INTERVAL ANALYSIS FOR GUARANTEED NONLINEAR PARAMETER ESTIMATION

Eric Walter and Michel Kieffer

Laboratoire des Signaux et Systèmes
CNRS – Supélec – Université Paris-Sud
Plateau de Moulon, F-91192 Gif-sur-Yvette, France
{walter, kieffer}@lss.supelec.fr

Abstract: Interval analysis was initially developed to analyze and control numerical errors in computers. It can now be used to solve problems that are at the core of nonlinear parameter estimation, such as the minimization of possibly nonconvex cost functions or the characterization of sets defined by nonlinear inequalities. The solutions obtained are approximate but guaranteed, in the sense that no solution can be lost, a definite advantage over the usual local iterative techniques. After recalling basic concepts of interval analysis, this introductory paper describes algorithmic tools that can be used for nonlinear parameter estimation and applies them to simple illustrative examples drawn from compartmental modeling, a formalism widely used in biology. Guaranteed numerical integrators and properties of cooperative systems make it possible to deal with differential models, even when they do not admit closed-form solutions. Pointers to freely downloadable software are provided.

Keywords: compartmental models, cooperativity, global optimization, identifiability, interval analysis, nonlinear estimation, parameter bounding.

1. INTRODUCTION

When the vector $\mathbf{p}$ of the parameters of a model has to be estimated from experimental data by minimizing a cost function that is quadratic in $\mathbf{p}$, for instance because this cost function is quadratic in an error that is itself affine in $\mathbf{p}$, it is well known that an explicit expression for the optimal estimate $\widehat{\mathbf{p}}$ as a function of the data can be derived, and that, under suitable hypotheses on measurement noise, it is possible to compute the covariance matrix of this estimate and thus to assess its uncertainty. Although this is an important special case, it is far from covering all situations of interest. Most knowledge-based models, for instance, have outputs that are nonlinear in their parameters, which usually makes it impossible to obtain such explicit estimates. Iterative local optimization procedures are then employed, which can provide no guarantee as to their results.

The purpose of this paper is to show how interval analysis, initially developed to analyze and control numerical errors in computers (Moore, 1959), allows such guaranteed results to be obtained in nonlinear parameter estimation. Interval analysis can indeed be used to solve sets of nonlinear equations or inequalities or to minimize nonconvex cost functions (Moore, 1979), (Ratschek and Rokne, 1988), (Neumaier, 1990), (Hansen, 1992), (Hammer *et al.*, 1995), (Kearfott, 1996). Numerical solutions are provided under the form of sets guaranteed to contain *all* actual solutions of the initial mathematical problem, which is a considerable advantage over the usual numerical methods that deliver point estimates by iterative local refinement of some initial guess. The basic concepts of interval analysis will be presented first. Some tools based on interval analysis and especially relevant for nonlinear parameter estimation will then be described and applied, before providing pointers to freely downloadable software. To sim-

plify presentation and save space, many details will be skipped, for instance on how interval computation can be implemented in a guaranteed way even when a floating-point representation of real numbers is used. Much more information may be found in (Jaulin *et al.*, 2001*b*).

2. BASIC CONCEPTS

An *interval* is a connected subset of $\mathbb{R}$. Here, it will be assumed to be closed and bounded, but open-ended and unbounded intervals may also be considered. Let $\mathbb{IR}$ be the set of all closed intervals. Denote the *lower bound* of $[x] \in \mathbb{IR}$ by $\underline{x}$ or $\mathrm{lb}([x])$, and its *upper bound* by $\overline{x}$ or $\mathrm{ub}([x])$, so

$$[x] = \{x \in \mathbb{R} \mid \underline{x} \leqslant x \leqslant \overline{x}\}.$$

The *width* of $[x]$ is

$$w([x]) = \overline{x} - \underline{x},$$

and its *midpoint* (or *center*) is

$$\mathrm{mid}([x]) = \frac{\underline{x}+\overline{x}}{2}.$$

When $\underline{x}$ and $\overline{x}$ are equal, $[x]$ is said to be *degenerate*. Any real number can thus be represented as a degenerate interval. A closed interval is entirely defined by its lower and upper bounds, so these intervals have a dual nature: they may be viewed as *sets*, to which set-theoretic operations such as intersection, union and Cartesian product apply, and as *pairs of real numbers*, on which an arithmetic can be built.

2.1 *Operations on intervals*

The *intersection* of two intervals

$$[x] \cap [y] = \{z \in \mathbb{R} \mid z \in [x] \text{ and } z \in [y]\}$$

is always an interval (provided that the empty set is considered as one). The *union* of two intervals

$$[x] \cup [y] = \{z \in \mathbb{R} \mid z \in [x] \text{ or } z \in [y]\}$$

is not necessarily an interval, hence the definition of the *interval union* of $[x]$ and $[y]$, denoted by $[x] \sqcup [y]$, which is the smallest interval that contains $[x] \cup [y]$. The Cartesian product of intervals is an *interval vector* (or *box*), see the next column.

Addition, subtraction, multiplication and division can be extended to intervals according to

$$[x] \diamond [y] = [\{x \diamond y \in \mathbb{R} \mid x \in [x] \text{ and } y \in [y]\}], \qquad (1)$$

where $\diamond \in \{+,-,\cdot,/\}$ and where $[\mathbb{S}]$ is the *interval hull* of the set $\mathbb{S}$, *i.e.*, the smallest interval that contains it. In (1), $[x]$ and $[y]$ are treated as corresponding to independent variables x and y, which is a major source of pessimism. For instance, $[x]-[x]$ is not equal to $[0,0]$, unless $[x]$ is degenerate. Addition and multiplication remain associative and commutative, but multiplication is no longer distributive with respect to addition. Instead

$$[x] \cdot ([y]+[z]) \subset [x] \cdot [y] + [x] \cdot [z],$$

a property known as *subdistributivity*. As a result, it is recommended to factorize expanded forms as much as possible, to make the width of the interval results as small as possible. For nonempty intervals,

$$[x]+[y] = [\underline{x}+\underline{y}, \overline{x}+\overline{y}],$$
$$[x]-[y] = [\underline{x}-\overline{y}, \overline{x}-\underline{y}],$$
$$[x] \cdot [y] = [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}],$$

and if $0 \notin [y]$ then

$$[x]/[y] = [x] \cdot [1/\overline{y}, 1/\underline{y}].$$

(Specific formulas involving unbounded intervals are available for division by an interval containing zero.) When applied to degenerate intervals, these rules simplify into those of real arithmetic, of which interval arithmetic can thus claim to be an extension.

The interval counterpart $[f]^*$ of a function f from $\mathbb{R}$ to $\mathbb{R}$ satisfies

$$[f]^*([x]) = [\{f(x) \mid x \in [x]\}].$$

For any continuous function, $[f]^*([x])$ is thus equal to the image set $f([x])$. Elementary interval functions can be expressed in terms of bounds. This is especially simple for monotonic functions. For instance, for any nonempty $[x]$,

$$[\exp]^*([x]) = [\exp(\underline{x}), \exp(\overline{x})].$$

For nonmonotonic functions, the situation is more complicated. For example, $[\sin]^*([0,\pi]) = [0,1]$, which differs from $[\sin(0), \sin(\pi)] = [0,0]$, and specific algorithms had to be built for trigonometric and hyperbolic functions.

An *interval vector* (or *box*) $[\mathbf{x}]$ is a Cartesian product of intervals. It will be written indifferently as

$$[\mathbf{x}] = [x_1] \times [x_2] \times \cdots \times [x_n],$$

or

$$[\mathbf{x}] = ([x_1], [x_2], \ldots, [x_n])^{\mathrm{T}}.$$

The set of all n-dimensional boxes will be denoted by $\mathbb{IR}^n$. Nonempty boxes of $\mathbb{IR}^n$ are n-dimensional axis-aligned parallelepipeds. Most notions introduced for intervals extend without difficulty to boxes. The *lower bound* of $[\mathbf{x}] \in \mathbb{IR}^n$ is the vector of the lower bounds of its interval components:

$$\underline{\mathbf{x}} = (\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n)^{\mathrm{T}}.$$

Similarly, the *upper bound* of $[\mathbf{x}]$ is

$$\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_n)^{\mathrm{T}}.$$

The *width* of $[\mathbf{x}]$ is

$$w([\mathbf{x}]) = \max_{1 \leqslant i \leqslant n} w([x_i]).$$

Its *midpoint* (or *center*) is

$$\mathrm{mid}([\mathbf{x}]) = (\mathrm{mid}([x_1]), \ldots, \mathrm{mid}([x_n]))^{\mathrm{T}}.$$

The *interval hull* $[\mathbb{S}]$ of a subset $\mathbb{S}$ of $\mathbb{R}^n$ is the smallest box of $\mathbb{IR}^n$ that contains it.

The intersection of the boxes $[\mathbf{x}]$ and $[\mathbf{y}]$ of $\mathbb{IR}^n$ satisfies

$$[\mathbf{x}] \cap [\mathbf{y}] = ([x_1] \cap [y_1]) \times \ldots \times ([x_n] \cap [y_n]),$$

provided that it is nonempty. Most often the union of two boxes $[\mathbf{x}]$ and $[\mathbf{y}]$ is not a box. Its interval hull can be computed as

$$[\mathbf{x}] \sqcup [\mathbf{y}] = ([x_1] \sqcup [y_1]) \times \ldots \times ([x_n] \sqcup [y_n]).$$

The test for the inclusion of $[\mathbf{x}]$ in $[\mathbf{y}]$ can be performed componentwise, since

$$[\mathbf{x}] \subset [\mathbf{y}] \Leftrightarrow \begin{cases} [x_1] \subset [y_1], \\ \quad\vdots \\ [x_n] \subset [y_n]. \end{cases}$$

Classical operations on vectors trivially extend to interval vectors. For instance, if $\alpha \in \mathbb{R}$,

$$\begin{aligned} \alpha[\mathbf{x}] &= (\alpha[x_1]) \times \cdots \times (\alpha[x_n]), \\ [\mathbf{x}]^{\mathrm{T}} \cdot [\mathbf{y}] &= [x_1] \cdot [y_1] + \cdots + [x_n] \cdot [y_n], \\ [\mathbf{x}] + [\mathbf{y}] &= ([x_1] + [y_1]) \times \cdots \times ([x_n] + [y_n]). \end{aligned}$$

Interval matrices can be similarly defined, and the notions of lower and upper bounds, width, midpoint and interval hull extend as for boxes. If $[\mathbf{A}]$ and $[\mathbf{B}]$ are intervals, interval vectors or interval matrices of appropriate dimensions and if $\diamond$ is a binary operator, then

$$[\mathbf{A}] \diamond [\mathbf{B}] = [\{\mathbf{A} \diamond \mathbf{B} \mid \mathbf{A} \in [\mathbf{A}] \text{ and } \mathbf{B} \in [\mathbf{B}]\}].$$

For example, if $[\mathbf{A}]$ and $[\mathbf{B}]$ are in $\mathbb{IR}^{n \times n}$ then

$$[\mathbf{A}] \cdot [\mathbf{B}] = \left(\sum_{k=1}^{n} [a_{ik}] \cdot [b_{kj}] \right)_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n}.$$

More sophisticated operations such as matrix inversion and the computation of eigenvalues and eigenvectors raise difficulties that go beyond this introductory paper. See (Neumaier, 1990) for details.

2.2 *Inclusion functions*

Let $\mathbf{f}$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^m$, which may be defined by an algorithm or even by a differential equation. The interval function $[\mathbf{f}]$ from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ is an *inclusion function* for $\mathbf{f}$ if

$$\forall [\mathbf{x}] \in \mathbb{IR}^n, \ \mathbf{f}([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{x}]).$$

The image set $\mathbf{f}([\mathbf{x}])$ may have any shape. It may be nonconvex, or even disconnected if $\mathbf{f}$ is discontinuous. Whatever this shape, an inclusion function $[\mathbf{f}]$ of $\mathbf{f}$ makes it possible to compute a box $[\mathbf{f}]([\mathbf{x}])$ guaranteed to contain $\mathbf{f}([\mathbf{x}])$. Actually this box may offer a very pessimistic vision of $\mathbf{f}([\mathbf{x}])$, but since it is far easier to manipulate boxes than generic sets, this is a very interesting tool for studying functions. One of the main challenges of interval analysis is to provide inclusion functions that can be evaluated quickly while keeping $[\mathbf{f}]([\mathbf{x}])$ as small as possible.

An inclusion function $[\mathbf{f}]$ for $\mathbf{f}$ is *convergent* if, for any sequence of boxes $[\mathbf{x}]_k$,

$$\lim_{k \to \infty} w([\mathbf{x}]_k) = 0 \Rightarrow \lim_{k \to \infty} w([\mathbf{f}]([\mathbf{x}]_k)) = 0.$$

It is *minimal* if for any $[\mathbf{x}]$, $[\mathbf{f}]([\mathbf{x}])$ is the smallest box that contains $\mathbf{f}([\mathbf{x}])$. The minimal inclusion function for $\mathbf{f}$ is unique and will be denoted by $[\mathbf{f}]^*$, as it corresponds to the interval counterpart of $\mathbf{f}$. $[\mathbf{f}]$ is *inclusion monotonic* if

$$[\mathbf{x}] \subset [\mathbf{y}] \Rightarrow [\mathbf{f}]([\mathbf{x}]) \subset [\mathbf{f}]([\mathbf{y}]).$$

A minimal inclusion function is inclusion monotonic but not necessarily convergent (because $\mathbf{f}$ may be discontinuous). A convergent inclusion function may not be inclusion monotonic.

The construction of inclusion functions for $\mathbf{f}$ can be cast into that of inclusion functions for each of its coordinate functions. This is why we shall focus attention on getting inclusion functions for real-valued functions $f : \mathbb{R}^n \to \mathbb{R}$. The first idea that comes to mind is to perform two optimizations to compute the *infimum* and *supremum* of f over the box $[\mathbf{x}]$ of interest. At least in principle, one should thus get the smallest interval containing $f([\mathbf{x}])$, denoted by $[f]^*([\mathbf{x}])$. However, these optimization problems themselves turn out to be far from trivial in general. An alternative and much more tractable approach is as follows. Consider a function f expressed as a finite composition of the operators $+, -, \cdot, /$ and elementary functions $\sin, \cos, \exp$, sqrt... An inclusion-monotonic inclusion function $[f] : \mathbb{IR}^n \to \mathbb{IR}$ for f is obtained by replacing each real variable x_i by an interval variable $[x_i]$ and each operator or elementary function by its interval counterpart. The result is called the *natural inclusion function* of f. If f involves only continuous operators and continuous elementary functions, then $[f]$ is convergent. If, moreover, each of the variables $x_1, \ldots, x_n$ occurs at most once in the formal expression of f then $[f]$ is minimal. Unfortunately, natural inclusion functions are not minimal in general, and their performance strongly depends on the formal expression of f, as illustrated by the next example.

Consider the following three formal expressions of the same function f:

$$\begin{aligned} f_1(x) &= x(x+2), \\ f_2(x) &= x^2 + 2x, \\ f_3(x) &= (x+1)^2 - 1. \end{aligned}$$

At $[x] = [-1, 1]$, their natural inclusion functions take the values

$$\begin{aligned} [f_1]([x]) &= [x]([x]+2) = [-3, 3], \\ [f_2]([x]) &= [x]^2 + 2[x] = [-2, 3], \\ [f_3]([x]) &= ([x]+1)^2 - 1 = [-1, 3]. \end{aligned}$$

$[f_2]$ is less pessimistic than $[f_1]$ because it takes advantage of the fact that the lower bound of a square cannot be negative. Since x occurs only once in f_3 and f_3 is

continuous, $[f_3]$ is minimal. Thus $[f_3]([x]) = f([x]) = [-1,3]$.

Natural inclusion functions are not always to be recommended. Their efficiency depends strongly on the number of occurrences of each variable, which is often difficult to reduce. An important field of investigation of interval analysis (Ratschek and Rokne, 1984) has thus been to propose other types of inclusion function that might provide less pessimistic results, such as those now briefly described.

Assume that f is differentiable over $[\mathbf{x}]$, and denote $\mathrm{mid}([\mathbf{x}])$ by $\mathbf{m}$. The mean-value theorem then implies that

$$\forall \mathbf{x} \in [\mathbf{x}], \exists \mathbf{z} \in [\mathbf{x}] \text{ such that } f(\mathbf{x}) = f(\mathbf{m}) + \mathbf{g}^{\mathrm{T}}(\mathbf{z}) \cdot (\mathbf{x} - \mathbf{m}),$$

where $\mathbf{g}$ is the gradient of f. Thus,

$$\forall \mathbf{x} \in [\mathbf{x}], f(\mathbf{x}) \in f(\mathbf{m}) + [\mathbf{g}^{\mathrm{T}}]([\mathbf{x}]) \cdot (\mathbf{x} - \mathbf{m}),$$

where $[\mathbf{g}^{\mathrm{T}}]$ is an inclusion function for $\mathbf{g}^{\mathrm{T}}$, so

$$f([\mathbf{x}]) \subset f(\mathbf{m}) + [\mathbf{g}^{\mathrm{T}}]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{m}).$$

Therefore, the interval function

$$[f_{\mathrm{c}}]([\mathbf{x}]) = f(\mathbf{m}) + [\mathbf{g}^{\mathrm{T}}]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{m})$$

is an inclusion function for f, called its *centered inclusion function*. This function becomes especially interesting when the width of $[\mathbf{x}]$ is small, because the pessimism resulting from the interval evaluation of $[\mathbf{g}]([\mathbf{x}])$ is reduced by the scalar product with $[\mathbf{x}] - \mathbf{m}$, which is a small interval centered on zero. The centered inclusion function can be significantly improved at the cost of a slightly more complicated formulation (Hansen, 1968). Iterating the reasoning that led to the centered inclusion function, one may think of using a Taylor series expansion to approximate f at a higher order. This leads to the second-order *Taylor inclusion function*

$$[f]_{\mathrm{T}}([\mathbf{x}]) = f(\mathbf{m}) + \mathbf{g}^{\mathrm{T}}(\mathbf{m}) \cdot ([\mathbf{x}] - \mathbf{m}) + \tfrac{1}{2}([\mathbf{x}] - \mathbf{m})^{\mathrm{T}} \cdot [\mathbf{H}]([\mathbf{x}]) \cdot ([\mathbf{x}] - \mathbf{m}),$$

where $[\mathbf{H}]$ is an inclusion function for the *Hessian matrix* of f.

The convergence rate of a natural inclusion function is at least linear, whereas those of the centered and Taylor inclusion functions are at least quadratic provided that the Taylor inclusion function is based on an expansion to order $k \geqslant 2$. Quadratic convergence looks more interesting than linear convergence, but it should be remembered that it only means that more accurate results will be obtained in the case of infinitesimal boxes. Nothing similar can be said on the behavior of these inclusion functions for large boxes. To the contrary, when the box involved is large enough, the natural inclusion function is often more satisfactory than the centered and Taylor inclusion functions. No approach to building an inclusion function can thus claim to be uniformly the best, and a compromise between complexity and efficiency must often be struck. One may also use several inclusion functions and take the intersection of their image sets to get a better approximation of the image set of the original function.

2.3 *Subpavings*

Intervals and boxes are not by themselves general enough to describe all sets of interest, which include, for instance, unions of disconnected subsets. This motivates the introduction of the notion of subpaving. A *subpaving* of a box $[\mathbf{x}]$ is a union of nonoverlapping subboxes of $[\mathbf{x}]$ with nonzero width. The set $\mathbb{S}$ of interest will be bracketed between inner and outer approximations by computing subpavings $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ such that

$$\underline{\mathbb{S}} \subset \mathbb{S} \subset \overline{\mathbb{S}}.$$

The distance between $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ is indicative of the quality of the approximation of $\mathbb{S}$. Computation on subpavings allows approximate computation on compact sets, and forms the basic ingredient of the parameter estimation algorithms to be presented below. The special class of *regular subpavings* simplifies representation and manipulation by computers as they can be represented by binary trees. A subpaving of $[\mathbf{x}]$ is regular if each of its boxes can be obtained from $[\mathbf{x}]$ by a finite succession of bisections and selections.

2.4 *Contractors*

Consider a vector $\mathbf{x}$ of n real variables x_i, linked by relations (or constraints) that can be written in vector form as

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \tag{2}$$

Assume that the prior domain for $\mathbf{x}$ is the box

$$[\mathbf{x}] = [x_1] \times \cdots \times [x_n].$$

Solving (2) for $\mathbf{x}$ in $[\mathbf{x}]$ is a *constraint satisfaction problem* (CSP) (Davis, 1987), (Hyvönen, 1992), (Sam-Haroud and Faltings, 1996), (van Hentenryck *et al.*, 1998), the solution set of which is

$$\mathbb{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\}.$$

Inequality constraints can be dealt with within this framework by introducing *slack variables*. Instead of looking for $\mathbb{S}$, which is an NP-complete problem, *contractors* aim more modestly at reducing the size of the prior domain without loosing any solution or performing any bisection. Many contractors are available (Jaulin *et al.*, 2001*b*) and we shall only present one of them, namely the *interval Newton contractor* classically used for global optimization. Assume that $\mathbf{f}$ is once differentiable, and let $\mathbf{J_f}$ be its Jacobian matrix and $\mathbf{m}$ be the midpoint of $[\mathbf{x}]$. The mean-value theorem implies that for any $\mathbf{x} \in [\mathbf{x}]$ there exists $\mathbf{z} \in [\mathbf{x}]$ such that

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{m}) + \mathbf{J_f}(\mathbf{z}) \cdot (\mathbf{x} - \mathbf{m}). \tag{3}$$

Now assume that $\widehat{\mathbf{x}} \in [\mathbf{x}]$ is a solution of (2), so $\mathbf{f}(\widehat{\mathbf{x}}) = \mathbf{0}$. Assume further that the number of constraints is equal to the number of variables, so $\mathbf{J_f}$ is square. Provided that it is also invertible, (3) implies that

$$\widehat{\mathbf{x}} = \mathbf{m} - \mathbf{J_f}^{-1}(\mathbf{z}) \cdot \mathbf{f}(\mathbf{m}).$$

Now $\mathbf{z}$ is known to belong to $[\mathbf{x}]$, so

$$\widehat{\mathbf{x}} \in \mathbf{m} - \mathbf{J_f}^{-1}([\mathbf{x}]) \cdot \mathbf{f}(\mathbf{m}).$$

Since $\widehat{\mathbf{x}}$ is also assumed to belong to $[\mathbf{x}]$, it must belong to

$$[\mathbf{x}_\mathrm{r}] = [\mathbf{x}] \cap \left(\mathbf{m} - \mathbf{J_f}^{-1}([\mathbf{x}]) \cdot \mathbf{f}(\mathbf{m})\right). \tag{4}$$

The interval Newton contractor thus replaces the prior box $[\mathbf{x}]$ by a possibly much smaller box $[\mathbf{x}_\mathrm{r}]$, which may even be empty if $[\mathbf{x}]$ does not contain any solution. This is obtained without bisection and thus escapes the curse of dimensionality. The often unrealistic assumption that $\mathbf{J_f}([\mathbf{x}])$ is invertible can be relaxed by replacing (4) by the computation of an outer approximation of the set of all solutions for $\widehat{\mathbf{x}}$ in $[\mathbf{x}]$ of the *linear* system of equations.

$$\mathbf{f}(\mathbf{m}) + \mathbf{J_f}([\mathbf{x}]) \cdot (\widehat{\mathbf{x}} - \mathbf{m}) = \mathbf{0}.$$

Specific methods involving *preconditioning* are used for this purpose (Hammer *et al.*, 1995).

3. BASIC TOOLS FOR NONLINEAR ESTIMATION

3.1 *Global optimization*

The problem to be considered now is the minimization of a cost function c over a compact set $\mathbb{P} \subset \mathbb{R}^{n_\mathrm{p}}$:

$$\min_{\mathbf{p} \in \mathbb{P}} c(\mathbf{p}).$$

It is, of course, always possible to transform a maximization problem into a minimization problem, for instance by multiplying $c(\mathbf{p})$ by -1. The *global minimum* will be denoted by $\widehat{c}$, and the set of all corresponding *global minimizers* by $\mathbb{S}$. Although interval analysis may also be used for constrained or minimax problems, we shall limit ourselves to *unconstrained minimization*. This means that $\mathbb{P}$ is only a (possibly very large) domain of interest, and that the minimizers are not expected to lie on its boundary. The main ideas of Hansen's algorithm will be presented. As this algorithm forms the main subject of his book, this will be an oversimplification and the reader is urged to consult (Hansen, 1992) for more details. For simplicity, the search domain $\mathbb{P}$ will be assumed to be a box $[\mathbf{p}]_0$ in parameter space. Let $\mathscr{L}$ be a list of boxes included in $\mathbb{P}$ that may contain global minimizers. Initially, $\mathscr{L}$ contains only $[\mathbf{p}]_0$. The basic structure of the algorithm is as follows, where $\varepsilon > 0$ is a tolerance parameter to be chosen by the user.

While $\mathscr{L}$ contains at least one box with width greater than ε, do {

(1) Pop the first such box $[\mathbf{p}]$ out of $\mathscr{L}$,
(2) Update the upper bound $\mathrm{ub}(\widehat{c})$ of the global minimum $\widehat{c}$ over $[\mathbf{p}]_0$.
(3) Try to eliminate $[\mathbf{p}]$ via the midpoint, monotonicity and convexity tests.
(4) Should this fails, try to eliminate or at least reduce $[\mathbf{p}]$ with the interval Newton contractor.
(5) If $[\mathbf{p}]$ resists elimination and $w([\mathbf{p}]) > \varepsilon$ then bisect it and push the two resulting boxes into $\mathscr{L}$. Otherwise $[\mathbf{p}]$ is deemed too small to be bisected and will be pushed back as is into $\mathscr{L}$. }

At last eliminate all boxes of $\mathscr{L}$ that no longer pass the midpoint test.

Except during the first iteration of the algorithm, when $\mathscr{L}$ contains only one box, the rule for selecting the box to be considered at Step 1 must be chosen carefully as it has much impact on performance. A good strategy is to take one of the boxes with the largest potential, *i.e.*, a box $[\mathbf{p}]$ such that for any other box $[\mathbf{q}]$ in $\mathscr{L}$ with width greater than ε

$$\mathrm{lb}([c]([\mathbf{p}]) \leqslant \mathrm{lb}([c]([\mathbf{q}]).$$

To update the upper bound of $\widehat{c}$ as required by Step 2, it suffices to compute $c(\mathrm{mid}([\mathbf{p}]))$, which is of course an upper bound of the optimal value of the cost, and to compare it with the best upper bound available so far.

Three methods for eliminating the box $[\mathbf{p}]$ under consideration are tried at Step 3. The *midpoint test* requires the evaluation of the image of $[\mathbf{p}]$ by an inclusion function $[c]$ of the cost function. If

$$\mathrm{lb}([c]([\mathbf{p}])) > \mathrm{ub}(\widehat{c}),$$

then no vector $\mathbf{p}$ in $[\mathbf{p}]$ can be a global minimizer of c over $\mathbb{P}$, so $[\mathbf{p}]$ can be eliminated. The *monotonicity test* takes advantage of the fact that if c is once differentiable then any unconstrained minimizer corresponds to a stationary point at which the gradient $\mathbf{g}$ takes the value zero. Thus if an inclusion function $[\mathbf{g}]$ is available for $\mathbf{g}$ and if $[\mathbf{p}]$ is such that

$$\mathbf{0} \notin [\mathbf{g}]([\mathbf{p}]),$$

then no vector $\mathbf{p}$ in $[\mathbf{p}]$ can be an unconstrained minimizer so $[\mathbf{p}]$ can be eliminated. Finally the *convexity test* is based on the fact that if c is twice differentiable, then a necessary condition for $\mathbf{p}$ to be an unconstrained minimizer is that the Hessian $\mathbf{H}$ of c be nonnegative definite at $\mathbf{p}$. A necessary condition for this to be true is that all diagonal entries of $\mathbf{H}(\mathbf{p})$ satisfy

$$h_{ii}(\mathbf{p}) \geqslant 0, i = 1, \ldots, n_\mathrm{p}.$$

So if inclusion functions $[h_{ii}]$ are available for the diagonal entries of the Hessian and if $[\mathbf{p}]$ is such that

$$\exists i \in \{1, \ldots, n_\mathrm{p}\} \mid [h_{ii}]([\mathbf{p}]) < 0,$$

then no $\mathbf{p}$ in $[\mathbf{p}]$ can be an unconstrained minimizer and $[\mathbf{p}]$ can be eliminated.

Automatic differentiation can be used to compute inclusion functions for $\mathbf{g}$ and $\mathbf{H}$ from the code evaluating $c(\mathbf{p})$ (Rall, 1981), (Griewank and Corliss, 1991),

(Hammer *et al.*, 1995), (Rall and Corliss, 1999), (Jaulin *et al.*, 2001*b*). Note that if the cost function c is not differentiable, or if it turns out that inclusion functions for its gradient or Hessian cannot be obtained, then parts of Step 3 may have to be dropped. In the limit, Step 3 may be based only on the midpoint test.

Step 4 uses an interval Newton contractor to reduce the size of $[\mathbf{p}]$ without having to bisect it. The problem is to find a box guaranteed to contain all solutions in $[\mathbf{p}]$ of

$$\mathbf{g}(\mathbf{p}) = \mathbf{0},$$

so $\mathbf{g}$ plays the role of $\mathbf{f}$ in Section 2.4, and the role of the Jacobian matrix of $\mathbf{f}$ is played by the Hessian of c. Step 4 can of course be implemented only if c is twice differentiable.

The simplest strategy for bisection at Step 5 is to cut $[\mathbf{p}]$ perpendicularly to one of its edges of maximum width, but more sophisticated policies taking into account information provided by the gradient may also be considered.

Since the tolerance parameter ε is strictly positive, the algorithm stops after a finite number of steps and the set $\mathbb{S}$ of all global minimizers of c is included in the union of the boxes in $\mathscr{L}$, each of which has now a width smaller than ε. It is also possible to compute an interval $[\widehat{c}]$ guaranteed to contain $\widehat{c}$, with its upper bound the best upper bound obtained at Step 2 and its lower bound given by

$$\underline{\widehat{c}} = \min_{[\mathbf{p}]\in\mathscr{L}} \mathrm{lb}([c]([\mathbf{p}])).$$

The algorithm can also be made to stop when

$$w([\widehat{c}]) < \eta,$$

where η is another tolerance parameter to be set by the user.

3.2 *Set inversion*

Let $\mathbf{f}$ be a possibly nonlinear function from $\mathbb{R}^{n_\mathrm{p}}$ to $\mathbb{R}^{n_\mathrm{y}}$ and $\mathbb{Y}$ be a subpaving of $\mathbb{R}^{n_\mathrm{y}}$. *Set inversion* is the characterization of the reciprocal image of $\mathbb{Y}$

$$\mathbb{S} = \{\mathbf{p} \in \mathbb{R}^{n_\mathrm{p}} \mid \mathbf{f}(\mathbf{p}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}). \tag{5}$$

Using an inclusion function $[\mathbf{f}]$ for $\mathbf{f}$, two regular subpavings $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ such that

$$\underline{\mathbb{S}} \subset \mathbb{S} \subset \overline{\mathbb{S}}$$

can be obtained with the algorithm SIVIA, for Set Inverter Via Interval Analysis (Jaulin and Walter, 1993).

A (possibly very large) search box $[\mathbf{p}]_0$ to which $\overline{\mathbb{S}}$ is guaranteed to belong must be provided by the user. It will be used to grow a subpaving by successive bisections and selections. Consider a box $[\mathbf{p}]$ of this subpaving. Four cases may be encountered.

(1) If $[\mathbf{f}]([\mathbf{p}])$ has a nonempty intersection with $\mathbb{Y}$, but is not entirely in $\mathbb{Y}$, then $[\mathbf{p}]$ may contain a part of $\mathbb{S}$ without being included in $\mathbb{S}$ and is said to be *undetermined*. If it has a width greater than a prespecified precision parameter ε, then this undetermined box should be bisected into the boxes $L[\mathbf{p}]$ and $R[\mathbf{p}]$, each of which should be investigated by the algorithm.
(2) If $[\mathbf{f}]([\mathbf{p}])$ has an empty intersection with $\mathbb{Y}$, then $[\mathbf{p}]$ has an empty intersection with $\mathbb{S}$ and should be discarded.
(3) If $[\mathbf{f}]([\mathbf{p}])$ is included in $\mathbb{Y}$, then $[\mathbf{p}]$ is included in $\mathbb{S}$, and should be stored in $\underline{\mathbb{S}}$ and in $\overline{\mathbb{S}}$.
(4) If the box considered is undetermined, but its width is lower than ε, then it is deemed too small to be bisected further and is stored as is in the outer approximation $\overline{\mathbb{S}}$ of $\mathbb{S}$.

The resulting recursive algorithm is given in Table 1, where the subpavings $\underline{\mathbb{S}}$ and $\overline{\mathbb{S}}$ should be initialized as empty before the first call.

Algorithm SIVIA(in: $\mathbf{f}, \mathbb{Y}, [\mathbf{p}], \varepsilon$; inout: $\underline{\mathbb{S}}, \overline{\mathbb{S}}$)

```
1  if [f]([p]) ∩ Y = ∅ return;
2  if [f]([p]) ⊂ Y then
3  {S̲ := S̲ ∪ [p]; S̄ := S̄ ∪ [p]; return;};
4  if w([p]) < ε then {S̄ := S̄ ∪ [p]; return;};
5  SIVIA(f, Y, L[p], ε, S̲, S̄);
     SIVIA(f, Y, R[p], ε, S̲, S̄).
```

Table 1. SIVIA

The *uncertainty layer* $\Delta\mathbb{S}$ consisting of all boxes of $\overline{\mathbb{S}}$ that are not in $\underline{\mathbb{S}}$ is a regular subpaving, all boxes of which have a width smaller than ε.

3.3 *Guaranteed numerical integration*

As seen above, Hansen's algorithm and SIVIA both require the availability of inclusion functions. When no closed-form expression or finite algorithm is available for the evaluation of $\mathbf{f}$, the construction of $[\mathbf{f}]$ becomes much more difficult. Such a situation often occurs, for example, when the evaluation of $\mathbf{f}$ requires the solution of some ordinary differential equation

$$\mathbf{x}' = \mathbf{g}(\mathbf{x}, \mathbf{p}, t), \text{ with } \mathbf{x}(0) = \mathbf{x}_0(\mathbf{p}), \tag{6}$$

where

$$\mathbf{x}' = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$$

and $\mathbf{p}$ is some parameter vector only known to belong to a box $[\mathbf{p}]$. At a given time instant t, let $\mathbf{f}(\mathbf{p}, t)$ be the solution of (6) for a given $\mathbf{p} \in [\mathbf{p}]$ and let $\mathbf{f}([\mathbf{p}], t)$ be the set of all values that can be reached by the solution of (6) for all $\mathbf{p}$ in $[\mathbf{p}]$.

Several numerical integrators based on interval analysis are readily available to compute boxes guaranteed to contain $\mathbf{f}([\mathbf{p}], t)$, *e.g.*, AWA (Lohner, 1992), COSY (Hoefkens *et al.*, 2001) or VNODE (Nedialkov and Jackson, 2001). The techniques employed usually consist of two steps. First, the existence of a solution for (6) over a given time interval is ensured using

Brouwer's fixed-point theorem. As a result, a box containing $\mathbf{f}([\mathbf{p}],t)$ is obtained. This box is then refined using a Taylor expansion of the solution or other similar schemes to get a smaller box $[\mathbf{f}]([\mathbf{p}],t)$ satisfying

$$\mathbf{f}([\mathbf{p}],t) \subset [\mathbf{f}]([\mathbf{p}],t).$$

Guaranteed numerical integration thus provides an inclusion function for $\mathbf{f}(.,t)$. This technique is well suited for systems such as (6), where $[\mathbf{p}]$ is a degenerate box with zero width. For large boxes as needed in the context of parameter estimation, the enclosure for $\mathbf{f}([\mathbf{p}],t)$ may become *very* pessimistic. However, when (6) can be bounded between *cooperative systems*, it remains possible to obtain an accurate inclusion function for $\mathbf{f}([\mathbf{p}],t)$, see (Gouzé *et al.*, 2000).

Definition 1. The dynamical system

$$\mathbf{x}' = \mathbf{g}(\mathbf{x},t),$$

where $\mathbf{x} \in \mathscr{D} \subset \mathbb{R}^n$, is *cooperative* over $\mathscr{D}$ if

$$\frac{\partial g_i(\mathbf{x},t)}{\partial x_j} \geqslant 0 \text{ for all } i \neq j,\ t \geqslant 0 \text{ and } \mathbf{x} \in \mathscr{D}.$$

The following theorem, which is a reformulation of a result described in (Smith, 1995), indicates how an enclosure of the solution of (6) can be obtained with cooperative systems when $\mathbf{p}$ is only known to belong to $[\mathbf{p}]$.

Theorem 1. If there exists a pair of cooperative systems

$$\underline{\mathbf{x}}' = \underline{\mathbf{g}}\left(\underline{\mathbf{x}},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right) \text{ and } \overline{\mathbf{x}}' = \overline{\mathbf{g}}\left(\overline{\mathbf{x}},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$$

satisfying

$$\underline{\mathbf{g}}\left(\mathbf{x},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right) \leqslant \mathbf{g}(\mathbf{x},\mathbf{p},t) \leqslant \overline{\mathbf{g}}\left(\mathbf{x},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$$

for all $\mathbf{p} \in \left[\underline{\mathbf{p}},\overline{\mathbf{p}}\right]$, $t \geqslant 0$ and $\mathbf{x} \in \mathscr{D}$, and if these cooperative systems can be given initial conditions $\underline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right)$ and $\overline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right)$ such that

$$\underline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right) \leqslant \mathbf{x}_0(\mathbf{p}) \leqslant \overline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right)$$

for all $\mathbf{p} \in \left[\underline{\mathbf{p}},\overline{\mathbf{p}}\right]$, then the solution of (6) satisfies

$$\underline{\mathbf{x}}(t) \leqslant \mathbf{x}(t) \leqslant \overline{\mathbf{x}}(t), \text{ for all } t \geqslant 0,$$

where $\underline{\mathbf{x}}(t) = \underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$ is the flow associated with

$$\left\{\underline{\mathbf{x}}' = \underline{\mathbf{g}}\left(\underline{\mathbf{x}},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right), \underline{\mathbf{x}}(0) = \underline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right)\right\}$$

and $\overline{\mathbf{x}}(t) = \overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$ is the flow associated with

$$\left\{\overline{\mathbf{x}}' = \overline{\mathbf{g}}\left(\overline{\mathbf{x}},\underline{\mathbf{p}},\overline{\mathbf{p}},t\right), \overline{\mathbf{x}}(0) = \overline{\mathbf{x}}_0\left(\underline{\mathbf{p}},\overline{\mathbf{p}}\right)\right\}.$$

■

For any given $t \geqslant 0$, the box-valued function

$$[\varphi]\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right) = \left[\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right), \overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)\right]$$

is thus an inclusion function for the solution of (6). Usually, however, no closed-form expressions are available for $\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$ and $\overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)$, but guaranteed integration makes it possible to compute enclosures for these flows at any $t \geqslant 0$ as

$$\left[\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)\right] = \left[\underline{\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}, \overline{\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}\right]$$

and

$$\left[\overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)\right] = \left[\underline{\overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}, \overline{\overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}\right].$$

The function

$$[\Phi]\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right) = \left[\underline{\underline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}, \overline{\overline{\varphi}\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right)}\right]$$

is then such that

$$[\mathbf{f}]([\mathbf{p}],t) \subset [\Phi]\left(\underline{\mathbf{p}},\overline{\mathbf{p}},t\right) \tag{7}$$

and is therefore an inclusion function for the solution of (6), which can be numerically evaluated for any $t \geqslant 0$.

Note that the right-hand side of (7) no longer involves boxes, contrary to its left-hand side. This allows guaranteed numerical integration to be used on degenerate boxes, with much more accurate results.

4. PARAMETER ESTIMATION

The interval tools previously presented will now be applied to two types of nonlinear parameter estimation problems, namely parameter optimization and parameter bounding.

4.1 *Parameter optimization*

Nonlinear parameter estimation often boils down to the optimization of a cost function, which may for instance result from information or hypotheses about the noise corrupting the data. This cost function is not convex in general, and local optimization techniques cannot be guaranteed to locate its global minimum and all corresponding global minimizers. Since computing time is finite, no global search method based on random exploration can offer any guarantee either. This makes deterministic global optimization algorithms such as that presented in Section 3.1 particularly attractive.

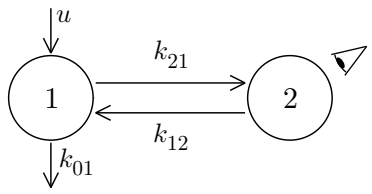

Fig. 1. Two-compartment model

To illustrate their potential and limitation, consider the compartmental model described by Figure 1. Such models are widely used in biology. They consist of finite sets of homogeneous subsystems, called compartments and represented by circles, which may exchange material as indicated by arrows. The equations

describing the behavior of the compartmental model are readily obtained by writing down conservation equations, under the form of a state equation. Let $\mathbf{x} = (x_1, x_2)^{\mathrm{T}}$ be the vector of the (positive) quantities of material in the two compartments of the model of Figure 1 and assume, for the time being, that all flows of material are proportional to the quantity of material in the origin compartment. The evolution of $\mathbf{x}$ is then governed by the linear state equation

$$\mathbf{x}' = \mathbf{g}(\mathbf{x}, \mathbf{p}, u), \tag{8}$$

where

$$\mathbf{p} = (k_{01}, k_{12}, k_{21})^{\mathrm{T}}$$

and

$$\mathbf{g}(\mathbf{x}, \mathbf{p}, u) = \begin{pmatrix} -(k_{21} + k_{01}) x_1 + k_{12} x_2 + u \\ k_{21} x_1 - k_{12} x_2 \end{pmatrix}.$$

Assume that the quantity x_2 of material in Compartment 2 is observed, so the model output is

$$y_{\mathrm{m}}(\mathbf{p}, t_i) = x_2(\mathbf{p}, t_i), \; i = 1, ..., n_{\mathrm{y}}. \tag{9}$$

Assume further that there is no input ($u \equiv 0$) and that the initial condition is $\mathbf{x}_0 = (1, 0)^{\mathrm{T}}$. Then, for $t_i \geqslant 0$,

$$y_{\mathrm{m}}(\mathbf{p}, t_i) = \alpha(\mathbf{p}) \left(\mathrm{e}^{\lambda_1(\mathbf{p}) t_i} - \mathrm{e}^{\lambda_2(\mathbf{p}) t_i} \right), \tag{10}$$

where

$$\alpha(\mathbf{p}) = \frac{k_{21}}{\sqrt{(k_{01} - k_{12} + k_{21})^2 + 4 k_{12} k_{21}}}, \tag{11}$$

$$\lambda_{1,2}(\mathbf{p}) = -\frac{1}{2} [(k_{01} + k_{12} + k_{21}) \pm ((k_{01} - k_{12} + k_{21})^2 + 4 k_{12} k_{21})^{1/2}]. \tag{12}$$

Although the model is linear, its output thus depends nonlinearly on its parameters.

The least-square estimate of $\mathbf{p}$ is obtained by minimizing the cost function

$$c(\mathbf{p}) = \sum_{i=1}^{n_{\mathrm{y}}} (y(t_i) - y_{\mathrm{m}}(\mathbf{p}, t_i))^2, \tag{13}$$

where $y(t_i)$ is the quantity of material measured in Compartment 2 at time t_i. Results obtained with Hansen's algorithm on this problem where reported in (Kieffer and Walter, 1998) and will only be summarized here. With sixteen data points, the search domain $\mathbb{P} = [0.01, 2.0] \times [0.05, 3.0] \times [0.05, 3.0]$ and a precision parameter $\varepsilon = 10^{-9}$, it took about a day on a Pentium at 200MHz to enclose the set of all global minimizers of (13) in the union of two very small boxes in parameter space. These two solutions are due to the fact that the parameters of this model are only locally identifiable, see, *e.g.*, (Walter, 1982), because the values of k_{12} and k_{01} can be exchanged without modifying input-output behavior. Note that no identifiability study was needed to reach this conclusion here. The disappointing length of these computations is due to multioccurrences of the parameters in the natural inclusion function for $y_{\mathrm{m}}(\mathbf{p}, t_i)$ and *a fortiori* in the natural inclusion function for c. Note that $\mathbf{p}$ appears n_{y} times in the formal expression of the cost function. Optimization time can be drastically decreased to about 90s by using an intermediate parametrization of $y_{\mathrm{m}}(\mathbf{p}, t_i)$ in terms of α, λ_1 and λ_2 and then solving (11) and (12) for $\mathbf{p}$ using an interval Newton solver.

4.2 *Parameter bounding*

Parameter bounding, see, *e.g.*, (Milanese *et al.*, 1996), represents an attractive alternative to parameter optimization. Instead of looking for an optimal value of $\mathbf{p}$, one looks for the *set* of all parameter vectors that are consistent (in a sense to be specified) with the experimental data, model structure and error bounds. It is assumed that each experimental datum $y(t_i)$ corresponds to a known interval $[\underline{e}_i, \overline{e}_i]$ of acceptable errors. A parameter vector $\mathbf{p} \in [\mathbf{p}]_0$ is deemed acceptable if $\underline{e}_i \leqslant y(t_i) - y_{\mathrm{m}}(\mathbf{p}, t_i) \leqslant \overline{e}_i$ for all $i = 1, \ldots, n_{\mathrm{y}}$. Parameter estimation then amounts to the characterization of the set

$$\begin{aligned} \mathbb{S} &= \{\mathbf{p} \in [\mathbf{p}]_0 \mid y(t_i) - y_{\mathrm{m}}(\mathbf{p}, t_i) \in [\underline{e}_i, \overline{e}_i], \\ &\qquad i = 1, \ldots, n_{\mathrm{y}}\} \\ &= \{\mathbf{p} \in [\mathbf{p}]_0 \mid \mathbf{y}_{\mathrm{m}}(\mathbf{p}) \in [\mathbf{y}]\}, \end{aligned} \tag{14}$$

with

$$\begin{aligned} [\mathbf{y}] = [y(t_1) - \overline{e}_1, y(t_1) - \underline{e}_1] \times \\ \cdots \times [y(t_{n_{\mathrm{y}}}) - \overline{e}_{n_{\mathrm{y}}}, y(t_{n_{\mathrm{y}}}) - \underline{e}_{n_{\mathrm{y}}}] \end{aligned}$$

and

$$\mathbf{y}_{\mathrm{m}}(\mathbf{p}) = \left(y_{\mathrm{m}}(\mathbf{p}, t_1), \ldots, y_{\mathrm{m}}(\mathbf{p}, t_{n_{\mathrm{y}}}) \right)^{\mathrm{T}}.$$

One of the advantages of this approach from the point of view of interval analysis is that the multioccurrences of $\mathbf{p}$ due to cost functions such as (13) are avoided, thus decreasing the pessimism of inclusion functions and the number of bisections required to reach a satisfactory conclusion.

A guaranteed enclosure of $\mathbb{S}$ can be obtained using SIVIA, described in Section 3.2, as (14) is a set-inversion problem similar to (5), with $\mathbb{Y} = [\mathbf{y}]$. Solving (14) with SIVIA requires an inclusion function for $\mathbf{y}_{\mathrm{m}}(\mathbf{p})$ to be provided. Two approaches may be considered depending on whether a closed-form expression is available for $y_{\mathrm{m}}(\mathbf{p}, t_i)$. They will be illustrated by considering again the model of Figure 1.

4.2.1. *Set inversion using a closed-form expression*

Assume that

$$y(t_i) = x_2(\mathbf{p}^*, t_i)(1 + b(t_i)), \; i = 1, ..., 16, \tag{15}$$

where $\mathbf{p}^*$ is the true value of the parameter vector and $b(t_i) \in [\underline{b}, \overline{b}]$. Noisy data have been obtained as

follows. First the noise-free response $x_2(\mathbf{p}^*, t_i)$ was computed by simulating (8) with the true value of the parameters to be estimated given by $(k_{01}^*, k_{12}^*, k_{21}^*) = (1, 0.25, 0.5)$. This noise-free response was then corrupted according to (15) with realizations of a pseudorandom noise $b(t_i)$ belonging to $[-0.05, 0.05]$. The resulting data points are indicated by dots on Figure 2.

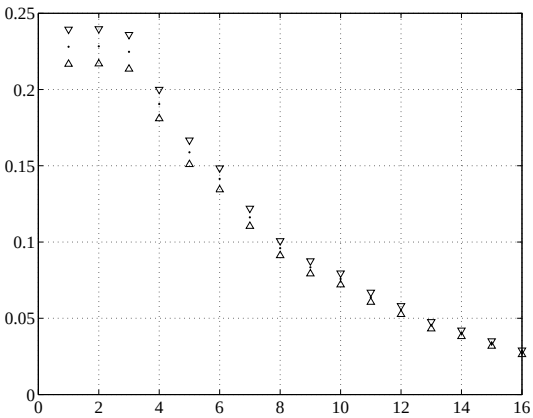

Fig. 2. Measured output (·), upper bounds (▽) and lower bounds (△) of the tolerance intervals (the true system is linear)

The intervals $[\underline{e}_i, \overline{e}_i]$ must be chosen such that

$$x_2(\mathbf{p}^*, t_i) \in [y(t_i) - \overline{e}_i, y(t_i) - \underline{e}_i], \; i = 1, ..., 16. \tag{16}$$

Equation (15) implies that

$$\begin{aligned} x_2(\mathbf{p}^*, t_i) &\in \frac{1}{1 + [-0.05, 0.05]} y(t_i) \\ &\subset [0.95238, 1.05264] y(t_i), \end{aligned}$$

so taking

$$\underline{e}_i = -0.05264 y(t_i) \tag{17}$$

and

$$\overline{e}_i = 0.04762 y(t_i) \tag{18}$$

ensures that (16) is satisfied.

Using SIVIA with the initial search box $[\mathbf{p}]_0 = [0, 5]^{\times 3}$ and various values of the precision parameter ε leads to the results summarized in Table 2 for computations on an Athlon 1800+.

ε	0.005	0.0025	0.00125
Comput. time (s)	9	14	24
Volume of $\overline{\mathbb{S}}$	$1.7 \cdot 10^{-3}$	$4 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$

Table 2. Estimation results using a closed-form expression

Figure 3 presents the projections of $\overline{\mathbb{S}}$ onto the (k_{01}, k_{12}) and (k_{12}, k_{21}) planes when $\varepsilon = 0.0025$. $\mathbb{S}$ consists of two disconnected subsets. Again, k_{01} and k_{12} can be exchanged without modifying the input-output behavior of the model.

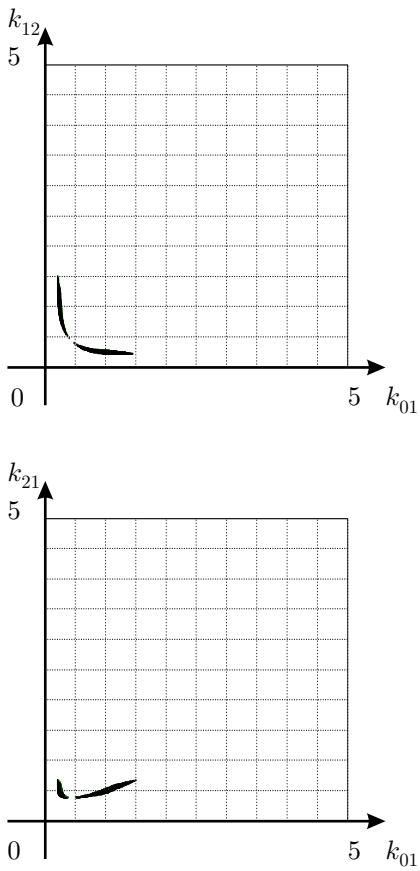

Fig. 3. Projections onto the (k_{01}, k_{12}) and (k_{01}, k_{21}) planes of $\overline{\mathbb{S}}$ found using a closed-form expression with $\varepsilon = 0.0025$

4.2.2. *Set inversion using guaranteed integration* With this approach, a closed-form expression for $y_m(\mathbf{p}, t_i)$ is no longer needed.

For a given box $[\mathbf{p}] = [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$ in parameter space such that $\underline{\mathbf{p}} \geqslant \mathbf{0}$, with

$$\underline{\mathbf{p}} = (\underline{k}_{01}, \underline{k}_{12}, \underline{k}_{21})^{\mathrm{T}} \text{ and } \overline{\mathbf{p}} = (\overline{k}_{01}, \overline{k}_{12}, \overline{k}_{21})^{\mathrm{T}},$$

it is possible to enclose $\mathbf{g}(\mathbf{x}, \mathbf{p}, u)$ in (8) between

$$\underline{\mathbf{g}}(\mathbf{x}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, u) = \begin{pmatrix} -(\overline{k}_{21} + \overline{k}_{01}) x_1 + \underline{k}_{12} x_2 + u \\ \underline{k}_{21} x_1 - \overline{k}_{12} x_2 \end{pmatrix}$$

and

$$\overline{\mathbf{g}}(\mathbf{x}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, u) = \begin{pmatrix} -(\underline{k}_{21} + \underline{k}_{01}) x_1 + \overline{k}_{12} x_2 + u \\ \overline{k}_{21} x_1 - \underline{k}_{12} x_2 \end{pmatrix}$$

As $\underline{\mathbf{p}} \geqslant \mathbf{0}$, it is easy to show that

$$\mathbf{x}' = \underline{\mathbf{g}}(\mathbf{x}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, u) \text{ and } \mathbf{x}' = \overline{\mathbf{g}}(\mathbf{x}, \underline{\mathbf{p}}, \overline{\mathbf{p}}, u)$$

are cooperative. The technique presented in Section 3.3 thus makes it possible to obtain an inclusion function for $y_m(\mathbf{p}, t_i)$. Since the initial condition of the model does not depend on $\mathbf{p}$, the bounding cooperative systems are given the same initial condition

$$\underline{\mathbf{x}}_0 = \mathbf{x}_0 = \overline{\mathbf{x}}_0.$$

For $[\underline{e}_i, \overline{e}_i]$, $i = 1, \ldots, 16$, the same intervals are taken as in Section 4.2.1. Using SIVIA and the guaranteed

integration toolbox VNODE with the same initial search box in parameter space now leads to the results summarized in Table 3. The shape and volume of the set obtained for $\varepsilon = 0.005$ are similar to those obtained in Section 4.2.1 for $\varepsilon = 0.0025$.

ε	0.01	0.005
Comput. time (s)	1300	1600
Volume of $\overline{\mathbb{S}}$	$2.5 \cdot 10^{-3}$	$6 \cdot 10^{-4}$

Table 3. Estimation results using guaranteed integration

Two remarks are in order when comparing Tables 2 and 3. First, with the same value of the precision parameter ε, the volume of $\overline{\mathbb{S}}$ is smaller when using guaranteed integration than with the closed-form expression used in Section 4.2.1. This is due to a less pessimistic inclusion function. However, if results of the same accuracy are compared, the computing time using guaranteed integration is more than 100 times larger than with the closed-form expression.

4.2.3. *Nonlinear system* Assume now that k_{01} in (8) depends on x_1 according to

$$k_{01}(x_1) = \frac{a}{1 + bx_1}.$$

This corresponds to a Michaelis-Menten nonlinearity (Godfrey, 1983).

The evolution of the quantities of material in the two compartments of the model of Figure 1 is now described by the nonlinear state equation

$$\mathbf{x}' = \mathbf{h}(\mathbf{x}, \mathbf{p}, u), \tag{19}$$

where

$$\mathbf{p} = (a, b, k_{12}, k_{21})^{\mathrm{T}}$$

and

$$\mathbf{h}(\mathbf{x}, \mathbf{p}, u) = \begin{pmatrix} -k_{21}x_1 - \dfrac{ax_1}{1 + bx_1} + k_{12}x_2 + u \\ k_{21}x_1 - k_{12}x_2 \end{pmatrix}.$$

Again, Compartment 2 is observed, with input and initial condition as in Sections 4.2.1 and 4.2.2. The observation equation is as in (15), with the same hypothesis about the $b(t_i)$s. An inclusion function based on guaranteed numerical integration will be employed.

For any $\mathbf{p} \in [\underline{\mathbf{p}}, \overline{\mathbf{p}}]$ such that $\underline{\mathbf{p}} \geqslant \mathbf{0}$, it is possible to bound $\mathbf{h}(\mathbf{x}, \mathbf{p}, u)$ in (19) between

$$\begin{pmatrix} -\left(\overline{k}_{21} + \dfrac{\overline{a}}{1 + \underline{b}x_1}\right)x_1 + \underline{k}_{12}x_2 + u \\ \underline{k}_{21}x_1 - \overline{k}_{12}x_2 \end{pmatrix}$$

and

$$\begin{pmatrix} -\left(\underline{k}_{21} + \dfrac{\underline{a}}{1 + \overline{b}x_1}\right)x_1 + \overline{k}_{12}x_2 + u \\ \overline{k}_{21}x_1 - \underline{k}_{12}x_2 \end{pmatrix}.$$

The resulting systems are cooperative, as $\underline{\mathbf{p}} \geqslant \mathbf{0}$. It is thus again simple to built an inclusion function for $y_{\mathrm{m}}(\mathbf{p}, t_i)$.

Two sets of data points have been considered. In the first one, the data points $y(t_i)$ and intervals $[\underline{e}_i, \overline{e}_i]$, $i = 1, \ldots, 16$, are as in Sections 4.2.1 and 4.2.2. SIVIA has been used here with the initial search box

$$[\mathbf{p}]_0 = [0,5] \times [0,5] \times [0.25, 0.25] \times [0.5, 0.5],$$

which means that k_{12} and k_{21} are now treated as known *a priori*. When $\varepsilon = 0.001$, in less than 55s on an Athlon 1800+, the subpaving $\overline{\mathbb{S}}$ presented on Figure 4 is obtained.

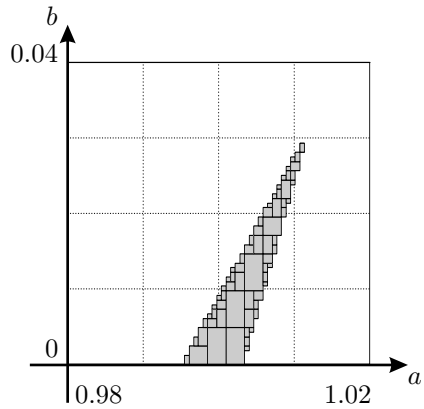

Fig. 4. Outer approximation of the solution set for (a, b) when the true system is linear

By projecting it on the axes of parameter space, one gets the following parameter uncertainty intervals

$$[a] = [0.9955, 1.0114]$$

and

$$[b] = [0, 0.02930].$$

Since the data have been generated with a linear model, it comes as no surprise that the parameter uncertainty interval for b includes 0.

The second set of data points has been obtained by corrupting the noise-free response $x_2(\mathbf{p}^*, t_i)$ of (19) where $(a^*, b^*, k_{12}^*, k_{21}^*) = (1, 4/3, 0.25, 0.5)$ with realizations of a pseudorandom noise $b(t_i)$ belonging to the interval $[-0.05, 0.05]$ according to (15), see Figure 5.

The intervals for $[\underline{e}_i, \overline{e}_i]$, $i = 1, \ldots, 16$, are computed as before, using (17) and (18), and the initial search box is as for the first set of data points.

When $\varepsilon = 0.01$, in less than 4mn, the subpaving $\overline{\mathbb{S}}$ presented in Figure 6 is obtained. As b cannot be zero, the data represented on Figure 5 could not have been generated by a linear model, given our hypotheses.

5. SOFTWARE

We shall limit ourselves to tools that can be downloaded freely, at least for academic use.

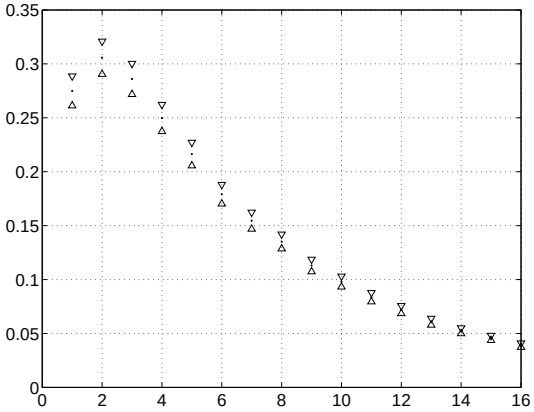

Fig. 5. Measured output (·), upper bounds (▽) and lower bounds (△) of the tolerance intervals (the true system is nonlinear)

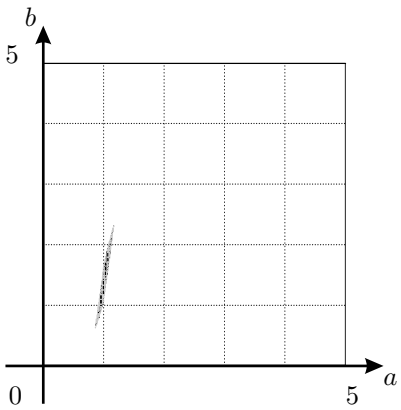

Fig. 6. Outer approximation of the solution set for (a, b) when the true system is nonlinear

Those familiar with MATLAB will find INTLAB, a library developed by S. M. Rump, particularly useful for experimenting with interval techniques. INTLAB runs under WINDOWS, UNIX and LINUX and may be downloaded from

`http://www.ti3.tu-harburg.de /english/index.html`

A number of interval algorithms cannot be implemented efficiently enough with an interpreted language such as MATLAB; this is why interval libraries built in compiled languages should also be useful.

C-XSC 2.0, a C++ class library developed by many people over more than a decade of an effort started at the University of Karlsruhe, is available at

`http://www.math.uni-wuppertal.de /~xsc/xsc/download.html`

together with a C++ toolbox for verified computing that works with C-XSC 2.0.

PROFIL/BIAS, also written in C++ and developed by O. Knüppel and S. M. Rump, is available at

`ftp://ti3sun.ti3.tu-harburg.de /pub/profil/`

Chapter 11 of (Jaulin *et al.*, 2001*b*) shows how a library such as PROFIL/BIAS could be built and how subpavings and algorithms such as SIVIA can then be implemented. The corresponding C++ source code is available at

`http://www.lss.supelec.fr /books/intervals`

A FORTRAN 90 software developed by R. B. Kearfott for global optimization is available at

`http://interval.louisiana.edu /GlobSol/download_GlobSol.html`

All of these products have benefited considerably from object-oriented programming and operator overloading, which allow intervals and boxes to be manipulated about as simply as standard data types.

Tools for guaranteed numerical integration have been developed in various languages. AWA uses PASCAL-XSC and may be found at

`ftp://ftp.iam.uni-karlsruhe.de /pub/awa/`

COSY was written in FORTRAN 90 and is available at

`http://cosy.pa.msu.edu/`

VNODE has been developed in C++ using PROFIL/BIAS and may be downloaded from

`http://www.cas.mcmaster.ca/~nedialk /Software/VNODE/VNODE.shtml`

A regularly updated list of pointers to software implementing interval analysis can be found at

`http://www.cs.utep.edu /interval-comp/main.html`

6. CONCLUSIONS

When nonlinear parameter estimation involves minimizing possibly nonconvex cost functions or solving sets of nonlinear inequalities, global deterministic methods based on interval analysis have definite advantages over more conventional local iterative methods that are unable to provide any guarantee as to their results. Structural identifiability studies can be bypassed since all solutions are provided, including (but not limited to) those due to a lack of global identifiability. Examples drawn from compartmental modeling have shown that it is possible to estimate the parameters of models defined by (possibly nonlinear) differential equations. The main challenge is to increase the complexity of the problems that can be considered as much as possible. Two allies in this endeavor have been briefly presented; the first one is the notion of contractor, which allows boxes to be reduced and sometimes eliminated without bisection, and the second is the property of cooperativity, which

allows efficient inclusion functions to be derived for important classes of differential models.

The ideas presented here in the context of parameter identification readily extend to state estimation or parameter tracking, see (Kieffer *et al.*, 1998), (Jaulin *et al.*, 2001*a*) and (Jaulin, 2002).

7. REFERENCES

Davis, E. (1987). Constraint propagation with interval labels. *Artificial Intelligence* **32**(3), 281–331.

Godfrey, K. (1983). *Compartimental Models and Their Application*. Academic Press. London.

Gouzé, J. L., A. Rapaport and Z. M. Hadj-Sadok (2000). Interval observers for uncertain biological systems. *Journal of Ecological Modelling* (133), 45–56.

Griewank, A. and Corliss, G. F., Eds.) (1991). *Automatic Differentiation of Algorithms: Theory, Implementation and Applications*. SIAM. Philadelphia, PA.

Hammer, R., M. Hocks, U. Kulisch and D. Ratz (1995). *C++ Toolbox for Verified Computing*. Springer-Verlag. Berlin.

Hansen, E. R. (1968). On solving systems of equations using interval arithmetic. *Mathematical Computing* **22**, 374–384.

Hansen, E. R. (1992). *Global Optimization Using Interval Analysis*. Marcel Dekker. New York, NY.

Hoefkens, J., M. Berz and K. Makino (2001). Efficient high-order methods for ODEs and DAEs. In: *Automatic Differentiation : From Simulation to Optimization* (G. Corliss, C. Faure and A. Griewank, Eds.). Springer-Verlag. New-York, NY. pp. 341–351.

Hyvönen, E. (1992). Constraint reasoning based on interval arithmetic; the tolerance propagation approach. *Artificial Intelligence* **58**(1-3), 71–112.

Jaulin, L. (2002). Nonlinear bounded-error state estimation of continuous-time systems. *Automatica* **38**, 1079–1082.

Jaulin, L. and E. Walter (1993). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica* **29**(4), 1053–1064.

Jaulin, L., M. Kieffer, I. Braems and E. Walter (2001*a*). Guaranteed nonlinear estimation using constraint propagation on sets. *International Journal of Control* **74**(18), 1772–1782.

Jaulin, L., M. Kieffer, O. Didrit and E. Walter (2001*b*). *Applied Interval Analysis*. Springer-Verlag. London.

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

Kieffer, M. and E. Walter (1998). Interval analysis for guaranteed nonlinear parameter estimation. In: *MODA 5-Advances in Model-Oriented Data Analysis and Experiment Design* (A. C. Atkinson, L. Pronzato and H. P. Wynn, Eds.). pp. 115–125. Physica-Verlag. Heidelberg.

Kieffer, M., L. Jaulin and E. Walter (1998). Guaranteed recursive nonlinear state estimation using interval analysis. In: *Proceedings of the 37th IEEE Conference on Decision and Control*. Tampa, FL. pp. 3966–3971.

Lohner, R. (1992). Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value-problem. In: *Computational Ordinary Differential Equations* (J. R. Cash and I. Gladwell, Eds.). Clarendon Press. Oxford. pp. 425–435.

Milanese, M., Norton, J., Piet-Lahanier, H. and Walter, E. (Eds.) (1996). *Bounding Approaches to System Identification*. Plenum Press. New York, NY.

Moore, R. E. (1959). Automatic error analysis in digital computation. Technical Report LMSD-48421 Lockheed Missiles and Space Co, Palo Alto, CA.

Moore, R. E. (1979). *Methods and Applications of Interval Analysis*. SIAM. Philadelphia, PA.

Nedialkov, N. S. and K. R. Jackson (2001). Methods for initial value problems for ordinary differential equations. In: *Perspectives on Enclosure Methods* (U. Kulisch, R. Lohner and A. Facius, Eds.). Springer-Verlag. Vienna. pp. 219–264.

Neumaier, A. (1990). *Interval Methods for Systems of Equations*. Cambridge University Press. Cambridge, UK.

Rall, L. B. (1981). *Automatic Differentiation: Techniques and Applications*. Vol. 120 of *Lecture Notes in Computer Science*. Springer-Verlag. Berlin.

Rall, L. B. and G. F. Corliss (1999). Automatic differentiation: Point and interval AD. In: *Encyclopedia of Optimization* (P. M. Pardalos and C. A. Floudas, Eds.). Kluwer. Dordrecht, the Netherlands.

Ratschek, H. and J. Rokne (1984). *Computer Methods for the Range of Functions*. Ellis Horwood. Chichester, UK.

Ratschek, H. and J. Rokne (1988). *New Computer Methods for Global Optimization*. Ellis Horwood. Chichester, UK.

Sam-Haroud, D. J. and B. Faltings (1996). Consistency techniques for continuous constraints. *Constraints* **1**(1-2), 85–118.

Smith, H. L. (1995). *Monotone Dynamical Systems: An Introduction to the Theory of Competitive and Cooperative Systems*. Vol. 41 of *Mathematical Surveys and Monographs*. American Mathematical Society. Providence, RI.

van Hentenryck, P., L. Michel and F. Benhamou (1998). `Newton`: Constraint programming over nonlinear constraints. *Science of Computer Programming* **30**(1–2), 83–118.

Walter, E. (1982). *Identifiability of State Space Models*. Springer-Verlag. Berlin.