Interval Computations
No 3, 1993

Methodologies for Tolerance Intervals

Birna P. Kristinsdottir*, Zelda B. Zabinsky*, Tibor Csendes†,
and Mark E. Tuttle*

Algorithms for finding large feasible n-dimensional intervals for constrained nonlinear optimization are presented. The n-dimensional interval is iteratively enlarged about a seed point while a checking routine maintains feasibility. Two checking routines are discussed: an interval subdivision method and a global optimization method. Both checking routines can be used in the overall methodology to generate a feasible suboptimal interval. Such an interval is useful when examining manufacturing tolerances in design optimization. Numerical results are presented for a practical application in the optimal design of a flat composite plate and a composite stiffened panel structure.

Методология нахождения интервалов допуска

Б. П. Кристиндоттир, З. Б. Забински,
Т. Чендеш, М. Э. Таттл

Предлагается алгоритм нахождения больших n-мерных интервалов допуска для нелинейной оптимизации с ограничениями. Рассматриваемый n-мерный интервал итерационно увеличивается около начальной точки, в то время как механизм проверки отслеживает допустимость процесса. Рассматриваются две проверочные процедуры: метод деления интервала и метод глобальной оптимизации. Обе проверочные процедуры могут применяться в рамках общей методологии для порождения приемлемого подоптимального интервала. Такой интервал полезен при исследовании производственных допусков в оптимизации проектирования. Представлены численные результаты практического применения в оптимальном проектировании плоской композитной пластины и композитной укрепленной панельной конструкции.

---

*The work of these authors has been supported in part by a National Science Foundation grant DDM-9211001, and by the Boeing Commercial Airplane Company and the NASA-Langley Research Center.

†The work of this author has been supported by the Grants OTKA 2879/1991 and OTKA 2675/1991, and by the Special Program East-Europe of the Swiss Federal Commission for Scholarships. On leave at the Institute of Informatics at Basel University, Switzerland.

1 Introduction

1.1 Manufacturing tolerance problem

This paper presents methods to incorporate manufacturing tolerances in an engineering design problem when the original design problem is in the form of a constrained optimization problem.

Consider the nonlinear (possibly global) optimization problem (P)

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \le 0 \qquad j = 1, 2, \ldots, m, \end{array}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}$ and the constraint functions, $g_j(x) : \mathbb{R}^n \to \mathbb{R}$ are continuous functions, and n is the dimension of the problem. Let us denote the set of feasible points by A, that is $A := \{x \in \mathbb{R}^n : g_j(x) \le 0$ for each $j = 1, 2, \ldots, m\}$. Also let x^* be an optimal solution for problem (P).

An example of a global optimization engineering design problem can be found in the optimal design of composite structures [8], [9] and [12]. The optimal solution, x^*, representing the optimal design, may be on an active constraint. Since manufacturing processes are not able to reproduce the optimal solution x^* exactly, if the point x^* were actually produced, it could fail. In order for x^* to be a practical design, we need to find a *feasible* interval, where $x \in A$ for all $x \in [x_i^* - \delta, x_i^* + \delta]$ for $i = 1, 2, \ldots, n$ and $\delta > 0$ is a specific manufacturing tolerance for each variable. It is also desirable that this interval be in the neighborhood of the global optimum (within ϵ of $f(x^*)$).

We restate our problem: find an n-dimensional interval X^* such that, for all $x \in X^*$,

$$f(x) \le f_\epsilon \equiv f(x^*) + \epsilon \qquad \text{and} \tag{1}$$

$$g_j(x) \le 0 \qquad \text{for } j = 1, 2, \ldots, m. \tag{2}$$

A methodology to solve the stated problem does not previously exist, but methods discussed in [3], [4] and [5], study related problems. Most of the literature within the optimization community that exists on sensitivity analysis is on the sensitivity of the optimum when some of the parameters of the problem are varied and/or the right hand side of the constraints are perturbed. This is often called parametric programming (e.g. see [7]). In [2] the Augmented Lagrangian function is used to derive sensitivity results in parametric programming. This approach is not applicable to our restated problem. In [10] a procedure that incorporates manufacturing tolerances as a part of engineering design optimization is introduced. The approach includes both parametric variations and manufacturing tolerances using a Taylor series approximation in the neigborhood of the optimum. This method does not check feasibility in the neighborhood around the point, so we do not have information on whether the interval is feasible. It also assumes that the tolerances are small enough for the Taylor approximation to be valid. We propose a different approach to incorporate manufacturing tolerances into engineering design optimization.

1.2 Two questions and approaches

Our approach to the manufacturing tolerance problem suggests two basic questions and algorithms. For both we assume we are given a design that is interior to the feasible region, called x^{seed}. This point may come from solving an earlier optimization problem. Formally we assume that x^{seed} satisfies the following conditions:

$$f(x^{\text{seed}}) < f_\epsilon \quad \text{and} \tag{3}$$
$$g_j(x^{\text{seed}}) < 0 \quad \text{for each } j = 1, 2, \ldots, m. \tag{4}$$

The first question is: given an interior point x^{seed}, and required tolerances of $\pm\delta$, does the tolerance interval of $[x_i^{\text{seed}} - \delta,\ x_i^{\text{seed}} + \delta]$, $i = 1, 2, \ldots, n$, lie entirely in the feasible region? We answer this question by directly checking the feasibility of the tolerance interval using one of two checking routines presented in Section 2.1.

The second question is: given an interior point x^{seed}, what are the largest tolerances around x^{seed}? We answer this question by iteratively growing a maximal feasible rectangle about x^{seed} using the main algorithm, presented in Section 3.1.

1.3 Composite structural design problem

Laminated composites are made from a stack of several plies, which are bonded together to form a composites laminate. A ply is a thin layer, made from long reinforced fibers, (e.g. graphite fibers), embedded within a weaker matrix material (e.g. epoxy). Within an individual ply, all fibers are oriented in the same direction. Composite laminates are usually fabricated such that fiber angles vary from ply to ply. Previous research, reported in [8], [9] and [12] has developed optimization software to aid in the design of composite aircraft structures. The program finds the optimal ply orientations and stiffener geometries, given material properties and loading conditions for a flat composite plate or stiffened panel structure. The software combines classical lamination theory with a random search global optimization algorithm, called Improving Hit-and-Run, see [12] and [13].

The composites optimization problem is a global optimization problem and can be formulated as (P). The objective function $f(x)$ can be the weight of the structure, the cost of the structure or a combination of cost and weight. The inequality constraints $g_j(x) \leq 0$ represent mechanical constraints such as strain and strength of the structure. The design variables x are the ply orientation angles in degrees and, in the case of a stiffened panel, also include stiffener geometry variables, such as stiffener spacing and stiffener height. A common size of the optimization problem is 25 dimensions.

When composite materials are designed, it is critical to account for possible variations in fiber angles during the design phase. During the manufacturing process the fiber angles may vary substantially from their optimal value, $\pm 2^{\circ}$ is not uncommon. This is large enough to make a Taylor series approximation [10] inexact. The composite structural design problem has motivated the research of our methodologies.

2 Checking a tolerance interval

2.1 Checking routines

The first question mentioned assumes that an interior point x^{seed}, and a tolerance interval $X = [x_i^{\text{seed}} - \delta,\ x_i^{\text{seed}} + \delta]$ for $i = 1, 2, \ldots, n$ and a $\delta > 0$ are specified. The problem is to check whether the tolerance interval is feasible and close to the optimum. That is, check whether the interval is contained in A and within ϵ of $f(x^*)$, i.e. for every $x \in X$ equations (1) and (2) are satisfied. In this paper, two checking routines are presented. The first checking routine uses interval arithmetic to check feasibility, as presented in [6]. The second checking routine uses a global optimization algorithm, IHR (Improving Hit-and-Run) [13] to check feasibility of the tolerance interval.

2.1.1 Interval checking routine

The interval checking routine (see [6]) is a version of the interval subdivision method modified to check whether an interval X is strongly feasible. The interval X is said to be *strongly feasible* if for every $x \in X$, $f(x) < f_\epsilon$ and $g_j(x) < 0$ for all $j = 1, 2, \ldots, m$. Suppose $f(x) : \mathbb{R}^n \to \mathbb{R}$ is a continuous function, and $\mathbb{I}^n$ is a set of n-dimensional compact real intervals. Then the function $F(X) : \mathbb{I}^n \to \mathbb{I}$ is called an *inclusion function* of $f(x)$, if for every $X \in \mathbb{I}^n$, $F(X) \supseteq \overline{f}(X)$, where $\overline{f}(X) = \{f(x) : x \in X\}$ is called the range of $f(x)$ over the n-dimensional interval X. The *width* of an n-dimensional interval, $X \in \mathbb{I}^n$, is defined as the maximum length of the edge of the interval, $w(X) = \max\{w(X_i) : i = 1, \ldots, n\}$ where X_i is the i-th coordinate interval of X. It is assumed that all inclusion functions are *isotone*, that is, for inclusion function $F : \mathbb{I}^n \to \mathbb{I}$, we have $X \subseteq Y$ implies $F(X) \subseteq F(Y)$ for all $X, Y \in \mathbb{I}^n$. For more information on interval arithmetic see [1] and [11]. It is assumed that there exist inclusion functions for the objective function and constraints.

The checking routine also requires a stopping criterion parameter γ, which is set to a small positive real value. The procedure to check whether an interval X is strongly feasible is as follows:

Interval checking routine

Step 0. Initialize a list L to be empty, and set $Y = X$.

Step 1. If the width of Y is less than γ, then go to **Step 7**.

Step 2. Evaluate the inclusion functions $F(Y)$ and $G_j(Y)$ for each $j = 1, 2, \ldots, m$.

Step 3. If $\max F(Y) \geq f_\epsilon$ or $\max G_j(Y) \geq 0$ for any $j = 1, 2, \ldots, m$, then go to **Step 5**.

Step 4. If the list L is empty, then go to **Step 6**, else put the last item of the list L into Y, delete this item from the list, and go to **Step 1**.

Step 5. Subdivide Y into subintervals U and V, set $Y = U$, put V into the list L as the last member, and go to **Step 1**. The subdivision should be made, such that the largest side of Y is halved.

Step 6. RETURN that the checked interval is strongly feasible.

Step 7. RETURN that the checked interval is not strongly feasible, and $Z = Y$.

If the checking routine indicates that the initial interval X is not strongly feasible, then it returns a very small subinterval Z that is not strongly feasible. By properly setting γ, Z indicates the place where one of the constraints or $f(x) < f_\epsilon$ is violated.

2.1.2 Global optimization checking routine

The second checking routine checks the feasibility of the tolerance interval by finding the worst point in the interval using a random search global optimization algorithm, Improving Hit-and-Run [13]. The interval X to be checked can be written as upper and lower bounds, $\min X_i \le x_i \le \max X_i$, for $i = 1, 2, \ldots, n$. The checking procedure is to find the largest deviation of the constraints and $f(x) < f_\epsilon$.

The checking procedure is as follows:

$$\begin{aligned} &\text{maximize} \quad h(x) = \max\{f(x) - f_\epsilon,\, g_1(x),\, g_2(x),\, \ldots,\, g_m(x)\} && (5)\\ &\text{subject to} \quad \min X_i \le x_i \le \max X_i \qquad \text{for } i = 1, 2, \ldots, n. && (6)\end{aligned}$$

The global optimization algorithm finds an approximation of the global optimum over the tolerance interval X. If the optimal value h^* is less than zero, the interval X is strongly feasible. If the optimal value h^* is greater than or equal to zero, then the interval X is not strongly feasible. Also, the algorithm provides a point $z = x^*$ where the constraints or $f(x) < f_\epsilon$ are violated.

The two checking routines are just about interchangeable. The interval checking routine has the advantage that a strongly feasible interval, containing the global optimum can be recognized in a finite number of steps, see Theorem 1 in [6]. The global optimization checking routine will converge with probability one to the global optimum. It is possible for the global optimization checking routine to stop prematurely and accept an interval as feasible when in fact it isn't. It is also possible for the interval checking routine to reject a feasible interval as infeasible if γ is not small enough. We will show in Section 2.2 that the global optimization routine has computational advantages. In practical use there is a tradeoff between absolute guarantee and computational efficiency.

2.2 Checking tolerance intervals in a composite laminates

In this section the two checking routines will be applied to check a tolerance interval around a given seed point. As a test problem, a 4 ply composite flat plate laminate is tested, subject to strain constraints. Analytic expression of the strain constraints can be found in [12]. To find the inclusion functions for the interval checking routine, the natural interval extension with outside rounding is used ([1, 11]). For our specific functions, one interval function evaluation costs approximately two times more computing time than a real function evaluation. The loading conditions used are: $N_x = 2000\,\text{lbs/in}$, $N_y = 1000\,\text{lbs/in}$ and $N_y = -500\,\text{lbs/in}$.

The seed point used is: $\theta = (-45^o, 14^o, 14^o, -45^o)$. An interval of the form $X = [x_i^{\text{seed}} - \delta,\, x_i^{\text{seed}} + \delta]$ for $i = 1,\, 2,\, \ldots,\, n$ and δ ranging from 0.5^o to 3.0^o is checked for feasibility. In the interval checking routine we used $\gamma = 0.01$. The same intervals were checked using IHR by checking each box 5 times, using a fixed number of function evaluations of 5000. For the case where IHR detected an interval as infeasible, the average number of iterations needed to detect infeasibility are reported. The results are shown in Table 1.

The interval checking routine gives conservative but reliable information about the feasibility of the tolerance intervals. That is, the interval checking routine may indicate an interval is infeasible when it really is feasible, but it will not indicate an interval is feasible when it is actually infeasible. In fact, the intervals given in Table 2 are all feasible intervals if they are checked with more precision. The interval checking routine made the error, because $\gamma = 0.01$ was not small enough to get a sharp correct indication. Unfortunately, using a smaller γ greatly increases computation, and was not practical for this problem. On the other hand, IHR may not always find the true global

	Interval checking routine		IHR	
δ	NFE	Feasible	NFE	Feasible
0.5º	60,980	Yes	25,000	Yes
1.0º	1,657,265	Yes	25,000	Yes
1.5º	18,351,740	Yes	25,000	Yes
2.0º *	2,417,171	No	25,000	Yes
2.5º	8,877,255	No	61	No
3.0º	10,304,657	No	28	No

* The 2.0 degree interval is actually feasible

Table 1: Checking interval $[x^{\text{seed}} \pm \delta]$ with both routines

δ	2.0º	2.5º	3.0º
$\underline{\theta}_1, \overline{\theta}_1$	-47.0000000,-46.9921875	-47.5000000,-47.4902344	-48.0000000,-47.9941406
$\underline{\theta}_2, \overline{\theta}_2$	12.0000000, 12.0078125	11.5000000, 11.5097656	11.0000000, 11.0058594
$\underline{\theta}_3, \overline{\theta}_3$	12.0000000, 12.0078125	11.5000000, 11.5097656	11.0000000, 11.0058594
$\underline{\theta}_4, \overline{\theta}_4$	-43.1406250,-43.1328125	-43.5546875,-43.5449219	-43.6757812,-43.6699219

Table 2: Intervals returned as infeasible by the interval checking routine

optimum and it may indicate that a tolerance interval is feasible, when it really is infeasible. But if the IHR checking routine indicates an interval is infeasible, it is always correct. As can be seen in Table 1, the interval checking routine requires a lot more computation than the IHR checking routine.

To use both routines to their best advantage, we are investigating a hybrid method that alternates between both routines. This hybrid method would stop if the interval portion indicates feasibility, or if the IHR portion indicates infeasibility. It is a promising way to reduce computation and maintain a guarantee of a correct conclusion.

3 Growing a tolerance interval

3.1 Main algorithm

The second question mentioned will be addressed using the Main Algorithm introduced in [6]. The algorithm requires an interior point x^{seed} satisfying conditions (3) and (4), and iteratively grows a strongly feasible interval around x^{seed}. This is an approximation of a maximal feasible tolerance interval containing x^{seed}. The algorithm uses a stopping criterion parameter η and step sizes $d(i,1)$ and $d(i,2)$ for $i = 1, 2, \ldots, n$, which are set at the beginning to positive reals. To start, $d(i,1)$ and $d(i,2)$ must be larger than η. The stopping criterion indicates that the algorithm should stop increasing the size of the actual box X, when the change along each coordinate is less than the threshold η in all directions.

Main algorithm

Step 0. Initialize interval vector $X_i = [x_i^{\text{seed}}, x_i^{\text{seed}}]$, and $d(i,j) \geq \eta > 0$ for all $i = 1, 2, \ldots, n$ and $j = 1, 2$.

Step 1. For $i = 1$ to n do:

Step 2. Set $Y_j = X_j$ for $j = 1, 2, \ldots, n$; $j \neq i$ and

$$Y_i = \Big[\min(X_i) - d(i,1),\ \min(X_i)\Big].$$

Step 3. Use the checking routine to check whether $f(y) < f_\epsilon$ and $g_j(y) < 0$ $(j = 1, 2, \ldots, m)$ for each $y \in Y$. If the answer is yes, then set $X = X \cup Y$. Otherwise $d(i,1) = (\min(X_i) - \max(Z_i))/2$, where Z is the interval passed back by the checking routine as not strongly feasible.

Step 4. Set $Y_j = X_j$ for $j = 1, 2, \ldots, n$; $j \neq i$ and

$$Y_i = \Big[\max(X_i),\ \max(X_i) + d(i,2)\Big].$$

Step 5. Use the checking routine to check whether $f(y) < f_\epsilon$ and $g_j(y) < 0$ $(j = 1, 2, \ldots, m)$ for each $y \in Y$. If the answer is yes, then set

$X = X \cup Y$. Otherwise $d(i,2) = (\min(Z_i) - \max(X_i))/2$, where Z is the interval passed back by the checking routine as not strongly feasible.

Step 6. End of i-loop

Step 7. Stopping criterion: if the number of inclusion function calls is less than 1,000,000, and there is an $i = 1, 2, \ldots, n$ such that either $d(i,1) \geq \eta$ or $d(i,2) \geq \eta$ then go to **Step 1**.

Step 8. Print X, and STOP.

3.2 Convergence results

Consider a fixed constrained nonlinear optimization problem as given in Section 1.1. Denote the result box calculated with the algorithm parameters γ and η by $X^*_{\gamma,\eta}$, and the level set belonging to the function value f_ϵ by S_{f_ϵ}. Further assume that

$$w\big(F(X)\big) \to 0 \quad \text{as } w(X) \to 0, \qquad \text{and} \tag{7}$$

$$w\big(G_j(X)\big) \to 0 \quad \text{as } w(X) \to 0 \tag{8}$$

for all $j = 1, 2, \ldots, m$.

The following theorems characterize the convergence properties of our algorithm with the interval checking routine:

Theorem 1. *Assume the set $S_{f_\epsilon} \cap A$ is bounded, the seed point x^{seed} fulfills the conditions (3) and (4), $d(i,j) > 0$, and the properties (7) and (8) hold for the inclusion functions $F(X)$ and $G(X)$. Then there exist threshold values $\gamma^T > 0$ and $\eta^T > 0$ such that for all γ: $0 < \gamma < \gamma^T$ and η: $0 < \eta < \eta^T$*
1) the algorithm stops after a finite number of steps,
2) the result interval $X^*_{\gamma,\eta}$ has a positive measure, and
3) the result interval $X^*_{\gamma,\eta}$ is strongly feasible: $X^*_{\gamma,\eta} \subset S_{f_\epsilon} \cap A$.

See [6] for proof. Using Theorem 1, similar results can be obtained when the global optimization checking routine is used by making certain assumptions. The results are summarized in the following corollary.

Corollary 1. *Assume the set $S_{f_\epsilon} \cap A$ is bounded, the seed point x^{seed} fulfills the conditions (3) and (4), $d(i,j) > 0$ and the optimization routine correctly*

returns whether an interval is strongly feasible or not. Then there exists an $\eta^T > 0$ *such that for all* η*:* $0 < \eta < \eta^T$
1) the algorithm stops after a finite number of checking routine calls,
2) the result interval X^*_η *has a positive measure, and*
3) the result interval X^*_η *is strongly feasible:* $X^*_\eta \subset S_{f_\epsilon} \cap A$.

Theorem 2 describes the limit of the result boxes when the algorithm parameters γ and η are equal and converge together to zero.

Theorem 2. *If the conditions of Theorem 1 are fulfilled, then the limiting interval* $X^* = \lim_{\gamma\to 0} X^*_{\gamma,\gamma}$ *exists, and* X^* *is maximal in the sense that for every box* X' *the relations* $X^* \subseteq X'$ *and* $X' \subseteq S_{f_\epsilon} \cap A$ *imply* $X' = X^*$.

See [6] for proof. The algorithm does not find a maximal volume box around the seed point in a finite number of steps. In our application to manufacturing tolerances this is not a disadvantage since we often want to be able to control the shape of the resulting maximal box.

3.3 Stiffened panel design

In this test a more realistic problem will be treated using the Main Algorithm to grow a maximal tolerance interval using the global optimization checking routine. The test problem used here is a stiffened panel design example from [8]. The material used is AS4/3501-6, graphite/epoxy and the structure is subjected to the same multiple loading condition as in [8]. The seed point used is obtained from a previous optimization by minimizing weight subject to strain and strength constraints using a margins of safety equal to 0.3. The seed point is the following:

$$\theta_{\text{skin}} = (-38^o, 35^o, -30^o, 49^o, -82^o, -82^o, 49^o, -30^o, 35^o, -38^o)$$

$$\theta_{\text{stiffener}} = (18^o, 18^o, -18^o, 62^o, -18^o, -62^o, -62^o, -18^o, 62^o, -18^o, 18^o, 18^o)$$

$$(\text{STS, WSTF, HSTF, WSTC, ASTW}) = (18.69,\ 1.00,\ 2.00,\ 1.99,\ 89.97^o)$$

θ_{skin} and $\theta_{\text{stiffener}}$ are fiber angles in degrees. Dimensions associated with the stiffener geometry are shown in Figure 1. STS stands for stiffener spacing in inches, WSTF for width of stiffener flange in inches, HSTW for height of stiffener web in inches, WSTC stands for width of stiffener cap in inches and ASTW for the angle of stiffener web in degrees.

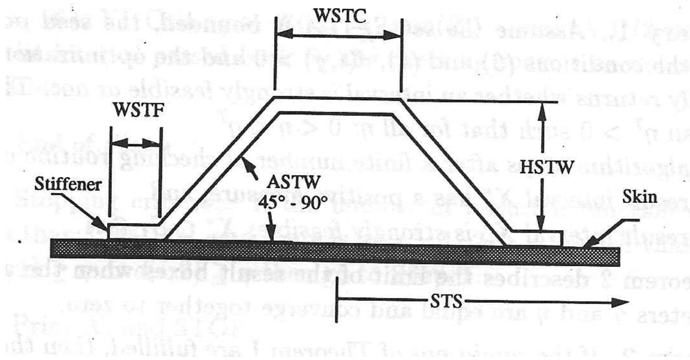

Figure 1: Hat stiffener geometry

Tolerance intervals are grown for the fiber angles in the skin and the stiffener. The weight of the structure only depends on the number of plies and geometry variables but not on fiber angles. Therefore we are not concerned with changes in weight, only concerned with the feasibility of the structure. The tolerances on the geometry variables will be assumed to be given and fixed, and they are ± 1.00 degree on the stiffener web angle and ± 0.05 inches on all the other geometry variables.

The resulting tolerances are shown in Figure 2, which demonstrates how the fiber angles can vary from their optimum value without making the structure infeasible. Notice that the tolerances are different for different variables. For instance the optimal value of the first fiber angle in the skin (-38^o) can range between -43^o and -33^o and therefore has a tolerance of 4^o in one direction and 5^o in the other direction. The optimal value of the third fiber angle in the skin (-30^o) can vary from -33^o to -27^o and therefore has a tolerance of approximately $\pm 3^o$. This is valuable tolerance information in practice because there are manufacturing situations where tolerances can be of different sizes for different variables, that is tolerances may be asymmetric. The initial setting of $d(i,j)$ in the Main Algorithm can be manipulated to influence the resultant tolerance interval.

4 Conclusions

We have shown that growing an interval using the Main Algorithm gives valuable tolerance information for a constrained optimization problem. The practical information can be used to simply check whether a tolerance interval is feasible, or to gain information about possibly asymmetric tolerances.

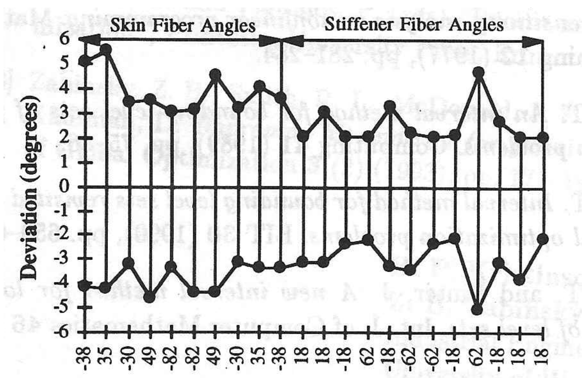

Figure 2: Tolerances on fiber angles in skin and stiffener

Two checking routines are discussed and compared. The interval checking routine has the advantage of giving a reliable indication of feasibility, while the IHR checking routine gives a reliable indication of infeasibility. Also, IHR appears to have computational advantages in detecting infeasibility. We hope to use both routines to their best advantage by constructing a hybrid checking routine that combines the reliability of the interval checking routine with the efficiency of the IHR checking routine. This would provide a truly practical scheme for evaluating tolerance intervals.

References

[1] Alefeld, G. and Herzberger, J. *Introduction to interval computations.* Academic Press, New York, 1983.

[2] Buys, J. D. and Gonin, F. *The use of augmented Lagrangian functions for sensitivity analysis in nonlinear programming.* Mathematical Programming **12** (1977), pp. 281–284.

[3] Csendes, T. *An interval method for bounding level sets of parameter estimation problems.* Computing **41** (1989), pp. 75–86.

[4] Csendes, T. *Interval method for bounding level sets revisited and tested with global optimization problems.* BIT **30** (1990), pp. 650–657.

[5] Csendes, T. and Pintér, J. *A new interval method for locating the boundary of level sets.* Int. J. of Computer Mathematics **46** (1993) (in printing).

[6] Csendes, T., Zabinsky, Z. B., and Kristinsdottir, B. P. *Constructing large feasible suboptimal intervals for constrained nonlinear optimization* (submitted for publication).

[7] Fiacco, A. V. *Sensitivity analysis for nonlinear programming using penalty methods.* Mathematical Programming **10** (1976), pp. 287–311.

[8] Graesser, D. L., Zabinsky, Z. B., Tuttle, M. E., and Kim, G. I. *Designing laminated composites using random search techniques.* Composite Structures **18** (1991), pp. 311–325.

[9] Graesser, D. L., Zabinsky, Z. B., Tuttle, M. E., and Kim, G. I. *Optimal design of composite structures.* Composite Structures **24** (1993), pp. 273–281.

[10] Parkinson, A., Sorensen, C., Free, J., and Canfield, B. *Tolerances and robustness in engineering design optimization.* Advances in Design Automation — 1990, Vol. 1, ASME Publication No. DE-Vol. 13–1. Proceedings of ASME Design Automation Conference, Chicago, IL, Sept. 16–19, 1990.

[11] Ratschek, H. and Rokne, J. *Computer methods for the range of functions.* Ellis Horwood, Chichester, 1984.

[12] Zabinsky, Z. B., Graesser, D. L., Tuttle, M. E., and Kim, G. I. *Global optimization of composite laminates using improving hit-and-run.* In: Floudas, C. and Pardalos, P. (eds) "Recent advances in global optimization", Princeton University Press, Princeton, 1992, pp. 343–368.

[13] Zabinsky, Z. B., Smith, R. L., McDonald, J. F., Romeijn, H. F., and Kaufman, D. *Improving hit-and-run for global optimization.* Journal of Global Optimization **3** (2) (1993), pp. 171–192.

B. P. Kristinsdottir,
Z. B. Zabinsky
Industrial Engineering Program
University of Washington,
Seattle,
USA

T. Csendes
Kalmár Laboratory
József Attila University,
Szeged,
Hungary

M. E. Tuttle
Mechanical Engineering Department
University of Washington,
Seattle,
USA