Symbolic Preconditioning with Taylor Models: Some Examples†

R. Baker Kearfott (rbk@louisiana.edu)
University of Louisiana at Lafayette

G. W. Walster (bill.walster@eng.sun.com)
Sun Microsystems

Abstract. Deterministic global optimization with interval analysis involves

– using interval enclosures for ranges of the constraints, objective, and gradient to reject infeasible regions, regions without global optima, and regions without critical points;

– using interval Newton methods to converge on optimum-containing regions and to verify global optima.

There are certain problems for which interval dependency leads to overestimation in the enclosures of the individual components, causing the optimization search to become prohibitively inefficient. As Hansen has observed earlier, in other problems, there isn't overestimation in the individual components, but overestimation is introduced in the preconditioning in the interval Newton method.

We examine these issues for a particular nonlinear systems problem that, to date, has defied numerical solution. To reduce overestimation, we use Taylor models. The Taylor models sometimes reduce individual overestimation but, consistent with Hansen's observations, especially reduce the overestimation due to preconditioning. From numerical experiments, we conclude that, in certain instances, Taylor models can greatly reduce both the number of subregions necessary to complete an exhaustive search and the total computational effort.

Keywords: interval analysis, global optimization, Taylor models, preconditioning

1. Introduction

Throughout, we assume some knowledge of branch-and-bound methods for global optimization and of interval computations. There are many excellent introductions to these topics, such as (Ratschek and Rokne, 1988) for both interval computations and branch-and-bound methods, (Pardalos and Rosen, 1987, Chapter 6) for branch and bound methods, (Neumaier, 1990) for interval computations and an advanced treatment of interval nonlinear systems, and (Kearfott, 1996) for an introduction to interval computations in the context here.

1.1. The Global Optimization Context

The context of our study is deterministic global optimization via exhaustive search. For example, if the problem is the equality-constrained problem

$$\boxed{\begin{array}{l} \text{minimize } \phi(x) \\ \text{subject to } c_i(x) = 0,\ i = 1, \ldots, m_1, \\ \qquad\qquad g_i(x) = 0,\ i = 1, \ldots, m_2, \\ \text{where } \phi : \mathbb{R}^n \to \mathbb{R} \text{ and } c_i, g_i : \mathbb{R}^n \to \mathbb{R}. \end{array}} \tag{1}$$

then bounding the range of ϕ over a small region known to contain feasible points gives an upper bound for the global minimum of ϕ over the region $\boldsymbol{x}$. Some method is then used to bound the range of ϕ over subregions $\tilde{\boldsymbol{x}} \subset \boldsymbol{x}$. If the lower bound $\underline{\phi}$, so obtained, for ϕ over $\tilde{\boldsymbol{x}}$ has $\underline{\phi} > \overline{\phi}(x)$, then $\tilde{\boldsymbol{x}}$ may be rejected as not containing any global optima. Similarly, if bounds on the range of any c_i over $\tilde{\boldsymbol{x}}$ cannot contain zero, then $\tilde{\boldsymbol{x}}$ can be rejected as not containing any feasible points.

A special case is where the objective function ϕ is constant or nonexistent, and where $m = n$, so we have a square nonlinear system of equations

$$\boxed{F(x) = (c_1(x), \ldots, c_n(x)) = 0.} \tag{2}$$

Here, we focus on this case.

1.2. Behavior of Interval Newton Methods

We use interval evaluations to obtain bounds on the ranges of the c_i. We also employ interval Newton methods, in which we use interval computations to bound the solution set $\boldsymbol{v}$ to systems of the form

$$Y\boldsymbol{A}v = -YF(\check{x}), \tag{3}$$

where $\boldsymbol{A}$ is a bound on the range $\{F'(x) \mid x \in \boldsymbol{x}\}$ or an *interval slope matrix*, and where Y is a *preconditioner matrix*, often chosen to be the inverse of the matrix of midpoints of the entries of $\boldsymbol{A}$. For example, if

$$\begin{array}{rcl} f_1(x) & = & x_1^2 - x_2^2 - 1 \\ f_2(x) & = & 2x_1x_2, \end{array} \qquad \text{with} \quad \boldsymbol{x} = \begin{pmatrix} [0.9, 1.2] \\ [-0.1, 0.1] \end{pmatrix}, \quad \check{x} = \begin{pmatrix} 1.05 \\ 0 \end{pmatrix},$$

then an interval extension of the Jacobi matrix for f is

$$\boldsymbol{F}'(\boldsymbol{x}) = \begin{pmatrix} 2\boldsymbol{x}_1 & -2\boldsymbol{x}_2 \\ 2\boldsymbol{x}_2 & 2\boldsymbol{x}_1 \end{pmatrix},$$

and its value at $\boldsymbol{x}$ is

$$\begin{pmatrix} [1.8, 2.4] & [-0.2, 0.2] \\ [-0.2, 0.2] & [1.8, 2.4] \end{pmatrix}.$$

The inverse midpoint preconditioner is

$$Y = \begin{pmatrix} 2.1 & 0 \\ 0 & 2.1 \end{pmatrix}^{-1} \approx \begin{pmatrix} 0.476 & 0 \\ 0 & 0.476 \end{pmatrix},$$

and the preconditioned system corresponding to (3), rounded out, is

$$\begin{pmatrix} [0.85, 1.15] & [-.096, .096] \\ [-.096, .096] & [0.85, 1.15] \end{pmatrix} \boldsymbol{v} = \begin{pmatrix} [-.0488, -.0487] \\ 0 \end{pmatrix}.$$

The interval Gauss–Seidel method is used to compute sharper bounds on $\boldsymbol{v} = \boldsymbol{x} - \check{x}$, beginning with $\boldsymbol{v} = \begin{pmatrix} [-0.15, 0.15] \\ [-0.1, 0.1] \end{pmatrix}$. That is,

$$\tilde{\boldsymbol{v}}_1 \subseteq \frac{[-.0488, -.0487] - [-.096, .096]\boldsymbol{v}_2}{[0.85, 1.15]} \subset [-0.0688, -.034].$$

Thus, the first component of $\mathbf{N}(f, \boldsymbol{x}, \check{x})$ is $\check{x}_1 + \boldsymbol{v}_1 \subset [0.981, 1.016]$. In the second step of the interval Gauss–Seidel method,

$$\tilde{\boldsymbol{v}}_2 = (0 - [-.096, .096]\tilde{\boldsymbol{v}}_1)/[.085, 1.15] \subset [-0.00778, 0.00778],$$

so, rounded out, $\mathbf{N}(f, \boldsymbol{x}, \check{x})$ is computed to be

$$\begin{pmatrix} [0.981, 1.016] \\ [-0.00778, 0.00778] \end{pmatrix} \subset \begin{pmatrix} [0.9, 1.2] \\ [-0.1, 0.1] \end{pmatrix}.$$

A general theorem on interval Newton methods (see e.g. (Neumaier, 1990)) states that, if $\mathbf{N}(f, \boldsymbol{x}, \check{x}) \subset \boldsymbol{x}$ as above, then this proves that there is a unique solution of $F(x)$ in $\boldsymbol{x}$.

Now, to illustrate a way this procedure runs into difficulties, consider

Example 1.

$$\begin{aligned} f_1(x) &= x_1^3 - x_2^3 \\ f_2(x) &= x_1^3 + x_2^3 - 2.1 \end{aligned} \qquad \text{with} \quad \boldsymbol{x} = \begin{pmatrix} [0.7, 1.3] \\ [0.7, 1.3] \end{pmatrix}, \quad \check{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

In Example 1, interval evaluation of the individual components gives the *exact* range to within roundout error (i.e. there is no overestimation), since each variable occurs only once in each expression. However, the values of the elements in the first and second rows of the interval Jacobi matrix $\boldsymbol{F}'(\boldsymbol{x}) = \begin{pmatrix} 3\boldsymbol{x}_1^2 & -3\boldsymbol{x}_2^2 \\ 3\boldsymbol{x}_1^2 & 3\boldsymbol{x}_1^2 \end{pmatrix}$ are not independent. Thus, evaluating $\boldsymbol{F}'(\boldsymbol{x})$ componentwise *before* preconditioning gives $\boldsymbol{F}'([0.7, 1.3], [0.7, 1.3]) = \begin{pmatrix} [1.47, 5.07] & [-5.07, 1.47] \\ [1.47, 5.07] & [1.47, 5.07] \end{pmatrix}$, and all information about the dependencies between the rows is lost. Preconditioning by the inverse of the midpoint matrix

$$Y \approx \begin{pmatrix} 0.1529 & 0.1529 \\ -0.1529 & 0.1529 \end{pmatrix}$$

then gives

$$\begin{aligned} Y\boldsymbol{F}'(\boldsymbol{x}) &\subset \begin{pmatrix} [0.4495, 1.5505] & [-0.5505, 0.5505] \\ [-0.5505, 0.5505] & [0.4495, 1.5505] \end{pmatrix}, \\ YF(\check{x}) &\approx \begin{pmatrix} -0.01529 \\ -0.01529 \end{pmatrix}. \end{aligned}$$

From this, the interval Gauss–Seidel method gives

$$\mathbf{N}(f, \boldsymbol{x}, \check{x}) \subseteq \begin{pmatrix} [0.6666, 1.4014] \\ [0.6666, 1.4014] \end{pmatrix} \not\subseteq \boldsymbol{x} = \begin{pmatrix} [0.7, 1.3] \\ [0.7, 1.3] \end{pmatrix},$$

despite the fact that there is a unique solution of $F(x) = 0$ within $\boldsymbol{x}$.

In contrast, assume that we can precondition the system in Example 1 *symbolically*, so that information about the tandem variation of the entries in a particular column of the Jacobi matrix F' is not lost. This preconditioning is carried out by applying the linear transformation to the coefficients in the function representation:

$$Y \begin{pmatrix} 3x_1^2 & -3x_2^2 \\ 3x_1^2 & 3x_2^2 \end{pmatrix} \approx \begin{pmatrix} 0.9174x_1^2 & 0 \\ 0 & 0.9174x_2^2 \end{pmatrix}.$$

Applying the interval Gauss–Seidel method to the symbolically preconditioned system gives

$$\mathbf{N}(f, \boldsymbol{x}, \check{x}) \approx \begin{pmatrix} 1 - \dfrac{-0.01529}{0.9174[0.7, 1.3]^2} \\ 1 - \dfrac{-0.01529}{0.9174[0.7, 1.3]^2} \end{pmatrix} \subseteq \begin{pmatrix} [1.0032, 1.0114] \\ [1.0032, 1.0114] \end{pmatrix} \subset \boldsymbol{x}.$$

Hansen perhaps first proposed the idea of symbolic preconditioning in (Hansen, 1997).

For symbolic preconditioning to be practical, a basis representation of the functions f_i should be chosen and manipulated automatically. This is the role of Taylor models and Taylor arithmetic.

1.3. Taylor Models

We will consider interval Taylor models for a function $f : \boldsymbol{x} \subseteq \mathbb{R}^n \to \mathbb{R}$ of the form

$$f(x) \in P_d(x - \check{x}) + \boldsymbol{I}_d, \tag{4}$$

where $P_d(x)$ is a degree-d polynomial in the n variables $x \in \mathbb{R}^n$, $\check{x}$ is a base point (often the midpoint of the interval vector $\boldsymbol{x}$), and $\boldsymbol{I}_d$ is an interval that encompasses the truncation error over the interval vector $\boldsymbol{x}$ and possible roundoff errors in computing the coefficients of

P_d. Although Taylor models have certain theoretical properties that are no better than simpler mean value extensions (see our comments in (Kearfott and Arazyan, 2000)), in many cases, especially when $\check{x}$ is the midpoint of $\boldsymbol{x}$, an interval evaluation

$$\{f(x) \mid x \in \boldsymbol{x}\} \subseteq P_d(\boldsymbol{x} - \check{x}) + \boldsymbol{I}_d \tag{5}$$

gives an orders-of-magnitude narrower interval enclosure for the range $\{f(x) \mid x \in \boldsymbol{x}\}$ than straightforward interval evaluation $\boldsymbol{f}(\boldsymbol{x})$ or than a mean value extension $f(\check{x})+\boldsymbol{\nabla f}(\boldsymbol{x})(\boldsymbol{x}-\check{x})$. (See (Kearfott and Arazyan, 2000; Makino and Berz, 1999).)

More than reduction of excess width in interval enclosures for the range of a single function, our interest in Taylor models lies in their use to implement symbolic preconditioning as we described below Example 1. In particular, a *Taylor arithmetic* can be defined on *Taylor objects* in such a way that evaluation of an expression for f in this arithmetic gives the Taylor polynomial with remainder term for f as in (5). For an introduction to these concepts and techniques, see (Berz and Hoffstätter, 1998).

Berz et al have a good Taylor model implementation in COSY-Infinity (Berz et al., 1996; Berz, 2000). Although the COSY-Infinity package has its own language designed especially for beam physics computations, Jens Hoefkens has recently developed a Fortran 90 module for general access to the COSY-Infinity Taylor arithmetic.

1.4. The Software Environment

The software environment within which we do the experiments below is GlobSol (Corliss, 1998; Corliss and Kearfott, 2000) combined with COSY-Infinity. Because of low-level implementation details, GlobSol, in many instances, executes several times more slowly than the most efficient possible software, for a given algorithm. Nonetheless, we have used GlobSol here due to its easy user interface and due to our familiarity with its structure (allowing us to make low-level modifications).

2. An Example Problem

The problem is an interesting variable-dimension problem originally examined by E. Hansen and G. W. Walster. GlobSol could complete with low-dimensional versions of this problem, but very inefficiently. The package Numerica (Van Hentenryck et al., 1997) also could not complete efficiently for any variants of this problem.

Problem 1. For $\nu \geq 4$, choose a_i, $2 \leq i \leq \nu - 1$ and x_j, $1 \leq j \leq \nu - 3$ such that

$$\begin{aligned}
1 + \sum_{i=2}^{\nu-1} a_i &= 0, \\
\nu + \sum_{i=2}^{\nu-1} i a_i &= 0 \\
\nu x_j^{\nu-1} + \sum_{i=1}^{\nu-1} i a_i x_j^{i-1} &= 0 \quad \text{for } 1 \leq j \leq \nu - 3, \\
\left(x_j^{\nu} + x_{j+1}^{\nu}\right) + \sum_{i=1}^{\nu-1} a_i \left(x_j^{i} + x_{j+1}^{i}\right) &= 0 \quad \text{for } 1 \leq j \leq \nu - 4,
\end{aligned}$$

where

$$0 < x_1, \quad x_j < x_{j+1} \text{ for } 1 \leq j \leq \nu - 4, \text{ and } \quad x_{\nu-3} < 1. \tag{6}$$

Thus, Problem 1 represents a system of $2\nu - 5$ equations in $2\nu - 5$ unknowns, with positive variables. The inequalities (6) prevent a combinatorial explosion of solutions due to symmetries. It is known that the problem, with the inequalities included, has a unique solution. The inequalities may be included either explicitly in the problem statement or by defining the search region appropriately.

3. Concerning the Experimental Environment

To provide a realistic test bed for the problem class (1), we modified GlobSol's main optimization routine `find_global_min`. In particular, for the problem class (1),

1. the interval Newton method used to reduce the size of subregions $\boldsymbol{x}$ is appropriate for systems of the form (2);

2. when $\boldsymbol{x}$ is split by bisecting a coordinate, the coordinate selection scheme is based on *maximal smear* for F as in (Kearfott, 1997), rather than the somewhat analogous form for $\nabla\phi$ as described in (Kearfott, 1996, (5.1)) or (Ratz and Csendes, 1995);

3. bound constraints are disabled;

4. the best-found upper bound on the objective is initialized to 0;

5. the algorithm to verify feasibility is changed appropriately.

Otherwise, the algorithm in `find_global_min` is as documented. A (possibly constant) objective should be supplied with minimum equal to zero at the solutions of $F(x) = 0$. If inequality constraints are supplied, they act as the "soft constraints" described in (Van Hentenryck et al., 1997).

Within the global optimization algorithm, the following evaluations could possibly benefit from Taylor evaluations:

1. evaluation of a the objective function $\boldsymbol{\phi}$;
2. evaluation of the gradient $\boldsymbol{\nabla\phi}$;
3. evaluation of the equality constraints $\boldsymbol{c}_i$;
4. evaluation of the gradients of the constraints $\boldsymbol{\nabla c}_i$;
5. use in symbolic preconditioning as explained in §1.2.

Implementation of Taylor arithmetic with optimal efficiency in this GlobSol would take prohibitively long for the exploratory experiments of this paper. Nonetheless, we implemented the capability to perform the experiments in a way that gives useful information. In particular, we provided separate routines for each of the above five tasks, utilizing Hoefkens' Fortran 90 COSY Infinity interface. The implementation is not efficient in the sense that the entire code list (see (Kearfott, 1995) or (Kearfott, 1996)), including computations for each component of each constraint and constraint gradient, is evaluated each time a particular constraint value or constraint gradient component value is required. Contact the authors for further details.

We have designed the experiments to detect the low-level efficiency of this implementation, to determine the benefits of the Taylor models in terms of total calls to the function, and to thus surmise advantages in more nearly optimal implementations.

4. The experiments

4.1. Timings for the Underlying Computations

To determine the ratio of a call to an interval routine versus a call to the corresponding Taylor model routine, we used Problem 1 with $n = 4$, and with objective ϕ equal to the sum of squares of the $2n - 5$ equation residuals. In separate runs, we evaluated $\boldsymbol{\phi}$, $\boldsymbol{\nabla\phi}$, $\boldsymbol{F}$ and $\boldsymbol{F}'$. We called these functions 1000 times within a loop. Table 4.1 represents the results on a 450MHz machine with DEC (Compaq) Fortran version

6.1, no optimization. For the interval arithmetic, we used the relatively slow INTLIB (Kearfott et al., 1994), distributed with GlobSol. (Faster, sharper packages, such as the intrinsic interval type in Sun's Fortran compiler, may give different results.) The times are elapsed times in

Table I. Timings for the underlying computations

Routine	plain interval	with Taylor
constraints	0.33	22.30
constraint gradients	0.33	22.69
objective	0.33	22.63
gradient	0.39	22.52

seconds. Taylor models of degree 5 were used. (Since the function components themselves are of degree 4, the Taylor models represent the functions exactly in point arithmetic.)

We see from Table 4.1 that an interval evaluation proceeds roughly 65 times faster than a Taylor model evaluation within our implementation. We also see that each routine, be it objective, gradient, constraints, or constraint gradient, completes in the same amount of time, to within the accuracy of the timing process. This is because, as indicated above, the entire code list is evaluated once for each quantity, and most of the work is in evaluation of the code list.

In a second test, we designed a non-trivial function for which interval evaluation and Taylor model evaluation will lead to the same number of boxes in the overall subdivision process. This function is

Problem 2.

$$\begin{aligned}
f_1(x) &= x_1^2 + \sin(x_2) - 1 = 0\\
f_2(x) &= x_3^2 + \sin(x_4) - 1 = 0\\
f_3(x) &= x_1 + x_4 = 0\\
f_4(x) &= x_2 + x_3 = 0
\end{aligned}$$

with initial box $\boldsymbol{x}^{(0)} = ([-2,2],[-2,2],[-2,2],[-2,2])^T$.

Problem 2 has four solutions within the initial box $\boldsymbol{x}^{(0)}$.

There is no overestimation in evaluation of the individual components f_i. Also, because each symbol x_i occurs in only one entry of the Jacobi matrix, preconditioning does not introduce any implicit overestimation. However, the Jacobi matrix is irreducible (i.e. the system is fully coupled), so an interval Newton method is necessary for efficiency within the overall search algorithm. The transcendental functions $\sin(x_2)$ and $\sin(x_4)$ slow down evaluation of this relatively simple function, to make timing comparisons more accurate and also ensure that the Taylor models are not trivial.

We used $\phi(x) = \sum_{i=1}^{4} f_i^2(x)$ as objective, and we used stopping tolerance `EPS_DOMAIN` $= 10^{-5}$. We ran the global search once without any Taylor model evaluation and once with Taylor model evaluation of the constraints, as well as use of Taylor models throughout the interval Newton method. In both cases, the global search considered exactly the same total number of boxes. The results appear in Table 4.1. Thus, for

Table II. Timings for the underlying computations

	plain interval	with Taylor
total # boxes	14	14
total elapsed time	0.33	1.64

Problem 2, the time penalty within the GlobSol environment for using degree 5 Taylor models when there are no benefits to do so appears to be approximately a factor of 5.

4.2. Results for Problem 1

In our first experiment, we solved Problem 1 with $\nu = 5$ (so the dimension n of the system is also 5). In this experiment, we took an initial box $\boldsymbol{x}_0$ in which (6) is automatically satisfied:

$$\boldsymbol{x}^{(0)} = (\boldsymbol{a}_2^{(0)}, \boldsymbol{a}_3^{(0)}, \boldsymbol{a}_4^{(0)}, \boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)})^T,$$

$$\text{where } \boldsymbol{x}_i^{(0)} = \left[\frac{i-1}{\nu-1} + 10^{-2}, \frac{i}{\nu-1} - 10^{-2}\right]$$
$$\text{and } \boldsymbol{a_i^{(0))}} = [-5, 5], \quad i = 2, 3, 4.$$

We ran the global search

1. without any Taylor models;
2. with Taylor models for the constraints only;
3. with Taylor models and symbolic preconditioning within the interval Newton method only;
4. with Taylor models both for the constraints and within the interval Newton method.

In all cases, we set the limits on CPU time, maximum number of boxes considered, etc. so the global search would complete successfully. The results appear in Table 4.2.

Table III. Performance for Problem 1 with $n = 5$.

	Total # boxes	Total elapsed time
interval only	121,151	1,762
Taylor constraints only	8,422	1,259
Taylor interval Newton only	3,562	154
Taylor interval Newton and constraints	2,961	395

Table 4.2 indicates that, for Problem 1, use of symbolic preconditioning in the interval Newton method is important. Taylor models for the individual components, although also useful, appear to be less important than the symbolic preconditioning. (We note that, when symbolic preconditioning is used in the interval Newton method, the constraint residuals are calculated with Taylor models and checked before the actual preconditioning is done. Thus, use of Taylor models on the individual constraints and in the symbolic preconditioning is not totally separated in this experiment.)

Unfortunately, we were unable to obtain results for larger values of n. (For example, with $\nu = 6$, corresponding to $n = 7$, using Taylor interval Newton only, the global search could not complete with less than 200,000 boxes considered.)

5. Is there a Good Heuristic for Use of Taylor Models?

Taylor models are beneficial for the individual constraints when there is significant interval dependency (and overestimation) in the individual constraints. Similarly, Taylor preconditioning is beneficial when there is significant overestimation in the preconditioned constraints. Three possibilities come to mind for detecting such overestimation, in either case.

1. Compute an ordinary interval inclusion, then actually compute the Taylor inclusion, and compare the widths.

2. Sample the function at a number of random points within the box, and compare the range to the range given by an ordinary interval inclusion.

3. Use inner bounds.

Actually computing the Taylor inclusion does not save any computational effort. However, this can be done once or twice at the beginning of the global search algorithm (with relatively large boxes), and we may then heuristically assume that all subsequent boxes have the same degree of overestimation. The main failings of this possibility would be different behavior in different subregions and less overestimation in the smaller sub-boxes produced as subdivision progresses.

Berz and Makino use random sampling to heuristically estimate the overestimation in ordinary interval inclusions and in Taylor models in COSY Infinity. There is clearly the possibility of coming to an incorrect conclusion through this process, although it seems to work fairly well in practice.

A final possibility is to compute rigorous inner estimations with, say, twin arithmetic as described in (Nesterov, 1997); Muñoz is developing a package for twin arithmetic, and has developed formulas for inner estimations for a number of functions and their slopes (Muñoz, 2001). Twin arithmetic gives rigorous inner estimations, and, although more costly than computation of ordinary interval inclusions, is much less costly than Taylor arithmetic. However, there are instances in which the inner estimations that twin arithmetic gives are much narrower than the actual range; see (Hertling, 2001).

6. Conclusions

The goal was to design and carry out initial experiments to demonstrate or refute the potential value of symbolic preconditioning and other uses of Taylor arithmetic in rigorous branch and bound algorithms for constrained global optimization. There was a factor of around 20 time penalty in individual objective or constraint evaluations and a factor of about 5 time penalty in the overall global optimization algorithm, when the total number of boxes in the global optimization algorithm remained constant. However, in problems where coupling between the equations leads to overestimation due to applying the preconditioner, symbolic preconditioning with the Taylor model can lead to an order-of-magnitude reduction in the total amount of work, and can make it practical to solve problems that were previously impractical to solve.

Nonetheless, straightforward application of Taylor models will not make it practical to solve all such problems.

Considering the experiments in this paper and previous experiments (such as in (Kearfott and Arazyan, 2000), (Makino and Berz, 1999), or (Makino and Berz, 2000)), we can give the following advice:

- Taylor models are not universally applicable, but, when used in specific ways for specific problems, can make certain unsolvable problems solvable.

- If one recognizes interval dependency among individual constraints, especially if the dependency occurs in a transcendental way, then Taylor models for these constraints can be useful.

- If one recognizes that preconditioning will introduce interval dependency due to coupling in a system of differential equations, then symbolic preconditioning with Taylor models will probably be useful.

- An order of magnitude or more increase in speed may be possible if Taylor arithmetic is incorporated at a low level in the global optimization algorithm. (Whether or not to do this would depend on the importance of the problem or problems being solved.)

References

Berz, M.: 2000, 'COSY INFINITY Web page'. `http://bt.nscl.msu.edu/cosy/`.

Berz, M. and G. Hoffstätter: 1998, 'Computation and Application of Taylor Polynomials with Interval Remainder Bounds'. *Reliable Computing* **4**(1), 83–97.

Berz, M., K. Makino, K. Shameiddine, G. H. Hoffstätter, and W. Wan: 1996, 'COSY INFINITY and Its Applications in Nonlinear Dynamics'. In: *Computational Differentiation, Techniques, Applications, and Tools.* Philadelphia, pp. 363–365, SIAM.

Corliss, G. F.: 1998, 'GlobSol Entry Page'. `http://www.mscs.mu.edu/~globsol/`.

Corliss, G. F. and R. B. Kearfott: 2000, 'Rigorous Global Search: Industrial Applications'. In: *Developments in Reliable Computing.* Dordrecht, Netherlands, pp. 1–16, Kluwer.

Hansen, E. R.: 1997, 'Preconditioning Linearized Equations'. *Computing* **58**, 187–196.

Hertling, P.: 2001, 'A Limitation for Underestimation Via Twin Arithmetic'. *Reliable Computing* **7**(2), 157–169.

Kearfott, R. B.: 1995, 'A Fortran 90 Environment for Research and Prototyping of Enclosure Algorithms for Nonlinear Equations and Global Optimization'. *ACM Trans. Math. Software* **21**(1), 63–78.

Kearfott, R. B.: 1996, *Rigorous Global Search: Continuous Problems.* Dordrecht, Netherlands: Kluwer.

Kearfott, R. B.: 1997, 'Empirical Evaluation of Innovations in Interval Branch and Bound Algorithms for Nonlinear Algebraic Systems'. *SIAM J. Sci. Comput.* **18**(2), 574–594.

Kearfott, R. B. and A. Arazyan: 2000, 'Taylor Series Models in Deterministic Global Optimization'. In: *Proceedings of Automatic Differentiation 2000: From Simulation to Optimization*. New York, Springer-Verlag.

Kearfott, R. B., M. Dawande, K.-S. Du, and C.-Y. Hu: 1994, 'Algorithm 737: INTLIB, A Portable FORTRAN 77 Interval Standard Function Library'. *ACM Trans. Math. Software* **20**(4), 447–459.

Makino, K. and M. Berz: 1999, 'Efficient Control of the Dependency Problem Based on Taylor Model Methods'. *Reliable Computing* **5**(1), 3–12.

Makino, K. and M. Berz: 2000, 'New Applications of Taylor Model Methods'. In: *Proceedings of Automatic Differentiation 2000: From Simulation to Optimization*. New York, Springer-Verlag.

Muñoz, H.: 2001, 'Theory and Practice in Verified Nonsmooth Optimization'. Ph.D. thesis, University of Louisiana at Lafayette.

Nesterov, V. M.: 1997, 'Interval and Twin Arithmetics'. *Reliable Computing* **3**(4), 369–380.

Neumaier, A.: 1990, *Interval Methods for Systems of Equations*. Cambridge, England: Cambridge University Press.

Pardalos, P. M. and J. B. Rosen: 1987, *Constrained Global Optimization: Algorithms and Applications*, Lecture Notes in Computer Science no. 268. New York: Springer-Verlag.

Ratschek, H. and J. Rokne: 1988, *New Computer Methods for Global Optimization*. New York: Wiley.

Ratz, D. and T. Csendes: 1995, 'On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization'. *J. Global Optim.* **7**, 183–207.

Van Hentenryck, P., L. Michel, and Y. Deville: 1997, *Numerica: A Modeling Language for Global Optimization*. Cambridge, MA: MIT Press.