Accelerating Consistency Techniques and Prony's Method for Reliable Parameter Estimation of Exponential Sums*

Jürgen Garloff[1], Laurent Granvilliers[2], and Andrew P. Smith[1]

[1] University of Applied Sciences / FH Konstanz
Postfach 100543, D-78405 Konstanz, Germany
{garloff,smith}@fh-konstanz.de

[2] LINA, University of Nantes
BP 92208, F-44322 Nantes cedex 3, France
granvilliers@lina.univ-nantes.fr

Abstract. In this paper the problem of parameter estimation for exponential sums is considered, i.e., of finding the set of parameters (amplitudes as well as decay constants) such that the exponential sum attains values in specified intervals at prescribed time data points. These intervals represent uncertainties in the measurements. An interval variant of Prony's method is given by which a box can be found containing all the consistent values of the parameters. Subsequently this box is tightened by the use of consistency techniques, which are accelerated by the introduction of redundant constraints. The use of interval arithmetic results in enclosures for the consistent values of the parameters which can be guaranteed also in the presence of rounding errors.

Keywords: Parameter estimation, exponential sum, Prony's method, interval arithmetic, constraint propagation, redundant constraint.

1 Introduction

The simulation of complex systems for a wide range of applications dates back to the early development of modern computers. Once a mathematical model is known, the system behaviour can be analysed without the need for practical experimentation. This approach is specifically useful to compute information which cannot easily be obtained in practice or to test extreme situations. It also becomes possible to predict the system behaviour or to optimize system components. In the following, we will consider a family of dynamical systems modeled by the function

$$y(t) = f(\mathrm{x}, t), \tag{1}$$

* This work has been supported by a PROCOPE project funded by the French Ministry of Education, the French Ministry of Foreign Affairs, the German Academic Exchange Service (DAAD) and the Ministry of Education and Research of the Federal Republic of Germany under contract no. 1705803.

where t represents time, and $\mathrm{x} \in \mathbf{R}^n$ is the vector of parameters. Each individual system leads to the problem of finding consistent values of parameters.

Let observations of the system be given, that is a series of data $(\tilde{y}_i, t_i)$, $i = 1, \ldots, m$, where $\tilde{y}_i$ is the system output at time t_i. The model-driven inverse problem (*parameter estimation problem*) consists of finding values of x such that the following equations hold:

$$\tilde{y}_i = f(\mathrm{x}, t_i), \quad i = 1, \ldots, m.$$

Unfortunately, this problem generally has no solution, since output values may be imprecise and uncertain. Therefore one tries to determine values of the model parameters that provide the best fit to the observed data, generally based on some type of maximum likelihood criterion, which results in minimizing the function

$$\sum_{i=1}^{m} w_i (f(x, t_i) - \tilde{y}_i)^2. \tag{2}$$

It is not uncommon for the objective function (2) to have multiple local optima in the area of interest. However, the standard methods used to solve this problem are local methods that offer no guarantee that the global optimum, and thus the best set of model parameters, has been found. In contrast, methods from global optimization [10, 11, 13] are capable of localizing the global optimum of (2). However, this approach does not take into account that the observed data are affected by uncertainty. Therefore the resulting models may be inconsistent with error bounds on the data.

To take uncertainty into account, we assume that the observed data are corrupted by errors, e.g. measurement errors, $\pm\varepsilon_i, \varepsilon_i \geq 0, i = 1, \ldots, m$. Then the correct value $y_i = f(x^*, \varepsilon_i)$ is within the interval $[\tilde{y}_i - \varepsilon_i, \tilde{y}_i + \varepsilon_i]$, $i = 1, \ldots, m$. More generally, we suppose that y_i is known to be contained in the interval $[a_i, b_i]$. The data driven inverse problem (*parameter set estimation problem*) consists of finding values of x subject to the following system of inequalities:

$$a_i \leqslant f(\mathrm{x}, t_i) \leqslant b_i, \quad i = 1, \ldots, m. \tag{3}$$

The aim is to compute a representation of the set Ω of the consistent values of the parameters that may help in decision making. Interval arithmetic and inclusion functions for the model functions are used in [17, 21] to find boxes generated by bisection which are contained in Ω; the union of these boxes constitutes an inner approximation of Ω. Also, boxes are identified which contain part of the boundary of Ω or contain only inconsistent values; boxes of this second category can be used to construct an outer approximation of the set of inconsistent values. However, such an approach can not handle large initial boxes or problems with many parameters. Therefore, interval constraint propagation techniques are introduced in [19] to drastically reduce the number of bisections.

In this paper, we concentrate on models of exponential sums arising in many applications such as, e.g., pharmacokinetics [14, 26]. It is well-known, e.g. [5], p. 242, and [22], that parameter estimation of exponential sums is notoriously sensitive to data perturbations. Two complementary techniques are applied. The first one is an interval variant of Prony's method [22, 25], which aims to compute an initial domain for the parameters to be estimated. The second one is applied after problem (3) is transformed into a set of equalities and is the symbolic generation of redundant constraints in order to accelerate constraint propagation. The challenge is to compute constraints leading to more precision in the numerical process, to control the amount of symbolic computations and to limit the number of redundancies in order to avoid slow-downs of the whole solving procedure.

The outline of this paper is as follows. The basics of interval arithmetic and constraint satisfaction techniques are presented in Section 2. The new methods are introduced in Section 3. A numerical example is given in Section 4. We finally conclude in Section 5.

2 Preliminaries

2.1 Interval Arithmetic

We consider the following sets: the set $\mathbf{R}$ of real numbers including the infinities, the finite set $\mathbf{F}$ of floating point numbers and the finite set $\mathbf{I}$ of closed intervals spanned by two floating point numbers. Every interval $\mathbf{x} \in \mathbf{I}$ is denoted by $[\underline{x}, \overline{x}]$ and is defined as the set of real numbers $\{x \in \mathbf{R} \mid \underline{x} \leqslant x \leqslant \overline{x}\}$.

Interval arithmetic [23] is a set theoretic extension of real arithmetic. The operations are implemented by floating-point computations with interval bounds according to monotonicity properties. For instance, the sum $[a, b] + [c, d]$ is equal to $[a + c, b + d]$, provided that the left bound is downward rounded and the right bound is upward rounded. Interval reasonings can be extended to complex functions using the so-called interval evaluation method. Given a function $f : \mathbf{R}^n \to \mathbf{R}$, let each real number in the expression of f be replaced by the interval spanned by floating point numbers obtained by rounding this real number downward and upward, each variable be replaced with its domain, and each operation be replaced with the corresponding interval operation. Then the interval expression can be evaluated using interval arithmetic, which results in a superset of the range of f over the domain of the variables.

2.2 Consistency Techniques

A numerical constraint satisfaction problem (NCSP) is given by a set of variables $\{x_1, \ldots, x_n\}$, each variable x_i lying in an interval domain $\mathbf{x}_i$, and a set of constraints over the real numbers $\{c_1, \ldots, c_m\}$. The solution set of a NCSP is defined as the set

$$\{a \in \mathbf{R}^n \mid c_1(a) \wedge \cdots \wedge c_m(a)\},$$

where each constraint c_j is considered as a relation.

Consistency techniques aim to reduce the Cartesian product of variables domains $\mathbf{x}_1 \times \cdots \times \mathbf{x}_n$, which defines the search space called a box. Most of the reduction algorithms are based on constraint projections. The projection of a constraint $c(x_1, \ldots, x_n)$ over a variable x_i is the set

$$\Pi_i(c) = \{a_i \in \mathbf{x}_i \mid \forall j \in \{1, \ldots, n\} \setminus \{i\}, \exists a_j \in \mathbf{x}_j : \; c(a_1, \ldots, a_n)\}.$$

It follows that the reduction step

$$\mathbf{x}_i := \Pi_i(c)$$

is reliable since each value belonging to the complementary set cannot be extended in a solution of the NCSP. In practice projections are reliably approximated by means of interval computations. For example the inversion algorithm uses the so-called relational interval arithmetic [8]. A numerical inversion procedure has been described as a chain rule in [16].

Example 1. Consider the constraint $2x_1 - x_2^2 = 4$, given $(x_1, x_2) \in [-3, 3] \times [1, 3]$. The computation of its projection over x_1 by the chain rule can be explained as follows. Define an equivalent constraint, where the left-hand term is reduced to x_1, namely $x_1 = (4 + x_2^2) \div 2$. Evaluate the right-hand term using interval arithmetic. The interval $[2.5, 6.5]$ is computed, and it is intersected with the domain of x_1. The new domain of x_1 is equal to $[2.5, 3]$. Thus, the set of values $[-3, 2.5)$ has been shown to be locally inconsistent with the given constraint.

Given a set of constraints, constraint projections have to be processed in sequence in order to obtain the consistency of the whole problem. The corresponding iterative algorithm is called constraint propagation. The result is a new box that contains the solution set. In order to separate the solutions, constraint propagation has to be embedded in a more general bisection algorithm. Boxes are reduced and then bisected until every box is sufficiently small.

2.3 Data Fitting Problems as NCSPs

Problem (3) should be transformed before propagation for two reasons. First, the variable y_i has to be explicit in order to reduce the error bounds. Second, each data value leads to two inequalities involving the term $f(\mathrm{x}, t_i)$. Since constraints are processed independently, an efficient approach consists of sharing computations over this term. Problem (3) is equivalent to the following set of existentially quantified equations

$$\exists y_i \in [a_i, b_i] : \; y_i = f(\mathrm{x}, t_i), \quad i = 1, \ldots, m. \tag{4}$$

Now, quantifiers can be removed, making the variables y_i first-class variables. This leads to Problem (5):

$$y_i = f(\mathrm{x}, t_i), \quad i = 1, \ldots, m. \tag{5}$$

Problems (4) and (5) are equivalent for computations of projections over the parameters. In fact, quantifiers just introduce an intermediary level of projections, which is of no benefit. It can clearly be seen that constraint propagation for Problem (5) is on average twice as fast as propagation for Problem (3).

2.4 Exponential Sums

We consider now a model with exponential sums, as follows:

$$f(\mathrm{x}, t) = \sum_{j=1}^{p} x_{2j-1} \exp(-x_{2j} t), \quad n = 2p. \tag{6}$$

In fact three problems occur when exponential sums are processed by consistency techniques. The first problem is the evaluation of the exponential function over positive real numbers far from 0. For instance consider a term $\exp(-tx)$ given $t = 100$ and suppose that x is negative. If x is smaller than -8 then $\exp(-tx)$ is evaluated to $+\infty$ on a 64-bit machine. In this case, interval-based methods are powerless. This weakness points out the needs for getting an *a priori* tight search space of parameters.

The second problem concerns slow convergences in constraint propagation. The cause is that two exponential sums from two different constraints have a similar shape. For instance consider the terms $f_1(x) = 0.2e^{0.3x} + 1$ and $f_2(x) = 0.5e^{0.4x}$, depicted in Figure 1 (f_2 has the largest slope). Domain reductions are numbered from 1. The first reduction concerns the upper bound of y using f_1. The eliminated box contains no solution of equation $y = f_1(x)$, i.e., no point of the curve of f_1. Then, the upper bound of x is contracted using f_2, and so on. A similar process leads to the reduction of the other bounds. In this case, the number of constraint processing steps using the chain rule is equal to 82.

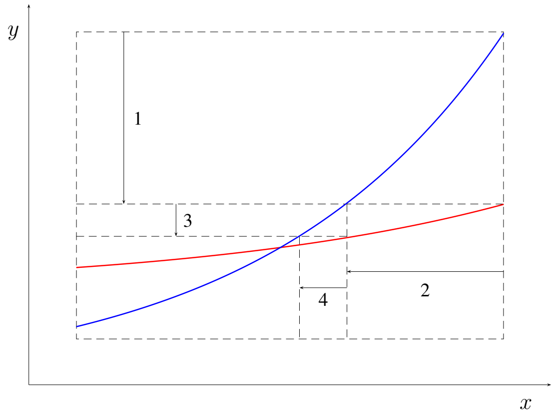

Fig. 1. Constraint Propagation over Two Exponential Terms.

In practice, the only difference is that the variables are not reduced to real values, but that they belong to real intervals. The intersection of curves becomes an intersection of surfaces. In this case, inefficiencies of constraint propagation remain.

The third problem is inherent to local approaches, since a sequence of local reasonings may not derive global information. Many techniques try to overcome this problem, one of them being the use of redundant constraints in the constraint propagation algorithm. A constraint is said to be *redundant* with respect to a set of constraints if it does not influence the solution set. Redundant constraints can be derived from the set using combination and simplification procedures, for instance Gröbner basis techniques for polynomials [7]. The interesting feature is that combination is a means for sharing information between constraints. The main challenge is to control the amount of symbolic computations, to compute constraints able to improve the precision of consistency techniques, and to limit the number of redundant constraints in order avoid slow-downs in constraint propagation.

3 Acceleration Methods

3.1 Prony's Method

Given the model (6), we wish to find *decay constants* x_{2j} and *amplitudes* x_{2j-1}, $j = 1, \ldots, p$, such that (5) is satisfied at equidistant $t_i = t_0 + ih$, $i = 1, \ldots, m$, with given stepsize h. A method to accomplish this task is Prony's method [25], cf. Chap. IV, §23 of [22], which dates back to the 18th century. This method relies on the observation that a function of the form (6) satisfies a linear difference equation with constant coefficients. We concentrate here on the case $p = 2$. We choose a fixed group of four time data points, selected from the set $\{1, \ldots, m\}$, say $\{1, 2, 3, 4\}$. Prony's method then first requires the solution of the following system of two linear equations in the unknowns ζ_1 and ζ_2.

$$\begin{pmatrix} y_1 & y_2 \\ y_2 & y_3 \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = - \begin{pmatrix} y_3 \\ y_4 \end{pmatrix}. \tag{7}$$

The solution (ζ_1, ζ_2) of this system provides the coefficients of a quadratic

$$q(u) = u^2 + \zeta_2 u + \zeta_1. \tag{8}$$

If the zeros u_1 and u_2 of q are distinct and positive then the decay constants are given by $\{x_2, x_4\} = \{\log(u_1)/h, \log(u_2)/h\}$. Finally, we obtain the amplitudes x_1 and x_3 from the solution of a second system of two linear equations

$$\begin{pmatrix} 1 & 1 \\ u_1 & u_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_3 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{9}$$

with $x_k = e^{-t_1 x_{k+1}} z_k$, $k = 1, 3$.

Now consider the interval problem (4). We want to find intervals $\mathbf{x}_1, \ldots, \mathbf{x}_4$, such that all $x_j \in \mathbf{x}_j$, $j = 1, \ldots, 4$, for which

$$f(\mathrm{x}, t_i) \in [a_i, b_i], \quad i = 1, \ldots, m. \tag{10}$$

By changing to the interval data given by (4), Prony's method now requires the solution of interval variants of the two linear systems (7) and (9) and the enclosure of the zero sets of the interval polynomial corresponding to (8) [1]. Special

[1] A preliminary version of the interval variant of Prony's method was given in Sect. 5.2 of [12].

care has to be taken to find tight intervals for the decay constants and amplitudes. To determine enclosures for the zero sets of q in the case that the roots are positive and can be separated, we compute an enclosure for the largest positive root by the well-known formula and a respective enclosure for the smallest positive root by an interval variant of Vietà's method.

For a system of p linear interval equations in p unknowns

$$[A]x = [b] \tag{11}$$

the (general) solution set is defined as the set

$$\Sigma = \{x \in \mathbf{R}^p \mid \exists A \in [A], b \in [b] \ : \ Ax = b\}. \tag{12}$$

Here we assume that the interval matrix is nonsingular, i.e., it contains only nonsingular real matrices. We are interested in the hull of the solution set, i.e., the smallest axis aligned box containing Σ.

For the system of two linear interval equations corresponding to (9), we can easily compute the hull of the solution set by the method presented in [3], cf. [24] p. 97. The system (7) exhibits two dependencies: The system matrix is symmetric and the coefficient in its bottom right corner is equal to the negation of the first entry of the right hand side. So it is natural to consider in the interval problem the symmetric solution set Σ_{sym} [1, 2], [24], Sect. 3.4, which is the solution set restricted to the systems with symmetric matrices, and the even smaller solution set, denoted by Σ^*_{sym}, obtained when in addition the dependency on the first entry of the right hand side is taken into account. With elementary computations (which are delegated to the Appendix) it is possible to determine the hulls of these structured solution sets. In Figure 2, these three solution sets together with their hulls for the following system

$$\begin{pmatrix} [1,3] & [0,1] \\ [0,1] & [-4,-1] \end{pmatrix} \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix} = - \begin{pmatrix} [-4,-1] \\ [-1,2] \end{pmatrix} \tag{13}$$

are displayed. The general solution set Σ consists of the whole shaded region and the symmetric solution set Σ_{sym} consists of the regions shaded in medium and dark grey. The dark grey region is the solution set Σ^*_{sym}. At least the first two solution sets can be obtained by analytical methods, cf. [1, 2], but are determined here by the computation of the solutions of a large number of real systems corresponding to boundary and interior points of the interval matrix and the interval right hand side.

If the interval system corresponding to (7) is singular, one should check whether the underlying problem is not better described by a single exponential term, i.e., we have $p = 1$ in (6). In fact, if

$$\tilde{y}_i := x_1 \exp(-x_2(t_0 + ih)) \in [a_i, b_i], \quad i = 1, 2, 3, \tag{14}$$

holds true, then it follows that

$$0 = \tilde{y}_1 \tilde{y}_3 - \tilde{y}_2^2. \tag{15}$$

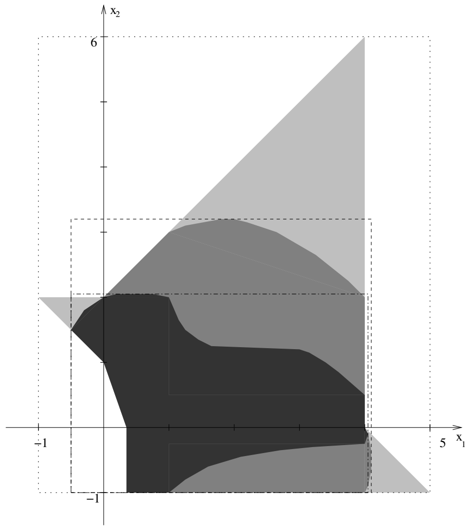

Fig. 2. The three solution sets Σ, Σ_{sym}, and Σ^{*}_{sym} and their hulls for system (13).

We mention two possibilities for tightening the enclosures for the parameters obtained in this way: we can choose another group of four time data points, compute again enclosures for the parameters and intersect with the enclosures obtained for the first group. Continuing in this way, we successively improve the quality of the enclosures. If an intersection becomes empty, we have then proven that there is no exponential function of the form (6) which solves the real interpolation problem with data taken from the intervals given in (4).

Another possible improvement is obtained as follows: If we plug on the right hand side of (6) the intervals $\mathbf{x}_j$ into x_j, $j = 1, \ldots, 4$, then we will obtain an interval function. If the evaluation of this function at a time data point results in an interval which is not equal to or a superset of the original data interval, we have proven that certain measurements are not possible. If this difference is large, we may conclude that measurements have not been made precisely enough.

A salient feature of the above approach is that if this method works, i.e., the two interval systems are nonsingular and the roots can be separated, we obtain an enclosure for the parameters without any prior information on the decay constants and amplitudes. Such prior information is normally required for the use of interval methods, e.g., [20]. Often one has to choose an unnecessarily wide starting box which is assumed to contain all feasible values of interest. Application of a subdivision method then results in a large number of subdivision steps. Therefore, Prony's method is predestinated to be used as a preprocessing step for more sophisticated methods. The amount of computational effort is negligible.

3.2 Redundant Computations

Several transformation techniques [18] of exponential sums have been proposed, mainly for the case of data equidistant in time, cf. Sect. 3.1. The other situation has been studied less. However, we will see that constraint propagation may be greatly improved if well-chosen redundant constraints are generated. Given Problem (5), the basic idea is that two terms in the same column can be divided to generate a redundant constraint, as follows:

$$\begin{cases} u_{ij} = x_{2j-1}\exp(-x_{2j}t_i), \\ u_{kj} = x_{2j-1}\exp(-x_{2j}t_k), \\ u_{ij} = u_{kj}\exp(x_{2j}(t_k - t_i)). \end{cases} \tag{16}$$

The simplification consists in eliminating variable x_{2j-1} from the last constraint. The system is then rewritten as follows:

$$\begin{cases} y_i = \sum_{j=1}^{p} u_{ij}, & i = 1, \ldots, m, \\ u_{ij} = x_{2j-1}\exp(-x_{2j}t_i), & i = 1, \ldots, m, \quad j = 1, \ldots, p, \\ u_{ij} = u_{kj}\exp(x_{2j}(t_k - t_i)), & 1 \leqslant i < k \leqslant m, \quad j = 1, \ldots, p. \end{cases} \tag{17}$$

The number of exponential terms in the system potentially grows from mp to $mp + 0.5m(m-1)p$. In fact the complexity is increased by a non-constant factor $O(m)$. Even if the precision of numerical computations is improved by the use of the redundant constraints, too many constraints to be considered during propagation may induce a slow-down. We then show how to keep the same complexity while filtering the necessary constraints. Consider the first three constraints from the initial system c_1, c_2, and c_3, and let j represent the j-th column. The symbolic step is an elimination procedure which combines two constraints in order to remove the variable x_{2j-1}. The aim is to derive a constraint whose projection over x_{2j} can be efficiently computed. As a consequence, a redundancy, e.g., between c_1 and c_2, is equivalent to an existentially quantified formula, as follows:

$$\exists x_{2j-1}\ c_1 \wedge c_2.$$

Now, suppose that the two following redundancies are available:

$$\exists x_{2j-1}\ c_1 \wedge c_2, \qquad \exists x_{2j-1}\ c_2 \wedge c_3.$$

It can be shown that the third redundancy c defined by $\exists x_{2j-1}\ c_1 \wedge c_3$ is useless for reducing the domain of x_{2j}. Suppose that one value of x_{2j} does not allow the satisfaction of c. Then either c_1 or c_3 is violated, and so do the first two redundancies. We then conclude that c is useless. As a consequence, it suffices to consider per column the redundancies between two consecutive rows. The number of redundant constraints is then equal to $(m-1)p$.

Example 2. Consider the following instance of (16), where variables u are computed by simulation, given the parameter values $(10, 0.5)$:

$$\begin{cases} (i, j, k) = (1, 1, 2) \\ (t_1, t_2) = (2, 5) \\ u_{11} = x_1\exp(-x_2t_1) \\ u_{21} = x_1\exp(-x_2t_2). \end{cases} \tag{18}$$

Now, find $x_1 \in \mathrm{x}_1$ and $x_2 \in \mathrm{x}_2$ such that the equations of (18) are satisfied. First of all, if the domains are such that the exponential terms are evaluated to $+\infty$, e.g., for $\mathrm{x}_1 = \mathrm{x}_2 = [-1000, 1000]$, then consistency techniques are powerless. If the domains are tighter, e.g., $\mathrm{x}_1 = \mathrm{x}_2 = [-100, 100]$, then one box enclosing the solution is derived after 94 calls to the chain rule:

$$[9.9999999963, 10.000000004] \times [0.49999999988, 0.50000000013].$$

The redundant constraint is

$$u_{11}/u_{21} = \exp(x_2(t_2 - t_1)).$$

If it is added to the system, the number of calls decreases to 5.

In fact more work can be done symbolically. Let I, J denote the domains of u_{ij} and u_{kj} and let K denote the domain of x_{2j}. Then a new domain for variable x_{2j} can be computed by the following interval expression:

$$x_{2j} := K \cap \left(\frac{1}{t_k - t_i} \cdot \log \left(\frac{I}{J} \right) \right). \tag{19}$$

4 A Numerical Example

Software. The software `RealPaver` [15] is used for the tests. Given a model of exponential sums and a series of measurements together with error bounds, the aim is to compute the convex hull of the set of consistent values of the unknowns. In the following, the same tuning of algorithms is used, namely a fixed number of boxes in the bisection process and a fixed maximum computation time. This way, the precision of resulting boxes can be compared for different input systems.

Benchmark. Consider the following problem, consisting of four time-equidistant measurements:

$$\begin{array}{rl} x_1 e^{4.387 x_2} + x_3 e^{4.387 x_4} & \in [-0.304, -0.298] \\ x_1 e^{12.069 x_2} + x_3 e^{12.069 x_4} & \in [21.43, 21.86] \\ x_1 e^{19.751 x_2} + x_3 e^{19.751 x_4} & \in [171.9, 175.3] \\ x_1 e^{27.434 x_2} + x_3 e^{27.434 x_4} & \in [1257, 1282] \end{array}$$

Results. Starting with an initial box

$$[-100, 100] \times [-10, 10] \times [-100, 100] \times [-10, 10]$$

`RealPaver` computes no reduction. If the 6 redundant constraints are used, then the box is reduced to

$$[-100, 100] \times [-1.326, 10] \times [-100, 100] \times [-1.326, 10].$$

There is clearly a need for using Prony's method to obtain a tight initial box. For the considered problem, Prony's method computes (within 0.01s) the following enclosures for the set of parameters:

$$[-6.673, -3.374] \times [-0.130, 0.014] \times [0.911, 1.344] \times [0.247, 0.266].$$

`RealPaver` then computes the following new box:

$$[-5.881, -3.618] \times [-0.124, 0.014] \times [0.977, 1.256] \times [0.251, 0.262].$$

This precision is improved if the redundant constraints are used, as follows:

$$[-5.872, -3.740] \times [-0.123, 0.014] \times [0.991, 1.223] \times [0.252, 0.262].$$

5 Conclusion

In this paper, we have shown that constraint satisfaction techniques have to be improved in order to process exponential-based models, which are often ill-conditioned. For this purpose, two techniques have been introduced, namely an interval variant of Prony's method and a symbolic procedure. The main goal is to improve the tightness of the bounds for the parameters, whilst keeping the computation time unchanged (or improved).

In a bounded-error context, the problem is to solve a set of inequalities. A powerful approach is to use inner computations to approximate the interior of the solution set, which is often a continuum, using boxes. For this purpose, we believe that three techniques should be combined: inner computations using constraint negations [6], an inner box extension method [9] and interior algorithms based on local search.

A serious limitation of Prony's method is that it requires equidistant time data points. However, many examples in the literature contain at least some equidistant data points. If the measurements provide a group of at least four such points, then we can apply Prony's method as a preprocessing step to deliver a suitable initial box. In a future paper, we will report on Prony's method for functions (6) comprising three exponential terms ($p = 3$).

References

1. G. Alefeld, V. Kreinovich, and G. Mayer. On the shape of the symmetric, persymmetric, and skew-symmetric solution set. *SIAM J. Matrix Anal. Appl.*, 18:693–705, 1997.
2. G. Alefeld and G. Mayer. On the symmetric and unsymmetric solution set of interval systems. *SIAM J. Matrix Anal. Appl.*, 16:1223–1240, 1995.
3. N. Apostolatos and U. Kulisch. Grundzüge einer Intervallrechnung für Matrizen und einige Anwendungen. *Elektron. Rechenanlagen*, 10:73–83, 1968.
4. H. Beeck. Über intervallanalytische Methoden bei linearen Gleichungssystemen mit Intervallkoeffizienten und Zusammenhänge mit der Fehleranalysis. Dissertation, Technical University Munich, 1971.

5. R. Bellmann. *Methods of Nonlinear Analysis*, volume I. Academic Press, New York, London, 1970.
6. F. Benhamou and F. Goualard. Universally quantified interval constraints. In R. Dechter, editor, *Proceedings of International Conference on Principles and Practice of Constraint Programming*, volume 1894 of *Lecture Notes in Computer Science*, pages 67–82, Singapore, 2000. Springer-Verlag.
7. F. Benhamou and L. Granvilliers. Automatic generation of numerical redundancies for nonlinear constraint solving. *Reliable Computing*, 3(3):335–344, 1997.
8. J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.
9. H. Collavizza, F. Delobel, and M. Rueher. Extending consistent domains of numeric CSPs. In T. Dean, editor, *Proceedings of International Joint Conference on Artificial Intelligence*, pages 406–413, Stockholm, Sweden, 1999. Morgan Kaufmann.
10. W. R. Esposito and C. A. Floudas. Global optimization in parameter estimation of nonlinear algebraic models via the error–in–variables approach. *Industrial and Engineering Chemistry Research*, 37:1841–1858, 1998.
11. W. R. Esposito and C. A. Floudas. Parameter estimation of nonlinear algebraic models via global optimization. *Computers and Chemical Engineering*, 22:S213–S220, 1998.
12. J. Garloff. Untersuchungen zur Intervallinterpolation. Dissertation, University of Freiburg, *Freiburger Intervall-Berichte* 80/5, 1980.
13. C.-Y. Gau and M. A. Stadtherr. Nonlinear parameter estimation using interval analysis. *AIChE Symp. Ser.*, 94(304):444–450, 1999.
14. M. Gibaldi and D. Perrier. *Pharmacokinetics.* Marcel Dekker, Inc., New York, 1982.
15. L. Granvilliers. *RealPaver User's Manual, version 0.3*, 2003. Available at www.sciences.univ-nantes.fr/info/perso/permanents/granvil/realpaver.
16. L. Granvilliers and F. Benhamou. Progress in the solving of a circuit design problem. *Journal of Global Optimization*, 20(2):155–168, 2001.
17. E. P. Hofer, B. Tibken, and M. Vlach. Traditional parameter estimation versus estimation of guaranteed parameter sets. In W. Krämer and J. Wolff von Gudden-berg, editors, *Scientific Computing, Validated Numerics, Interval Methods*, pages 241–254. Kluwer Academic Publishers, Boston, Dordrecht, London, 2001.
18. K. Holmström and J. Petersson. A review of the parameter estimation problem of fitting positive exponential sums to empirical data. *Applied Mathematics and Computations*, 126:31–61, 2002.
19. L. Jaulin. Interval constraint propagation with application to bounded-error estimation. *Automatica*, 36:1547–1552, 2000.
20. L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis.* Springer, London, Berlin, Heidelberg, 2001.
21. L. Jaulin and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.
22. C. Lanczos. *Applied Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1956.
23. R. E. Moore. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1966.
24. A. Neumaier. *Interval Methods for Systems of Equations.* Cambridge University Press (Encyclopedia of Mathematics and its Applications), Cambridge, 1990.
25. R. Prony. Essai experimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *Journal de l'Ecole Polytechnique*, 1(2):24–76, 1795.
26. M. Weiss. *Theoretische Pharmakokinetik.* Verlag Gesundheit, Berlin, 1990.

Appendix

Determination of the hulls of the three solution sets of the linear interval system appearing in Prony's method

It is well-known, e.g. [4], that the hull of the (general) solution set of (11) can be obtained as the hull of the solutions of all the vertex systems of (11), i.e., the systems of real equations with coefficients being identical to endpoints of the respective coefficient intervals.[2] Therefore, in the case $p = 2$ we have to solve 2^6 point systems. Consider now the symmetric system

$$\begin{pmatrix} [\underline{a}_1, \overline{a}_1] & [\underline{a}_2, \overline{a}_2] \\ [\underline{a}_2, \overline{a}_2] & [\underline{a}_3, \overline{a}_3] \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} [\underline{b}_1, \overline{b}_1] \\ [\underline{b}_2, \overline{b}_2] \end{pmatrix}$$

and one of its point systems

$$\begin{pmatrix} a_1 & a_2 \\ a_2 & a_3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Assume that the matrix is nonsingular. Then it is easy to see that both components of the solution vector (x_1 and x_2) are monotonic with respect to a_1, a_3, b_1, and b_2. Therefore, x_1 and x_2 can attain their minimum and maximum only at the endpoints of the intervals $[a_1]$, $[a_3]$, $[b_1]$, and $[b_2]$. Since

$$\frac{\partial x_1}{\partial a_2} = \frac{-b_2 a_2^2 + 2a_3 b_1 a_2 - a_1 a_3 b_2}{(a_1 a_3 - a_2^2)^2}$$

x_1 can only take its minimum and maximum when $a_2 \in \{\underline{a}_2, \overline{a}_2\}$ or

$$b_2 a_2^2 - 2a_3 b_1 a_2 + a_1 a_3 b_2 = 0. \tag{20}$$

Similarly, x_2 can only take its extreme values when $a_2 \in \{\underline{a}_2, \overline{a}_2\}$ or

$$b_1 a_2^2 - 2a_1 b_2 a_2 + a_1 a_3 b_1 = 0. \tag{21}$$

So we have to solve all possible 2^5 vertex systems. We additionally have to consider point systems generated as follows: For each of the 2^4 possible combinations

$$a_1 \in \{\underline{a}_1, \overline{a}_1\}, a_3 \in \{\underline{a}_3, \overline{a}_3\}, b_1 \in \{\underline{b}_1, \overline{b}_1\}, b_2 \in \{\underline{b}_2, \overline{b}_2\},$$

solve the two quadratic equations (20) and (21); this gives up to four values $a_2^{(i)}$, $i = 1, 2, 3, 4$. Discard any $a_2^{(i)}$ for which $a_2^{(i)} \notin [\underline{a}_2, \overline{a}_2]$. Solve the point systems for the remaining $a_2^{(i)}$. Thus we need to solve at most $4 * 2^4$ extra point systems altogether. After at most 96 point systems are solved, we have to compute the smallest box containing all the solutions (x_1, x_2) generated in this way. This box provides the hull of the symmetric solution set.

[2] For a more tractable approach see Chap. 6 in [24].

We consider now the linear interval system

$$\begin{pmatrix} [\underline{a}_1, \overline{a}_1] & [\underline{a}_2, \overline{a}_2] \\ [\underline{a}_2, \overline{a}_2] & [\underline{b}_1, \overline{b}_1] \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} [\underline{b}_1, \overline{b}_1] \\ [\underline{b}_2, \overline{b}_2] \end{pmatrix}. \tag{22}$$

This is the same system as before, except that an extra dependency, viz. $a_3 = b_1$ has been introduced. Note that we have suppressed the minus-sign appearing on the right hand side of (7) for simplicity. Affixing a minus-sign on the right hand side of (7) results in a reflection of the solution set at the origin. Consider the point system

$$\begin{pmatrix} a_1 & a_2 \\ a_2 & b_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}.$$

Again, assume that the matrix is nonsingular. As before, we have that x_1 and x_2 are monotonic with respect to a_1 and b_2. In addition, x_2 is also monotonic with respect to b_1. This leaves

$$\frac{\partial x_1}{\partial a_2} = \frac{-b_2 a_2^2 + 2 b_1^2 a_2 - a_1 b_1 b_2}{(a_1 b_1 - a_2^2)^2}, \tag{23}$$

$$\frac{\partial x_1}{\partial b_1} = \frac{a_1 b_1^2 - 2 a_2^2 b_1 + a_1 a_2 b_2}{(a_1 b_1 - a_2^2)^2}, \tag{24}$$

$$\frac{\partial x_2}{\partial a_2} = \frac{-b_1 a_2^2 + 2 a_1 b_2 a_2 - a_1 b_1^2}{(a_1 b_1 - a_2^2)^2}. \tag{25}$$

We have to solve a number of point systems, which fall into four categories (see below). After these point systems are solved, as before we have a set of solution pairs (x_1, x_2). The hull of all these solutions provides the hull of Σ^*_{sym}.

1. Solve all 2^4 vertex systems of (22).

2. Solve all possible point systems, where for each of the eight choices of the vertices of $[a_1]$, $[b_1]$, $[b_2]$ we determine a finite number of values taken from $(\underline{a}_2, \overline{a}_2)$, where x_1 and x_2 may plausibly take their maximum or minimum. Up to four such values are generated by (separately) solving the two quadratic equations which are obtained by setting the numerators in (23) and (25) equal to zero, i.e.,

$$b_2 a_2^2 - 2 b_1^2 a_2 + a_1 b_1 b_2 = 0, \tag{26}$$

$$b_1 a_2^2 - 2 a_1 b_2 a_2 + a_1 b_1^2 = 0. \tag{27}$$

3. Solve all possible point systems, where for each of the eight choices of the vertices of $[a_1]$, $[a_2]$, $[b_2]$ we determine a finite number of values taken from

$(\underline{b}_1, \overline{b}_1)$, where x_1 may plausibly take its maximum or minimum. Up to two such values are generated by solving the equation, cf. (24),

$$a_1 b_1^2 - 2a_2^2 b_1 + a_1 a_2 b_2 = 0. \tag{28}$$

4. Solve all possible point systems, where for each of the four choices of the vertices of $[a_1]$ and $[b_2]$ we need to determine a finite number of values taken from $(\underline{a}_2, \overline{a}_2)$ and from $(\underline{b}_1, \overline{b}_1)$, where x_1 may plausibly take its extreme values.
   We seek points a_2 and b_1 which jointly satisfy equations (26) and (28). If we solve (28) for b_1 and plug its two solutions into (26), we end up with the condition
   $$a_2 c(d-c)(8d+c) = 0,$$
   where $c = a_1^2 b_2$ and $d = a_2^3$. Therefore, possibly valid values for a_2 are
   $$a_2^{(1)} = 0, \quad a_2^{(2)} = \sqrt[3]{c}, \quad a_2^{(3)} = \frac{1}{2}\sqrt[3]{-c}.$$
   However, $c = 0$ is a degenerate case. So if either $a_1 = 0$ or $b_2 = 0$ we must work alternatively:
   If $a_1 = 0$, we may conclude from (28) that either $a_2 = 0$ or $b_1 = 0$. However, due to nonsingularity, we have $a_2 \neq 0$. Therefore $b_1 = 0$, and from (26) it follows that $b_2 = 0$, too, whence $0 \in \Sigma_{sym}^*$. Similarly, if $b_2 = 0$, we may conclude from (26) that either $a_2 = 0$ or $b_1 = 0$. If $b_1 = 0$ we have again $0 \in \Sigma_{sym}^*$.

The numbers of point systems to be solved given above are only in the worst case. In general, these will be a lot less. Certainly these numbers are not minimal and can be optimized.