quad v' = Z + W, \quad t' = 1/(1 + u'/v'),$$
$$r = \text{linearInt}(\text{ratsin}(x/y), \text{ratsin}(-z/w), t'),$$
$$u'' = X + Z, \quad v'' = Y + W, \quad t'' = 1/(1 + u''/v''),$$
$$s = \text{linearInt}(\text{ratsin}(x/z), \text{ratsin}(y/w), t''),$$

with linearInt and ratsin as defined in (1) and (8). The enclosure obtained is $[-3.4000, 8.9575]$, which has 112.7% of the optimal width. Since ratsin and linearInt need 5 and 4 effective interval operations, respectively, Pryce's recipe takes 57 effective interval operations. But p, t', and t'' may be computed instead by (2) and (3), saving some operations and leading to an equivalent algorithm that takes only 54 effective interval operations.

Digression: A degenerate global optimization problem. As part of playing with the approach of Pryce, we (Mihaly Markot, Hermann Schichl and myself) looked at the innocent-looking subproblem

$$\begin{array}{ll} \min & s = 2(xz + wy)/(w^2 + x^2 + y^2 + z^2) \\ \text{s.t.} & w \in [-0.9, -0.6], \quad x \in [-0.1, 0.2], \\ & y \in [0.3, 0.7], \quad z \in [-0.2, 0.1] \end{array}$$

for getting the optimal lower bound for s. Its solution turned out to be remarkably difficult for several of our solvers; so we analyzed it more closely. It turned out that the problem has a 2-dimensional continuum of global minimizers with minimum -1, attained at any point with $z = -x \in [-0.2, 0.1]$ and $y = -w \in [0.6, 0.7]$. This makes standard branch and bound codes extremely slow.

On the other hand, the above recipe

$$\mathbf{u} = \mathbf{x}^2 + \mathbf{z}^2; \; \mathbf{v} = \mathbf{y}^2 + \mathbf{w}^2; \; t = [\underline{v}/(\underline{v} + \overline{u}), \overline{v}/(\overline{v} + \underline{u})];$$
$$s = \text{linearInt}(\text{ratsin}(x/z), \text{ratsin}(y/w), t);$$

gives the optimal lower bound without any branching. Thus a prior analysis of expressions defining a global optimization problem may significantly contribute to its easy solvability.

Using centered forms. Lubomir Kolev noted that one may consider the squares of w, x, y, z as independent variables and express w, x, y, z in terms of these and signs. Since x

and z may change signs, this leads to four distinct cases to be considered. Using G-interval techniques, linearizing the square roots by a min-range approximation, and taking for a, b, c the 8 possible endpoint combinations, Kolev reported the enclosure $[-3.2472, 8.4992]$, which is 107.1% of the optimal width, but with over 200 effective interval operations.

Using implied equations and inequalities. The final algorithm to be discussed uses very specific properties of the particular expression, and has the best performance among all known solutions to the challenge. Based on the discussion of (13)–(14) and (7), we get the desired enclosure as follows:

```
% Neumaier's Gaussian recipe
% 7 interval + 27 real ops = 20.5 effective ops
disp('Gaussian')
u=w^2+x^2;v=y^2+z^2;                          % 2 x (+,2 sqr)

% use the Gaussian product formula uv=(xy-wz)^2+(xz+wy)^2
% and the Cauchy-Schwarz inequality to get |d|<=sqrt(uv(b^2+c^2)),
% hence 2d/u = e*sqrt(kq) for some e in [-1,1], where
k=4*(mag(b)^2+mag(c)^2)                       % real(2 max,2 sqr,+*)
q=v/u;infsup(q)                               % /

% now f in (a(1-q)+sqrt(kq)*[-1,1])/(1+q)
% simple sufficient monotonicity condition
if inf(a)>0 & sup(q)<=1+16*inf(a)^2/k & sup(q)>1 & inf(q)<1,
                                              % real(sqr,*/,4+)
  % f is monotone falling in q since a>0
  % and both extremes are extremal at sup(a)
  sq=sup(q);
  finf=(sup(a)*(1-sq)-sqrt(k*sq))/(1+sq);
  iq=inf(q);
  fsup=(sup(a)*(1-iq)+sqrt(k*iq))/(1+iq);    % 2 x real(3+,2x,/,sqrt)
else
  % simple monotonicity test fails
  % more complex version not programmed
  finf=-inf;fsup=inf;
end;
ff=infsup(finf,fsup);
```

The result is the enclosure $[-3.1073, 8.1089]$, which has 102.3% of the optimal width). The algorithm used 7 interval operations and 27 real operations, hence 20.5 effective interval operations. Thus, quite unexpected by me, it is even cheaper than the ordinary interval evaluation!

A fully rigorous enclosure would also have to set correct rounding modes, which was not done here for better readability. Note that parallelizing the Gaussian recipe essentially halves the time involved in the computation, and speculative evaluation of the main branch (explicitly programmed above) may effectively eliminate the overhead for the branching statement.

Other ideas. John Pryce noticed that f can be represented as the Rayleigh quotient $f = k^*Mk/k^*k$ of the complex Hermitian matrix $M = \begin{pmatrix} a & c+ib \\ c-ib & -a \end{pmatrix}$ at the vector $k = \begin{pmatrix} w+ix \\ y+iz \end{pmatrix}$, where $i = \sqrt{-1}$. The eigenvalues of M are $\mp\sqrt{a^2+b^2+c^2}$; hence

$$|k^*Mk| \le \sqrt{(a^2+b^2+c^2)(w^2+x^2+y^2+z^2)},$$

$$|f| \le \sqrt{(a^2+b^2+c^2)/(w^2+x^2+y^2+z^2)},$$

which gives the poor enclosure $[-13.5810, 13.5810]$. But perhaps the idea can be better exploited in another way.

Further challenges. For narrow input intervals, the quality of a particular method may change considerably, and other methods (related to centered forms) will probably perform better. This should be investigated.

Also, the question of how to best compute an enclosure in the completely general case (without sign restrictions on the input) should be addressed, and the question of how an algorithm treats different input boxes related to each other by one of the symmetries. Also, cheap and good enclosures for *all* components of $v = Q[r_0, r]u$ simultaneously are of interest.

Finally, it would be interesting to find global a priori bounds on the amount of overestimation of some of the better methods.

References

[1] I. Alolyan, Real roots of quadratic interval polynomials, Int. J. Math. Analysis 1 (2007), 1041–1050.
`http://docs.ksu.edu.sa/PDF/Articles17/Article170540.pdf` [8]

[2] W. Barth and E. Nuding, Optimale Lösung von Intervallgleichungssystemen, Computing 12 (1974), 117-125. [4, 14]

[3] E. Baumann, Optimal centered forms, BIT 28 (1988), 80-87. [11]

[4] B. Buchberger and F. Winkler, Groebner Bases: Theory and Applications, Cambridge University Press, 1998. [11]

[5] F. Domes, GloptLab, a configurable framework for the rigorous global solution of quadratic constraint satisfaction problems, Manuscript (2008).
`http://www.mat.univie.ac.at/~dferi/publ/Gloptlab.pdf` [11]

[6] F. Domes and A. Neumaier, Quadratic constraint propagation. Manuscript (2008).
`http://www.mat.univie.ac.at/~dferi/publ/Propag.pdf` [8]

[7] J.E. Flores Diaz, Improvement in the ray tracing of implicit surfaces based on interval arithmetic, Ph.D. Thesis, Girona, Spain, November 2008. [2]

[8] J. Garloff and A.P. Smith, Rigorous affine lower bounds for multivariate polynomials and their use in global optimization, pp. 199–211 in: Lecture Notes in Management Science 1, Springer 2008. [9]

[9] C.R. Gwaltney, Y. Lin, L.D. Simoni and M.A. Stadtherr, Interval methods for nonlinear equation solving applications, in Handbook of Granular Computing (W. Pedrycz et al., eds.), Wiley (2008).
`http://www.nd.edu/~markst/granular-chapter.pdf` [2]

[10] N.T. Hayes, System and method to compute narrow bounds on a modal interval polynomial function, U.S. Patent 20080256155 (2008).
`http://www.faqs.org/patents/app/20080256155` [13]

[11] N.T. Hayes, System and method to compute narrow bounds on a modal interval spherical projection, U.S. Patent 20080307027 (2008).
`http://www.faqs.org/patents/app/20080307027` [11]

[12] W. Huyer and A. Neumaier, Integral approximation of rays and verification of feasibility, Reliable Computing 10 (2004), 195–207. [11]

[13] C. Jansson, Interval linear systems with symmetric matrices, skew-symmetric matrices and dependencies in the right hand side, Computing 46 (1991), 265–274. [12]

[14] C. Jansson, VSDP: Verified SemiDefinite Programming, Web site (2008).
`http://www.ti3.tu-harburg.de/jansson/vsdp/index.html` [2, 14]

[15] R.B. Kearfott, M.T. Nakao, A. Neumaier, S.M. Rump, S.P. Shary and P. van Hentenryck, Standardized notation in interval analysis, pp. 106-113 in: Proc. XIII Baikal International School-seminar "Optimization methods and their applications", Irkutsk, Baikal, July 2-8, 2005. Vol. 4 "Interval analysis". Irkutsk: Institute of Energy Systems SB RAS, 2005.
`http://www.mat.univie.ac.at/~neum/papers.html#notation` [3]

[16] C. Keil, Lurupa – rigorous error bounds in linear programming, Web page (2008).
`http://www.ti3.tu-harburg.de/~keil/#software` [2, 14]

[17] L. Kolev, Use of interval slopes for the irrational part of factorable functions, Reliable Computing 3 (1997), 83–93. [9]

[18] L. Kolev, A new method for global solution of nonlinear equations, Reliable Computing 4 (1998), 125–146. [9]

[19] L. Kolev, Automatic computation of a linear interval enclosure, Reliable Computing 7 (2001), 17–28. [9]

[20] L.V. Kolev, An improved interval linearization for solving non-linear problems, Numerical Algorithms, 37 (2004), 213–224. [9]

[21] R. Krawczyk and A. Neumaier, Interval slopes for rational functions and associated centered forms, SIAM J. Numer. Anal. 22 (1985), 604–616. [9]

[22] K. Makino, Rigorous analysis of nonlinear motion in particle accelerators, Ph.D. Thesis, Dept. of Physics and Astronomy, Michigan State University, 1998.
`http://bt.pa.msu.edu/makino/phd.html` [13]

[23] J.-P. Merlet, Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries, Int. J. Robotics Res. 18 (1999), 902–916.
`http://ijr.sagepub.com/cgi/content/abstract/18/9/902` [2]

[24] J.-P. Merlet, Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis, Int. J. Robotics Res. 23 (2004), 221–235.
`http://ijr.sagepub.com/cgi/content/abstract/23/3/221` [2]

[25] R.E. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966. [2, 8]

[26] R.E. Moore, Methods and Applications of Interval Analysis, SIAM, Philadelphia, 1979. [2]

[27] R. Moore, On reducing overestimation of ranges of multinomials without splitting boxes, Manuscript (2006).
`http://www.mat.univie.ac.at/~neum/ms/Moore-ranges.pdf` [8]

[28] R.E. Moore, R B. Kearfott and M.J. Cloud, Introduction to Interval Analysis, SIAM, Philadelphia, 2009. [2]

[29] A. Neumaier, Interval Methods for Systems of Equations, Encyclopedia of Mathematics and its Applications 37, Cambridge Univ. Press, Cambridge 1990. [2, 8, 9, 10, 12, 14]

[30] A. Neumaier, A simple derivation of the Hansen-Bliek-Rohn-Ning-Kearfott enclosure for linear interval equations, Reliable Computing 5 (1999), 131-136. Erratum, Reliable Computing 6 (2000), 227.
`http://www.mat.univie.ac.at/~neum/papers.html#ning` [14]

[31] A. Neumaier, Complete search in continuous global optimization and constraint satisfaction, pp. 271–369 in: Acta Numerica 2004 (A. Iserles, ed.), Cambridge University Press 2004.
`http://www.mat.univie.ac.at/~neum/papers.html#glopt03` [2, 13]

[32] A. Neumaier, Taylor forms - use and limits, Reliable Computing 9 (2002), 43-79. http://www.mat.univie.ac.at/~neum/papers.html#taylor [9, 13]

[33] A. Neumaier, Computer-assisted proofs, in: (W. Luther and W. Otten, eds.) Proc. 12th GAMM-IMACS Symp. Sci. Comp., (SCAN 2006). IEEE Computer Society, 2007. http://www.mat.univie.ac.at/~neum/papers.html#caps [2]

[34] A. Neumaier, Certified error bounds for uncertain elliptic equations, J. Comput. Appl. Math. 218 (2008), 125–136. http://www.mat.univie.ac.at/~neum/papers.html#pdebounds [2]

[35] A. Neumaier, Computer graphics, linear interpolation, and nonstandard intervals, Manuscript (2008). http://www.mat.univie.ac.at/~neum/papers.html#nonstandard [2, 4, 9, 11, 13]

[36] A. Neumaier, Vienna proposal for interval standardization, Manuscript (2008). http://www.mat.univie.ac.at/~neum/papers.html#1788 [3, 13, 14, 17]

[37] A. Neumaier and J.-P. Merlet, Constraint satisfaction and global optimization in robotics, Slides, 2002. http://www.mat.univie.ac.at/~neum/ms/robslides.pdf [2]

[38] A. Neumaier and A. Pownuk, Linear systems with large uncertainties, with applications to truss structures, Reliable Computing 13 (2007), 149–172. http://www.mat.univie.ac.at/~neum/papers.html#linunc [2, 4, 12]

[39] A. Neumaier and O. Shcherbina, Safe bounds in linear and mixed-integer programming, Math. Programming A 99 (2004), 283-296. http://www.mat.univie.ac.at/~neum/papers.html#mip [14]

[40] A. Neumaier, O. Shcherbina, W. Huyer and T. Vinko, A comparison of complete global optimization solvers, Math. Programming B 103 (2005), 335–356. http://www.mat.univie.ac.at/~neum/papers.html#test [2, 13]

[41] P.A. Parrilo, Semidefinite programming relaxations for semialgebraic problems. Math. Programming B 96 (2003), 293–320. [11]

[42] E. Popova, On the solution of parametrised linear systems, pp. 127–138 in: Scientific Computing, Validated Numerics, Interval Methods (W. Kraemer, J. Wolff von Gudenberg, eds.), Kluwer 2001. http://www.math.bas.bg/~epopova/papers/SCAN2000Popova.pdf [12]

[43] E. Popova, Solving linear systems whose input data are rational functions of interval parameters, pp. 345-352 in: Numerical Methods and Applications (T. Boyanov et al., eds.), Lecture Notes in Computer Science 4310, Springer, Berlin 2007. Extended version in: http://www.math.bas.bg/~epopova/papers/05PreprintEP.pdf [12]

[44] E. Popova, R. Iankov and Z. Bonev, Bounding the response of mechanical structures with uncertainties in all the parameters, pp. 245–265 in Proc. NSF Workshop Rel. Eng. Comp. Savannah, Georgia, USA, Feb. 22-24, 2006 (R.L. Muhannah and R.L. Mullen, eds).
http://www.math.bas.bg/~epopova/papers/06recPopovaEtAl.pdf [12]

[45] E. Popova and W. Kraemer, Inner and outer bounds for parametric linear systems, J. Comput. Appl. Math. 199 (2007), 310–316. [12]

[46] A. Pownuk, Efficient method of solution of large scale engineering problems with interval parameters based on sensitivity analysis, pp. 305–316 in: Proc. NSF Workshop Reliable Engineering Computing, Sept. 15–17, 2004, Savannah, GA (R.L. Muhanna and R.L. Mullen, eds.). www.gtsav.gatech.edu/rec/recworkshop/presentations/presentations.html [10, 12]

[47] J.D. Pryce and G.F. Corliss, Interval arithmetic with containment sets, Computing 78 (2006), 251–276. http://homepage.ntlworld.com/j.d.pryce/isloct05/IntvlArithCsets.pdf [5]

[48] R. Ramamoorthi and A.H. Barr, Fast construction of accurate quaternion splines, pp. 287–292 in: Proc. 24th ann. conf. Computer graphics interactive techniques, 1997.
http://www.cs.columbia.edu/cg/pdfs/64_sig97.pdf [16]

[49] S. Ratschan, Efficient solving of quantified inequality constraints over the real numbers, ACM Trans. Comput. Logic 7 (2006), 723–748.
http://arxiv.org/pdf/cs/0211016 [2]

[50] Reliable Computing mailing list.
http://interval.louisiana.edu/reliable_computing.txt [15]

[51] J. Rohn, Systems of linear interval equations, Linear Algebra Appl. 126 (1989), 39–78. [14]

[52] S.M. Rump, INTLAB - INTerval LABoratory, pp. 77–104 in: Developments in Reliable Computing (T. Csendes, ed.), Kluwer, Dordrecht 1999.
http://www.ti3.tu-harburg.de/rump/intlab/ [18]

[53] H. Schichl and A. Neumaier, Transposition theorems and qualification-free optimality conditions, Siam J. Optimization 17 (2006), 1035–1055. [11]

[54] M. Schnurr, Steigungen höherer Ordnung zur verifizierten globalen Optimierung, Universittsverlag Karlsruhe, Karlsruhe 2007.
http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/22480 [9]

[55] M. Schnurr, Computing slope enclosures by exploiting a unique point of inflections, Appl. Math. Comput. 204 (2008), 249–256. [9]

[56] Z. Shen and M.A. Wolfe, On interval enclosures using slope arithmetic, Appl. Math. Comput. 39 (1990) 89–105. [9]

[57] K. Shoemake, Animating rotation with quaternion curves, pp. 245–254 in: SIGGRAPH 85 proceedings, 1985. [16]

[58] J. Stolfi and L.H. de Figueiredo, Self-Validated Numerical Methods and Applications, 21st Brazilian Mathematics Colloquium, IMPA, 1997. http://www-sbras.nsc.ru/interval/Library/InteBooks/Stolfi/Stolfi.pdf [9]

[59] F. Tonon, Using extended interval algebra in discrete mechanics, In: NSF Workshop Reliable Engineering Computing 2006. http://www.gtsav.gatech.edu/workshop/rec06/papers/Tonon_paper.pdf [12]